

MOBIILISOVELLUKSEN KEHITYS ILMOITUSTEN OHJELMISTOLLE



Ammattikorkeakoulututkinnon opinnäytetyö

Riihimäki, Tieto- ja viestintäteknikka

Kevät 2018

Ville Rahkonen

Tieto- ja viestintäteknikka
Riihimäen kampus

Tekijä	Ville Rahkonen	Vuosi 2018
Työn nimi	Mobiilisovelluksen kehitys ilmoitusten ohjelmistolle	
Työn ohjaaja/t	Toni Laitinen	

TIIVISTELMÄ

Opinnäytetyön tavoite oli suunnitella ja kehittää toimeksiantajalle mobiilisovellus ilmoitusten ohjelmistolle. Mobiilisovelluksen tehtävä on toimia ilmoitusten raportoinnin työkaluna erilaisten organisaatioiden työntekijöiden ja esimiesten välillä. Mobiilisovelluksen tekniikan ja ominaisuuksien avulla toimeksiantaja toivoi saavansa käyttöönsä tehokkaan ja käyttäjäystävällisen ohjelmiston.

Mobiilisovellus kehitettiin Android-käyttöjärjestelmälle käyttäen Java-ohjelmointikieltä. Ohjelmistokehityksen palvelintekniikkoina käytettiin Apache-verkkopalvelinta ja MySQL-tietokantapalvelinta, jonka ohjelmointikielenä toimi PHP. Ohjelmistokehityksessä käytettiin useita erilaisia ohjelmia, kuten Android Studio, Postman, Notepad++ ja Adobe Photoshop.

Ohjelmistokehityksen suunnitellut vaiheet valmistuivat aikataulussaan, ja mobiilisovellukseen saatiin kehitettyä kaikki toimeksiantajan toivomat ominaisuudet. Lisäksi ohjelmistokehityksen kaikissa vaiheissa otettiin huomioon mobiilisovelluksen mahdollinen jatkokehitys. Ohjelmistokehityksen päättyessä mobiilisovellus oli täysin toimiva, ja valmis toimeksiantajan omalle jatkokehitykselle.

Avainsanat mobiilisovellus, ohjelmistokehitys, ohjelmointi

Sivut 27 sivua, joista liitteitä 7 sivua

Information and Communication Technology
Riihimäki

Author	Ville Rahkonen	Year 2018
Subject	Mobile application development for notification software	
Supervisors	Toni Laitinen	

ABSTRACT

The purpose of this thesis was to design and develop a mobile application for a notification software for the commissioner. The purpose of the mobile application is to work in different organizations as a notification reporting tool between the employees and their superiors. The commissioner wished that the technology and features of the mobile application would give them an effective and user-friendly software.

The mobile application was developed for the Android operating system using the Java programming language. The Apache-network server and MySQL-database server with the PHP-programming language were used as server technologies in the software development. Many different forms of software's were used in the software development process, for example: Android Studio, Postman, Notepad++ and Adobe Photoshop.

All the planned phases in the software development were successfully completed on schedule and the mobile application included all the features that the commissioner had hoped for. On top of that each phase in the software development process was completed with further development in mind. As the software development process was finished, the mobile application was fully functional and ready for the commissioner to use for further development.

Keywords mobile application, programming, software development

Pages 27 pages including appendices 7 pages

SISÄLLYS

1	JOHDANTO.....	1
2	TAUSTAA.....	1
2.1	Kehitysprojektin suunnittelu.....	1
2.2	Mobiilisovellukselle asetetut tavoitteet	1
2.3	Ohjelmistokehityksen aikataulutus.....	2
2.4	Mobiilisovelluksen jatkokehitys	3
3	TEKNOLOGIAT JA TYÖKALUT	3
3.1	Android.....	3
3.1.1	Android Studio.....	4
3.1.2	Emulaattori	5
3.1.3	OnePlus 3.....	6
3.2	Java.....	6
3.3	WAMP	7
3.3.1	Käyttöjärjestelmä	7
3.3.2	Apache	8
3.3.3	MySQL.....	8
3.3.4	PHP	9
3.4	Muut työkalut.....	9
3.4.1	Notepad++	9
3.4.2	Postman.....	9
3.4.3	Verkkoselain	10
3.4.4	Adobe Photoshop.....	10
4	MOBIILISOVELLUKSEN SUUNNITTELU JA OHJELMISTOKEHITYS	11
4.1	Ohjelmistokehityksen laitteisto	11
4.2	Mobiilisovelluksen käyttöjärjestelmä	11
4.3	Ohjelmistokehityksen palvelinympäristö.....	11
4.4	Ohjelmointiympäristöön tutustuminen	12
4.5	Mobiilisovelluksen rakenne ja kehitystyö.....	12
4.5.1	Tietokannan luominen.....	13
4.5.2	Käyttäjän rekisteröityminen ja sisäänkirjautuminen	14
4.5.3	Käyttöliittymä ja ulkoasu	15
4.5.4	Uusi ilmoitus ja tiedonsiirto.....	16
4.5.5	Kamera ja tiedostonsiirto	17
4.6	Siirto toimeksiantajan palvelimelle.....	18
5	YHTEENVETO	19
	LÄHTEET.....	20

Liitteet

Liite 1

Android ohjelmistokehityspakkauksen lisenssisopimus

1 JOHDANTO

Opinnäytetyön aiheena on mobiilisovelluksen suunnittelu ja kehitys ilmoitusten ohjelmistolle. Opinnäytetyön toimeksiantaja on suomalainen organisaatio, jolle mobiilisovellus kehitetään yksinkertaisena ja toimivana kokonaisuutena. Ilmoitusten ohjelmiston tavoitteena on tarjota erilaisille organisaatioille työkalu, jonka avulla työntekijät voivat raportoida esimiehilleen erilaisia ilmoituksia.

Toimeksiantaja ei halua organisaationsa nimeä mainittavan opinnäytetyössä. Opinnäytetyö sisältää kuvia, tiedostojen nimiä ja otteita lähdekoodista, joissa aineisto on korvattu todellista toteutusta teoriassa vastaavalla aineistolla toimeksiantajan yksityisyyden suojaamiseksi.

Opinnäytetyö esittelee kehitysprojektin suunnittelu- ja kehitysvaiheet, käytetyt teknologiat ja työkalut, mobiilisovelluksen toiminnot sekä pohdintoja kehitysprojektin aikana tehdyistä valinnoista. Kehitysprojektin kaikki eri kehitysvaiheet antavat kokonaisuudessaan hyvän kuvan uuden mobiilisovelluksen kehitystyöstä nopeassa aikataulussa.

2 TAUSTAA

2.1 Kehitysprojektin suunnittelu

Mobiilisovelluksen suunnittelun alkuvaiheilla tuli ottaa huomioon eri kohderyhmien tarpeita. Ensimmäisenä toimeksiantajan asettamat toiveet ja vaatimukset, jotka pohjautuivat heidän kattavaan tietoon ja aiempiin kokemuksiinsa. Myös ohjelmistokehittäjän oma henkilökohtainen osaaminen täytyi huomioida, jotta ohjelmistokehitys oli sujuvaa ja tehokasta mutta samalla myös taitoja kehittävä. Kehitysprojekti oli minulle ensimmäinen mobiilisovelluksen kehitystyö, mutta sen käyttämät teknologiat ja laaja kysyntä työmarkkinoilla tekivät siitä erittäin mielenkiintoisen kehitystyön kohteen. Tässä yksittäisessä kehitysprojektissa ohjelmistokehityksen aikataulut oli erittäin tärkeää koska aikaa oli käytössä rajallisesti, ja etukäteen tarkasteltuna projekti piti sisällään paljon uusia opeteltavia taitoja.

2.2 Mobiilisovellukselle asetetut tavoitteet

Mobiilisovellukselle oli asetettu selkeät tekniset tavoitteet yhdessä toimeksiantajan kanssa. Mobiilisovelluksen tuli sisältää virheettömästi toimivat tekniset ratkaisut sisäänkirjautumisprosessiin, turvalliseen tiedonsiirtoon, mobiililaitteen kameran käyttöön ja tallennetun tiedon

jälleenkäyttöön. Lisäksi huomiota tuli kiinnittää sovelluksen käyttöliittymään ja ulkoasuun, jotta varmistettiin käyttäjien tyytyväisyys sovelluksen toimivuuden ja helppokäyttöisyyden suhteen. Mobiilisovelluksen täytyi myös olla helposti jatkokehitettävissä, sekä olla ohjelmoitu kaikilta osaltaan puhtaasti ilman ylimääräistä koodia. Kehitysprojektille luotiin myös graafinen konsepti, jossa havainnollistuu yksinkertaistettuna mobiilisovelluksen tavoiteltu toiminnallisuus (Kuva 1).



Kuva 1. Kehitysprojektille suunniteltu graafinen konsepti.

2.3 Ohjelmistokehityksen aikataulus

Kehitysprojektissa varattiin ohjelmistokehitykselle aikaa viisi viikkoa. Jokaiselle ohjelmistokehityksen viikolle oli etukäteen asetetut tavoitteet. Ensimmäisen viikon tavoitteet olivat ohjelmointiympäristöön, eli Android Studio -ohjelmaan liittyviä. En ollut aiemmin käyttänyt kyseistä ohjelmaa, mutta sen perustoiminnallisuus oli hyvin samankaltaista monien muiden jo tuttujen ohjelmointiympäristöjen kanssa. Lisäksi ensimmäisellä viikolla tuli tehdä oman älypuhelinlaitteen konfigurointi kehitysalustaksi ohjelmointitietokoneen kanssa. Toisen viikon tavoite oli mobiilisovelluksen rakenteen, käyttöliittymän ja ulkoasun kehitys. Kolmantena viikkona puolestaan keskityttiin palvelintekniikkaan ja tiedonsiirtoon. Neljäs viikko oli kehitysprojektin haastavinta aikaa, kun tuolloin tavoitteena oli kehittää mobiilisovelluksen varsinaisia toimintoja kuten kameran käyttö ja kuvan siirto. Ohjelmistokehityksen viimeisenä eli viidentenä viikkona mobiilisovellus viimeisteltiin, ja sen palvelinyhteys siirrettiin paikalliselta testipalvelimelta toimeksiantajan palvelimelle.

2.4 Mobiilisovelluksen jatkokehitys

Toimeksiantajan toiveiden mukaisesti ohjelmistokehityksen kaikissa vaiheissa tuli ottaa huomioon mobiilisovelluksen jatkokehitys. Jatkokehityksellä tarkoitetaan ohjelmistokehitystä, joka tapahtuu tämän kehitysprojektin jälkeen toimeksiantajan toimesta. Esimerkiksi mobiilisovelluksen valikoita luodessa, tuli mahdollisille uusille jatkokehityksessä luotaville ominaisuuksille varata tyhjää tilaa. Myös mobiilisovelluksen rakennetta suunniteltaessa jatkokehityksen mahdolliset lisäykset tuli ottaa huomioon.

3 TEKNOLOGIAT JA TYÖKALUT

Mobiilisovelluksen kehitys vaatii usean erilaisen teknologian ja työkalun käyttöä. Ohjelmistokehitykselle on nykyään tarjolla todella paljon erilaisia vaihtoehtoja, oli sitten kyseessä ohjelmointikieli, -ympäristö tai palvelinteknologia. Opinnäytetyössä käyttämäni vaihtoehdot ovat kaikki maailmanlaajuisesti suosittuja sekä korkeasti arvostettuja teknologioita ja työkaluja.

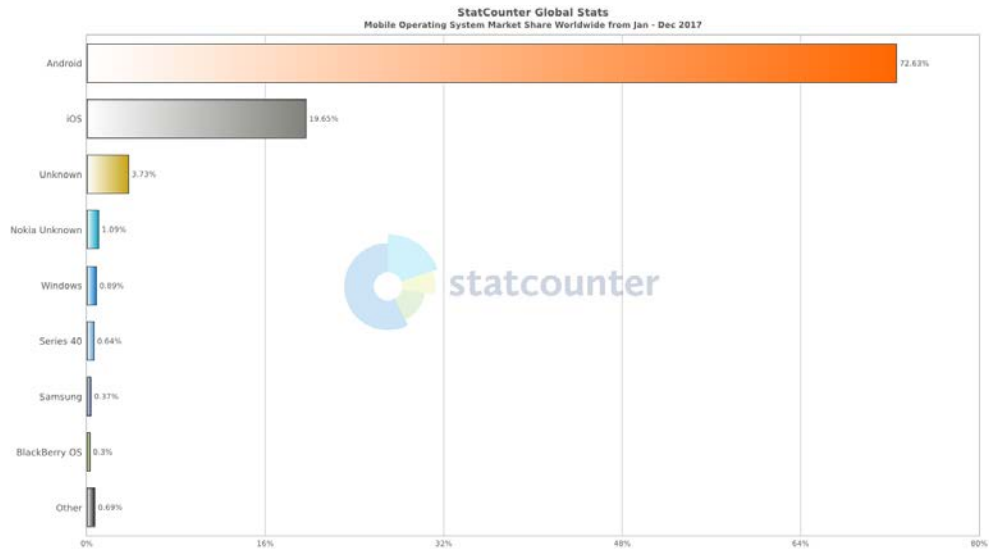
3.1 Android

Android on ohjelmistopino, joka sisältää väliohjelmistoja, käyttäjän perussovelluksia ja tärkeimpänä käyttöjärjestelmän. Android on kohdistettu älypuhelinlaitteille ja muille mobiililaitteille. Androidia kehitti ensin Android Inc., mutta Google osti sen ennen kuin Android julkaistiin yleisölle. Ohjelmistopinon virallinen julkaisu tapahtui vuonna 2008. (Wikipedia 2018 a.)

Androidia koodataan Java-ohjelmointikielellä, ja sen Java-kirjastot ovat Googlen kehittämiä. Androidin ohjelmistopino koostuu Linuxista, Bionic-C kirjastosta ja muista järjestelmäkirjastoista, oliokielellä kehitetystä ohjelmistokehityksestä, järjestelmäsovelluksista sekä sovellusohjelmista, jotka suoritetaan Dalvik-virtuaalikoneella. Ohjelmistokehitys sijaitsee Java-kirjastojen päällä, jotka puolestaan toimivat Dalvik-virtuaalikoneella. (Wikipedia 2018 a.)

Useimmiten Android-laitteet eivät käytä muokkaamatonta Android-käyttöliittymää. Laittekehittäjillä on tapana esiasentaa mobiililaitteisiinsa oma käyttöliittymänsä, ja erilaisia oletussovelluksia. Vaikka eri laitekehittäjien käyttöliittymät siis poikkeaisivat ulkoasultaan toisistaan, on kyseessä Android-käyttöjärjestelmä.

Android on maailman ylivoimaisesti suosituin mobiililaitteiden käyttöjärjestelmä (Kuva 2). Vuonna 2017 noin 72,63 % kaikista maailman omistetuista mobiililaitteista käyttivät jotain Androidin versiota käyttöjärjestelmänään (StatCounter Global Stats 2018 a).

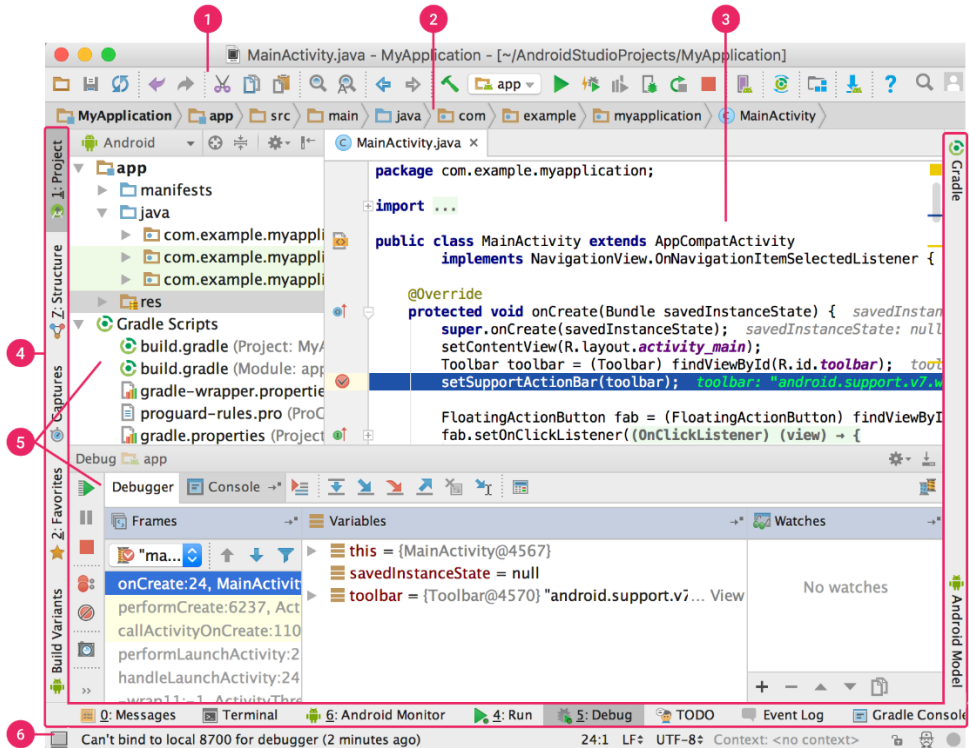


Kuva 2. Statistiikka kertoo Androidin ylivoimaisesta asemasta maailman mobiilimarkkinoilla vuonna 2017 (StatCounter Global Stats 2018 a).

3.1.1 Android Studio

Android Studio on virallinen ohjelmointiympäristö Android-mobiilisovellusten kehittämiseen. Se pohjautuu IntelliJ IDEA -ohjelmointiympäristöön, mutta sisältää sen lisäksi paljon ohjelmistokehitystä tehostavia toimintoja. Opinnäytetyön mobiilisovelluksen ohjelmistokehitys on suoritettu Android ohjelmistokehityspakkauksen lisenssisopimuksen (Liite 1) alaisilla ehdoilla.

Android Studion päänäkymästä voi nähdä ohjelmointiympäristön tärkeimmät työkalut (Kuva 3). Ohjelmointiympäristön yksi ehdottomia vahvuuksia on sen suuret kustomointi mahdollisuudet. ”Voit organisoida päänäkymää antaaksesi itsellesi enemmän ruututilaa piilottamalla tai liikkuttamalla työkalupalkkeja ja -ikkunoita” (Android Developers 2018). Yläreunan työkaluvalikosta löytyy muun muassa projektin tallennus ja sovelluksen testaus. Työkaluvalikon alapuolella on navigointipalkki, jonka avulla käyttäjä tietää mikä projektin osa on sillä hetkellä valittuna. Editori-ikkuna on jokaiselle ohjelmoijalle tuttu näky, paikka jossa varsinainen ohjelmointi ja työ tehdään. Vasemmasta reunasta löytyvät projektiin kuuluvat tiedostot. Koko ohjelmointiympäristön päänäkymää ympäröi erilaiset ikkunavalikot, joista pääsee tarkastelemaan esimerkiksi virhekoodeja ja tapahtumalokia. Näkymän alareunassa on statuspalkki, joka kertoo olennaisia tietoja projektin ja ohjelmointiympäristön nykyisestä tilasta.



Kuva 3. Android Studion päänäkö (Android Developers 2018).

3.1.2 Emulaattori

Mobiilisovelluksia kehitettäessä on tärkeää testata sovellusta säännöllisin väliajoin. Oikeaa fyysistä mobiililaitetta ei nykyään tarvitse testauksessa välttämättä käyttää, koska erilaiset emulaattorit tekevät testausprosessista vaivattomampia ja tuloksellisesti lähes yhtä tarkkoja. Emulaattorit simuloivat toiminnallaan käyttöjärjestelmiä ja toimintaympäristöjä tarkasti, ja usein pienet kehitystyön testaukset saadaan suoritettua emulaattorilla nopeammin kuin oikeilla mobiililaitteilla. Emulaattoreiden ulkoasukin pyrkii simuloimaan oikeaa älypuhelinlaitetta (Kuva 4).

On olemassa todella monia aspekteja, joita emulaattoritestaaminen ei kuitenkaan kerro käyttäjälleen yhtä tarkasti kuin oikea mobiililaitte. Esimerkiksi sovelluksen latausajat ja suorituskyky eivät välttämättä aina vastaa todellisuutta. Toisena voisi mainita eri käyttöliittymän osien käytännöllisyyden ja tuntuman. Erilaisten nappien ja tarkennus työkalujen testaaminen ei onnistu samalla tavalla emulaattorilla, koska emulaattorissa navigointi tapahtuu hiirellä ja näppäimistöllä sormenpäiden sijaan. Mobiilisovelluksen testaus olisi siis suotavaa tehdä oikeilla mobiililaitteilla, varsinkin kehitystyön loppuvaiheilla.



Kuva 4. Ruutukaappaus kehitystyössä käytetystä Android-emulaattorista.

3.1.3 OnePlus 3

OnePlus 3 on kesäkuussa 2016 julkaistu älypuhelinlaite. Sen on valmistanut kiinalainen OnePlus. OnePlus 3 on varustettu 5,5 tuuman AMOLED-näytöllä, 6 gigatavun keskusmuistilla, Qualcomm Snapdragon 820-prosessorilla ja Adreno 530-näytönohjaimella. (OnePlus, 2018). Sen käyttöjärjestelmänä toimii viimeisimpien päivitysten myötä Android-versio 8.0.0. Kyseisen älypuhelinmallin suurimmaksi vahvuudeksi on maailmanlaajuisesti arvostettu sen hyvä hinta-laatusuhde. OnePlus 3 -älypuhelinlaite käytettiin tämän ohjelmistokehitysprojektin päätoimisena testauslaitteena.

3.2 Java

Java on Sun Microsystemsin kehittämä teknologiaperhe- ja ohjelmistoalusta, johon kuuluu muun muassa laitteistoriippumaton oliopohjainen ohjelmointikieli sekä ajoaikainen ympäristö virtuaalikoneineen ja luokkakirjastoineen. Alun perin kaupallisesti lisensoitu Java on vuodesta 2007 alkaen ollut vapaan GNU GPL -lisenssin alainen. Java-alusta on käytössä noin 3,8 miljardissa laitteessa matkapuhelimista supertietokoneisiin. (Wikipedia 2018 b.)

Kehitysprojektin mobiilisovellus on ohjelmoitu Java-kielellä. Java on erittäin käyttäjäystävällinen sekä johdonmukainen ohjelmointikieli, ja se pohjautuu C-kieleen (Kuva 5). Mikäli käyttäjällä on aikaisempaa kokemusta esimerkiksi C, C++ tai C# -ohjelmointikielistä, on Java todennäköisesti helppo omaksua nopeasti.

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         // "Hello, world!" tekstin tulostus  
4         System.out.println("Hello, world!");  
5     }  
6 }  
7
```

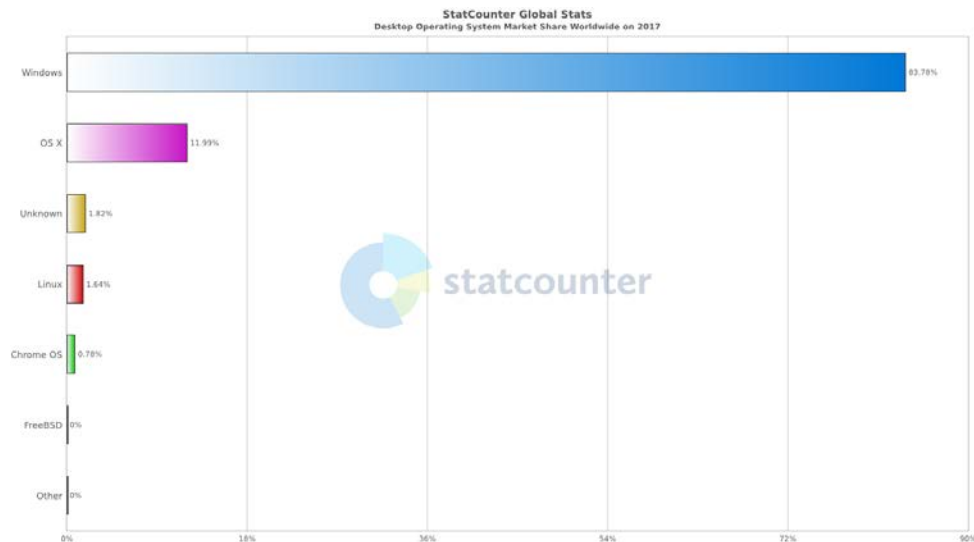
Kuva 5. Yksinkertainen tekstin tulostus esimerkki Java-ohjelmointikielellä muistuttaa rakenteeltaan vahvasti C-ohjelmointikieltä.

3.3 WAMP

WAMP on ohjelmistokokonaisuus, joka koostuu useasta eri ohjelmasta. Yhdessä ohjelmat muodostavat WWW-palvelimen, jonka alla voidaan ajaa esimerkiksi tietokantoja tai dynaamisia websivuja. WAMP on vastaava kokonaisuus kuin sitä suositumpi LAMP. Niiden ero tulee käytetystä käyttöjärjestelmästä. WAMP käyttää Windowsia, kun taas LAMP Linuxia. Linuxin parempi soveltuvuus palvelinympäristön käyttöjärjestelmäksi tekee LAMP-ohjelmistokokonaisuudesta yleisesti suositumman. Ohjelmistokokonaisuuksien nimitys (WAMP ja LAMP) tulee niihin sisältyvien ohjelmien nimien ensimmäisistä kirjaimista. WAMP sisältää Windows käyttöjärjestelmän, Apache webpalvelimen, MySQL tietokantarajapinnan ja PHP, Perl tai Python ohjelmointikielen.

3.3.1 Käyttöjärjestelmä

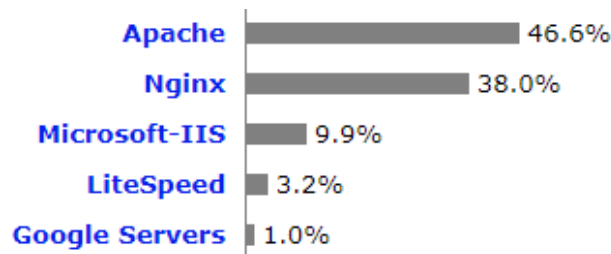
Ohjelmistokehityksen käyttöjärjestelmänä toimi Microsoft Windows 10. Windows on maailman käytetyin tietokonekäyttöjärjestelmä. Windowsista tekee suosituksen käyttöjärjestelmän sen selkeä graafinen käyttöliittymä, jonka juuret ovat lähtöisin jo huomattavan paljon vanhemmista Windows-versioista. Sen asema työpöytä-tietokoneiden käyttöjärjestelmänä on todella vahva, minkä voi huomata kuvasta 6. Vuonna 2017 noin 84 % maailman työpöytä-tietokoneista operoi Windowsin käyttöjärjestelmällä (StatCounter Global Stats 2018 b).



Kuva 6. Statistiikka osoittaa Windowsin aseman työpöytätietokoneiden valtiaana vuonna 2017 (StatCounter Global Stats 2018 b).

3.3.2 Apache

Apache on HTTP-palvelinohjelma, joka perustuu avoimeen lähdekoodiin. Apache julkaistiin vuonna 1995, ja se on ollut suosituin HTTP-palvelin jo vuodesta 1996 tähän päivään saakka. Huhtikuussa 2018 Apachea käyttää palvelinohjelmanaan 46,6 % kaikista maailman websivuista (Q-Success 2018). Määrä on suuri, mutta Apachen kovin kilpailija Nginx on saavuttanut Apachea säännöllisesti viime vuosien ajan (Kuva 7).



Kuva 7. Maailman viisi eniten käytettyä HTTP-palvelinohjelmaa huhtikuussa 2018 (Q-Success 2018).

3.3.3 MySQL

MySQL on WAMP-ohjelmistokokonaisuuden käyttämä relaatiotietokantaohjelmisto. Sen kehityksestä vastasi vuoteen 2008 asti vuonna 1995 perustettu ruotsalainen yritys MySQL AB. Toinen MySQL:n kehittäjistä oli suomalainen Michael Widenius. Vuonna 2008 Java-ohjelmointikielenkin kehittänyt Sun Microsystems kuitenkin osti MySQL AB:n, ja vuonna 2009 Oracle Corporation puolestaan osti Sun Microsystemsin. (Wikipedia 2017.)

3.3.4 PHP

Hypertext Preprocessor, lyhenteeltään PHP, on erityisesti web-palvelinympäristöissä käytettävä ohjelmointikieli. PHP julkaistiin vuonna 1995. Se on komentosarjakieli eli ammattisanastossa niin sanottu skriptikieli, joka tulkitsee koodinsa vasta ohjelman suoritusvaiheessa. Suuri osa PHP:n syntaksista on lainattu muista ohjelmointikielistä kuten C, Java ja Perl, mutta se sisältää myös omia uniikkeja ominaisuuksia. Sen tärkeimpiä ominaisuuksia ovat tehokas muuttujien käyttö, ehdolliset suoritukset ja komentojen toisto. (The PHP Group 2018.)

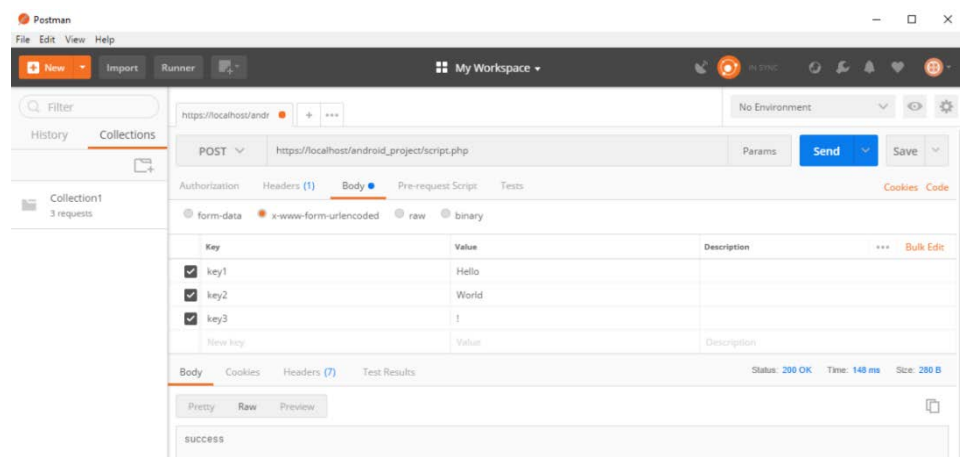
3.4 Muut työkalut

3.4.1 Notepad++

Tekstieditori on ohjelmoijan tärkein työkalu. Notepad++ on Microsoft Windowsille kehitetty avoimen lähdekoodin tekstieditori, joka tukee lähes kaikkien suosituimpien ohjelmointikielien syntaksin korostusta. Sen ensimmäinen versio julkaistiin vuonna 2003. (Wikipedia, 2016.) Opinnäytetyön ohjelmistokehityksen palvelinpuolen ohjelmoinnissa käytettiin Notepad++:n versiota 7.2.

3.4.2 Postman

Postman on ohjelmistorajapinnan kehitystyökalu, jonka avulla voidaan muun muassa lähettää http-pyyntöjä haluttuun www-osoitteeseen tai testata palvelimien ohjelmakoodien toimivuutta (Kuva 8). Postman-työkalua käyttää viisi miljoonaa ohjelmistokehittäjää ja yli 100 000 yritystä. Sen ensimmäinen versio kehitettiin sivuprojektina, joka julkaistiin lokakuussa 2012. Ohjelman suosio kasvoi kuitenkin räjähdysmäisesti, ja vuonna 2014 sen ympärille perustettiin yritys jatkokehittämään sitä. (Postdot Technologies, Inc. 2018.)

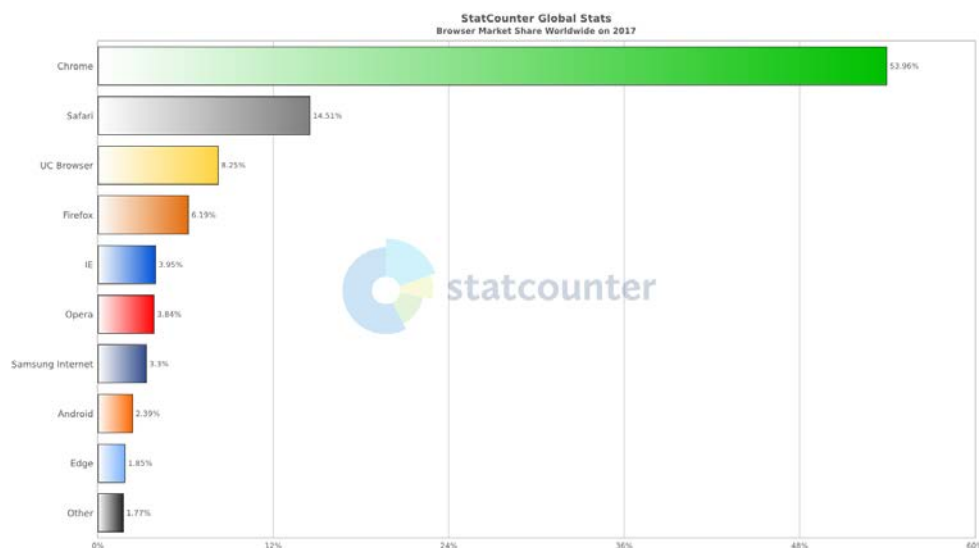


Kuva 8. Näkymä Postman-ohjelman http-pyyntöjen toiminnasta.

3.4.3 Verkkoselain

Verkkoselain on ohjelma, joka kääntää www-palvelimille sijoitetut tiedostot luettavaan muotoon. Palvelimille sijoitetut tiedostot ovat yleisimmin html-tekstiedostoja, joihin on kirjoitettu sivustolle tarkoitetun tekstin lisäksi muun muassa sen graafiset ulkoasuasetukset ja tiedostosisällöt. Verkkoselaimet ovat myös kommunikoijia, jotka luovat yhteyksiä palvelimien kanssa. Kommunikointi tapahtuu pääasiassa http-protokollaa käyttämällä. Sen avulla selaimet ja palvelimet voivat lähettää tietoa toisilleen.

Google Chrome on ollut käytetyin verkkoselain jo vuodesta 2011 lähtien, ja vuonna 2017 sitä käytti jo yli 50 % kaikista verkkoselainten käyttäjistä (Kuva 9). Google Chrome oli myös opinnäytetyön ohjelmistokehityksessä käytetty verkkoselain. Sitä käytettiin muun muassa tietokantojen tarkasteluun ja hallinnoimiseen phpMyAdmin-rajapinnassa.



Kuva 9. Yli puolet käytetyistä verkkoselaimista olivat Google Chrome vuonna 2017 (StatCounter Global Stats 2018 c).

3.4.4 Adobe Photoshop

Adobe Systemsin kehittämä Adobe Photoshop on kuvankäsittelyohjelma, joka mahdollistaa useiden eri kuvaformaattien käsittelyn useilla erilaisilla tekniikoilla. Adobe Photoshop on kaupallisten markkinoiden ylivoimaisesti käytetyin kuvankäsittelyohjelma, ja se on saatavilla suosituimmille tietokonekäyttöjärjestelmille kuten Windows ja Mac OS. Ensimmäinen Photoshop-ohjelma julkaistiin Macintoshille jo vuonna 1990, ja sitä on kehitetty tiiviisti läpi vuosien. Kirjoitushetkellä uusin versio ohjelmasta oli ilmestynyt huhtikuussa 2018, ja se kantaa nimeä Adobe Photoshop CC 2018 (19.1.3.606). (Wikipedia 2018 c.)

4 MOBIILISOVELLUKSEN SUUNNITTELU JA OHJELMISTOKEHITYS

4.1 Ohjelmistokehityksen laitteisto

Mobiilisovelluksen ohjelmistokehityksessä käytetty kehityslaitteisto sisälsi tehokkaan pöytätietokoneen ja OnePlus 3-mobiililaitteen. Kehitysprojektin pääkehityslaitte oli kyseinen pöytätietokone. Pöytätietokoneen käyttöjärjestelmänä toimi 64-bittinen Microsoft Windows 10 Home, ja sen suorittimena oli Intel i5-6600K 4.20 GHz nopeudella. Lisäksi tietokoneessa oli 8 gigatavua keskusmuistia, 1 teratavu tallennustilaa ja GeForce GTX 960 -näytönohjain. Pöytätietokoneeseen asennettiin ennen ohjelmistokehityksen alkua kaikki ohjelmistokehityksessä käytettävät ohjelmat.

OnePlus 3-mobiililaitteen tehtävä oli toimia kehitettävän mobiilisovelluksen päätestauslaitteena. Ohjelmistokehityksessä pienimmätkin muutokset sovelluksessa testattiin joko OnePlus 3-mobiililaitella tai ohjelmointiympäristön tarjoamalla Android-emulaattorilla. Yleisenä toimintatapana läpi ohjelmistokehityksen oli, että pienemmät muutokset kuten käyttöliittymän toiminnot testattiin emulaattorilla. Suuremmat muutokset ja tietokantayhteydet testattiin päätestauslaitteella. Ohjelmistokehityksen loppuvaiheissa kaikki testaukset suoritettiin päätestauslaitteella.

4.2 Mobiilisovelluksen käyttöjärjestelmä

Mobiilisovelluksen suunnittelu aloitettiin kehitysympäristön teknologioiden valitsemisella. Suurin mutta myös helpoin päätös oli valita sovelluksen kohde käyttöjärjestelmä. Käyttöjärjestelmäksi valittiin Android, koska se on ylivoimaisesti käytetyin mobiililaitteiden käyttöjärjestelmä maailmassa. Kehitystyön aikana omistin myös itse Android-käyttöjärjestelmää käyttävän älypuhelimien, jota käytettiin mobiilisovelluksen testilaitteena. Lisäksi Android on ohjelmistokehityksen kannalta kehittäjäystävällinen, koska se käyttää ohjelmointikielensä suosittua Javaa, sisältää erinomaisen ohjelmointiympäristön Android Studio ja Androidille kehitetyt sovellukset ovat myöhemmin helposti käännettävissä myös muille käyttöjärjestelmille.

4.3 Ohjelmistokehityksen palvelinympäristö

Kehitystyön aikaiseksi palvelimeksi suoraviivaisin ratkaisu oli pystyttää samaan verkkoon pääkehityslaitteen, eli pöytätietokoneen kanssa www-palvelin. Tästä aiheesta minulla oli aiemmilta opiskeluvuosiltani kokemusta, kun koulutuksissa oli käytetty LAMP-ohjelmistopinoa. Koska pääkehityslaitteella oli käyttöjärjestelmänä Microsoft Windows 10, valitsin luonnollisesti LAMP-ohjelmistopinoa vastaavan mutta Windowsille asennettavan WAMP-kokonaisuuden. Näin ollen ylimääräiselle virtuaalisen palvelimen pystytykselle ei ollut tarvetta.

WAMP-ohjelmistopinon valinnan myötä ohjelmistokehityksen tietokantapalvelimena toimi MySQL-versio 5.7.19, ja verkkopalvelimena operoi Apache-versio 2.4.27. Sovelluksen ja palvelimen välistä kommunikointia ohjasi komentosarjakieli PHP:llä koodatut skriptit. PHP:n versio oli 5.6.31. Ohjelmistokehityksen tietokantojen hallinta tapahtui phpMyAdmin-ympäristössä, jonka versio oli kehitysvaiheessa 4.7.4.

4.4 Ohjelmointiympäristöön tutustuminen

Ohjelmistokehityksen ohjelmointiympäristönä toimi Androidille yksinomaaisesti kehitetty Android Studio. Kyseisestä ohjelmointiympäristöstä ei ollut aikaisempaa käyttökokemusta, ja siihen tutustuminen vei aikaa noin viiden työpäivän verran. Android Studion ulkoasu ja toiminnallisuus kuitenkin muistuttavat vahvasti ennestään tuttuja ohjelmistokehitykseen tarkoitettuja ohjelmointiympäristöjä, kuten IntelliJ IDEA ja Microsoft Visual Studio.

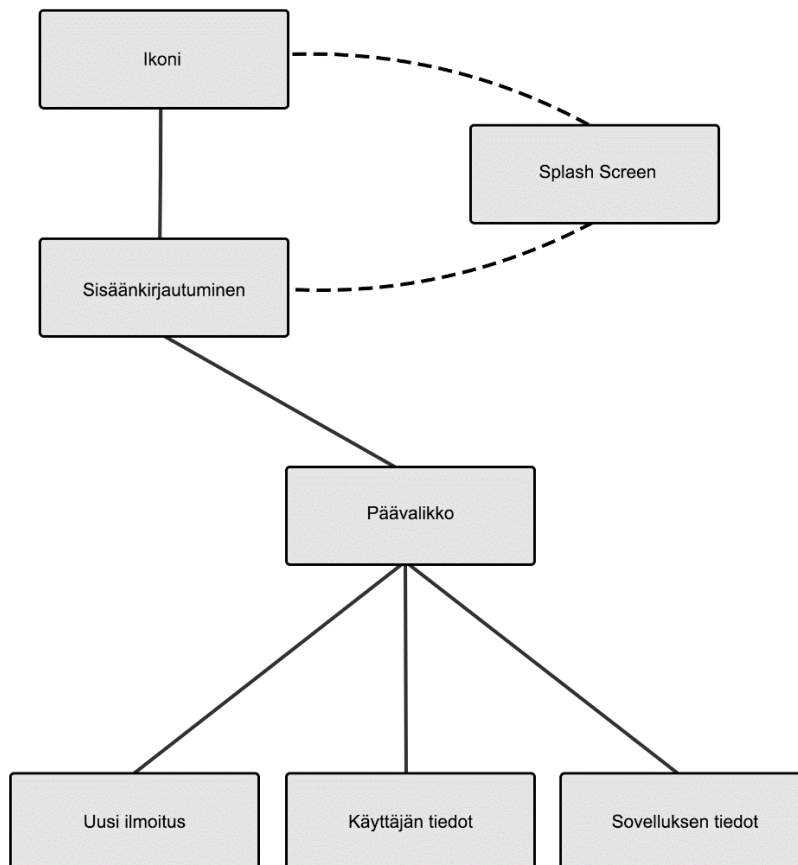
4.5 Mobiilisovelluksen rakenne ja kehitystyö

Mobiilisovelluksen rakennetta suunnitellessa oli tärkeää ensin ymmärtää, miten kyseiset sovellukset rakentuvat. Android-sovellukset käyttävät aktiviteeteiksi (activity) kutsuttavia näkymiä, jotka ovat ikään kuin sovelluksen välilehtiä. Mobiilisovellus käynnistyy, kun käyttäjä painaa sovelluksen ikonin. Yleinen käynnistymisen käytäntö Android mobiilisovelluksilla on aloittaa aktiviteetista nimeltä pääaktiviteetti (MainActivity). Sen jälkeen on tapana lisätä erilaisia painikkeita, joihin ohjelmoidaan ohjaus seuraavaan painikkeeseen yhdistettyyn aktiviteettiin. Tätä ajatusmaailmaa sovelletaan läpi koko sovellushierarkian. Toiseen aktiviteettiin ohjauksen ei tarvitse aina tapahtua painikkeesta, vaan sen voi myös koodata tapahtuvan esimerkiksi jonkin tietyn ehtolauseen toteutuessa.

Koska ilmoitusohjelmistolle kehitettävä mobiilisovellus suunniteltiin usean eri käyttäjän käyttöön, oli luonnollista kehittää ensimmäisestä eli sovelluksen aloittavasta pääaktiviteetista sisäänkirjautumisen näkymä. Kun käyttäjä kirjautuu sovellukseen onnistuneesti sisään, ohjaa "KIRJAUDU"-painike käyttäjän seuraavaksi päävalikkoaktiviteettiin (MenuActivity). Päävalikosta käyttäjä voi siirtyä sovelluksen asetuksiin, varsinaiseen toimintoon eli ilmoituksen tekoon tai kirjautua ulos. Asetukset- ja Uusi Ilmoitus-valinnat ohjaavat käyttäjän taas uusiin niille tarkoitettuihin aktiviteetteihin (SettingsActivity ja FormActivity), kun taas ulos kirjautuminen vie käyttäjän takaisin sisäänkirjautumisen aktiviteettiin.

Ensimmäinen suunnitelma oli asettaa suoraan päävalikkoon kolme painiketta; uusi ilmoitus, käyttäjän tiedot ja sovelluksen tiedot (Kuva 10). Kehitystyön edetessä päävalikkoon kuitenkin päätyi vain vaihtoehto uusi ilmoitus, koska se oli sovelluksen varsinainen ominaisuus. Käyttäjän ja sovelluksen tiedot sijoitettiin ylävalikon lisätoimintoihin ja asetukset-

aktiviteettiin. Myös alkuperäisen suunnitelman mukainen splash screen-latausnäkyvä jäi ajanpuutteen vuoksi kehittämättä. Sen tarkoitus olisi korvata sovelluksen käynnistyksen yhteydessä näkyvä musta ruutu, jolloin käyttäjä tietää käynnistäneensä sovelluksen.




Kuva 10. Mobiilisovelluksen ensimmäinen rakennesuunnitelma.

4.5.1 Tietokannan luominen

Ennen kuin varsinaista mobiilisovellusta alettiin kehittämään, täytyi ohjelmistolle luoda tietokanta, johon sovelluksen käyttäjien tiedot tallennetaan. Kun mobiilisovelluksen rakenne ja halutut toiminnot oli suunniteltu huolella etukäteen, oli helppoa viedä aiempien suunnitelmien rakenne myös tietokantoihin.

Ohjelmistokehityksessä käytetylle MySQL-tietokantapalvelimelle luotiin uusi tietokanta, jolle annettiin nimeksi android. Tähän tietokantaan luotiin kolme taulukkoa; käyttäjät, data ja kuvat. Kuvassa 11 esitetään data-taulukon luomisessa käytetty SQL-kysely. Taulukko käyttäjät pitää sisällään tiedot rekisteröityneistä käyttäjätunnuksista. Data taulukkoon tallennetaan käyttäjien ilmoituksissa täyttämät tiedot, ja kuvat-taulukkoon vietään tietoa käyttäjien palvelimelle lataamista valokuvista.

Suorita SQL-kyselyjä tietokannassa android: 

```

1 CREATE TABLE `data` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `aihe` varchar(100) NOT NULL,
4   `otsikko` varchar(100) NOT NULL,
5   `kommentti` varchar(9999) NOT NULL,
6   `tapahtumapvm` varchar(100) NOT NULL,
7   `esimies` varchar(100) NOT NULL,
8   PRIMARY KEY (`id`)
9 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1

```

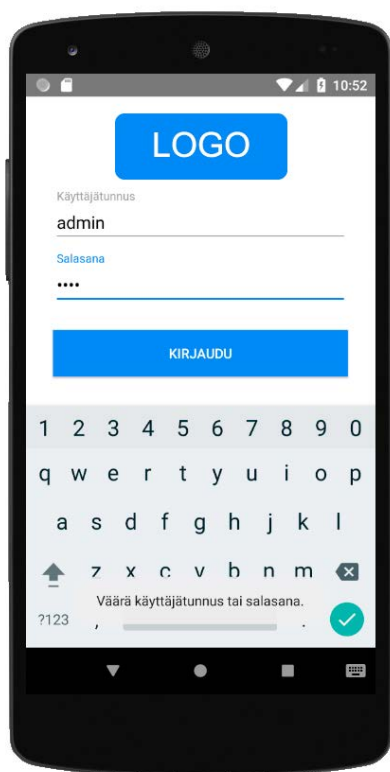
Kuva 11. Data taulukon luomisessa käytetty SQL-kysely.

4.5.2 Käyttäjän rekisteröityminen ja sisäänkirjautuminen

Mobiilisovelluksessa käyttäjällä ei ole mahdollisuutta luoda uusia käyttäjätunnuksia, eli rekisteröityä. Rekisteröinti tapahtuu sovelluksen ulkopuolisella PHP-skriptillä, jota sovelluksen tietokantapalvelinta hallitseva henkilö voi käyttää. Tällä toimenpiteellä on tarkoitus estää mobiilisovelluksen käyttö henkilöiltä, joilla ei ole sopimusta sovelluksen käytöstä sen omistajan kanssa. Sovellus on tarkoitettu ladata Google Play -kauppaan mistä sen voi kuka tahansa ladata, mutta käyttäjätunnuksen saa vasta käyttösopimuksen teon yhteydessä.

Rekisteröitymisprosessi tapahtuu siis edellä mainitun PHP-skriptin avulla, jossa määritetään uuden käyttäjätunnuksen rekisteröitymiseen tarvittavat tiedot. Sen tarkempia yksityiskohtia rekisteröitymisprosessista ei käsitellä käyttäjätietojen turvallisuuden vuoksi.

Kun mobiilisovellus avataan mobiililaitteella sovelluksen ikonia painamalla, aukeaa käyttäjälle sisäänkirjautumisnäkyvä (Kuva 12). Näkyvä sisältää organisaation logon, tekstikentät käyttäjätunnukseksi ja salasanalle sekä kirjautumispainikkeen. Painiketta painettaessa sovellus tarkistaa ensin tietokannasta ovatko täytetyt tiedot (käyttäjätunnus ja salasana) oikein, ja luo sitten yhteyden palvelimeen. Jos täytetyissä tiedoissa on virheitä, sovellus ilmoittaa siitä ponnahdusikkunalla eikä kirjautuminen onnistu. Jos kirjautumistiedot ovat oikein, käyttäjä kirjautuu sovellukseen ja siirtyy automaattisesti päävalikkoaktiiviteettiin. Lisäksi sovellus muistaa sisäänkirjautumisen kyseisessä laitteessa niin kauan, kunnes käyttäjä itse kirjautuu ulos. Näin ollen kirjautumista ei tarvita jokaisella mobiilisovelluksen käynnistyskerralla, mikä tekee sovelluksen käytöstä nopeampaa.



Kuva 12. Mobiilisovelluksen kirjautumisenäkymä ja ponnahdusikkunaviesti.

4.5.3 Käyttöliittymä ja ulkoasu

Sovelluksen käyttöliittymästä haluttiin helppo ja yksinkertainen. Käyttöliittymässä noudatettiin pitkälti yleisiä tekniikoita, joihin mobiililaitteiden käyttäjät ovat jo tottuneet suosituimpien sovellusten parissa. Sovelluksen painikkeet aseteltiin johdonmukaisesti paikkoihin, ja ylimääräisiä ärsykeitä tai painikkeita ei ole.

Päävalikon jäsentely kuvastaa hyvin koko sovelluksen käyttöliittymän painikkeiden asettelua (Kuva 13). Yläpalkin otsikkoteksti kertoo nykyisen aktiviteetin, yläpalkin vasemman reunan nuoli toimii takaisin painikkeena ja yläpalkin oikean reunan pisteet avaavat listan vähemmän käytettävistä ominaisuuksista, kuten asetukset tai uloskirjautuminen. Näytön suuri keskialue on tarkoitettu sovelluksen pääominaisuuksille, eli tässä tapauksessa uuden ilmoituksen tekemiselle. Mobiilisovelluksen ulkoasu haluttiin pitää siistinä ja selkeänä. Tekstiosioissa pyrittiin käyttämään mahdollisimman paljon tummaa tekstiä vaalealla pohjalla. Ulkoasun kehityksessä oli myös otettava huomioon sovelluksen taittuminen eri kokoisille resoluutioille. Koska vanhemmissa älypuhelimissa saatetaan käyttää jopa puolet pienempää resoluutiota kuin useimmissa älypuhelimissa, tuli jokaista resoluutiota testata jokaisessa valikossa. Android Studio emulaattori työkalut tekivät eri resoluutioiden testauksesta kuitenkin suhteellisen vaivattonta.



Kuva 13. Päävalikko ja yläpalkista aukeava lisävalikko.

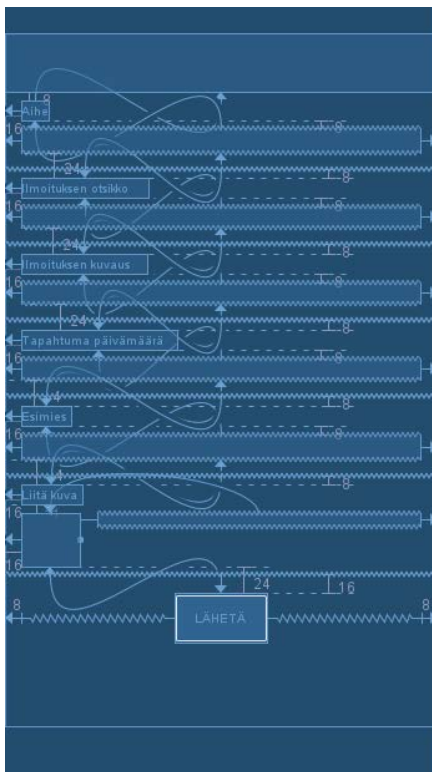
Ulkoasun taustakuvat logoineen tuotettiin Adoben Photoshop-kuvamuokkausohjelmistolla. Mobiilisovelluksen värimaailmaksi ei haluttu mitään liian moniväristä ja räikeää, ja siksi ulkoasun teemaksi muodostui yksivärinen järjestelmä. Yksivärisellä järjestelmällä tarkoitetaan värikokoelmaa, jossa kaikki värit ovat yhden perusvärin alaisia sävyjä (Refsnes Data 2018). Tässä projektissa perusväriksi valittiin sininen. Sinisen useiden eri sävyjen lisäksi mobiilisovelluksen ulkoasussa käytettiin ainoastaan mustaa, valkoista ja harmaata (Kuva 14).



Kuva 14. Mobiilisovelluksen ulkoasussa käytetyt sinisen eri värisävyt HTML-värikoodeineen.

4.5.4 Uusi ilmoitus ja tiedonsiirto

Mobiilisovelluksen pääominaisuus, eli uusi ilmoitus tapahtuu selkeällä lomakkeen täyttämällä. Ominaisuus aukeaa sovelluksen päävalikosta painamalla Uusi Ilmoitus-painiketta. Ilmoituksen lomakkeen rakenne muodostui yleisen ilmoituslomakemallin pohjalta. Lomakkeen rakenne sisältää tekstikentät ilmoituksen otsikolle ja kuvaukselle, pudotuslistat aiheelle ja esimiehen valinnalle, numerokentän päivämäärälle sekä selkeät painikkeet valokuvan lisäämiselle ja ilmoituksen lähettämiseksi (Kuva 15). Rakenne on yleispätevä ja sopii kaikenlaisten organisaatioiden käytettäväksi.



Kuva 15. Uuden ilmoituksen pohjapiirros.

Kun käyttäjä on täyttänyt ilmoituksen ja painaa lähetä-painiketta, siirtää mobiilisovellus täytetyt tiedot tietokantapalvelimelle. Tiedot siirtyvät salasanasuojatun yhteyden yli android-tietokannan data taulukkoon, jossa ne sijoitetaan PHP-skriptin ohjaamana omaan tietueeseensa. Tietueessa on jokaiselle ilmoituksen lomakkeen tietokentälle omat sarakkeet, joista tietoa on kehittäjien helppo käyttää jatkokehityksessä.

4.5.5 Kamera ja tiedostonsiirto

Ohjelmistokehityksen haastavin vaihe oli tuoda mobiililaitteiden kamera mukaan mobiilisovellukseen. Android-sovelluksissa täytyy ensin määrittää sovelluksen manifestiin (AndroidManifest.xml) parametri, joka kertoo käyttöjärjestelmälle sovelluksen sisältävän kameralle tarkoitettuja ominaisuuksia (Kuva 16). Kun parametri on lisätty manifestiin, osaa mobiilisovellus automaattisesti kysyä mobiililaitteen käyttäjältä lupaa kyseisen kamerasen käyttöön. Sovelluksen kameratoiminnot eivät voi käynnistyä ennen kuin käyttäjä on antanut sovellukselle luvan käyttää kameraa.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-feature android:name="android.hardware.camera"
  android:required="false" />
```

Kuva 16. Osa AndroidManifest.xml -tiedoston käyttäjälupamäärittämisistä.

Kun käyttäjä painaa kamera painiketta ilmoituksen lomakkeessa, käynnistää se Android-laitteen oman kameraympäristön. Käyttäjä ottaa ilmoitukseensa liittyvän valokuvan, ja hyväksyy ottamansa kuvan painamalla hyväksy painiketta. Siinä hetkessä mobiilisovellus luo käyttäjän mobiililaitteeseen otetusta valokuvasta uuden JPEG-kuvatiedoston, ja nimeää sen kuvanotto hetken mukaisesti. Nimeämisprosessi on toteutettu mobiilisovelluksen koodissa automatisoidulla skriptillä. Valokuvan hyväksytyään käyttäjä palaa takaisin ilmoitusnäkömään. Käyttäjä voi havaita valokuvan liittämisen ilmoitukseen onnistuneen, kun kuvatiedoston tiedostopolku on ilmestynyt lomakkeen loppuosaan.

Ilmoituksen ollessa valmis ja käyttäjän painaessa lähetä-painiketta, mobiilisovellus käynnistää www-palvelimella sijaitsevan PHP-skriptin. Tämä skripti koodaa mobiilisovellukseen liitetyn kuvatiedoston base64-muotoon, joka voidaan siinä muodossa helposti ohjata skriptin avulla kopioitavaksi mobiililaitteelta palvelimen hakemistoon (Kuva 17). Kuvatiedosto säilyttää koodaus- ja kopioimisprosessissa laatunsa, nimensä ja tiedostomuotonsa.

```
$encoded_string = $_POST["encoded_string"];
$image_name = $_POST["image_name"];
...
$decoded_string = base64_decode($encoded_string);

$path = 'images/' . $image_name;

$file = fopen($path, 'wb');

$is_written = fwrite($file, $decoded_string);
fclose($file);
```

Kuva 17. Osa kuvatiedoston kopiointiprosessin PHP-skriptistä.

4.6 Siirto toimeksiantajan palvelimelle

Vajaan viiden viikon ohjelmistokehityksen jälkeen mobiilisovellus oli suunnitelmien mukaisesti asetettujen tavoitteiden tasolla. Toimeksiantajan tietokantapalvelimelle luotiin mobiilisovelluksen vaatima tietokantarakenne. Tietokannan ja taulukoiden nimitykset olivat aiemmin paikallisella palvelimelläni käyttämistä nimityksistä poikkeavia. Lisäksi mobiilisovelluksen tietokantayhteyksien skriptit ja osoitteet muokattiin vastaamaan toimeksiantajan palvelimen tietoja, jotta sovellus osaisi jatkossa muodostaa yhteydet toimeksiantajan palvelimelle kehityksessä käytetyn paikallisen palvelimen sijaan. Skriptitiedostot siirrettiin myös toimeksiantajan määrittämään hakemistoon.

Siirtoprosessin aikana toimeksiantajan tietokantapalvelimen ja ohjelmistokehityksessä käytetyn tietokantapalvelimen välillä havaittiin pieniä

eroavaisuuksia. Tietokantapalvelinten versiot eivät olleet täsmälleen samat, ja siksi palvelimilla täytyi suorittaa pieniä päivityksiä. Nämä eroavaisuudet aiheuttivat mobiilisovelluksessa toiminnallisuushäiriöitä, jotka eivät mahdollistaneet sovelluksen käyttöä. Tietokantapalvelimien päivityksien jälkeen sovellus alkoi kuitenkin toimia toimeksiantajan palvelinympäristössä halutulla tavalla.

Tietokannan luomisen, yhteyksien muokkausten ja toiminnallisuusongelmien ratkaisujen jälkeen mobiilisovellukseen luotiin onnistuneesti toimeksiantajalle testikäyttäjätunnukset. Mobiilisovellus ladattiin kehityslaitteistosta useampien henkilöiden mobiililaitteisiin, ja se annettiin useamman heidän testattavaksi. Sovelluksen todettiin toimivan alkuperäisten suunnitelmien sekä tavoitteiden mukaisesti, ja ohjelmistokehityksen lopputulos oli valmis toimeksiantajan omalle jatkokehitykselle.

5 YHTEENVETO

Kehitysprojekti oli kokonaisuudessaan erittäin onnistunut. Projektin alkuvaiheessa huolellisesti laaditut suunnitelmat antoivat ohjelmistokehityksen suorittamiselle selkeät kehitysohjeet. Ohjelmistokehityksen aikataulutuksen laatiminen viikonmittaisiin vaiheisiin oli helppoa, ja laaditussa aikataulussa pysyttiin suhteellisen hyvin läpi projektin. Mobiilisovellukseen ohjelmoitu koodi koettiin hyväksi ja puhtaaksi koko sovelluksen osalta, ja sen rakenteesta muodostui erittäin selkeä.

Mikäli kehitysprojektille olisi ollut enemmän aikaa, olisi mobiilisovelluksen toimintojen kehitystyön jälkeen varattu lisää työtunteja sovelluksen käyttöliittymän ja erityisesti ulkoasun kehitykselle. Mobiilisovelluksen ulkoasusta muodostui selkeä ja toimiva, mutta erilaisten graafisten osien lisäysten avulla siitä olisi saatu enemmän ammattimaisen tuotteen näköinen. Myös ohjelmistokehityksessä käytettyjen palvelinten yhteensopi- vuus olisi kannattanut tarkistaa jo kehitysprojektin alkuvaiheilla.

Sovelluksen lopputulos tarjoaa erittäin toimivan kokonaisuuden, jolla pystytään toteuttamaan kaikki sille etukäteen asetetut toiminnalliset tavoitteet. Lisäksi perusteellisesti suoritettu ohjelmistokehitys mahdollistaa mobiilisovelluksen kattavan jatkokehityksen tulevaisuudessa. Kokonaisuudessaan kehitysprojekti oli mielenkiintoinen, ja opetti paljon uusia tai- toja koskien mobiilisovelluksien ohjelmistokehitystä, kokonaisvaltaista suunnittelua sekä ohjelmointia Java ja PHP-ohjelmointikielillä.

LÄHTEET

Android Developers (2018). Android Studio Intro. Haettu 26.4.2018 osoitteesta <https://developer.android.com/studio/intro/>

OnePlus (2018). OnePlus 3. Haettu 26.4.2018 osoitteesta <https://www.oneplus.com/fi/3>

Postdot Technologies, Inc. (2018). Company. Haettu 21.4.2018 osoitteesta <https://www.getpostman.com/company>

Q-Success (2018). Usage of web servers for websites. Haettu 21.4.2018 osoitteesta https://w3techs.com/technologies/overview/web_server/all

Refsnes Data (2018). Monochromatic Colors. Haettu 24.4.2018 osoitteesta https://www.w3schools.com/colors/colors_monochromatic.asp

StatCounter Global Stats (2018 a). Mobile Operating System Market Share Yearly 2017. Haettu 26.4.2018 osoitteesta <http://gs.statcounter.com/os-market-share/mobile/worldwide#yearly-2017-2017-bar>

StatCounter Global Stats (2018 b). Operating System Market Share Yearly 2017. Haettu 26.4.2018 osoitteesta <http://gs.statcounter.com/os-market-share/desktop/worldwide#yearly-2017-2017-bar>

StatCounter Global Stats (2018 c). Browser Market Share Yearly 2017. Haettu 26.4.2018 osoitteesta <http://gs.statcounter.com/browser-market-share#yearly-2017-2017-bar>

The PHP Group (2018). General Information. Haettu 26.4.2018 osoitteesta <http://fi2.php.net/manual/en/faq.general.php#faq.general.what>

Wikipedia (2016). Notepad++. Haettu 21.4.2018 osoitteesta <https://fi.wikipedia.org/wiki/Notepad%2B%2B>

Wikipedia (2017). MySQL. Haettu 21.4.2018 osoitteesta <https://fi.wikipedia.org/wiki/MySQL>

Wikipedia (2018 a). Android. Haettu 21.4.2018 osoitteesta <https://fi.wikipedia.org/wiki/Android>

Wikipedia (2018 b). Java. Haettu 21.4.2018 osoitteesta <https://fi.wikipedia.org/wiki/Java>

Wikipedia (2018 c). Adobe Photoshop. Haettu 21.4.2018 osoitteesta https://fi.wikipedia.org/wiki/Adobe_Photoshop

ANDROID OHJELMISTOKEHITYSPAKKAUKSEN LISENSISOPIMUS

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, as updated from time to time.

1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (<http://source.android.com/compatibility>) and which may be updated from time to time; and (ii) successfully passes the Android Compatibility Test Suite (CTS).

1.4 "Google" means Google LLC, a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

2. Accepting this License Agreement

2.1 In order to use the SDK, you must first agree to the License Agreement. You may not use the SDK if you do not accept the License Agreement.

2.2 By clicking to accept, you hereby agree to the terms of the License Agreement.

2.3 You may not use the SDK and may not accept the License Agreement if you are a person barred from receiving the SDK under the laws of the United States or other countries, including the country in which you are resident or from which you use the SDK.

2.4 If you are agreeing to be bound by the License Agreement on behalf of your employer or other entity, you represent and warrant that you have full legal authority to bind your employer or such entity to the License Agreement. If you do not have the requisite authority, you may not

accept the License Agreement or use the SDK on behalf of your employer or other entity.

3. SDK License from Google

3.1 Subject to the terms of the License Agreement, Google grants you a limited, worldwide, royalty-free, non-assignable, non-exclusive, and non-sublicensable license to use the SDK solely to develop applications for compatible implementations of Android.

3.2 You may not use this SDK to develop applications for other platforms (including non-compatible implementations of Android) or to develop another SDK. You are of course free to develop applications for other platforms, including non-compatible implementations of Android, provided that this SDK is not used for that purpose.

3.3 You agree that Google or third parties own all legal right, title and interest in and to the SDK, including any Intellectual Property Rights that subsist in the SDK. "Intellectual Property Rights" means any and all rights under patent law, copyright law, trade secret law, trademark law, and any and all other proprietary rights. Google reserves all rights not expressly granted to you.

3.4 You may not use the SDK for any purpose not expressly permitted by the License Agreement. Except to the extent required by applicable third party licenses, you may not copy (except for backup purposes), modify, adapt, redistribute, decompile, reverse engineer, disassemble, or create derivative works of the SDK or any part of the SDK.

3.5 Use, reproduction and distribution of components of the SDK licensed under an open source software license are governed solely by the terms of that open source software license and not the License Agreement.

3.6 You agree that the form and nature of the SDK that Google provides may change without prior notice to you and that future versions of the SDK may be incompatible with applications developed on previous versions of the SDK. You agree that Google may stop (permanently or temporarily) providing the SDK (or any features within the SDK) to you or to users generally at Google's sole discretion, without prior notice to you.

3.7 Nothing in the License Agreement gives you a right to use any of Google's trade names, trademarks, service marks, logos, domain names, or other distinctive brand features.

3.8 You agree that you will not remove, obscure, or alter any proprietary rights notices (including copyright and trademark notices) that may be affixed to or contained within the SDK.

4. Use of the SDK by You

4.1 Google agrees that it obtains no right, title or interest from you (or your licensors) under the License Agreement in or to any software applications that you develop using the SDK, including any intellectual property rights that subsist in those applications.

4.2 You agree to use the SDK and write applications only for purposes that are permitted by (a) the License Agreement and (b) any applicable law, regulation or generally accepted practices or guidelines in the relevant jurisdictions (including any laws regarding the export of data or software to and from the United States or other relevant countries).

4.3 You agree that if you use the SDK to develop applications for general public users, you will protect the privacy and legal rights of those users. If the users provide you with user names, passwords, or other login information or personal information, you must make the users aware that the information will be available to your application, and you must provide legally adequate privacy notice and protection for those users. If your application stores personal or sensitive information provided by users, it must do so securely. If the user provides your application with Google Account information, your application may only use that information to access the user's Google Account when, and for the limited purposes for which, the user has given you permission to do so.

4.4 You agree that you will not engage in any activity with the SDK, including the development or distribution of an application, that interferes with, disrupts, damages, or accesses in an unauthorized manner the servers, networks, or other properties or services of any third party including, but not limited to, Google or any mobile communications carrier.

4.5 You agree that you are solely responsible for (and that Google has no responsibility to you or to any third party for) any data, content, or resources that you create, transmit or display through Android and/or applications for Android, and for the consequences of your actions (including any loss or damage which Google may suffer) by doing so.

4.6 You agree that you are solely responsible for (and that Google has no responsibility to you or to any third party for) any breach of your obligations under the License Agreement, any applicable third party contract or Terms of Service, or any applicable law or regulation, and for the consequences (including any loss or damage which Google or any third party may suffer) of any such breach.

5. Your Developer Credentials

5.1 You agree that you are responsible for maintaining the confidentiality of any developer credentials that may be issued to you by Google or

which you may choose yourself and that you will be solely responsible for all applications that are developed under your developer credentials.

6. Privacy and Information

6.1 In order to continually innovate and improve the SDK, Google may collect certain usage statistics from the software including but not limited to a unique identifier, associated IP address, version number of the software, and information on which tools and/or services in the SDK are being used and how they are being used. Before any of this information is collected, the SDK will notify you and seek your consent. If you withhold consent, the information will not be collected.

6.2 The data collected is examined in the aggregate to improve the SDK and is maintained in accordance with Google's Privacy Policy.

7. Third Party Applications

7.1 If you use the SDK to run applications developed by a third party or that access data, content or resources provided by a third party, you agree that Google is not responsible for those applications, data, content, or resources. You understand that all data, content or resources which you may access through such third party applications are the sole responsibility of the person from which they originated and that Google is not liable for any loss or damage that you may experience as a result of the use or access of any of those third party applications, data, content, or resources.

7.2 You should be aware the data, content, and resources presented to you through such a third party application may be protected by intellectual property rights which are owned by the providers (or by other persons or companies on their behalf). You may not modify, rent, lease, loan, sell, distribute or create derivative works based on these data, content, or resources (either in whole or in part) unless you have been specifically given permission to do so by the relevant owners.

7.3 You acknowledge that your use of such third party applications, data, content, or resources may be subject to separate terms between you and the relevant third party. In that case, the License Agreement does not affect your legal relationship with these third parties.

8. Using Android APIs

8.1 Google Data APIs

8.1.1 If you use any API to retrieve data from Google, you acknowledge that the data may be protected by intellectual property rights which are owned by Google or those parties that provide the data (or by other

persons or companies on their behalf). Your use of any such API may be subject to additional Terms of Service. You may not modify, rent, lease, loan, sell, distribute or create derivative works based on this data (either in whole or in part) unless allowed by the relevant Terms of Service.

8.1.2 If you use any API to retrieve a user's data from Google, you acknowledge and agree that you shall retrieve data only with the user's explicit consent and only when, and for the limited purposes for which, the user has given you permission to do so.

9. Terminating this License Agreement

9.1 The License Agreement will continue to apply until terminated by either you or Google as set out below.

9.2 If you want to terminate the License Agreement, you may do so by ceasing your use of the SDK and any relevant developer credentials.

9.3 Google may at any time, terminate the License Agreement with you if: (A) you have breached any provision of the License Agreement; or (B) Google is required to do so by law; or (C) the partner with whom Google offered certain parts of SDK (such as APIs) to you has terminated its relationship with Google or ceased to offer certain parts of the SDK to you; or (D) Google decides to no longer provide the SDK or certain parts of the SDK to users in the country in which you are resident or from which you use the service, or the provision of the SDK or certain SDK services to you by Google is, in Google's sole discretion, no longer commercially viable.

9.4 When the License Agreement comes to an end, all of the legal rights, obligations and liabilities that you and Google have benefited from, been subject to (or which have accrued over time whilst the License Agreement has been in force) or which are expressed to continue indefinitely, shall be unaffected by this cessation, and the provisions of paragraph 14.7 shall continue to apply to such rights, obligations and liabilities indefinitely.

10. DISCLAIMER OF WARRANTIES

10.1 YOU EXPRESSLY UNDERSTAND AND AGREE THAT YOUR USE OF THE SDK IS AT YOUR SOLE RISK AND THAT THE SDK IS PROVIDED "AS IS" AND "AS AVAILABLE" WITHOUT WARRANTY OF ANY KIND FROM GOOGLE.

10.2 YOUR USE OF THE SDK AND ANY MATERIAL DOWNLOADED OR OTHERWISE OBTAINED THROUGH THE USE OF THE SDK IS AT YOUR OWN DISCRETION AND RISK AND YOU ARE SOLELY RESPONSIBLE FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR OTHER DEVICE OR LOSS OF DATA THAT RESULTS FROM SUCH USE.

10.3 GOOGLE FURTHER EXPRESSLY DISCLAIMS ALL WARRANTIES AND CONDITIONS OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

11. LIMITATION OF LIABILITY

11.1 YOU EXPRESSLY UNDERSTAND AND AGREE THAT GOOGLE, ITS SUBSIDIARIES AND AFFILIATES, AND ITS LICENSORS SHALL NOT BE LIABLE TO YOU UNDER ANY THEORY OF LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES THAT MAY BE INCURRED BY YOU, INCLUDING ANY LOSS OF DATA, WHETHER OR NOT GOOGLE OR ITS REPRESENTATIVES HAVE BEEN ADVISED OF OR SHOULD HAVE BEEN AWARE OF THE POSSIBILITY OF ANY SUCH LOSSES ARISING.

12. Indemnification

12.1 To the maximum extent permitted by law, you agree to defend, indemnify and hold harmless Google, its affiliates and their respective directors, officers, employees and agents from and against any and all claims, actions, suits or proceedings, as well as any and all losses, liabilities, damages, costs and expenses (including reasonable attorneys fees) arising out of or accruing from (a) your use of the SDK, (b) any application you develop on the SDK that infringes any copyright, trademark, trade secret, trade dress, patent or other intellectual property right of any person or defames any person or violates their rights of publicity or privacy, and (c) any non-compliance by you with the License Agreement.

13. Changes to the License Agreement

13.1 Google may make changes to the License Agreement as it distributes new versions of the SDK. When these changes are made, Google will make a new version of the License Agreement available on the website where the SDK is made available.

14. General Legal Terms

14.1 The License Agreement constitutes the whole legal agreement between you and Google and governs your use of the SDK (excluding any services which Google may provide to you under a separate written agreement), and completely replaces any prior agreements between you and Google in relation to the SDK.

14.2 You agree that if Google does not exercise or enforce any legal right or remedy which is contained in the License Agreement (or which Google has the benefit of under any applicable law), this will not be taken to be a

formal waiver of Google's rights and that those rights or remedies will still be available to Google.

14.3 If any court of law, having the jurisdiction to decide on this matter, rules that any provision of the License Agreement is invalid, then that provision will be removed from the License Agreement without affecting the rest of the License Agreement. The remaining provisions of the License Agreement will continue to be valid and enforceable.

14.4 You acknowledge and agree that each member of the group of companies of which Google is the parent shall be third party beneficiaries to the License Agreement and that such other companies shall be entitled to directly enforce, and rely upon, any provision of the License Agreement that confers a benefit on (or rights in favor of) them. Other than this, no other person or company shall be third party beneficiaries to the License Agreement.

14.5 EXPORT RESTRICTIONS. THE SDK IS SUBJECT TO UNITED STATES EXPORT LAWS AND REGULATIONS. YOU MUST COMPLY WITH ALL DOMESTIC AND INTERNATIONAL EXPORT LAWS AND REGULATIONS THAT APPLY TO THE SDK. THESE LAWS INCLUDE RESTRICTIONS ON DESTINATIONS, END USERS AND END USE.

14.6 The rights granted in the License Agreement may not be assigned or transferred by either you or Google without the prior written approval of the other party. Neither you nor Google shall be permitted to delegate their responsibilities or obligations under the License Agreement without the prior written approval of the other party.

14.7 The License Agreement, and your relationship with Google under the License Agreement, shall be governed by the laws of the State of California without regard to its conflict of laws provisions. You and Google agree to submit to the exclusive jurisdiction of the courts located within the county of Santa Clara, California to resolve any legal matter arising from the License Agreement. Notwithstanding this, you agree that Google shall still be allowed to apply for injunctive remedies (or an equivalent type of urgent legal relief) in any jurisdiction.