



TAMPEREEN
AMMATTIKORKEAKOULU

MOBIILIKEHITYS

Eri alustat ja niiden erot

Santeri Hetekivi

Opinnäytetyö
Huhtikuu 2017
Tietotekniikka, Insinööri (AMK)
Ohjelmistotekniikka



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka, Insinööri (AMK)
Ohjelmistotekniikka

HETEKIVI SANTERI:
Mobiilikehitys
Eri alustat ja niiden erot

Opinnäytetyö 37 sivua, joista liitteitä 0 sivua
Huhtikuu 2017

Tämän opinnäytetyön aiheena oli mobiilikehitys kaikilla yleisimmillä alustoilla ja niiden erot. Työn tarkoituksena oli selvittää lukijalle millä ohjelmointitapa ja -alusta yhdistelmällä hänen kannattaisi ruveta ohjelmoimaan mobiilisovellustaan. Työssä käytiin läpi Android, iOS, ja Windows Phone / UWP käyttöjärjestelmät ja näille käyttöjärjestelmille kehitetyt mobiililaitteet.

Työssä selvitettiin ero natiiveilla- ja hybridikehitystavoilla. Käytiin läpi myös yleisimpiä ohjelmointikieliä mobiilikehitykseen, joihin lukeutuvat Java, Swift, Objective-C, C# ja JavaScript.

Kehitettiin kaksi mobiilisovellusta Epic Invoicing Oy:lle. Toinen täysin natiiveilla tavoilla, jokaiselle alustalle omansa ja näiden omilla työkaluilla ja toinen täysin hybridinä.

Nettilasku oli täysin natiivi mobiilisovellus, joka kehitettiin iOS alustalle Swift, Androidille Java ja Windows Phonelle C# ohjelmointikielellä. Projektissa käytettiin käyttöjärjestelmävalmistajien omia työkaluja. iOS-alustalla ohjelmoitiin Xcode-kehitysympäristössä, Androidille Android Studiolla ja Windows Phonelle Microsoftin Visual Studio -kehitysympäristössä.

Nettikuitti.fi oli hybridi mobiilisovellus, joka kehitettiin täysin Microsoftin Xamarin monialusta kehitysalustalla. Ohjelmointikielenä käytettiin C# ohjelmointikieltä, jolloin voitiin käyttää osaa Nettilasku sovelluksen Windows Phone -koodikannasta.

Nettikuitti.fi projekti saatiin toteutettua täysin ilman laitekohtaisia kirjastoja. Tämä auttaa tulevaisuudessa, kun tehdään toimintoja ja korjauksia, koska tarvitsee muuttaa vain yhtä koodikantaa. Nettilasku projektissa joutui tekemään kaikki muutokset kolmella eri kielellä kolmelle eri alustalle, joka vei aikaa ja näin myös yrityksen rahoja.

Lopputuloksena todetaan, että mobiilikehitys kannattaa tehdä hybridi kehitys tavalla. Kehittäessäsi vain mobiilisovelluksen Xamarin on hyvä työkalu, web-sovelluksen muuttamisessa mobiilisovellukseksi ja React tai Angular kehitys React Native ja NativeScript ovat vahvoja työkaluja. Näin jokainen muutos voidaan tehdä vain kerran ja taataan että kaikilla alustoilla sovelluksesi sisältää samat ominaisuudet.

Asiasanat: ohjelmointi, mobiili, Android, iOS, UWP, natiivi, hybridi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Information Technology, Bachelor of Engineering (Polytechnic)
Software engineering

HETEKIVI SANTERI:
Mobile development
Different platforms and their differences

Bachelor's thesis 37 pages, appendices 0 pages
April 2017

The subject of this thesis was the development for all the major mobile platforms and their differences. Thesis's purpose was to investigate which programming type and platform combination would be best for the reader to start to program the mobile applications. The thesis went through Android, iOS, and Windows Phone / UWP operating systems and devices developed for these operating systems.

The thesis studied the difference between native and hybrid development methods. It went through the most common programming languages for mobile development, which include Java, Swift, Objective-C, C #, and JavaScript.

There were developed two mobile apps for Epic Invoicing Oy. One with completely native ways with one app for each platform with their own tools and another fully hybrid.

Nettilasku was fully native mobile application, which was developed in the iOS platform with Swift, Android with Java and for Windows Phone with C # programming language. In the project only manufacturers' own operating system tools were used. iOS platform was programmed in Xcode development environment, Android in Android Studio and Windows Phone in Microsoft's Visual Studio development environment.

Nettikuitti.fi was a hybrid mobile application, which was developed entirely by Microsoft's Xamarin multi-platform development platform. The programming language used was the C # programming language and that allowed use of parts of Nettilasku app's Windows Phone code base.

Nettikuitti.fi project was implemented without any device-specific libraries. This will help in the future when making development and repairs, because the need of the only one code base. Nettilasku project had to make any changes in three different languages on three different platforms, which took time and thus also the company's money.

The result indicates that mobile development should be done hybrid manner. When developing a mobile application Xamarin is a good tool. For transforming web applications to mobile React and Angular framework's React Native and NativeScript are powerful tools. Thus, every change can be made only once and to ensure that in all platforms app includes the same features.

Key words: programming, mobile, Android, iOS, UWP, native, hybrid

SISÄLLYS

| | | |
|-------|--|----|
| 1 | JOHDANTO..... | 7 |
| 2 | Käyttöjärjestelmät ja laitteet..... | 8 |
| 2.1 | Android..... | 8 |
| 2.1.1 | Käyttöjärjestelmä..... | 8 |
| 2.1.2 | Laitteet..... | 9 |
| 2.2 | IOS..... | 10 |
| 2.2.1 | Käyttöjärjestelmä..... | 10 |
| 2.2.2 | Laitteet..... | 11 |
| 2.3 | Windows Phone / UWP..... | 15 |
| 2.3.1 | Käyttöjärjestelmä..... | 15 |
| 2.3.2 | Laitteet..... | 16 |
| 3 | Sovelluskehitys..... | 18 |
| 3.1 | NATIIVIT OHJELMOINTIKIELET..... | 18 |
| 3.1.1 | Java..... | 18 |
| 3.1.2 | Swift..... | 18 |
| 3.1.3 | Objective-C..... | 19 |
| 3.1.4 | C#..... | 19 |
| 3.2 | HYBRIDIOHJELMOINTIKIELET..... | 19 |
| 3.2.1 | JavaScript..... | 20 |
| 3.2.2 | C#..... | 20 |
| 4 | JULKAISU JA MARKKINOINTI..... | 21 |
| 4.1 | Android..... | 21 |
| 4.1.1 | Google Play..... | 21 |
| 4.1.2 | Amazon Appstore..... | 21 |
| 4.1.3 | F-Droid..... | 22 |
| 4.1.4 | MiKandi..... | 22 |
| 4.2 | IOS ja App Store..... | 23 |
| 4.3 | Windows Phone ja Windows Phone -kauppa..... | 23 |
| 5 | Natiivi ohjelmaesimerkki (Nettilasku)..... | 25 |
| 5.1 | Nettilasku mobiilisovellus Android, iOS ja Windows Phone alustoille..... | 25 |
| 5.2 | Android-versio..... | 27 |
| 5.3 | Windows Phone -versio..... | 27 |
| 5.4 | iOS-versio..... | 27 |
| 5.5 | Lopullinen sovellus..... | 27 |
| 6 | Hybridi ohjelmaesimerkki (Nettikuitti.fi)..... | 28 |
| 6.1 | Nettikuitti.fi mobiilisovellus Android, iOS ja UWP -alustoille..... | 28 |

| | | |
|-------|--|----|
| 6.2 | Suunnitellut uudistukset | 28 |
| 6.2.1 | Dokumentin kuvaus ja tietojen täyttö | 28 |
| 6.2.2 | Tiedon nouto palvelimelta | 28 |
| 6.2.3 | Tiedostojen lähetys palvelimelle..... | 28 |
| 6.3 | Kehitys..... | 29 |
| 6.3.1 | Xamarin..... | 29 |
| 6.3.2 | Kirjastot..... | 29 |
| 6.4 | Julkaisu | 29 |
| 6.4.1 | Google Play..... | 30 |
| 6.4.2 | App Store | 30 |
| 6.4.3 | Windows Phone –kauppa..... | 30 |
| 6.5 | Lopullinen sovellus..... | 30 |
| 7 | POHDINTA | 35 |
| | LÄHTEET..... | 36 |

LYHENTEET JA TERMIT

| | |
|------|---|
| TAMK | Tampereen ammattikorkeakoulu |
| IDE | integroitu ohjelmistoympäristö |
| UWP | Universal Windows Platform |
| .apk | Android sovelluksen tiedostotyyppi |
| AR | Lisätty todellisuus (Augmented Reality) |

1 JOHDANTO

Työn tarkoituksen oli kertoa eri mobiililaitteista, -käyttöjärjestelmistä ja kehitystavoista. Tarkkailtiin kolmea suurinta mobiilikäyttöjärjestelmää Androidia, iOS:ssä ja Windows Phonea. Näistä annettiin lukijalle tarvittavat tiedot, jotta hän tietää perusasiat kaikista ja voi tehdä järkevän päätöksen mille alustoille aloittaa kehittämään sovellusta.

Käytiin myös läpi kaikki suosituimmat markkinapaikat eri alustoilla, jotta lukija osaa valita itsellensä sopivan julkaisupaikan sovellukselleen.

Työssä syvennyttiin natiivin ja hybridi mobiiliohjelmointitapojen eroihin ja kerrottiin miten, sama sovellus kehitettiin ensin natiivein menetelmin Android, iOS ja Windows Phone alustoilla ja myöhemmin myös hybridimenetelmin samoille alustoille.

Esimerkkisovelluksena käytettiin Epic Invoicing Oy:n Nettilasku.fi palvelun Nettilasku ja myöhemmin uudelleen nimettyä Nettikuitti.fi sovellusta.

Työn aikana kehitettiin ensin jokaiselle alustalle alustojen natiiveilla ohjelmointikielillä ja työn loppupuolella uudestaan hybridikehitystavalla Xamarinilla kaikille alustoille Nettikuitti.fi versio, jossa oli mukana myös monia uusia ominaisuuksia, jotka puuttuivat natiivista versiosta.

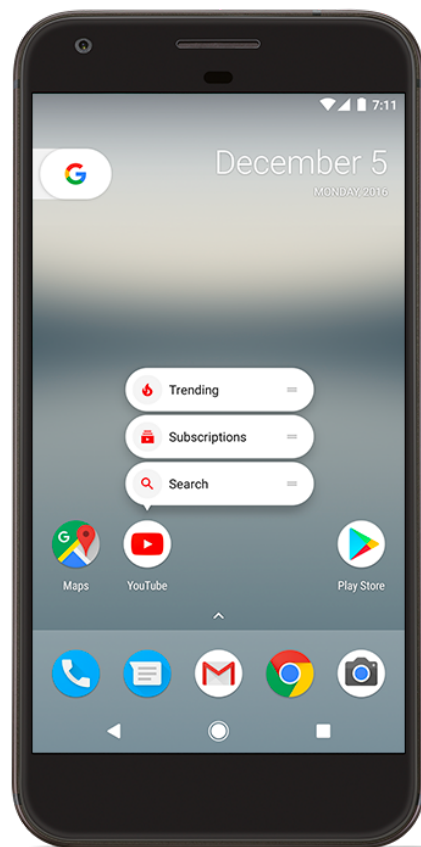
Xamarian versiosta pyrittiin toteuttamaan mahdollisimman vähän laitekohtaista koodia käytävä sovellus, jolloin muutokset pystyttiin tekemään vain kerran.

2 KÄYTTÖJÄRJESTELMÄT JA LAITTEET

Nykyään jatkuvassa kehityksessä on 6 eri käyttöjärjestelmää mobiililaitteille. Neljä näistä perustuu Linux ytimeen, Android, Tizen, Sailfish OS ja Ubuntu Touch. iOS perustuu Applen omaan XNU-ytimeen, mutta sekin kuuluu unixin kaltaisiin ytimiin Linux ytimen kanssa ja näin sisältää paljon samaa perustoiminnallisuutta sen kanssa.

2.1 Android

2.1.1 Käyttöjärjestelmä



KUVA 1. Android 7.1.1 käyttöjärjestelmä

Android on Googlen kehittämä käyttöjärjestelmä, joka pohjautuu muokattuun versioon Linus Torvaldin Linux-ytimeestä (KUVA 1). Android-käyttöjärjestelmän koodi on avointa

ja lisensoitu Apache Software License, Versio 2.0-lisenssillä (Android. Licenses). Androidin avoimuuden ansiosta monet harrastelijat ja yritykset ovat kehittäneet omia versioitaan Android käyttöjärjestelmästä ja jatkaneet vanhojen laitteiden tukemista kääntämällä niille muokattuja versioita uudemmista Android-versioista tai muokkaamalla tietoturvapäivitykset toimimaan vanhoissa Android-versioissa.

2.1.2 Laitteet

Android-laitteita kehittää monet eri valmistajat ja tämän takia on yli 24 000 erilaista laitetta, jotka käyttävät Android-käyttöjärjestelmää (OpenSignal. 2015).



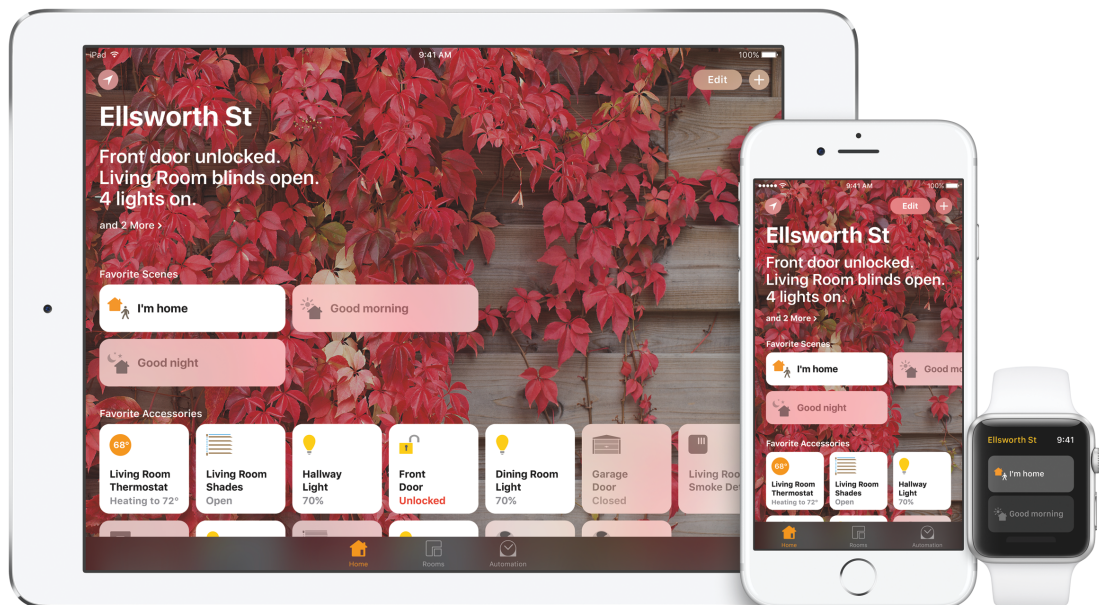
KUVA 2. Google Pixel XL

Ennen Google itse tuki ja mainosti tiettyjä Android-laitteita virallisina Android-lippulainvoina. Nämä laitteet olivat Nexus-sarjan laitteet. Vaikka laitesarjalla oli yhdistävä nimi, niillä oli eri valmistajat aina Asuksesta HTC:hen asti.

Nykyään Google valmistaa itse Pixel-sarjan laitteita ja on korvannut Nexus-sarjan uudella Pixel-sarjalla (KUVA 2).

2.2 IOS

2.2.1 Käyttöjärjestelmä



KUVA 3. iOS käyttöjärjestelmä

iOS on Applen mobiilikäyttöjärjestelmä, joka tarjoaa yhdistystapoja Applen tietokone käyttöjärjestelmän macOS/OS X kanssa (KUVA 3).

Viime aikoina Apple on suunnitellut yhdistää kaksi käyttöjärjestelmäänsä yhdeksi näin sallien kaikkien sovelluksien pyörivän molemmilla. Tästä merkkejä ovat muun muassa Applen omien mobiilisuorittimien päätyminen vähempitehoisiin MacBook-kannettaviin, näin sallien saman prosessori arkkitehtuurin molemmilla alustoilla.

2.2.2 Laitteet

Apple itse kehittää käyttöjärjestelmänsä ja laitteensa, jotka käyttävät sitä. Tämän takia, kun kehitetään iOS-alustalle, tiedetään tarkkaan millä laitteilla sitä tullaan ajamaan ja näin voi sovelluksen testata etukäteen, vaikka jokaisella eri laitteella. Sama on mahdotonta Android kehityksessä, koska Android-laitteita on aivan liian paljon, jotta kaikilla testaaminen olisi järkevää.



KUVA 4. iPhone

iPhone on Applen iOS kehityksen lippulaiva ja uudet versiot iOS-käyttöjärjestelmästä tulee samalla uusien iPhone mallien kanssa (KUVA 4).

Ensimmäisen iPhone julkaisua pidetään älypuhelimien alkuna, ja se tuhosi vanhat puhelinvalmistajajätit jättäen rinnalleen vain ne, jotka olivat valmiina nopeasti innovoimaan. Yksi näistä oli Googlen Android, jonka ensimmäinen kuluttaja laite julkaistiin samoihin

aikoihin ja on vuosien varrella kirinyt iPhonet kiinni ja ohittanut ne markkinaosuudessa vuonna 2010 (Gartner, Inc. 12.8.2010).



KUVA 5. iPad

iPad on Applen tablettitietokone (KUVA 5), joka toi tabletti tietokoneet koko kansan tietoisuuteen, vaikka muut valmistajat muun muassa Microsoft ovat kehittäneet tabletti tietokoneita vuosia ennen Applen ensimmäistä iPad-tablettia (Bort. 2013).



KUVA 6. iPod

iPod on Applen ensimmäinen mobiililaite (KUVA 6). Vuosia yritys oli valmistanut tietokoneita ja niihin lisäosia, mutta vuonna 2001 Apple julkaisi ensimmäinen iPodinsa, joka valloitti kansan ja sai saman aseman kannettavana musiikinsoittimena, kun Sony'n Walkman ennen sitä ja iPhone puhelimien osalta sen jälkeen (Edwards. 2011). Nykyään iPodin kehitys on hidastunut ja viimeisin versio julkaistiin vuonna 2012.



KUVA 7. Apple TV

Apple TV on Applen oma viihdekeskus laatikko (KUVA 7), joka oli vuosia ilman uutta versiota, kunnes vuonna 2015 Apple vihdoin julkaisi uuden laitteen.

Apple TV laitteita pyörittää tvOS, joka on television ruudulla käytettäväksi muokattu versio iOS käyttöjärjestelmästä. Tämän takia pystyy hyvin pienin muutoksin kehittää iPhone sovelluksista Apple TV laitteille sopivia versioita.



KUVA 8. Apple Watch

Uusin tulokas Apple laiteperheeseen on Apple Watch älykello (KUVA 8), jota pyörittää watchOS-käyttöjärjestelmä, joka on kellon näytölle suunniteltu versio iOS-käyttöjärjestelmästä. Apple Watch laitteille pystyy kehittämään samoja sovelluksia kuin muillekin iOS pohjaisia käyttöjärjestelmiä pyörittäville laitteille. Kehityksessä täytyy vain ottaa huomioon älykellojen rajoitettu kosketusnäyttö.

2.3 Windows Phone / UWP

2.3.1 Käyttöjärjestelmä

Windows Phone, nykyään UWP on Microsoftin vastaus Applen iOS ja Googlen Android mobiilikäyttöjärjestelmille.

Vuosien varrella Windows Phone käyttöjärjestelmä on pyrkinyt yhä lähemmin vastaamaan tietokoneiden Windows järjestelmää ja nykyään Microsoftin mobiililaitteet pyörittävät Universal Windows Platform alustaa, jonka ansiosta voit kehittää yhden sovelluksen ja se toimii tietokoneilla, puhelimilla, konsoleilla ja jopa Microsoftin omalla HoloLens AR-laseilla.

Viime vuosina Microsoft on vähentänyt panostustaan mobiilikäyttöjärjestelmäänsä ja laitteisiin, koska vuosien yrittämisellä ne eivät ole onnistuneet kaivertamaan koloa Android ja iOS laitteiden hallitsevilla mobiililaitemarkkinoilla (Keizer. 2016). Tähän suurimpana syynä on sovelluksien puute Windows Phone -alustalla, koska kehittäjät eivät näe niin pieni markkinaosuuden omaavan alustan tukemisen rahojensa ja aikansa arvoisena.

2.3.2 Laitteet



KUVA 9. Lumia 950 XL

Windows Phone laitteita on kehitetty, jos minkälaisia, mutta kuuluisimpina on Lumia sarja (KUVA 9). Lumia sarja alkoi Nokian Windows Phone sarjana, mutta nykyään kun

Nokia tavaramerkki on takaisin Microsoftilta, Lumia sarjaa markkinoidaan Microsoft-puhelimina ja ne toimivat Windows Phone laitteiden lippulaivoina.

3 SOVELLUSKEHITYS

3.1 NATIIVIT OHJELMOINTIKIELET

Natiivit ohjelmointikielet ovat ohjelmointikieliä, joilla eri alustoille on tarkoitus kehittää sovelluksia ja muut ohjelmointikielet yleensä kääntyvät näiksi, kun niillä kehitetään alustalle sovelluksia.

Kehittäessä ainoastaan yhdelle alustalle ja halutessaan mahdollisimman paljon tehoja irti ohjelmastaan, kannattaa kehittää tämän alustan natiivilla ohjelmointikielellä.

3.1.1 Java



KUVA 10. Java logo

Java (KUVA 10) on Googlen Android käyttöjärjestelmän natiivina käyttämä ohjelmointikieli ja on vuosia hallinnut mobiilikehitystä aina vanhoista Nokian laitteista lähtien. Javaa käytetään myös jopa servereillä ja melkein jokaisessa laitteessa, jossa on koodia, on luultavasti edes vähän Javaa.

3.1.2 Swift



KUVA 11. Swift logo

Swift on Applen oma ohjelmointikieli (KUVA 11), joka korvasi Objective-C kielen iOS kehityksessä. Kieli alkoi hyvin C-kielen tyyllisenä, mutta on vuosien varrella muotoutunut omaksensa.

3.1.3 Objective-C

Objective-C on Applen vanha ohjelmointikieli, joka nimensä mukaan pohjautuu C-kielen. Ennen Swift kieltä kaikki iOS kehitys tehtiin Objective-C kielellä ja tämän takia monet vanhat ohjeet ja koodiesimerkit ovat Objective-C kielisiä.

3.1.4 C#

C# on Microsoftin versio korkeamman tason C-kielestä. Kieli löytyy käytöstä melkein kaikilla Microsoftin laitteilla ja monet muutkin ovat valinneet sen ohjelmointikielekseen. Esimerkiksi pelimoottori Unity käyttää pääkielenään C# kieltä, vaikkakin nykyään sallii myös muokatulla JavaScript-kielillä ohjelmoinnin.

3.2 HYBRIDIOHJELMOINTIKIELET

Hybriditohjelmointikieliset ovat kieliä, joilla voidaan kirjoittaa sovelluksia monille eri alustoille, yleensä kääntäjän tai tulkin avulla.

Kääntäjä kääntää hybridi ohjelmointikielen alustan natiiville ohjelmointikielelle, jolloin ei menetetä yhtään tehoa, mutta mahdollisesti joitakin natiivin kielen monimutkaisempia kikkoja, joita kääntäjä ei mahdollisesti osaa jäljentämään.

Tulkit tulkkavat lennossa hybridikieltä natiiville kielelle. Tämä aiheuttaa hitautta ja mahdollisesti ongelmia, jos tulee tulkkausongelma kesken suorituksen.

3.2.1 JavaScript

Monet hybridi mobiilikehitysalustat käyttävät JavaScriptiä skriptikielenään. Ulkoasu yleensä näissä kirjoitetaan HTML- ja CSS-kielillä. Näin web-kehittäjät voivat jo tuntemillaan kielillä kehittämään myös mobiilisovelluksia ja muutkin kuin mobiilikehitysalustat ovat alkaneet omaksua JavaScriptiä ohjelmointikielekseen juuri sen suosion takia, yksi näistä on Unity-pelimoottori, jolle on kehitetty oma versio JavaScript-kielestä.

JavaScript-kehykset tarjoavat myös omat työkalut mobiilikehitykseen. Näistä suosituimmat ja suurimmat ovat Reactin React Native ja Angularin NativeScript.

3.2.2 C#

C# kielen suosio hybridimobiilikehityksessä on melkein täysin nykyään Microsoftin omistaman Xamarin alustan ansiota. Tämän alustan ansiosta voit kehittää C# kielellä Android, iOS ja UWP ohjelmia, kun ennen C# kielellä oli mahdollista kehittää vain Windows Phone ja UWP mobiili alustoille ohjelmille.

4 JULKAISU JA MARKKINOINTI

4.1 Android

Android-käyttöjärjestelmän avoimuuden takia sille on kehitetty monia sovelluksien kauppapaikkoja. Android-laitteille pystyy myös asentamaan sovelluksia suoraan .apk tiedostoista.

4.1.1 Google Play



KUVA 12. Google Play logo

Androidin sovelluskauppapaikoista suosituin on Google Play (KUVA 12), koska se tulee asennettuna kaikissa puhtaan Android-käyttöjärjestelmän omaavissa laitteissa ja suuressa osassa kolmannen osapuolen Android-versioissa, koska Android on avointa lähdekoodia, monet harrastelijoiden kehittämät Android versiot tulevat ilman Googlen sovelluksia ja näin ilman Google Play kauppaa, mutta se pystytään lataamaan laitteelle esimerkiksi Open GApps (The Open Google Apps Project) projektin tarjoamilla paketeilla.

4.1.2 Amazon Appstore



KUVA 13. Amazon Appstore logo

Amazon Appstore on Amazonin vastike Googlen Google Play kaupalle (KUVA 13). Amazon kehitti oman kauppapaikkansa, kun he alkoivat valmistaa omia Android pohjaisia puhelimia ja tabletteja Amazon Fire sarjassa. Näissä Amazonin omissa laitteissa oletus kauppapaikkana on Amazon Appstore ja vaikka Google Play kaupan saa laitteille, se vaatii isompaa ponnistelua.

Vaikkakin Amazon Appstore ei sisällä läheskään samaa määrää sovelluksia Google Play kauppaan verrattuna, niin suurin osa isoimmista ja tärkeimmistä sovelluksista löytyy ja jopa muutamia sovelluksia, joita ei löydy Google Play kaupasta.

4.1.3 F-Droid



KUVA 14. F-Droid logo

F-Droid on avoimen lähdekoodiin perustuva sovelluskauppa Android alustalle ja se sisältää vain ilmaisia sovelluksia (KUVA 14). F-Droid kauppapaikalle on tärkeää heidän käyttäjiensä yksityisyys. He eivät edes salli käyttäjää seuraavien sovelluksien asennusta, jos käyttäjä ei ole asetuksista itse sitä sallinut. (F-Droid. F-Droid kauppapaikan lisätietosivu)

4.1.4 MiKandi

MiKandi sovelluskauppa erikoistuu aikuisviihdepeleihin ja muihin, jotka eivät pääse Google Play kauppaan kyseenalaisen sisältönsä takia.

4.2 IOS ja App Store



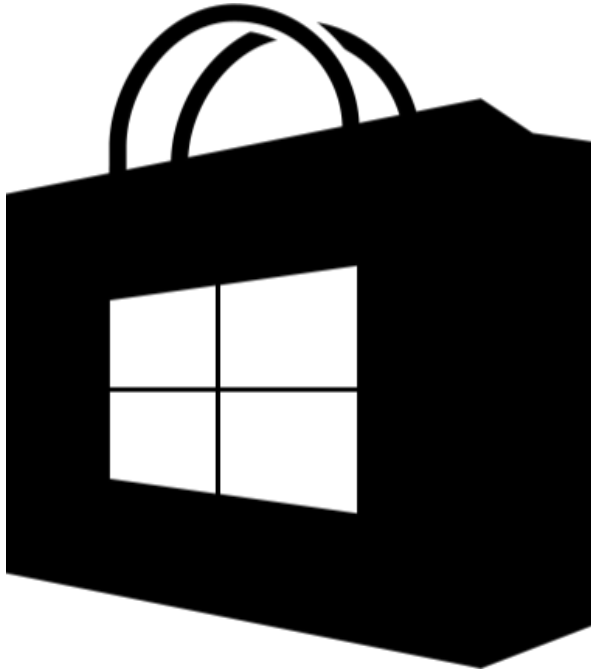
KUVA 15. App Store logo

Applen App Store on iOS-alustan oma kauppapaikka (KUVA 15). App Store on myös ainut paikka, josta voit saada sovelluksia iOS-laitteeseesi, jos et rupea avaamaan laitettasi ja sallimaan kolmannen osapuolen sovelluksien asentamisen.

Tämän monopoliaseman takia Applella on täysi valta mitkä sovellukset se päästää alustalleen. Applella on myös kaikista alustoista tarkin ensitestaus sovellukselle ja se on yleensä viimeinen alusta, jolla sovellus on valmis julkaisuun.

Näiden syiden takia hyvin harvoin App Storeen pääsee haitallisia sovelluksia, mutta se myös tarkoittaa sitä, että voit joutua odottamaan viikkoja, että sovelluksesi hyväksytään julkaisuun.

4.3 Windows Phone ja Windows Phone -kauppa



KUVA 16. Windows Phone -kaupan logo

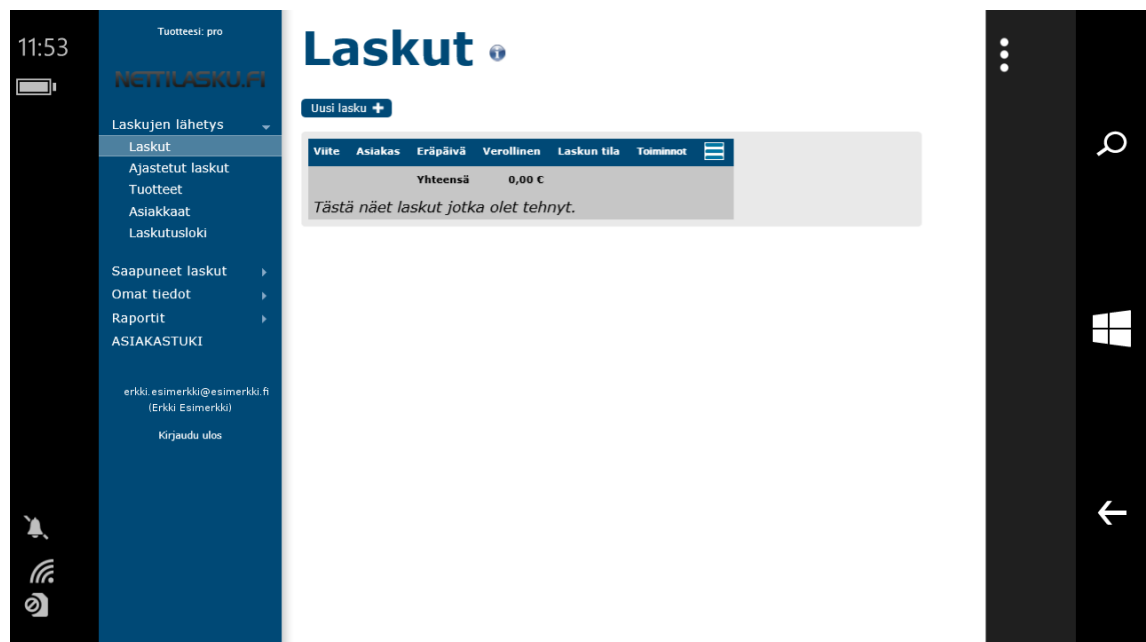
Windows Phone -kauppa sisältää alustojen pääsovelluskaupoista huomattavasti vähiten sovelluksia, mutta silti tarkistus ja sovelluksen julkaisu kestää huomattavasti Google Play kaupaa kauemmin, mutta lyhyemmin kuin Applen App Storessa. Julkaisutilanteessa Microsoftilta voidaan myös soittaa julkaisijalle varmistaakseen, että kyseessä on oikeasti virallinen henkilö, varsinkin jos julkaisee yrityksen nimissä.

5 NATIIVI OHJELMAESIMERKKI (NETTILASKU)

5.1 Nettilasku mobiilisovellus Android, iOS ja Windows Phone alustoille

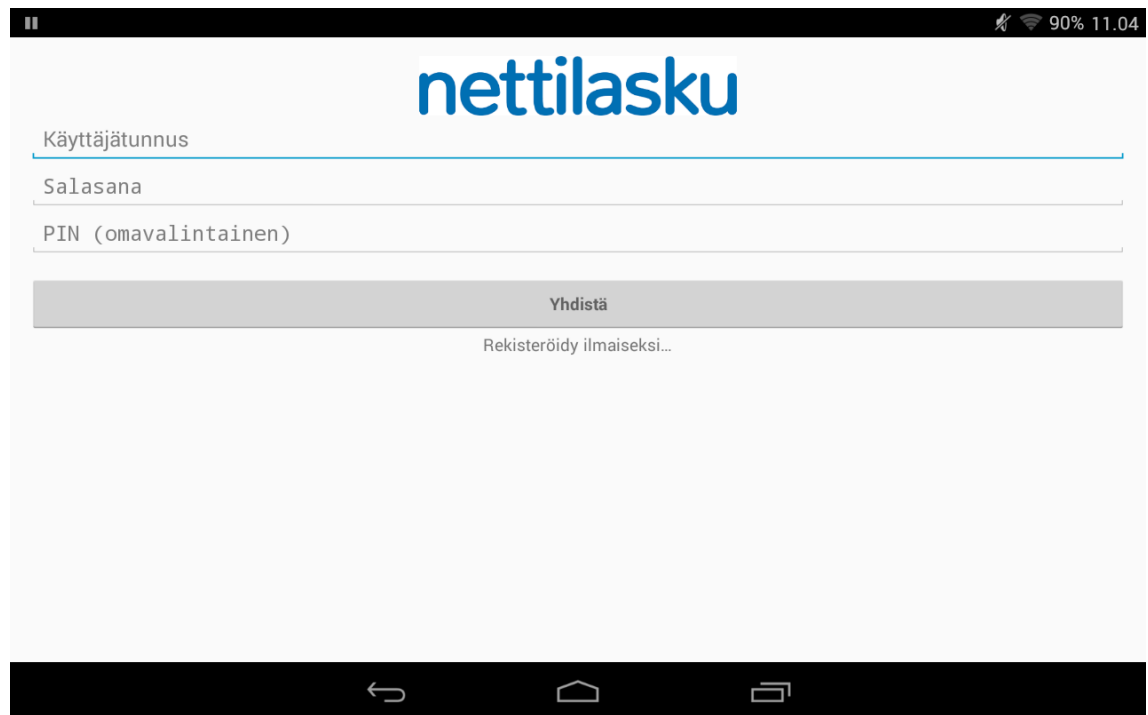
Keväällä 2014 PHP Solutions Oy antoi tehtävän tarkastella eri tapoja kehittää mobiili sovellus Epic Invoicing Oy:n Nettilasku.fi palvelulle.

Sovelluksen tarkoitus oli luoda helppo PIN numeroon ja vahvaan laitekohtaiseen tunnistautumiseen pohjautuva mobiili sisäänkirjautumisalusta Nettilasku.fi-palvelulle (KUVA 17).



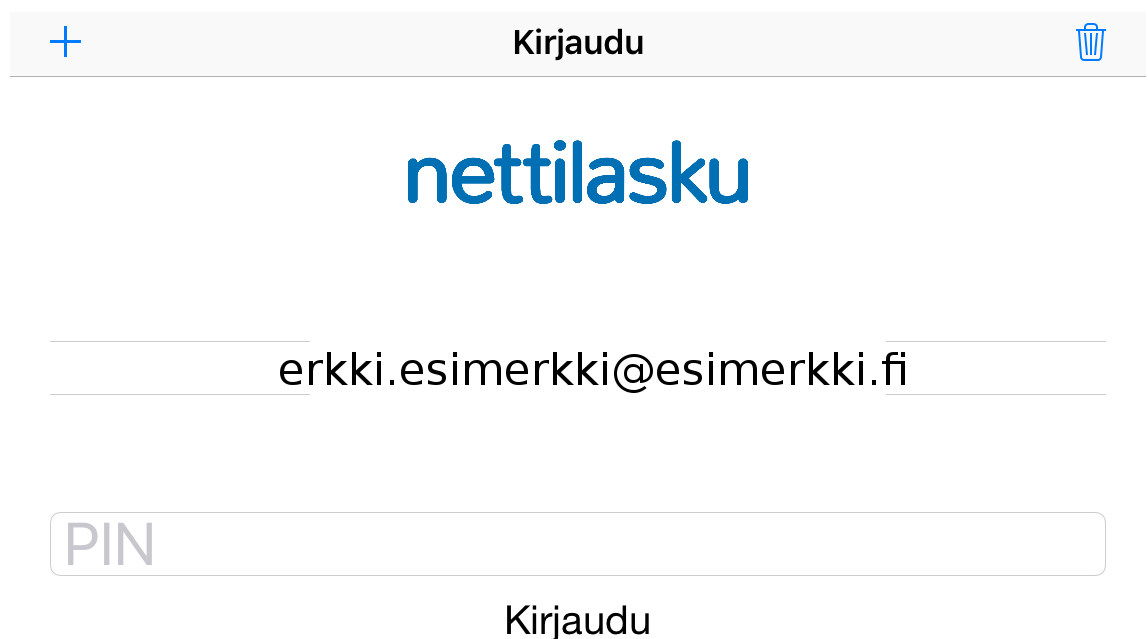
KUVA 17. Nettilasku sovellus kirjautuneena Nettilasku.fi-palveluun Windows Phone -versiossa

Ensimmäisellä kerralla käyttäjä tunnustautuisi täysillä tiedoillaan (KUVA 18).



KUVA 18. Nettilasku sovellus vahva tunnistautuminen Android-versiossa

Tämän jälkeen jälkeen tarvitsisi enää antamansa PIN numeron (KUVA 19).



KUVA 19. Nettilasku sovellus PIN tunnistautuminen iOS-versiossa

Sovellus salli myös monien käyttäjien lisäämisen ja poistamisen, joten henkilö, jolla oli monta yritystä Nettilasku.fi palvelussa pystyi käyttämään yhtä sovellusta kaikkiin kirjautumiseen.

5.2 Android-versio

Android-versio kehitettiin vielä beta-kehityksessä olevalla Googlen Android Studio IDE -alustalla ja Androidin pääkehityskielellä Javalla, johon oltiin lisätty Googlen kehittämät Android-kirjastot.

5.3 Windows Phone -versio

Windows Phone -versio kehitettiin Microsoftin Visual Studio IDE alustalla ja heidän C# -ohjelmointikielellä.

5.4 iOS-versio

iOS-versio kehitettiin Applen omalla Xcode IDE -alustalla ja heidän Swift-ohjelmointikielellänsä.

5.5 Lopullinen sovellus

Sovelluksesta julkaistiin ensin Android- ja Windows Phone -versiot ja vuoden 2016 alussa näiden pohjalta kehitetty iOS-sovellus, joka oli kirjoitettu täysin Applen uudella Swift-kielellä. Sovellus sai tasaisen käyttäjäkunnan, eikä sisältänyt suuria vikoja. Koko elämänsä aikana sovellukseen tehtiin vain kaksi isompaa päivitystä, jotka korjasivat pieniä ongelmia ja nopeuttivat varsinkin vanhinta Android-versiota.

Vuoden 2016 loppu puolella alkoivat puheet kuittien kuvaus toiminnallisuudesta. Android-versiossa oltiin jo kehityksen aikana testattu kuvien ottoa kameralla ja siitä serverille niiden lähettämistä. Toiminnon kaikille kolmelle alustalle erikseen kehittäminen veisi kolminkertaisen verran rahaa ja aikaa, joten päädyttiin kehittämään sovellusta uudestaan Windows Phone version C# koodin pohjalta Xamarin alustalle.

6 HYBRIDI OHJELMAESIMERKKI (NETTIKUITTI.FI)

6.1 Nettikuitti.fi mobiilisovellus Android, iOS ja UWP -alustoille

Vuoden 2016 loppu puolella palkattiin mobiilikehittäjä PHP Solutions Oy yritykseen ja hän ilmaisi intonsa mobiilikehitykseen. Tämän pohjalta päätettiin olla tekemättä isoa päivitystä Nettikuitti-sovellukseen vaan luoda sen pohjalta täysin uusi sovellus, ja tällä kertaa kehittää se Xamarinilla, jotta pieniä muutoksia ei tarvitsisi tehdä kolmeen kertaan.

Projektin tarkoitus oli kehittää mahdollisimman paljon alustaitseenäistä koodia, ja tuoda kaikki suunnitellut uudistukset mukaan sovellukseen.

6.2 Suunnitellut uudistukset

Alun erin projekti lähti liikkeelle vain isosta päivityksestä vanhaan Nettilasku sovellukseen, mutta kasvoi palavereiden myötä kokonaan uudeksi sovellukseksi.

6.2.1 Dokumentin kuvaus ja tietojen täyttö

Tarkoituksena oli kehittää toiminto, jolla käyttäjä pystyisi ottamaan kuvan dokumentista ja täyttää tämän tiedot. Ohjelma mahdollisti kaikenlaisien dokumenttien kuvauksen ja tietojen täytön. Tällaisia dokumentteja olivat muun muassa ostolaskut, kuitit ja käyntikortit.

6.2.2 Tiedon nouto palvelimelta

Ohjelman oli tarkoitus pystyä noutamaan tietoa Nettilasku.fi palvelun palvelimelta ja täyttää niiden avulla käyttäjälle valmiiksi eri kenttiä ja vaihtoehtoa.

6.2.3 Tiedostojen lähetys palvelimelle

Ohjelmaan tuleva dokumenttien kuvaus toiminto tarvitsi keinon lähettää kuvat ja tiedoista muodostuneet JSON-tiedostot palvelimelle. Tätä varten tarvitsi kehittää toiminnallisuus,

jonka avulla mobiilialustan muodostuvat tiedostot voidaan lähettää palvelimelle PHP-scriptiin, joka sitten käsittelee tiedostot ja muodostaa niistä halutut dokumentit Nettilasku.fi-palveluun.

6.3 Kehitys

6.3.1 Xamarin

Kehitysalustaksi valittiin Xamarin, koska vaikka on monia erillaisia hybridi mobiilisovellus kehitysalustoja, Xamarin on kuitenkin Microsoftin takaama. Lisäksi pääsi kirjoittamaan C#-kielellä, joka oli kehittäjille mieluisempaa, kuin JavaScript-kielellä kehitys.

6.3.2 Kirjastot

Nettikuitti.fi sovelluksen kehityksessä pystyttiin käyttämään täysin hybridejä kirjastoja ja niiden avulla pystyttiin saavuttamaan sovellus, jossa ei ollut riviäkään itse kirjoitettua alustakohtaista koodia, vain kaikki toiminnot suoritettiin hybridikirjastojen avulla.

Tulevaisuudessa pyritään myös jatkamaan kehitystä ilman laitekohtaista koodia, niin pitkälle kuin se on mahdollista, näin eliminoiden turhan koodin toistamisen eri alustoilla.

6.4 Julkaisu

Sovellus oli tarkoitus julkaista 15.2.2017, mutta viivästyi Applen tarkistuksen ja viime hetken muutos toiveiden takia, noin kuukauden. Tarkoituksena oli myös julkaista iOS-, Android- ja UWP-alustoille yhtä aikaisesti, mutta UWP-alustan julkaisu tarkistus palvelussa oli ongelmia, joten se jätettiin myöhemmällä.

Sovelluksen lopullinen julkaisupäivä oli 31.3.2017, jolloin iOS- ja Android-versiot sovelluksesta saapuvat markkinoille. Lopullisen versionumero oli 1.2.2, koska alkuperäisestä 1.0.0 version jälkeen uusittiin kokonaan sovelluksen dokumenttisivu, korjattiin ongelmia ja luotiin paljon uudistuksia, joita alkuperäisessä suunnitelmassa ei ollut mukana.

6.4.1 Google Play

Google Play oli helpoin julkaisu alusta toimia. Tarvitsi vain täyttää tiedot ja antaa allekirjoitettu .apk tiedosto. Tämän jälkeen automaattit tarkistivat, että ohjelma kääntyi ja julkaisivat ohjelman kauppaan noin muutamassa tunnissa julkaisu napin painalluksesta.

6.4.2 App Store

Itse tarkistukseen lähetys oli yksinkertaista, mutta tarkistuksessa kesti, jopa viikkoja ja tarkistus keskeytyi, kun tarkistajat kyselivät tietoja hinnoittelusta ja rekisteröitymisestä, mitkä aivan yhtä hyvin olisi voinut olla julkaisulomakkeessa.

6.4.3 Windows Phone –kauppa

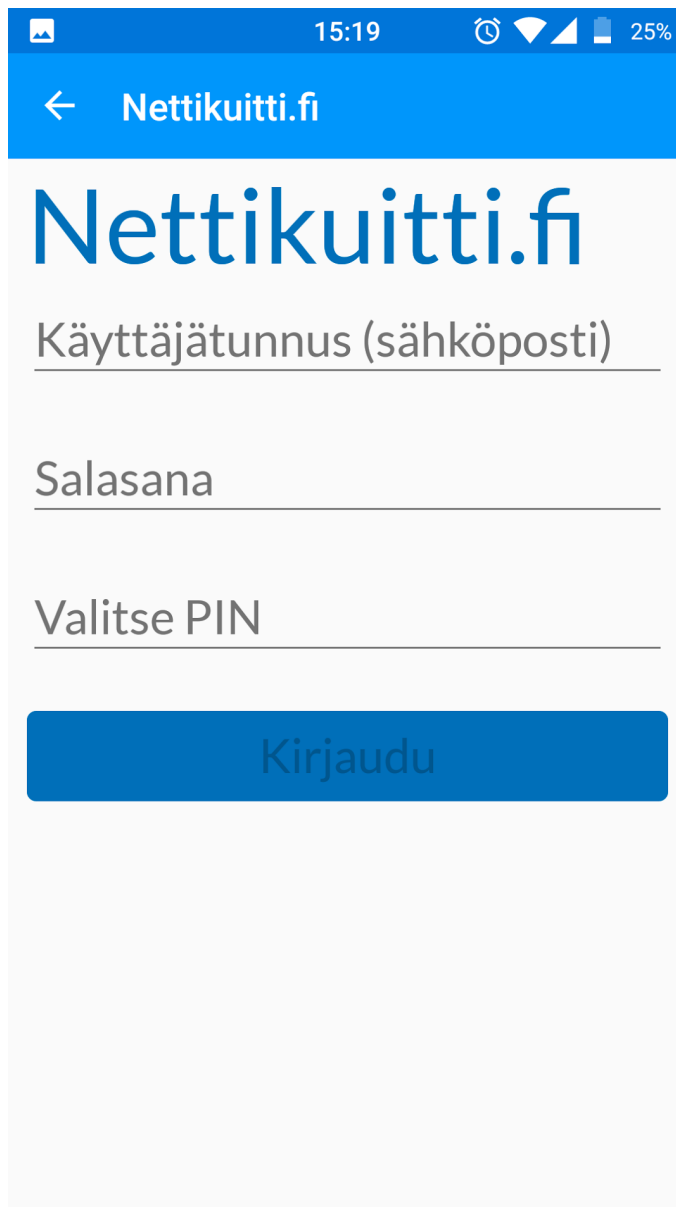
Windows Phone kauppa aiheutti eniten ongelmia ja tämän takia päädyimme siirtämään UWP-alustalle kehitettyä versiota myöhemmälle. Visual Studion ajamat testit ohjelma läpäisi ilman ongelmia, mutta kun julkaisutilanteessa ajettiin samat testit, niin ne epäonnistuivat Microsoftin omien kirjastojen takia.

Asiakastukikaaan ei ollut avuksi, koska ohjasivat kehittäjä chattiin, joka oli auki klo 17 – 00 ja kun muutamaan kertaan odotettiin tuo aika, ohjelma vain koko ajan kertoi, että olen jonossa paikalla yksi ja kehotti odottamaan.

6.5 Lopullinen sovellus

Lopullinen sovellus tuli täyttämään kaikki suunnitellut uudistukset ja saavutti vielä paljon kesken kehityksen tulevia parannuksia mukaansa.

Sovelluksessa oli alussa vahva tunnistautuminen, niin kuin Nettilasku sovelluksessakin (KUVA 20).



15:19 25%

← Nettikuitti.fi

Nettikuitti.fi

Käyttäjätunnus (sähköposti)

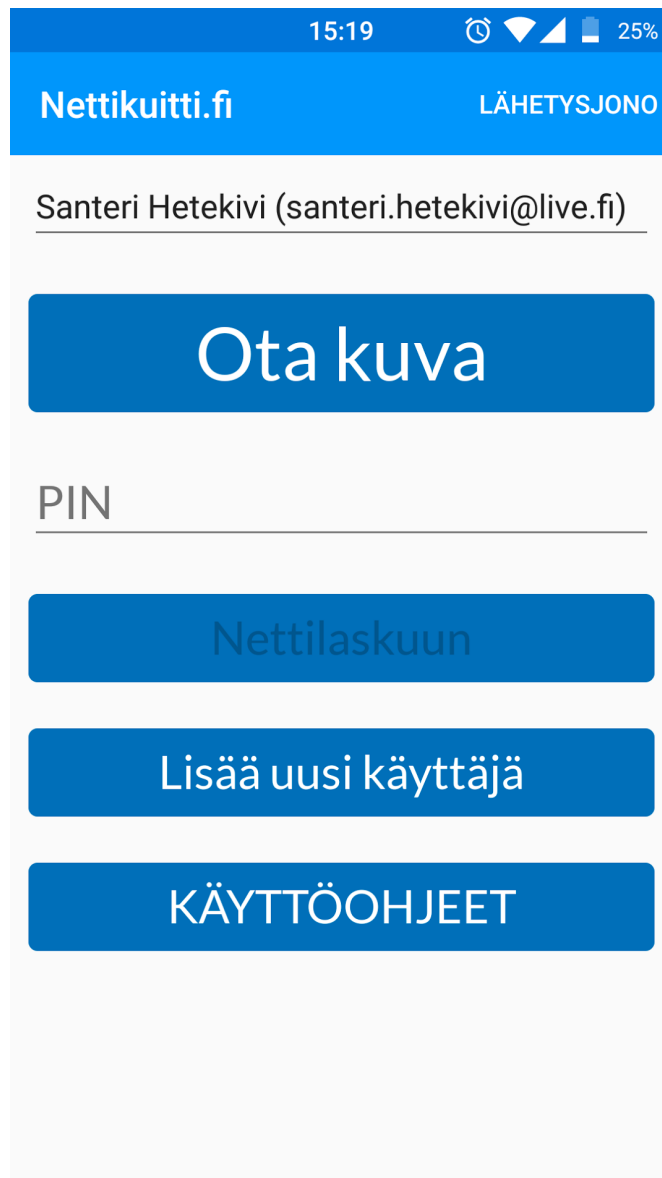
Salasana

Valitse PIN

Kirjaudu

KUVA 20. Nettikuitti.fi sovelluksen vahva tunnistautumis sivu

Kerran tunnistautuessaan käyttäjä pystyi seuraavilla kerroilla kirjautumaan Nettilasku.fi palveluun pelkällä PIN-koodilla (KUVA 21).



KUVA 21. Nettikuitti.fi sovelluksen PIN etusivu ja PIN tunnistautuminen.

Tältä etusivulta käyttäjä pystyi siirtymään Nettilasku.fi-palveluun, lukemaan käyttöohjeita ja uutena isoimpana muutoksena lähettämään dokumentteja Ota kuva -nappulalla ja seuraamaan niitä lähetyksjonosta.

15:20 24%

←

Kuvan tiedot

Käyttäjä

Santeri Hetekivi (santeri.hetekivi@live.fi)

Dokumentin tyyppi

Ostolasku

Täytä tiedot myöhemmin

Ostolasku:

Päivämäärä:

3/30/2017

Bruttosumma (€):

KUVA 22. Netti.fi sovelluksen dokumenttisivu.

Kuvan otettuaan käyttäjä pääsi täyttämään sen tiedot ja valitsemaan tyyppin. Tämä sivu oli rakennettu täysin modulaariseksi ja jokainen kenttä on oma olionsa (KUVA 22). Näin pystyttiin muokkaamaan helposti, mitä kenttiä milläkin dokumentilla on ja tulevaisuudessa, jopa tuomaan niitä suoraan palvelimelta.

Valinta kentät täyttyivät automaattisesti Nettilasku.fi-palvelimelta, käyttäjän vaihtoehdoilla ja tallentuivat puhelimen muistiin, jotta niitä voidaan käyttää, kun käyttäjän laitteella ei ole internet yhteyttä.

Ohjelman toimiminen ilman internet yhteyttä oli muutenkin hyvin tärkeässä osassa ja tämän takia kaikki dokumentit ja kuvat jäävät odottamaan lähetysjonoon ja lähtevät Nettilasku.fi palvelimelle vasta, kun käyttäjän laitteella on yhteys. Tämä mahdollisti kuvan ottamisen kuitista missä tahansa ja saavuttua internet yhteyden päähän näet kuitenkin Nettilasku.fi palvelussa.

7 POHDINTA

Kehitys vain yhdelle alustalle sen natiivilla ohjelmointikielellä ja työkaluilla on suositeltavaa vain, jos ohjelma on suunniteltu vain sille alustalle. Tällaisia tilanteita ovat esimerkiksi, jos tehdas haluaa Android-sovelluksen, joka on suunniteltu juuri tietylle Android-laitteelle, jota tehtaassa käytetään pelkästään.

Kaikki kuluttajapuolelle suunnatut sovellukset kannattaa kehittää jollain hybridi tekniikalla, koska samalla määrällä työtä saat paljon suuremman osan mahdollisista asiakkaista sovelluksellesi. Näin tehdessään saattaa menettää muutamia erikoistoimintoja, mutta esimerkiksi Xamarinilla voi kirjoittaa silti laitekohtaista koodia.

Vain mobiilisovelluksen kehitykseen Xamarin on hyvä työkalu, koska se on helppo oppia ja sallii kirjoittaa myös laitekohtaista koodia tarvittaessa. Sovelluksen julkaisukin on yksinkertaista. Tarvitset vain painaa nappia ja vastata kysymyksiin, niin sovellus on jo Google Play storessa. iOS julkaisu pitää tehdä vielä Applen työkaluilla, mutta Xamarin osaa luoda tarvittavan paketin ja jopa avata tarvittavan työkalun julkaisuun.

Nettisivun kääntämisessä mobiilisovellukseksi kannattaa harkita koko nettisivun tekemistä React tai Angular kehyksillä, koska näille on omat mobiilikehitys työkalunsa. Angular koodin voit muuttaa mobiilisovellukseksi NativeScript:illä ja Reactin React Native:llä. Näin pystyy säästämään paljon aikaa ja vaivaa. Lisäksi NativeScriptillä ja React Native:llä kehittäessä, pystyy käyttämään samoja taitoja, mitä on jo opetellut Reactilla tai Angularilla ohjelmoidessa.

LÄHTEET

Android. Licenses. Luettu 3.4.2017. <https://source.android.com/source/licenses.html>

Applen viralliset tuotekuvat. <http://www.apple.com/pr/products/>

Bort, J. 30.4.2013. Microsoft Invented A Tablet A Decade Before Apple And Totally Blew It. Luettu 3.4.2017. <http://www.businessinsider.com/heres-visual-proof-of-just-how-badly-microsoft-blew-it-with-tablets-2013-5?r=US&IR=T&IR=T>

Edwards, B. 23.10.2011. The iPod: How Apple's legendary portable music player came to be. Luettu 3.4.2017. <http://www.macworld.com/article/1163181/consumer-electronics/the-birth-of-the-ipod.html>

F-Droid. F-Droid kaupapaikan lisätietosivu. Luettu 15.3.2017. <https://f-droid.org/about/>

Gartner, Inc. 12.8.2010. Gartner Says Worldwide Mobile Device Sales Grew 13.8 Percent in Second Quarter of 2010, But Competition Drove Prices Down. Luettu 2.4.2017. <http://www.gartner.com/newsroom/id/1421013>

Googlen virallinen kuva Pixel XL puhelimesta, https://storage.googleapis.com/made-bygoog/v1/phone/specs/marlin-black-en_US.jpg

Googlen virallinen kuva Android 7.1.1 versiosta, https://storage.googleapis.com/gweb-uniblog-publish-prod/images/App-Shortcuts_YouTube_FINAL.width-581.png

Keizer, G. 19.8.2016. Windows Phone sales collapse as Microsoft ravages Nokia remnants. Luettu 3.4.2017. <http://www.pcworld.com/article/3109697/phones/windows-phone-sales-collapse-as-microsoft-ravages-nokia-remnants.html>

Olmstead, K. Atkinson, M. Apps Permissions in the Google Play Store. Pew Research Center. Julkaistu 10.11.2017. Luettu 11.2.2017. <http://www.pewinternet.org/2015/11/10/apps-permissions-in-the-google-play-store/>

Lumia 950 XL mainoskuva. <https://compass-ssl.microsoft.com/assets/43/b2/43b267f7-736d-4ae8-9d0e-f92079602f18.jpg?n=hero-desktop.jpg>

OpenSignal. Android Fragmentation Visualized. Julkaistu 08.2015. Luettu 12.02.2017, <https://opensignal.com/reports/2015/08/android-fragmentation/>

Virallinen App Store logo. https://upload.wikimedia.org/wikipedia/en/1/1f/App_Store_Logo.png

Virallinen Google Play logo. https://upload.wikimedia.org/wikipedia/commons/thumb/0/0d/Google_Play_logo.svg/2000px-Google_Play_logo.svg.png

Virallinen F-Droid logo. https://upload.wikimedia.org/wikipedia/commons/thumb/5/55/F-Droid_Logo_3.svg/2000px-F-Droid_Logo_3.svg.png

Virallinen Swift logo. https://devimages.apple.com.edgekey.net/assets/elements/icons/swift/swift-64x64_2x.png

Virallinen Windows Phone-kaupan logo. https://upload.wikimedia.org/wikipedia/en/thumb/1/1b/Windows_Store_Logo.svg/378px-Windows_Store_Logo.svg.png