

---

# **MOBIILISOVELLUSTYYPIT JA NIILLÄ KEHITTÄMISEN VAIHTOEHDOT**

Case: Hämeen Ammattikorkeakoulun mobiilipalvelut



Ammattikorkeakoulun opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Visamäki, kevät 2016

Mikael Lehto



VISAMÄKI  
Tietojenkäsittely  
Systemityö

---

<b>Tekijä</b>	Mikael Lehto	<b>Vuosi</b> 2016
<b>Työn nimi</b>	Mobiilisovellustyypit ja niillä kehittämisen vaihtoehdot	

---

## TIIVISTELMÄ

Opinnäytetyön toimeksiantaja oli Hämeen Ammattikorkeakoulu. Työn tavoitteena oli tuoda HAMKille uutta tietoa mobiilisovellusten kehittämisen vaihtoehdoista ja pohtia, mitkä niistä vaihtoehdoista sopisivat HAMKille.

Opinnäytetyötä varten ei toteutettu mobiilisovellusta. Tietoa web appin kehittamisestä sai kuitenkin HAMKin Oiva-mobiilipalvelusta. Oiva toteutettiin opiskelijatyönä vuoden 2015 lopussa ja 2016 alussa. Tärkeimpänä tiedonlähteenä toimivat erilaiset verkkosivut, joista haettiin mobiilisovellustyyppisiin ja työvälineisiin liittyvä tieto. HAMKin mobiilisovellusten tilanteeseen ja niiden vaatimuksiin liittyvät tiedot saatiin HAMKin mobiilistrategiadokumenteista ja haastattelusta asiakkaan kanssa.

Opinnäytetyössä selvisi, että hybridisovelluksien kehittämistapoja ja välineitä on paljon ja niillä on paljon kilpailua. Tutkimuksen tuloksena voidaan todeta, että hybridisovellus-muoto sopii parhaiten HAMKin tarpeisiin. HTML5-tekniikoilla tehty mobiilisovellus, eli web app, on myös hyvä vaihtoehto kevyihin mobiilipalveluihin. Natiivisovellusten kehittäminen ja ylläpito on kallista. HAMKin ei kannata vaivautua niiden tuottamiseen, kun on olemassa halvempia ja helpompia vaihtoehtoja jotka toimivat lähes yhtä hyvin.

**Avainsanat** hybridi, natiivi, web app, mobiilisovellus, kehitysjono

**Sivut** 27 s. + liitteet 9 s.

VISAMÄKI  
Information Business Technology  
Systemityö

---

<b>Author</b>	Mikael Lehto	<b>Year</b> 2016
<b>Subject of Bachelor's thesis</b>	Different mobile app types and their development options	

---

## ABSTRACT

The client of this thesis is HAMK and the goal is to provide HAMK with new information of ways to create mobile apps, and to ponder which of these choices would work best for HAMK's purposes.

No mobile apps were made for this thesis, however information regarding how to create web apps was obtained from HAMK's Oiva mobile service. Oiva was made as a student project in late 2015 and early 2016. The most important source of knowledge came from the Internet, from which information about different mobile app types and tools was gathered. Information regarding HAMK's situation with mobile apps and their requirements was gathered from HAMK's mobile strategy documents and from an interview with the client.

Through this thesis it was found that there are many ways and a lot of tools to create hybrid apps, and that they have a lot of competition amongst each-other. A hybrid app – type suits HAMK's needs the best. A web app is also a great option when making light mobile services. The development and up keep of native apps is expensive and therefore HAMK should not bother with making them as there are cheaper and easier options that work nearly as well.

**Keywords** hybrid, native, web app, mobile app, development queue

**Pages** 27 p. + appendices 9 p.

# SISÄLLYS

1	JOHDANTO.....	1
2	HAMKIN MOBIILIPALVELUIDEN TILANNE .....	3
2.1	Rajapinnat ja tietoturva .....	3
2.2	Mobiilitarjooma ja meneillä olevat hankkeet.....	4
2.3	Kehitysjono .....	5
3	NATIIVISOVELLUS.....	7
3.1	Android .....	7
3.2	iOS.....	7
3.3	Windows Phone .....	8
4	HYBRIDISOVELLUKSET JA CROSS-PLATFORM KEHITTÄMINEN .....	10
4.1	Sovelluskehikset ja pakkaustyökalut.....	10
4.2	Rapid Application Development.....	11
4.2.1	OutSystems Platform.....	12
4.2.2	Mendix Platform.....	13
4.2.3	AppGyver ja Composer 2.....	13
4.3	Muita cross-platform menetelmiä .....	14
4.3.1	Xamarin .....	14
4.3.2	Windows Bridge for iOS .....	15
5	WEB APP .....	16
6	HAMKIN TULEVIEN MOBIILISOVELLUSTEN LUOKITTELU .....	18
6.1	Paljon mediaa sisältävät palvelut .....	18
6.2	Aktiivisesti käytössä olevat palvelut.....	18
6.3	Laitteen natiivitoimintoja käyttävät palvelut.....	18
6.4	Yleisön koko .....	19
6.5	Kirjautumista vaativat palvelut .....	19
6.6	Verkkoyhteyttä vaativat palvelut .....	19
7	HYBRIDI VAI NATIIVI?.....	20
7.1	Natiivisovellus ja sen haitat .....	20
7.2	Cross-platform työkalut .....	20
7.3	Palveluiden tarpeet ja ratkaisut .....	21
8	WEB APP VAI NATIIVI? .....	23
8.1	Web app työkalut .....	23
8.2	Palveluiden tarpeet ja ratkaisut .....	24
9	LOPPUTULOS.....	25
10	YHTEENVETO.....	27
	LÄHTEET .....	29

---

Liite 1	Visual Studion maksulliset lisenssit
Liite 2	AppGyverin lisenssit
Liite 3	Mendix Platformin lisenssit
Liite 4	Xamarinin lisenssit

---

## KÄSITTEET

Avoin lähdekoodi: Ohjelman koodi on käyttäjän tarkasteltavissa (engl. Open Source)

CSS: Verkkosivujen muotoilukieli

Debuggaus: Koodin testaus jossa paikallistetaan virheet

Ekosysteemi: Käsittää mobiililaitteen käyttöjärjestelmän, sen tukemat sovellukset ja sovelluskaupat (esim. Android, iOS ja Windows Phone)

HTML5: Uusin versio hypertekstin merkintäkielestä. Käytetään verkkosivujen ohjelmointikielenä

Integroitu ohjelmointiympäristö: Ohjelma, jolla voi ohjelmoida (engl. IDE, Integrated Development Environment)

JavaScript: Ohjelmointikieli

Ohjelmointirajapinta tai pelkkä rajapinta: Eri ohjelmien välinen määritelmä jonka avulla ne voivat vaihtaa tietoa keskenään (engl. API, Application Programming Interface)

Runtime: Ohjelma joka tukee ohjelmien ajoa

SDK: Software Development Kit, eli ohjelmien kehittämistyökalut, joilla luodaan sovelluksia tietyille ohjelmistopaketeille

Selainmoottori: Ohjelma jonka tehtävä on noutaa dataa internetistä ja saada HTML- ja CSS-kielinen koodi ja muuntaa se verkkosivuksi (engl. Web Engine)

WWW-palvelin: Palauttaa www-selaimen pyytämän sisällön, kuten verkkosivut, ja tekee sen tarkasteltavaksi selaimen avulla.

## 1 JOHDANTO

Opinnäytetyön toimeksiantajana toimi Hämeen Ammattikorkeakoulu. Opinnäytetyössä käsitellään mobiilisovellustyyppejä ja niiden sopivuutta HAMKin tuleviin mobiilisovelluksiin. Opinnäytetyön tavoitteena on tuoda HAMKille tietoa mobiilisovellusten kehittämisen vaihtoehdoista. Oppilaitos on tutkinut aihetta jo aikaisemmin, joten tarkoitus on tuoda esiin asioita, mitä he eivät vielä tienneet. Työssä ei ole tarkoitus kiinnittää huomiota pilvipalveluihin ja siksi käsitellään vain HAMKin tukemia ekosysteemejä, jotka ovat Android, iOS ja Windows. Tutkimuksessa käytetty tieto on ulkoisista lähteistä, kuten internetistä ja haastatteluista. Vaikka opinnäytetyötä varten ei ole luotu mobiilisovellusta, niin aiheesta on kuitenkin kokemusta HAMKin Oiva-palvelun kehittämisen kautta.

Luku 2 käsittelee HAMKin mobiilipalveluiden tilannetta. Tarkoituksena on käsitellä HAMKilta saatuja mobiilistrategiadokumentteja ja tarkastella niissä olevia vaatimuksia ja ohjaavia tekijöitä mobiilisovelluksille. Luvussa tutustutaan myös HAMKin mobiilisovellusten kehitysjonoon, jota käsitellään vielä myöhemmin tässä opinnäytetyössä.

Luku 3 käsittelee mitä natiivisovellus tarkoittaa ja mitä ominaisuuksia niillä on. Luvussa käsitellään myös HAMKin tukemat ekosysteemit ja millainen julkaisuprosessi mobiilisovelluksilla on sovelluskaupoissa. Luku 4 käsittelee sitä, mitä hybridisovellukset ovat, miten ne toimivat ja millaisia kehittämisvaihtoehtoja niissä on. Luvussa käsitellään ensin hybridisovellusten kehittäminen sovelluskehysillä, jonka jälkeen siirrytään toisiin kehittämisen muotoihin. Luku 5 käsittelee mitä web appit ovat ja mitä ominaisuuksia, etuja ja rajoitteita niillä on. Luvussa käsitellään myös työkaluja joilla niitä voi tehdä ja miten. Tieto kehittämisvaihtoehdoista ja työkaluista on kerätty Internetistä.

Luvussa 6 luokitellaan HAMKin kehitysjonossa olevat sovellukset, jotta niitä voidaan käsitellä myöhemmin. Tässä luvussa käsitellään myös luvun 2 aiheita, jotka auttavat mobiilisovellusten luokittelussa. Luokittelun on tarkoitus helpottaa sovelluksen kehittämistavan löytämistä. Luokkia voidaan käyttää uudestaan jatkossa, kun kehitysjonoon lisätään uusia palveluita. Tämä on huomattavasti tehokkaampaa kuin kehitysjonon mobiilisovellusten yksittäin käsittely.

Luvussa 7 vertaillaan natiivi- ja hybridisovellusten ominaisuuksia ja kehittämisvaihtoehtoja. Lopuksi pohditaan miten hybridisovellus sopisi HAMKin tarpeisiin parhaiten käyttäen luvussa 6 luotuja luokkia. Luvussa 8 vertaillaan natiivisovelluksien ja web appien ominaisuuksia ja kehittämisvaihtoehtoja. Lopuksi pohditaan, missä tilanteissa web app olisi parempi vaihtoehto kuin natiivisovellus.

Luvussa 9 käsitellään tutkimuksen lopputuloksia, käsitellään kehitysjonon sovellukset ja suositellaan niille sovellustyyppi. Lopuksi pohditaan millä työkaluilla tulisi toteuttaa hybridisovellukset ja web appit. Opinnäytetyö viihteistellään yhteenvedolla luvussa 10.

Opinnäytetyö vastaa seuraaviin tutkimuskysymyksiin:

- Mikä on HAMKin mobiilisovellusten nykyinen tilanne?
- Mitä sovellustyyppijä on tarjolla ja miten ne eroavat toisistaan?
- Mitä mobiilisovelluksia HAMK aikoo tehdä ja miten ne voidaan toteuttaa?
- Miten sovellustyypit sopivat HAMKin tarkoituksiin?

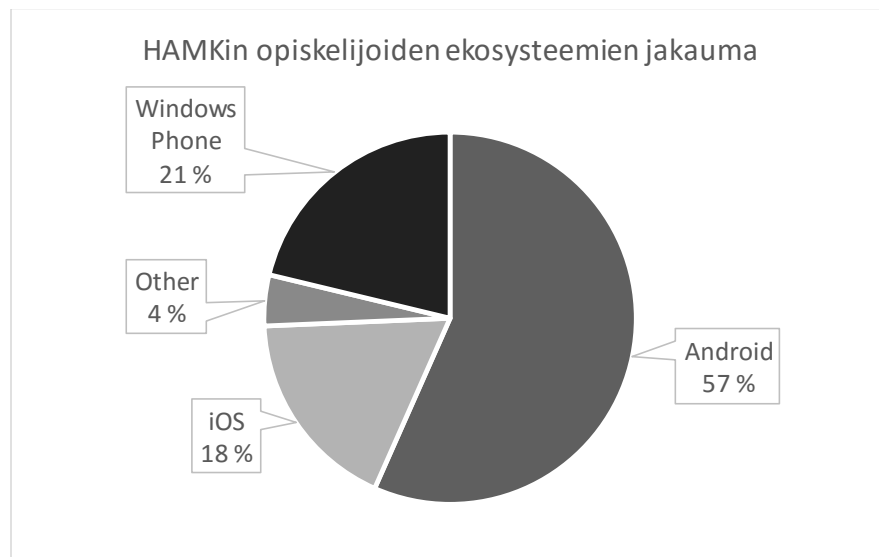


## 2 HAMKIN MOBIILIPALVELUIDEN TILANNE

Hämeen ammattikorkeakoulu teetti mobiilistrategian ja mobiilipalveluiden toimintasuunnitelman vuosille 2016–2020. Mobiilistrategian tarkoitus on asettaa palveluiden kehittämisen tavoitteet ja keinot niiden tavoitteiden saavuttamiseksi, ja tuottaa käyttäjille ja HAMKille mahdollisimman paljon lisäarvoa. Toimintasuunnitelman tarkoitus on ohjata HAMK:n mobiilipalveluiden kehitystä. Palveluiden on tarkoitus tehostaa opiskelijoiden ja henkilökunnan toimintaa. Ilman ohjaavaa ja yhteistä päämäärää, syntyy suuri riski palveluiden kehittämiseen käytetyn rahan hukkaan menemisestä. Myös tietoturvaratkaisujen suhteen tehdään tapauskohtaisia ratkaisuja. Ilman yhteistä näkemystä niiden suhteen lisääntyy tietoturvariskejä. HAMK:n mobiilipalveluiden on tarkoitus olla helppokäyttöisiä, yksinkertaisia, nopeita ja tarvelähtöisiä. Niiden on tarkoitus tukea opiskelijan aidon henkilökohtaisen opiskelijapolun rakentamista. (Rintala 2016; Rintala, haastattelu 17.2.2016.)

### 2.1 Rajapinnat ja tietoturva

Toimintasuunnitelmassa on määritelty, että HAMK tukee vain Android-, iOS- ja Windows Phone -ekosysteemiä. Nämä kolme ekosysteemiä ovat käytetyimpiä ja yleisimpiä Suomessa ja koko maailmassa. (HAMK:n mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015)



Kaavio 1. HAMK:n opiskelijoiden älypuhelinien ekosysteemien jakauma (Hytönen H-K 2016.)

Web-tekniikoita käyttävät mobiilipalvelut tukevat mahdollisuuksien mukaan myös muita laitteita. HAMK ei aio toteuttaa tai teettää räätälöityjä native-applikaatioita kustannussyistä, ellei ole erityisen hyvin perusteltua syytä. Asiakasryhmät joille mobiilipalvelut on suunnattu, ovat tärkeysjärjestyksessä: opiskelijoille, henkilökunnalle, hakijoille ja kumppaneille. (HAMK:n mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015.)

HAMK suosii avoimien ja reaaliaikaisten rajapintojen käyttämistä, sillä ne mahdollistavat mobiilikäyttöliittymien ja integraatioiden toteuttamisen. Myös opiskelijaprojektit, jotka käyttävät HAMK:n avoimia rajapintoja, ovat mahdollisia. Muuten sovellukset teetetään itse tai tilataan ulkopuolelta. Mobiilipalveluilla ei saa olla suoraa yhteyttä tietokantoihin ja niiden tulee käyttää HAMK:n järjestelmien dataa rajapintojen kautta. (HAMK:n mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015.)

HAMKilla ei ole tällä hetkellä kyvykkyyttä toteuttaa mobiilisovelluksia perinteisillä menetelmillä eikä RAD-välineillä. HAMK ei aio myöskään ostaa uusia kokonaisia mobiilisovelluksia ulkopuolelta kustannussyistä ja kilpailutuksen hitauden takia. Kyvykkyys sovellusten tekoon tulee hankkia itse. Henkilöresursseja kohdistetaan tarvittaessa mobiilisovelluksen suunnitteluun, sisältöjen laatimiseen ja prosessien kehittämiseen. RAD-välineillä kehittäminen olisi ehkä mahdollista, mutta perinteisillä tavoilla sovelluksen kehittämiseen ei ole kuitenkaan henkilöresursseja. Rajapintojen tekoon löytyy kyvykkyyttä ja resursseja, joten niitä käyttävien opiskelijaprojektien teko on kuitenkin mahdollisia. (HAMK:n mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015; Rintala, Haastattelu 17.2.2016.)

Toimintasuunnitelmassa on määritelty mitkä tiedot ovat julkista ja luottamuksellista. Kyseiset määritelmät tulee ottaa huomioon mobiilipalvelua kehitettäessä. HAMK:n toimintaan liittyvät tiedot, kuten tapahtumatiedot, opetussuunnitelmat ja opetustarjonta, on julkista. HAMK:n henkilöstöön ja opiskelijoihin liittyvät henkilötiedot ovat kuitenkin pidettävä salassa. (HAMK:n mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015)

### 2.2 Mobiilitarjooma ja meneillä olevat hankkeet

HAMKilla on ollut syksyn 2015 jälkeen organisaation laajuisesti käytössä seuraavat mobiilipalvelut: HAMK.fi ja HAMI.fi (responsiiviset verkkosivut), mobiilityöajanseuranta, Office 365-osapalvelut (johon kuuluu sähköposti, kalenteri, Yammer ja Intra), Moodle (responsiiviset verkkosivut), Skype for Business, opintasuoritukset sähköpostiin, mobiilikyselyt ja tuntiperuutukset tekstiviestillä. (HAMK:n mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015.)

Keväällä 2016 valmistui HAMKille opiskelijaprojektina teetetty Oiva-mobiilipalvelu. Sen päätoiminto on lukujärjestys ja sen lisäksi palveluun on lisätty vapaiden tilojen hakupalvelu, kampuksen ravintolan ruokalistojen haku ja lukuvuosi kalenteri. Palvelu käyttää avointa dataa ja eikä siis vaadi kirjautumista. Oiva on toteutettu responsiivisena web-palveluna.

Meneillä oleviin hankkeisiin kuuluu Peppi-järjestelmä, Perusrekisteri-järjestelmä, Pakki-järjestelmä ja biotalouden digitoimintojen kehittämishanke. Peppi-järjestelmä tarjoaa avoimet rajapinnat tilanvarauksiin ja opetussuunnitelmiin. Esimerkiksi Oiva käyttää näitä rajapintoja. Perusrekisteri-järjestelmä tuo opintosuhteiden ja opettajien selainpohjaiset työpöydät tabletille. Pakki-järjestelmä on opiskelijan selainpohjainen työpöytä, joka toteutetaan

responsiivisena verkkopalveluna. Se sisältää mm. ilmoittautumiset ja arviointit. Biotalous digitoiminnan kehittämishanke sisältää biotalouden opiskelijoille tarkoitettuja opetusvälineitä, jotka helpottavat ulkona opiskelua. (HAMKin mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015.)

### 2.3 Kehitysjono

HAMKin mobiilipalveluiden toimintasuunnitelmassa määriteltiin myös kehitysjono, jossa kerrotaan mitä sovelluksia HAMK on suunnitellut kehittävänsä vuoteen 2020 mennessä. Kehitysjonoon kuuluu: opintosuoritukset mobiiliin, vapaiden tilojen hakupalvelu, ohje- ja opetusvideot – opiskelijan käyttöliittymä, HAMKin IT-palveluiden liikennevalot, HelpDeskin asiakasliittymä mobiiliksi, Moodlen mobiilikäyttö opiskelijalle, opiskelijan ilmoittautumispalvelu ja opetuksen mobiilipalvelu. (HAMKin mobiilipalveluiden kehittämisen toimintasuunnitelma 2016, 2015.)

*Opintosuoritukset mobiiliin* esittää opiskelijan muutamia viimeisimmistä suorituksista. Suoritukset haetaan opiskelijanumerolla. Sovellus vaatii joko kirjautumisjärjestelmän tai PIN-koodin, sillä tieto ei ole täysin avointa. (Rintala, haastattelu 17.2.2016.)

Vapaiden tilojen hakupalvelu on palvelu, jolla haetaan nimensä mukaisesti vapaita tiloja (kuten luokkia, kokoustiloja tai laboratorioita) halutusta kampuksista ja rakennuksesta. Tieto on avointa dataa ja ei vaadi kirjautumista tai muita tunnistautumistoimintoja. Palvelu on liitetty opiskelijoiden ICT-Projektina tehtyyn Oiva-mobiilipalveluun. (Rintala, haastattelu 17.2.2016.)

Ohje- ja opetusvideot – opiskelijan käyttöliittymälle löytyy jo natiivisovellus, josta voi ottaa mallia kehittäessä HAMKille omaa versiota. Kultura on ohje- ja opetusvideopalvelu, josta opiskelijat pystyvät katsomaan opetuskäyttöön tarkoitettuja videoita. Palvelu toimii HAMKin, LAMKin ja Laurean välillä. Jokaisella korkeakoululla on omat syötteensä, mutta haut toimivat ristiin. Palvelu vaatii kirjautumisen HAMKin käyttäjätunnuksella. (Rintala, haastattelu 17.2.2016.)

HAMKin IT-palveluiden liikennevaloista pystyy tarkastamaan HAMKin järjestelmien tilat. Tämä palvelu on suunnattu henkilökunnalle. Palvelu vaatii kirjautumisen HAMKin käyttäjätunnuksella. (Rintala, haastattelu 17.2.2016.)

HelpDeskin asiakasliittymästä pystyy lähettämään korjauspyyntöjä virheilmoituksista. Palvelu vaatii kirjautumisen HAMKin käyttäjätunnuksilla. (Rintala, haastattelu 17.2.2016.)

Opiskelijan ilmoittautumispalvelussa opiskelija pääsee ilmoittautumaan kursseille ja lukuvuodelle läsnä olevaksi. Palvelu vaatii kirjautumisen HAMKin käyttäjätunnuksella. (Rintala, haastattelu 17.2.2016.)

Opetuksen mobiilipalvelu voisi olla esimerkiksi opiskelijan tai tuutorin visuaalinen HOPS-väline, jolla voi ajoittaa opintonsa. Voi sisältää myös

opiskelijoiden syrjäytymistutkan. Sisältö tullaan suunnittelemaan myöhemmin yhteistyössä koulutuksien kanssa. Palvelu vaatii kirjautumisen HAM-Kin käyttäjätunnuksella. (Rintala, haastattelu 17.2.2016.)

### 3 NATIIVISOVELLUS

Natiivisovellus on asennettava mobiilisovellus ja ne tulee kehittää erikseen aina jokaiselle halutulle laitealustalle siihen kohdistetuilla kehittämistyökaluilla ja ohjelmointikielillä. Esimerkiksi Facebook ja LinkedIn ovat natiivisovelluksia. Lataaminen tapahtuu yleensä laitealustan omasta sovelluskaupasta. (Vuorinen 2014; Premayogan 2015.)

Natiivisovellus ja sen sisältö ovat varastoituna paikallisesti laitteeseen, joten kuvat ja muut tiedot latautuvat nopeasti ja animaatiot pyörivät sulavasti (Montecuallo 2014). Sovellukset toimivat offlinessa eli ilman verkkoyhteyttä, mutta tietoa voi hakea myös Internetistä. Sovelluksilla on yhteydet laitealustan tuoreimpiin ohjelmointirajapintoihin ja pystyy täten käyttämään esimerkiksi kameraa, paikallistamisjärjestelmää (GPS), puhelu- ja tekstiviestitoimintoa. Natiivisovellus kykenee käyttämään kaikkia laitteiston resursseja parantaakseen toimintaansa ja ovat täten tehokkaampia ja nopeampia kuin muut sovellusmuodot. (Vuorinen 2014; Kiran 2013.)

#### 3.1 Android

Android on Googlen mobiililaitteille suunnattu käyttöjärjestelmä. Android mobiilisovellukset ohjelmoidaan Java-ohjelmointikielillä ja kehitysalustana yleensä käytetään joko Eclipseä tai Androidin omaa Android Studiota. (Kiran 2013.) Molemmat kehitysalustoista ovat avoimen lähdekoodin projekteja ja ladattavissa ilmaiseksi valmistajansa verkkosivuilta. (Android Studio n.d.; Eclipse n.d.)

Android mobiilisovelluksen voi julkaista eri tavoilla, kuten lähettämällä sen sähköpostilla, pistämällä verkkosivulle tai laittamalla esimerkiksi Google Play -kauppaan (Publishing Overview n.d). Julkaisu vaatii Google Play –kehittäjätilin, Kehittäjien jakelusopimuksen hyväksynnän ja \$25 rekisteröintimaksun. (Google Playn kehittäjäkonsolin käyttö n.d.) Google ei ota kehittäjän myymistä sovelluksista tai niiden sisällä tehdyistä ostoista rojalteja. (Google Play -kehittäjien jakelusopimus 2015.)

Mikäli käyttäjä haluaa veloittaa sovelluksestaan, sovelluksen sisäisistä ostoksista tai tilausmaksuista, tulee tehdä Google Payments Merchant –käyttäjätunnus. Tämän tunnuksen voi tehdä koska vaan ja se vaatii vain yritystiedot. (Get Started with Publishing n.d.)

Sovelluksen julkaisu Google Playssä vaatii Googlen automatisoidun testauksen läpi pääsemistä. Testissä tarkastetaan, ettei sovelluksessa ole viruksia tai haittaohjelmia. Sisällössä ei saa olla myöskään muita sääntöjen vastaisia rikkomuksia. (Perez 2015.)

#### 3.2 iOS

iOS on Applen mobiililaitteille suunnattu käyttöjärjestelmä, jota käytetään vain kahdella alustalla: iPhoneissa ja iPadeissa. Ohjelmointikielissä on kaksi vaihtoehtoa: Objective-C ja Swift. Kehitysalustana käytetään Applen

Xcodea, joka toimii vain OS X käyttöjärjestelmällä. Xcoden saa ladattua ilmaiseksi valmistajan sivuilta.

Kun liittyy Applen kehittäjäjäseneksi, pääsee myymään tai jakamaan sovelluksia App Storessa. Jäsenyys maksaa organisaatioille \$99 vuodessa (Apple Developer Program) tai \$299 vuodessa (Apple Developer Enterprise Program) jos haluaa organisaation kehittämät mobiilisovellukset jakeluun pelkästään omille työntekijöille käyttöön. (Choosing a Membership n.d.) Apple ei ota rojalteja myydyistä mobiilisovelluksista, mutta ottaa 30 % sovelluksen sisäisistä maksuista. (In-App Purchase for Developers n.d.)

Kun iOS-sovellus on valmis ja testattu, se tulee lähettää kauppaan iTunes Connectin avulla. iTunes Connect on työkalu, jolla voi merkitä oman appsinsa tiedot kauppaan. Se tallettaa metatiedot, eli kuvailevan ja määrittelevän tiedon sovelluksesta, Xcodella lähettämistä sovelluksen versioista. Sovelluksen voi tämän jälkeen lähettää testattavaksi TestFlightilla, jolla voi kutsua testajia osallistumaan beta-testaukseen. Testauksen pystyy suorittamaan joko 25 sisäisellä testaajalla tai 1000 ulkopuolisella testaajalla. Sisäiset testaajat ovat iTunes Connectin käyttäjiä. Ulkoiset testaajat ovat saaneet sähköpostikutsun testaukseen. Lopuksi tulee päästä läpi iTunes Connectin validointi testistä. (Submitting Your App to the Store n.d.)

Validointi suoritetaan Archives-ohjelmassa Xcodessa. Apple Developer Enterprise Program -lisenssillä toimivat eivät pysty validoimaan sovelluksiaan ja jakelu tapahtuu kaupan ulkopuolella. (Validating Your App n.d.; Distributing Apple Developer Enterprise Program Apps n.d.)

### 3.3 Windows Phone

Windows Phone on Microsoftin valmistama mobiilikäyttöjärjestelmä, jota käytetään Microsoftin älypuhelimilla. Ohjelmointikielenä käytetään C# tai C++, ja kehitysalustana Visual Studiota. (Kiran. 2013)

Visual Studiosta saa ladattua ilmaisen Community-version valmistajan verkkosivuilta, joka kelpaa mobiilisovellusten kehittämiseen. Visual Studio Professional 2015 standalone-lisenssi maksaa \$499. Microsoft Developer Networkin kanssa lisenssi maksaa \$1199 ja kuukausimaksun joka vaihtelee \$50–150 välillä paketin mukaan. (Visual Studio Purchasing Options n.d.; Subscriber Benefits n.d.) HAMKilla on erillinen jatkuvasti muuttuva ja päivittyvä laskutus Visual Studiosta (Rintala, haastattelu 17.2.2016).

Sovellusten jakelu tapahtuu Windows Phone – kaupassa. Sitä varten tulee luoda kehittäjätili, jonka rekisteröinti maksaa yksityiselle henkilölle 14 € ja yritykselle 75 €. (Account types, locations and fees n.d.) Windows Phone -sovelluksen julkaisu tehdään Windows Dev Centerin kojelaudalla (engl. Dashboard), jossa pystyy hallitsemaan sovelluksia (Publish Windows apps n.d.). Julkaisu vaatii sertifiointiprosessin läpäisyä. Sertifiointiprosessi alkaa alkukäsittelyllä, jossa käydään sovelluksen paketit läpi ja tarkistetaan ne virheitä. Tämän jälkeen siirrytään sertifiointiin, jossa suoritetaan useita testejä. Testeihin kuuluu turvallisuustestit, testit teknisen toimivuuden ja sisällön kannalta. Turvallisuustesteissä tarkistetaan sovellus virusten ja muiden

haittaohjelmien varalta. Sertifiointitestien jälkeen sovellus on valmis julkaisuun. (The app certification process n.d.)

Julkaisua ennen tulee varata sovelluksen nimi ja päättää hinta, jakelu ja näkyvyys, sekä muita asetuksia. Julkaisusta pitää löytyä seuraavat asiat: hinta, kategoria, ikäluokitus, sovelluspaketit, sovelluksen kuvaus ja joihinkin sovelluksiin yksityisyyskäytäntö. Windows Dev Centeristä löytyy Packages-sivu jonne voi ladata sovelluspaketit. (App submission n.d.; Upload app packages n.d.)

Ennen virallista julkaisua sovelluksen saa laitettua suljettuun testaukseen piilottamalla sovelluksen ja estämästä sen saannin. Sovelluksen saa kuitenkin ladattua Windows 10 laitteille eräänlaisilla mainoskoodeilla (promotional code), joita kehittäjä voi halutessaan luoda. Sovelluksen voi myös jakaa kohdistetulle ryhmälle Windows Dev Centerin kautta. (Beta testing and targeted distribution n.d.)

## 4 HYBRIDISOVELLUKSET JA CROSS-PLATFORM KEHITTÄMINEN

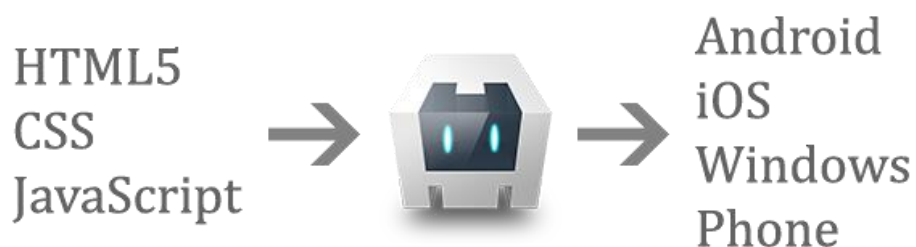
Tämä luku käsittelee hybridisovelluksia ja cross-platform menetelmiä, eli sovelluksien kehittämisestä jotka toimivat usealla ekosysteemillä. Hybridisovellukset rakennetaan käyttäen HTML5-tekniikoita, jotka toimivat kaikilla alustoilla. Sovellus käyttää WebView-näkymää, ”joka on kuin sovellukseen sisäänrakennettu Internet-selain fullscreen moodissa, eli ilman osoitepalkkia, takaisin-nappia ja muita selaimen toimintoja.” (Vuorinen 2014.) Kun haluaa käyttää laitteen eri toimintoja, kuten kameraa, tarvitaan pääsyä laitteen rajapintoihin. Yhteydenotto näihin rajapintoihin tapahtuu JavaScriptillä. (Vuorinen 2014.)

Tietotekniikan tutkimus- ja konsulttiyritys Gartner (2013) ennustaa, että vuonna 2016, 50 % mobiilisovelluksista ovat hybridisovelluksia.

### 4.1 Sovelluskehikset ja pakkaustyökalut

Tässä aluvuossa käsitellään, mitä sovelluskehikset ja sovelluksen pakkaustyökalut ovat ja millaisia vaihtoehtoja niissä on. Sovelluskehys on runko, jonka päälle rakennetaan ohjelma tai sovellus. Hybridisovelluksen rakentamiseen on useita sovelluskehiksiä, joista suosituimpia ovat: IONIC, Appcelerator Titanium ja Sencha Touch (Raj 2014).

Sovellus pakataan natiivisovellukseksi käyttäen pakkaustyökalua. iOS-sovellukset pakataan IPA-tiedostoksi (iOS Application Archive), Android-sovellukset APK-tiedostoksi (Android Package) ja Windows Phone-sovellukset XAP-tiedostoiksi (Application Package) (Trice 2012). Pakkaamisen jälkeen sovellus on valmis lisättäväksi sovelluskauppaan (Vuorinen 2014). Yleisin ja tunnetuin pakkaustyökalu on Apache Cordova eli entinen PhoneGap (Reynolds 2014).



Kuva 2. Ohjelmointikielen, pakkaustyökalun (Apache Cordova) ja ekosysteemien suhde.

Ionic on ilmainen avoimen lähdekoodin SDK. Se on rakennettu käyttäen AngularJS ohjelmistokehystä ja CSS-muotoilukielen laajennusta Sass. Kehittäminen ohjelmalla vaatii vain tuntemusta HTML-, CSS- ja JavaScript-kielistä. Windowsia käyttäville kehitysalustaksi suositellaan Visual Studiota johon saa ladattua Ionicin ja pakkaustyökalu Apache Cordovan. iOS kehitys vaatii tietokoneen, jossa on Mac OSX käyttöjärjestelmä. (Raj 2014; Getting Started with Ionic n.d.)



Sencha Touch on Senchan ilmainen JavaScript-sovelluskehys. Touchilla kehittäminen vaatii Java Runtime Environmentin (JRE) ja Rubyn. JRE:tä tarvitaan Sencha Cmd:n, eli Senchan komentokehoteen pyörittämiseen. iOS mobiilisovelluksen voi rakentaa JRE:n avulla, mutta Android sovellukset vaativat Java SDK:n. Rubyä taas tarvitaan Sencha Touchin CSS:n luontiin. (Getting Started with Sencha Touch n.d.)

AppGyverillä on kolme eri sovelluskehystä: Steroids, Supersonic ja Composer 1. Steroids on komentokehote (Command-Line Interface) jolla voi luoda uusia projekteja, ajaa paikallisia kehittämisspalvelimia, debugata projekteja ja luoda hybridisovelluksia. (Steroids n.d.)

Supersonic on käyttöliittymän sovelluskehys, jonka avulla voi luoda hybridisovelluksia. Se käyttää optimoitua versiota AngularJS:stä tukena, vaikka on muuten täysin itsenäinen sovelluskehys. (Supersonic n.d.)

Composer 1 on bootstrap eli HTML5, CSS ja JavaScript sovelluskehys. Sillä voi luoda hybridisovelluksen käyttöliittymän ja logiikan graafisessa käyttöliittymässä. Nämä tuotteet ovat ilmaisia ja ladattavissa valmistajan sivuilta. (Composer n.d.)

Vaikka hybridisovelluksen asennus tapahtuu samalla tavalla kuin natiivisovelluksen, ja se on pakattu natiiviksi, se ei silti ole natiivisovellus. Selainmoottori on ohjelma, jonka tehtävä on noutaa dataa Internetistä ja saada HTML/CSS-kielinen koodi ja muuntaa sen verkkosivuksi. Toiminnot suoritetaan alustan selainmoottorilla ja lisää uuden kerroksen käyttäjän ja sovelluksen väliin täten hidastaen toimintaa. Hybridisovelluksen suorituskyky ei siis vastaa natiivisovellusta. (Kiran 2013.)

Taulukko 1. Älypuhelimien selainten selainmoottorit ja niiden alustat (Web browser engine n.d.)

Alustat	Selain	Selainmoottori
Android	Android Web Browser	WebKit
iOS	Safari	WebKit
Windows Phone	Internet Explorer	Trident
Kaikki alustat	Google Chrome	Blink

## 4.2 Rapid Application Development

Rapid Application Development, eli RAD, on nopean kehittämisen ja korkean laadun tuottamisen konsepti, joka saavutetaan tiettyjen menetelmien avulla. Näihin menetelmiin kuuluu esimerkiksi vaatimusten määrittäminen kohderyhmien tai työpajojen avulla. Sovellusta kehittäessä tulisi luoda prototyyppi ja aikainen, uudestaan toistettava testausmalli, ja ohjelmistokomponentteja, eli esimerkiksi koodia, tulisi käyttää uudelleen. Aikataulu tulisi olla tahditettu kiinteästi ja siinä tulee ottaa huomioon seuraavan version parannukset. Muodollisuuksia tulisi vähentää tarkasteluissa ja kommunikoinnissa tehokkuuden vuoksi. (Rouse 2007.)

### 4.2.1 OutSystems Platform

OutSystems Platform on RAD-työkalu, jolla voi rakentaa sekä ottaa käyttöön hybridisovelluksia iOS, Android ja Windows Phone mobiililustoille, sekä verkkosivuja. Ohjelmoinnin määrää on vähennetty graafisella käyttöliittymän kehitysympäristöllä. Sovellukset rakennetaan OutSystemsin Silk UI – sovelluskehiksen päälle, joka on kosketuskäyttöön sopiva ja tekee sovelluksen responsiiviseksi. (OutSystems n.d.; OutSystems Evaluation Guide a n.d.)

Poikkeuksena useimpiin hybridisovellusten kehittämistyökaluihin, OutSystemsillä tehdyt sovellukset tukevat kaikkia mobiililaitteiden natiivitoimintoja. Näitä toimintoja ei tarvitse ohjelmoida natiivisti, sillä OutSystems on tehnyt sen valmiiksi. (OutSystems Evaluation Guide b n.d.; outsystems 2015.)

OutSystems Now on avoimen lähdekoodin projekti, johon käyttäjä saa ladata haluamansa OutSystemsillä tehdyt sovellukset. Käyttäjä voi rakentaa sovelluksesta yritykselleen sopivan version ja lisätä sinne tarvittavat sovellukset. OutSystems Now mahdollistaa sen, että kaikki platformilla tehdyt sovellukset ovat yhden yhteisen kirjautumisen takana. Ylläpitäjä voi päättää organisaationsa sisällä mitkä sovellukset ovat saatavilla ja keille. palvelun käyttäminen vaatii palvelimen joka voi löytyä omasta takaa. (OutSystems Now n.d.; OutSystems Evaluation Guide c n.d.; OutSystems Evaluation Guide d n.d.)

OutSystems Platformin tilausmaksu suoritetaan vuosittain. Hinta määräytyy alustan kapasiteetin mukaan. Tämä lasketaan koko OutSystems sovelluksen sisällön koon mukaan ja kyseisen sisällön käyttäjien määrän mukaan. OutSystemsillä on myös ilmainen testausympäristö, jossa on rajatut ominaisuudet. (OutSystems Platform Subscriptions n.d.)

Maksava asiakas saa useampia käyttäjiä projekteihin, julkaisumallin hallintatyökalut, sovellusportfolion hallintatyökalut, monitorointi- ja toimivuuden analysointityökalut, useita kehitys- ja testausympäristöjä ja tuen sähköpostin ja puhelimen välityksellä. Julkaisumallin hallintatyökaluilla voi hallita sovelluksen infrastruktuuria sovelluksen tietoturvan ja toimivuuden eristämiseksi. Sovellusportfolion hallintatyökaluilla voi määrittää ketkä pääsevät muuttamaan sovelluksen tai palvelun sisältöä. Monitorointi- ja analysointityökaluilla voi tarkkailla sovellusten terveyttä, eli kuinka hyvin ne toimivat, ja löytää virheitä. (OutSystems Platform Subscriptions n.d.)

OutSystemsin hinnoittelu perustuu platformin käyttöön. Peruslisenssiin sisältyy tietty määrä AO:ita, käyttäjiä ja ympäristöjä. Tämän jälkeen asiakas voi hankkia lisää näitä tarpeensa mukaan. AO:lla tarkoitetaan Application Objecteja eli tietokantatauluja, web-sivuja ja integraatiopisteitä. Sillä saa kuvattua hyvin sovelluksen laajuutta. Käyttäjiin voi kuulua esimerkiksi kehittäjät, työntekijät ja ulkoiset käyttäjät. Käyttäjätyyppit hinnoitellaan eri tavoilla. Peruslisenssin hinnoittelu suoritetaan asiakaskohtaisesti. (Kiviranta, Haastattelu 29.2.2016.)

### 4.2.2 Mendix Platform

Mendix Platform on OutSystemsin kaltainen RAD-työkalu, joka on tarkoitettu yritysten käyttöön (Mendix n.d). Sillä voi rakentaa ja ottaa käyttöön hybridisovelluksia sekä mobiililaitteille, että tietokoneelle. Mendixillä tehdyt sovellukset käyttävät sovelluskehityksenään Mendix UI Frameworkia. Tämä sovelluskehitys käyttää myös Ionic-kappaleessa mainittua CSS-muotoilukielen laajennusta Sass. Mendix Platform mahdollistaa visuaalisen sovelluksen kehittämisen ja projektin hallinnan siihen tarkoitetuilla työkaluilla. (Platform Overview n.d, Setup Mendix UI Framework with Koala n.d.)

Mendixillä on kolme eri versiota: Free, Pro ja Enterprise. Free-version saa nimensä mukaisesti ilmaiseksi. Se sisältää kaiken tarvittavan sovelluksien suunnitteluun ja rakentamiseen. Ilmaisversiossa saa vain yhden kehitysympäristön ja 10 sisäistä käyttäjää per sovellus. Tuen joutuu itse hakemaan Community Supportin puolelta. Jokaisessa versioissa voi rakentaa niin monta sovellusta kuin itse haluaa. (Editions and Pricing n.d.)

Pro-version hinta on 1250 € per kuukausi, joka laskutetaan vuosittain. Kuten Enterprise-versio, tämä sisältää testi ja tuotanto ympäristöt ja mahdollistaa 50 sisäisen käyttäjän tai 500 ulkoisen käyttäjän määrän. Version käyttäjät saavat platform tason tukea. (Editions and Pricing n.d.)

Enterprise-version hinta on 1650 € per kuukausi, joka laskutetaan vuosittain. Enterprise-versio sisältää Platinum-tason tuen, rajattoman määrän kehitysympäristöjä ja muita ominaisuuksia vain kyseisen version omistaville. (Editions and Pricing n.d.)

### 4.2.3 AppGyver ja Composer 2

AppGyver tuottaa yrityksille suunnattuja mobiilisovellusten kehitystyökaluja, joilla voi luoda hybridisovelluksia ja web appeja. Heidän tuotteisiinsa kuuluu Composer 2, Smart Forms, Data Access ja Workflow. (About AppGyver n.d.)

Composer 2 on työkalu, jolla suunnitellaan, rakennetaan ja laitetaan tuotantoon mobiilisovelluksia. Tämän työkalun käyttäminen ei vaadi ohjelmointia, vaan sovellukset tehdään graafisessa kehitysympäristössä. Sillä tehdyt sovellukset koostuvat datasta, käyttöliittymästä ja säännöistä, joiden yhteistyöllä määritetään sovelluksen toiminnot. (Composer 2 n.d.)

Smart Formsin tarkoitus on saada asiakkaat hankkiutumaan eroon paperilomakkeista ja siirtää ne sähköiseen muotoon. Tätä voi käyttää yhdessä Composerin kanssa sovellusta luodessa. Lomakkeelle kerätyn tiedon saa siirrettyä sähköisesti reaaliajassa palvelimelle. (Smart Forms n.d.)

Data Access on yrityksen datan käsittely työkalu, jolla voi integroida datan sovellukseen rajapintojen tai täyden systeemin integroimisen avulla. Datan käsittelyyn käytetään Two-way connectivityä, joka pitää datan ajan tasalla.

Sovellukset toimivat suorana laajenuksena backendistä ja palveluista. (Data Access n.d.)

Workflowin avulla pystyy luomaan prosessidiagrammeja ja käyttämään niitä käytännössä mobiilisovellusta luodessa. Prosessi diagrammeilla luodaan logiikka sovelluksen toiminnoille ja niiden kulku. (Workflow n.d.)

AppGyverin tuotteilla kehittäminen ilmaista. Käyttäjä voi käyttää niin paljon aikaa tuotteensa kehittämiseen kuin haluaa, mutta ei pysty julkaisemaan sitä ilman Production-lisenssiä, jonka hinta alkaa \$ 9 per kuukausi ja käyttäjä. Production paketilla sovelluksen voi pistää testaukseen oikeille käyttäjille 30 päivän jaksolle, jonka jälkeen sovelluksen voi julkaista. Sovelluksen käyttäjämäärää ei ole rajoitettu, mutta kasvattaa hintaa. Hintakatto per sovellus on \$1499 per kuukausi. Tukea saa foorumeilta ja sähköpostitse. Jokaiselle appille tulee oma laskutus. (Pricing n.d.)

AppGyverillä on myös Enterprise -lisenssi. Enterprise lisenssiin kuuluu muun muassa koulutus ohjelman käyttöön, omistautunut tuki ja sovelluksen yhdessä kehittäminen ja konsultointi. Hinta tähän tulee neuvotella erikseen AppGyverin kanssa. (Pricing n.d.)

### 4.3 Muita cross-platform menetelmiä

Toinen tapa kehittää hybridisovelluksia on esimerkiksi ohjelmilla Xamarin ja Windows Bridge. Nämä ohjelmat eivät käytä HTML-tekniikoita sovel-luskehitykseen vaan sovellukset kirjoitetaan yhdellä kielellä ja käännetään muihin alustoihin sopivaksi. Nämä sovellukset ovat natiivisovelluksia. Ne käyttävät laitteen resursseja toimintojensa suorittamiseen selainmoottorin sijasta. (Introduction to Mobile Development n.d.)

#### 4.3.1 Xamarin

Xamarin käyttää C#-ohjelmointikieltä, luokkakirjastoa ja runtimea, jotka toimivat kolmella eri mobiilialustalla: iOS, Android ja Windows Phone. Windows Phone käyttää C#-ohjelmointikieltä jo valmiiksi, joten Xamarinilla on kaksi markkinoitavaa tuotetta: Xamarin.iOS ja Xamarin.Android. Molemmat ovat rakennettu Mono:n päälle, joka on avoimen lähdekoodin versio .NET-ohjelmistokehyksestä. (Introduction to Mobile Development n.d.)

iOS:llä Xamarinin Ahead-of-Time –kääntäjä kääntää Xamarin.iOS-sovellukset natiiviksi ARM-kasauskoodiksi. ARM-kasauskieli on alemman tason konekieli, joka on raakaa 1:tä ja 0:aa mitä laitteen prosessori lukee (Galloway 2013). Androidilla Xamarinin kääntäjä kääntää ensin Intermediate Language -ohjelmointikieleen, joka sitten ”Just-in-Time” -kääntää natiivikasaukseen kun sovellus käynnistyy. Just-in-Time –kääntäminen tarkoittaa kääntämistä, jossa kääntäminen tapahtuu ohjelmaa ajaessa. Kun Xamarin sovellukset ovat käännetty, lopputulos on joko app-tiedosto iOS:lle tai apk-tiedosto Androidille. Näitä tiedostoja on mahdoton erottaa normaali-

leista natiivisovelluksista, jotka on rakennettu alustojen vakio ohjelmointiympäristöissä. Sovelluksen käyttöönotto tapahtuu samalla tavalla kuin natiivisovelluksen. (Introduction to Mobile Development n.d.)

Xamarinissa on iOS ja Android ekosysteemeille valmiit sidokset (binding) alustojen kehitystyökaluihin (SDK). Sidoksilla tarkoitetaan ohjelmointikielen yhdistämistä kehitystyökalun kirjastoon (library), jonne on ohjelmoitu erilaisia käyttäytymistomintoja. Objective-C-, Java- tai C/C++-ohjelmointikielillä kirjoitettujen olemassa olevien iOS ja Android kirjastojen käyttäminen on mahdollista. Xamarin tarjoaa myös modernit integroidut kehitysympäristöt (IDE) sovellusten kehittämiseen. Max OS -käyttöjärjestelmän käyttäjät voivat käyttää Xamarin Studiota. Windows -käyttöjärjestelmän käyttäjät voivat taas käyttää sekä Xamarin Studiota, että Visual Studiota. (Introduction to Mobile Development n.d.)

Xamarinin hinta Indie-paketti yksityiselle kehittäjälle on \$25 kuukaudessa. Business-paketti on tarkoitettu organisaation käyttöön ja se maksaa \$999 vuodessa. Tämä paketti sisältää myös Visual Studio integraation. Kallein paketti on Enterprise-paketti joka maksaa \$1899 vuodessa. (Simple and transparent pricing. n.d.)

### 4.3.2 Windows Bridge for iOS

Tämä luku käsittelee elokuussa 2015 julkaistua Windows Bridge for iOS siltausta. Windows Bridge for Android on vielä suljetussa testauksessa, joten sitä ei käsitellä tässä luvussa (Windows Bridges for Android n.d). Windows Bridge for iOS on Microsoftin kehittämä avoimen lähdekoodin projekti, joka kääntää valmiiksi kirjoitetun iOS-sovelluksen natiiviksi Windows Phone -sovellukseksi. Itse ohjelmointi tehdään Objective-C:llä käyttäen Windowsin rajapintoja. (Windows Bridges for iOS n.d.; Salmaan 2015.)

Objective-C koodin kääntämisessä käytetään kustomoitua kääntäjää, joka koostuu Microsoftin C ja C++ kääntäjistä clang ja cl. Käännetty koodi ja syntynyt lopputulos linkitetään yhteen. Tämä mahdollistaa Objective-C:n ja C++/CX kielten yhdessä olemisen saman projektin alla. Kielet voivat myös kutsua toisiansa käyttäen C tai C++ rajapintoja. (Compiler Options n.d.; Salmaan 2015.)

Clang -kääntämistyökalu ei ymmärrä CX -ohjelmointikielen laajennuksia, joita tarvitaan Windows rajapintojen kutsumiseen. Käyttäjän tulisi luoda cpp tiedostoja joissa manuaalisesti yhdistetään Objective-C ja C++/CX maailmat, jotta voi käyttää Universal Windows Platform -rajapintoja. Tämän sijaan käytetään projektioita, jotka lähettävät Windowsin rajapintoja toiseen ohjelmointikielen. Tämä mahdollistaa Windows rajapintojen käyttämisen Objective-C:llä. (Salmaan 2015.)

Windows Bridge for iOS on ladattavissa ilmaiseksi GitHubista. Ohjelma mahdollistaa alkuperäisten Xcode projektien siirtämisen ja sitomisen Visual Studioon ja sen kehitystyökaluihin. Tämän takia Windows Bridgen käyttämiseen suositellaan Visual Studiota. (Salmaan 2015.)

## 5 WEB APP

Tässä luvussa käsitellään web appeja, niiden ominaisuuksia ja miten ne eroavat natiivi- ja hybridisovelluksista. Web app on käytännössä responsiivinen verkkosivu, joka siis sopii mobiilialustan selaimeen ja muistuttaa mobiilisovellusta. Web app pystyy myös käyttämään multi-touch toimintoja, mikäli selain tukee niitä. Multi-touch toiminnot ovat sormilla tehtäviä toimintoja, joissa hyödynnetään kosketusnäyttöä, kuten esimerkiksi zoomaus. Kehityksessä käytetään normaaleja web-kehitystekniikoita: HTML5, JavaScript ja CSS. (Vuorinen 2014; Smus 2011.)

Web appit ovat verkkosivuja, joten niiden tulisi toimia jokaisen mobiilialustan selaimella. Koska uusin versio tarvitsee vain ladata palvelimelle, eikä sovelluskauppaan missä tulisi päästä läpi useista testeistä, myös sisällön päivittäminen voi tapahtua reaaliajassa, ja sisältöä ei koske sovelluskauppojen säännöt. Web appit eivät pysty käyttämään kaikkia laitteen ominaisuuksia, muutamaa poikkeusta lukuun ottamatta, toisin kuin hybridi- ja natiivisovellukset. (Vuorinen 2014; Montecuoillo 2014)

Web appeja ei pysty vielä integroimaan natiivilaitteen sovelluksiin ja dataan, kuten iCloudiin ja kontakteihin, jotka toimivat järjestelmätasolla. Web appit eivät pysty myöskään vielä suorittamaan taustaprosessointia tai käyttämään push-notifikaatioita. (Vuorinen 2014; Tello 2015.)

Web appeilla pystyy kuitenkin seuraaviin toimintoihin: GPS, kosketusliikkeet (kuten napautus, nipistäminen, pyyhkäisy), laitteen liikkeiden lukeminen, WebAudio (hoitaa äänen käytön), WebSocketit (muodostaa jatkuvan yhteyden selaimen ja palvelimen välille), push-notifikaatiot eli ilmoitukset (Androidilla), osittainen kameratuki ja mobiililaitteen kotivalikkoon/työpöydälle pikanäppäimen lisäys. (Tello 2015.)

Myös rajallinen offline-tuki on mahdollista web appeissa, mutta se ei vastaa natiivisovelluksien tasoa. Offline-sisältö tallennetaan mobiililaitteen paikallismuistiin ja app cacheen. Tallennettavaksi sisällöksi ei kannata pistää kuvia tai videoita, eikä suuria määriä web appin koodista. Sisällöksi tulisi pistää sen verran koodia, että saadaan web appi käynnistymään. Jos koodissa on virhe tai se on korruptoitunut, se estää sovellusta aukeamasta rikinäisen koodin kanssa. (Andrews 2014.)

Web appin tekoon on hyvä käyttää sovelluskehystä, joka on optimoitu kosketusnäytön käyttöön. Hyviä esimerkkejä kyseisistä ohjelmista ovat jQuery Mobile, Mobile Angular UI ja hybridi luvussa mainitut IONIC ja Sencha Touch. jQuery Mobile ja Mobile Angular UI sopivat myös hybridisovellusten sovellusalustoiksi. Vaihtoehtoisesti luomiseen voi käyttää Rapid Application Development -työkaluja, jos on valmis maksamaan niiden käytöstä.

jQuery Mobile kosketuskäyttöön suunniteltu ilmainen avoimen lähdekoodin sovelluskehys, joka on päivitys normaaliin jQueryyn. Sillä voi tehdä HTML5-pohjaisia responsiivisia käyttöliittymiä verkkosivuille ja älylaitteiden sovelluksille. Vaikka jQuery Mobile on suunniteltu mobiililaitteille, toimii se myös tietokoneiden selaimilla. jQuery Mobile sisältää myös useita

teemoja, joita voi muokata ThemeRoller -ohjelman avulla. (Grisogono 2014; jQuery Mobile n.d.)



Kuva 3. Lähes 70 % maailman verkkosivuista käyttää jQueryä verkkosivuillaan.

(Usage Statistics and market share of JQuery for Websites n.d).

Mobile Angular UI on ilmainen avoimen lähdekoodin projekti, joka käyttää Ionicin tapaan AngularJS ohjelmistokehystä. AngularJS:n lisäksi Mobile Angular UI käyttää front-endin työkalua Bootstrap, joka tarjoaa teemoja ja tekee sovelluksen responsiiviseksi. Mobile Angular UI:n on ladattavissa GitHubista. (Mobile Angular UI n.d; Bootstrap n.d.)

## 6 HAMKIN TULEVIEN MOBIILISOVELLUSTEN LUOKITTELU

Tässä luvussa käydään läpi tämän opinnäytetyön toisessa luvussa mainittujen HAMK:n kehitysjonossa olevien sovellusten luokittelu ja perustelut annetuille luokituksille. Luokittelu tulee tehdä siksi, että mobiilityyppien vertailussa ei tarvitse käsitellä yksittäisiä sovelluksia vaan voidaan pohtia kokonaisuuksia. Samaa luokittelua voidaan soveltaa myös jatkossa, kun keksitään tarpeita uusille sovelluksille. Luokittelu tehdään sovelluksen sisällön ja käytettävyyden perusteella.

### 6.1 Paljon mediaa sisältävät palvelut

Paljon mediaa sisältävillä palveluilla tarkoitetaan palveluita, joissa on paljon kuvia, ääniä tai videoita. Suuri määrä mediaa sovelluksessa tekee siitä raskaamman. Natiivi- ja hybridisovellukset sopivat niihin parhaiten, etenkin natiivisovellukset. (Vuorinen 2014.)

Opiskelijan käyttöliittymä opetuksen videopalveluun kuuluu tähän luokkaan, koska palvelu keskittyy videoihin. Moodlen mobiilikäyttöliittymä kuuluu tähän luokkaan, koska Moodlessa voi olla paikoitellen melko paljon kuvia ja muuta mediaa.

### 6.2 Aktiivisesti käytössä olevat palvelut

Useimmat mobiililaitteen käyttäjät omistavat jo kaikki heidän tarvitsemansa mobiilisovellukset. Tästä syystä mobiilisovellusten lataaminen on vähentynyt. Sovellukselle tulee olla aitoa tarvetta, että se on lataamisen arvoisen. Harva jaksaa ladata ja asentaa mobiilisovellusta yhden tai kahden käyttökerran takia. (Li 2014.)

Ohje- ja opetusvideoiden opiskelijoiden käyttöliittymä on palvelu, jolle voi olla paljon aktiivista käyttöä, riippuen miten tämä palvelu on otettu mukaan HAMK:n kurssien sisältöön. Moodlen mobiilikäyttöliittymä opiskelijalle voi olla myös aktiivisessa käytössä, mikäli opiskelijalla ei usein ole koulutehtäviä tehdessä tietokonetta lähellä. Tämä riippuu paljon koulutusalaista ja kurssista. Opetuksen mobiilipalvelun sisältö on vielä suunnitteluvaiheessa, ja käytön aktiivisuus riippuu siitä, miten sitä käytetään. Mikäli sovellus sisältää syrjäytymistutkan, sen tulisi olla aktiivisessa käytössä.

### 6.3 Laitteen natiivitoimintoja käyttävät palvelut

Natiivi- ja hybridisovellukset pääsevät käsiksi kaikkiin laitteiden toimintoihin, kuten kameraan tai GPS:ään. Vain osa niistä toimii web-sivustoissa, joten natiivitoimintojen tuki voi tulla esteeksi sovellusta kehittäessä.

Käsiteltävissä palveluissa ei missään ollut suoranaista käyttöä mobiililaitteiden natiivitoiminnoille. HelpDeskin asiakasliittymään tosin voisi yhdistää kamerasovelluksen, jos haluaa lähettää virheraportin yhteydessä tilannetta tarkentavan kuvan.



### 6.4 Yleisön koko

Mikäli mobiilisovelluksen sisältö on suunnattu kaikille HAMK:n toiminnasta kiinnostuneille henkilöille ja ryhmille, kannattaa harkita web appin tekemistä. Suuremman yleisön on helpompi lähestyä web appia kuin ladattavaa mobiilisovellusta. Web appiin on myös helpompi ohjata ihmisiä, sillä se on verkkosivusto. Tässä luvussa luokitellaan sovellukset joko pienelle tai suurelle yleisölle kohdistuvaksi. Pienen yleisöön voisi kuulua vain HAMK:n opiskelijat ja/tai henkilökunta, ja suureen yleisöön kaikki HAMK:n toiminnasta kiinnostuneet henkilöt ja ryhmät.

Kehitysjonon kaikki mobiilipalvelut ovat suunnattu HAMK:n opiskelijoille ja henkilökunnalle, eli täten pienelle yleisölle. Vapaiden tilojen hakupalvelu, vaikka onkin avointa dataa ja sen yleisön laajuutta ei ole rajoitettu, on suunnattu pelkästään HAMK:n opiskelijoille ja henkilökunnalle. Pois lukien vapaiden tilojen hakupalvelu, kehitysjonon sovellusten sisältö ei ole julkista.

### 6.5 Kirjautumista vaativat palvelut

Tähän luokkaan kuuluu palvelut, jotka vaativat joko kevyen tai raskaan kirjautumistoiminnon. Kevyt kirjautumistoiminto on esimerkiksi opiskelijanumero + PIN-koodi. Raskas kirjautuminen käyttää taas HAMK:n käyttäjätunnuksia. Kirjautumistoimintoa voi käyttää missä tahansa sovellustyyppissä.

Tähän luokkaan kuuluu kehitysjonosta kaikki muut paitsi vapaiden tilojen hakupalvelu, jonka sisältö on avointa. Suurin osa palveluista vaatii raskaan kirjautumisen, mutta opintosuoritusten mobiilipalvelu voi toimia kevyellä kirjautumisella.

### 6.6 Verkkoyhteyttä vaativat palvelut

Tähän luokkaan kuuluu kaikki palvelut, jotka hakevat tai lähettävät sisältöä verkkoyhteyden välityksellä. Nämä palvelut vaativat verkkoyhteyden jokaisen sovelluksen toiminnon käyttämiseen eivätkä sisällä offline-tukea. Kaikkiin sovellustyyppisiin saa offline-tuen, mutta natiivi- ja hybridisovellukset suoriutuvat siinä parhaiten.

Jokainen kehitysjonossa oleva palvelu vaatii verkkoyhteyden toimiakseen. Jotkut niistä palveluista ovat kuitenkin vähemmän riippuvaisia siitä. Joihinkin palveluihin voi lisätä offline-tuen, jotta voidaan käyttää aiemmin ladattua sisältöä. Esimerkki tällaisesta palvelusta voisi olla opiskelijoiden ohje- ja opetusvideopalvelu, jossa videoita voisi ladata paikalliseen muistiin myöhemmin katsottavaksi.

## 7 HYBRIDI VAI NATIIVI?

Tässä luvussa pohditaan miksi hybridisovellukset sopisivat HAMK:n tarkoituksiin paremmin kuin natiivisovellukset. Tarkoitus on vertailla hybridi- ja natiivisovelluksien sopivuutta luvussa 6 tehtyjen luokkien avulla ja pohdita, millä työkaluilla ne olisi hyvä tehdä.

### 7.1 Natiivisovellus ja sen haitat

Natiivisovellusten kehittäminen tulee kalliiksi, jos suuntaa ne kolmelle eri alustalle. Natiivisovellusten kehittämistyökalut ovat pääosin ilmaisia, mutta sovelluksen kehittämiseen tulee palkata osaavia kehittäjiä joiden palvelut maksavat. HAMK on todennut, ettei heillä ole kyvykkyyttä tai resursseja tuottaa eikä varaa ostaa tai teettää uusia natiivisovelluksia.

Lisää kustannuksia tulee alustojen kehittäjälisensseistä, mutta niistä joutuu maksamaan joka tapauksessa, olkoon kyseessä hybridi- tai natiivisovellus. Androidilta ja Windowsilta tarvitaan normaalit kehittäjälisenssit ja Applelta joko Pro tai Enterprise -lisenssit riippuen haluaako jakaa sovelluksia pienessä vai julkisessa piirissä. HAMKille nämä summat voivat olla suhteellisen pieniä.

### 7.2 Cross-platform työkalut

Cross-platform työkaluissa on paljon erilaisia vaihtoehtoja. RAD-työkalut vaikuttavat kaikista tehokkaimmilta työkaluilta ketterään mobiilisovellusten kehittämiseen, graafisten kehitysympäristöjen ja ohjelmoinnin tarpeettomuuden ansiosta. Ne ovat kalliita ja niiden hinta skaalautuu usein käyttäjämäärän perusteella ja suoritetaan yleensä kuukausimaksuina. Nämä työkalut ovat suunnattu pääosin yritysten hyötysovellusten käyttöön ja tuovat sen mukana tiettyjä rajoitteita. Esimerkiksi Mendixin sovellusten ulkoisen käyttäjämäärän katto on 500, joka sulkee sen pois vaihtoehdoista, koska HAMKilla on noin 7200 opiskelijaa ja 750 opettajaa ja muuta henkilöstöä (Tietoa HAMKista n.d.). OutSystems Platformilla ja AppGyverin Composer 2 tehdyillä sovelluksilla ei ole käyttäjämäärän kattoa. Molemmilla on ilmainen kokeiluversio maksullisesta versiosta, jossa pääsee itse tutustumaan tarkemmin tuotteeseen. OutSystems Platformin hinta neuvotellaan asiakkaan kanssa erikseen, jonka takia sitä ei pysty vertaamaan Composer 2:n kanssa.

RAD-työvälineiden käyttö tulee opetella, mutta sen on oletettavasti helpompaa, koska niissä ei tarvitse koodata juuri yhtään. HAMK voi kohdistaa resursseja näiden opetteluun tai laittaa opiskelijoille tehtäväksi opetella ja toteuttaa niillä mobiilisovellus kurssin yhteydessä samalla tavalla kuin Oivan kanssa tehtiin.

Mikäli HAMK haluaa mennä täysin ilmaisella kehityslinjalla hybridisovellusten kanssa, lukuun ottamatta laitealustojen kehittäjälisenssejä, kannattaa käyttää ilmaisia sovelluskehyskiä ja pakkaustyökaluja. Sovelluskehyskiissä

on paljon valikoimaa, mutta voi olla viisasta kuitenkin pysyä tunnetuimmissa tuotteissa kuten Ionic, tai AppGyverin vanhemmat ilmaistuotteet kuten Steroids. Paketointi työkaluissa on vähemmän vaihtoehtoja, joten se tulisi tehdä Apache Cordovalla (ent. PhoneGap). Hybridisovellusten kehittämiseen käytetään HTML5-tekniikoita, joiden pitäisi olla jokaiselle ohjelmoijalle tuttuja. HAMK voisi kohdistaa henkilöresursseja tai opiskelijatöitä tähän tutustumiseen ja tuotantoon.

Xamarinin kohdalla vaaditaan tuntemusta natiivisovellusten ohjelmoinnista pelkästään C#-ohjelmointikielellä. Vaikka HAMK on todennut, ettei heillä ole resursseja tai kyvykkyyttä tuottaa natiivisovelluksia, olisi tämä silti harkitsemisen arvoinen vaihtoehto. Sillä kehittäminen maksaa \$1000 - 1800 vuodessa, riippuen lisenssistä, ja vastineeksi saa natiivisovellukset jokaiselle alustalle. Xamarinin opettelun ja sillä kehittämisen työmäärää on vaikeampi arvioida HAMK:n kehitysjonon sovellusten kohdalla.

Microsoft Bridges for iOS:llä voi tehdä Windows Phone sovelluksia käyttäen olemassa olevaa iOS-sovelluksen koodia. Siltaus Androidille on tulos vasta myöhemmin. Tilanne jossa HAMKilla sattuu olemaan sovelluksesta iOS-versio, muttei muille ekosysteemeille, kuulostaa hyvin epätodennäköiseltä. iOS sovelluksen saa käännettyä vain Windows Phonelle, muttei Androidille sopivaksi. Sama logiikka pätee myös Androidin siltauksessa. Kääntämismahdollisuuksien puutteiden takia, tähän ei kannattaisi kohdistaa resursseja. Microsoft Bridges on ilmainen, joten se on helppo ottaa käyttöön, jos tarve löytyy ja on aikaa opetella. Käyttämisen vaikeutta on vaikea arvioida.

### 7.3 Palveluiden tarpeet ja ratkaisut

Palvelut joissa on paljon kuvia ja muuta ladattavaa, ovat raskaita käyttää. Natiivisovellukset pystyvät varastoimaan tämän paikallisesti laitteeseen ja kutsumaan ja lataamaan niitä nopeasti käyttämällä laitteen omia resursseja. Hybridisovellukset käyttävät selainmoottoria, joka ei vastaa natiivisovellusten tehokkuutta. Sisällön, mitä ei haeta Internetistä, saa kuitenkin varastoitua paikallisesti mobiililaitteeseen. Tämä tekee siitä kuitenkin nopeamman kuin mitä se olisi muuten.

Palveluille joille ei oletettavasti ole jatkuvaa käyttöä, ei moni jaksaa ladata. Täten sekä natiivi- tai hybridisovellukset eivät ole välttämättä hyviä vaihtoehtoja. Niistä paras vaihtoehto on kuitenkin hybridisovellus koska sen kehittäminen on helpompaa ja halvaa. Palveluille joille on jatkuvaa käyttöä, olisi hyvä toimia nopeasti ja tehokkaasti. Koska hybridisovellukset ovat lähes yhtä tehokkaita kuin natiivisovellukset, mutta huomattavasti halvempia ja helpompia toteuttaa, ovat ne myös paras vaihtoehto.

Sillä hybridisovellus voi käyttää samoja natiivitoimintoja kuin natiivisovellus, on se myös paras vaihtoehto. HTML5-tekniikoilla kehittäminen on huomattavasti helpompaa. Jokaisessa hybridisovelluksessa ei ole välttämättä edes toimintoja, jotka vaatisivat rajapinnoilla yhdistämistä laitteen toimintoihin.

Suurelle yleisölle, eli tässä tapauksessa kaikille HAMK:n toiminnasta kiinnostuneille osapuolille, kannattaa olla mahdollisimman saatavilla oleva sovellus. Hybridisovellus sopisi tälle ryhmälle paremmin kuin natiivisovellus kustannussyistä, vaikka sekään ei ole paras vaihtoehto.

Pienemmälle yleisölle, hyötykäyttöön tarkoitettu mobiilisovellus olisi hyvä toteuttaa hybridisovelluksena. HAMK:n mobiilisovellusten kehitysjonossa olevat sovellukset ovat kaikki toteutettavissa web-tekniikoilla, joten hybridisovellus olisi parempi vaihtoehto kuin natiivisovellus. Kehitysjonon mobiilisovelluksiin pääsyä tulee rajata usein kirjautumisella, sillä ne ovat suunnattu vain HAMK:n henkilökunnalle ja opiskelijoille. Kirjautumistoimintoa ei ole rajoitettu sovellustyypeillä.

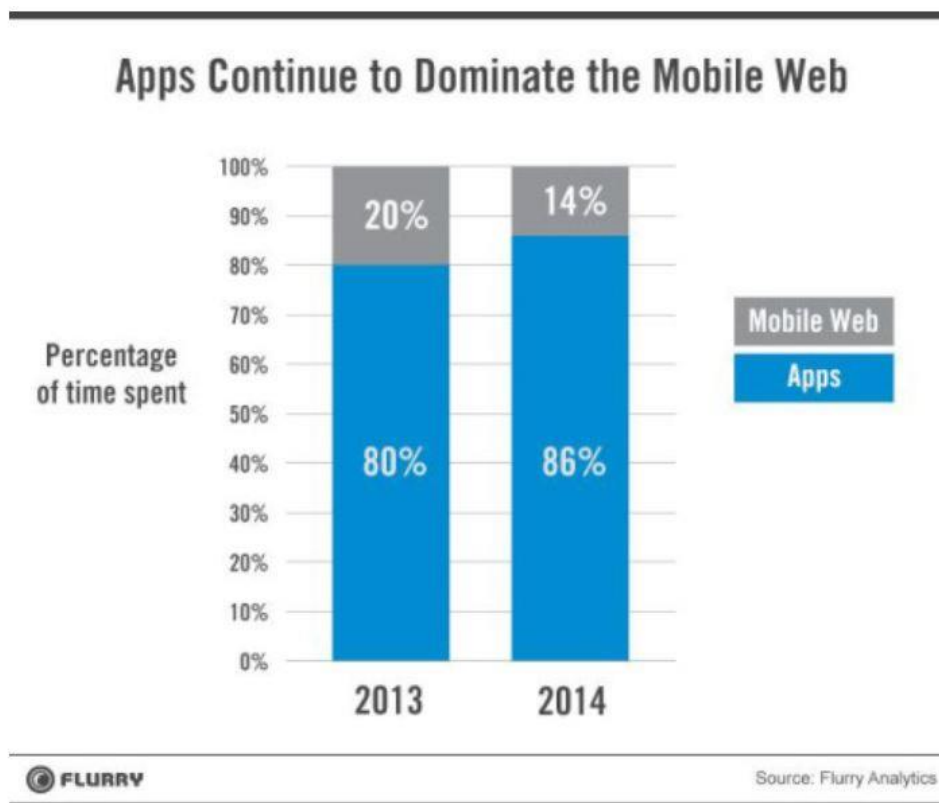
Offline-tuen saa parhaiten natiivisovelluksiin, sillä suurin osa sovelluksista on asennettu paikallisesti puhelimeen. Verkkoyhteyttä tarvitaan kun, sovellusten toimintoihin tarvittua dataa haetaan rajapintojen kautta HAMK:n tietokannoista. Kaikki kehitysjonossa olevat sovellukset vaativat verkkoyhteyden tästä syystä.

## 8 WEB APP VAI NATIIVI?

Tässä luvussa pohditaan miksi web appit sopisivat HAMK:n tarkoituksiin paremmin kuin natiivisovellukset. Tarkoitus on vertailla web appien ja natiivisovelluksien sopivuutta luvussa 6 tehtyjen luokkien avulla ja pohtia millä työkaluilla ne olisi hyvä tehdä.

Web appin etuihin kuuluu se, että niiden kehittäminen on täysin ilmaista ja niiden ei tarvitse käydä sovelluskauppojen validointitestejä julkaisua varten. Niiden kehittämisestä ei tarvitse maksaa kehittäjämaksuja kuten natiivi- ja hybridisovelluksien kehityksestä. Web appien sisältöä on mahdollista päivittää reaaliaikaisesti ja tuoreinta versiota ei tarvitse ladata sovelluskaupoista.

Web appit ovat muuten yleensä hitaita käyttää, jonka takia ihmiset suosivat mieluummin natiivi- tai hybridisovelluksia.



Kuva 4. Vuonna 2014 Yhdysvalloissa noin 86 % ihmisistä käyttää aikansa mieluummin mobiilisovelluksia kuin selaimessa. (Khalaf 2014.)

### 8.1 Web app työkalut

Web appien tekoon tarvitsee vain haluamansa sovelluskehityksen. Mikäli maksaa RAD-työkaluista niin niilläkin voi tehdä, mutta varta vasten web appin tekoon ei niistä kannata maksaa. Sovelluskehitykseksi käy mikä tahansa sovelluskehitys, myös ne mitä käytetään hybridisovellusten tekoon. Harkitseminen arvoisia vaihtoehtoja ovat: Ionic, jQuery Mobile, Sencha Touch ja AppGyverin Supersonic. Parasta olisi valita sovelluskehitys mistä kehittäjällä on eniten kokemusta.

Jos kehittäjällä ei ole kokemusta sovelluskehityksistä, niin olisi hyvä pohtia joko Ionicia tai jQuery Mobilea, sillä ne ovat nimiä jotka nousevat usein esille keskusteluissa web appien kehityksestä. Ionicilla on aktiivisin yhteisö internetissä, josta saa apua tarvittaessa. Sovelluskehityksenä Ionic toimii nopeammin kuin jQuery Mobile. jQuery Mobile tukee kolmannen osapuolen lisäosia, jota Ionic ei tee. jQuery Mobilessa on myös parempi teemavali-koima ja sen käyttämisen oppii nopeasti. Ionicia pidetään silti parempana ja nopeampana sovelluskehityksenä kuin jQuery Mobilea. Jos jQuerystä ei löydy aiempaa kokoemusta, niin Ionic olisi parempi vaihtoehto. (Moderaattori Gajotres, Forum-päivitys 7/2015.)

### 8.2 Palveluiden tarpeet ja ratkaisut

Paljon mediaa sisältävät palvelut ovat raskaita web appeille. Tästä syystä ne tulisi toteuttaa mieluummin natiivi- tai hybridisovelluksena, jos haluaa sovelluksen toimivan nopeasti. Web appeihin saa rajallisen offline-tuen, jossa tallennetaan tietoa paikallisesti puhelimeen, jonka avulla ne toimivat nopeammin. Esimerkiksi videoita ja kuvia ei kuitenkaan kannata tallentaa paikallisesti. Mikään ei muuten estä tekemästä paljon kuvia ja videoita sisältävää palvelua web appina.

Palvelut joille ei ole välttämättä aktiivista käyttöä ja jolle on erittäin pieni käyttäjämäärä, olisi hyvä tehdä web appina, sillä natiivisovellusten kehittäminen tulee muuten kalliiksi. Toinen vaihtoehto niillä on liittää ne suurempiin palvelukokonaisuuksiin, kuten vapaiden tilojen hakutoiminto. Suurelle yleisölle suunnatut palvelut, johon kuuluu kaikki HAMK:n toiminnasta kiinnostuneet henkilöt tai ryhmät, olisi hyvä toteuttaa myös web appina. Web appilla saa tavoitettua suuremman yleisön, kuin hybridi- tai natiivisovelluksella, koska sitä ei tarvitse etsiä ja ladata sovelluskaupasta, ja asentaa. Kirjautumistoimintoa ei ole rajoitettu sovellustyypeillä, joten web appeissa voi olla myös kirjautumistoiminto, mikäli sellaisen päätyy tekemään.

Web appit pääsevät käsiksi vain rajallisesti laitteen natiivitoimintoihin. Ne mitkä vaativat niihin pääsyä, tulisi toteuttaa natiivi- tai hybridisovelluksena. Useimmat HAMK:n kehitysjonossa olevat mobiilisovellukset eivät vaadi pääsyä näihin puuttuviin natiivitoimintoihin, täten ne voi toteuttaa myös web appina.

## 9 LOPPUTULOS

Opintosuoritukset mobiiliin: Hybridisovellus ja web app käyvät molemmat. Hybridisovellus mieluummin, jos palvelu toteutetaan osana suurempaa palvelua. Pienelle palvelulle, jossa on vain yksi toiminto, on hieman turhaa tehdä omaa kokonaista sovellusta.

Vapaiden tilojen hakupalvelu: Tämä palvelu on jo toteutettuna web appina osana suurempaa kokonaisuutta. Tämä voidaan toteuttaa myös hybridisovelluksena, esimerkiksi yhdessä vapaiden tilojen varaupalvelun kanssa.

Ohje- ja opetusvideot – opiskelijan käyttöliittymä: Toiminnollisuutta ajatellen paras vaihtoehto tähän olisi natiivisovellus, mutta koska HAMK ei aio toteuttaa natiivisovelluksia tulee tyytyä hybridisovellukseen, joka ei ole huono vaihtoehto.

HAMKin IT-palveluiden liikennevalot: Tämä palvelu on yksinkertainen ja se voidaan toteuttaa sekä hybridisovelluksena että web appina. Jos palvelu koostuu pelkästään IT-palveluiden liikennevaloista, niin ei välttämättä kannata nähdä vaivaa hybridisovelluksen tekemiseen. Osana suurempaa kokonaisuutta se olisi hyvä toteuttaa hybridisovelluksena.

HelpDeskin asiakasliittymä mobiiliksi: Voi olettaa, että moni opiskelija ei lähetä virheilmoituksia tai tukipyyntöjä montaa kertaa opiskelujensa aikana. Jos käyttäjällä ei löydy palvelulle aktiivista käyttöä, hän tuskin viitsii ladata sitä yhden ilmoituksen takia. Tämän takia palvelu olisi hyvä toteuttaa web appina hybridisovelluksen sijasta.

Moodlen mobiilikäyttö opiskelijalle: Moodlesta on jo responsiivinen versio olemassa, joten hybridisovellus on paras vaihtoehto käytettävyyden kannalta.

Opiskelijan ilmoittautumispalvelu: Tämän palvelun voi toteuttaa hybridisovelluksena ja web appina. Parempi vaihtoehto olisi tehdä hybridisovelluksena sen tehokkuuden takia. Voisi kuvitella, että ilmoittautumispalvelu olisi osana suurempaa sovellusta. Oletettavasti olisi kätevää, jos kaikki opiskelijan ja henkilökunnan tarvitsemat toiminnot olisivat yhden kirjautumista vaativaa sovelluksen takana.

Opetuksen mobiilipalvelu: Näihin mobiilipalveluihin kuuluvat sovellukset voidaan toteuttaa sekä hybridisovelluksena että web appina. Hybridisovellus on parempi vaihtoehto sovelluksen laajuuden takia, sillä suuren osan sovelluksesta saa paikallisesti mobiililaitteeseen.

Jos HAMKin valitsemasta RAD-työkalusta saadaan sovittua oppilaitoksen budjettiin sopiva hinta, niin sillä voi kehittää kaikki hybridisovellukset ja web appit. Muuten hybridisovellukset kannattaa toteuttaa käyttämällä Ionic-sovelluskehystä ja Apache Cordova -pakkaustyökalua esimerkiksi opiskelijatöinä. HAMKin on hyvä myös tutustua itse tähän ratkaisutapaan lä-

hemmin. Web appit saa kehitettyä samoilla työkaluilla. Pienemmissä projekteissa kannattaa harkita jQuery Mobilea sen helppokäyttöisyyden vuoksi. HAMKin Oiva -palvelu on esimerkiksi tehty käyttäen jQuery Mobilea. Moni kehitysjonon sovelluksista on toteutettu valmiiksi verkkosivuina, kuten: HelpDesk ja Moodle. Niiden toimintojen logiikka on siis tehty jo valmiiksi. Mobiilisovellusten käyttöliittymä tulisi tehdä enää sovelluskehityksen kanssa, että niistä tulisi responsiivisia.



## 10 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli auttaa toimeksiantajaa, eli Hämeen ammattikorkeakoulua, suunnittelemaan ja valitsemaan millä menetelmillä he aikovat kehittää mobiilisovelluksensa jatkossa. Tavoitteena oli tuoda esiin asioita joita HAMKilla ei ollut vielä tiedossa, sillä HAMK:n mobiilistrategian kehitystiimi on tutkinut aihetta jo aikaisemminkin.

Mobiilisovellusten on tarkoitus tukea ja tehostaa opiskelijoiden ja henkilökunnan toimintaa. Opiskelijat ja henkilökunta saavat parempia tuloksia aikaiseksi, kun he ovat tyytyväisiä ja tehokkaita. Nämä tulokset ovat perusta HAMK:n toiminnalle mobiilistrategian mukaan.

Natiivisovellukset ovat tehokkaita ja muut sovellusvaihtoehdot eivät ole välttämättä parempia. Natiivisovellusten kehittäminen on kallista, varsinkin kolmelle eri ekosysteemille. Kehittäjiltä se vaatii jokaisen alustan tuntemista. Välillä asiakkaat joutuvat palkkaamaan jokaiselle ekosysteemille oman natiivisovelluksen kehittäjän. Kuten luvussa 2 viitattiin, HAMKilla ei ole aikeita teettää tai ostaa uusia natiivisovelluksia, ellei syytä ole perusteltu erittäin hyvin. HAMKilla on olemassa olevia natiivisovelluksia, joihin tämä sääntö ei koske.

Natiivisovellusten kehittäminen on hankalaa ja kallista, joten cross-platform menetelmät on loistava vaihtoehto. Cross-platform menetelmät ovat kehittyneet pitkälle ja hybridisovelluksia suositaan varsinkin yritysten hyötysovellusten tekoon. Hybridisovellukset eivät ole yhtä tehokkaita toimivuudeltaan kuin natiivisovellukset, mutta kehitysjonossa olevat sovellukset eivät vaadikaan paljoa laitteiston resursseista, lukuun ottamatta opiskelijan videopalvelun käyttöliittymää.

Web appit, vaikka soveltuvat parhaiten pyörittämään vain kevyitä palveluita, eivät ole huono vaihtoehto mobiilisovelluksen kehittämiseen. Web appien mahdollisuuksia kehitetään jatkuvasti ja sen myötä ne pystyvät suorittamaan toiminnoista, jotka ovat olleet aiemmin mahdollisia vain natiivi- ja hybridisovelluksilla.

Jokaisesta sovelluskehiksestä ja työkaluista saisi kirjoitettua oman opinnäytetyönsä, minkä takia niiden käsittely oli vaikeaa. Niistä tuli saada tarpeeksi tietoa, jotta niistä pystyi kertomaan, mutta niitä ei päästy käsittelemään tarpeeksi syvästi. Sovelluskehiksen todellisen eron huomaa vasta kun niitä pääsee käyttämään ja tutkii niiden dokumentaatioita. Internetistä löytyy useita vertailuja niistä, mutta nekään eivät oikeastaan kerro tarkasti minkä takia toinen niistä on parempi. Eräässä vertailussa kerrottiin useita asioita missä jQuery Mobile on parempi kuin Ionic, mutta lopuksi päädyttiin siihen, että Ionic on parempi, ilman mitään varsinaista perustelua sille. Tämä ilmeni monissa muissakin vertailuissa. Sovelluskehiksillä eksperimentointi on helppo aloittaa, sillä ne ovat ilmaisia.

Opinnäytetyön aihe oli erittäin mielenkiintoinen ja voidaan olettaa, että tutkimuksesta on hyötyä. Tiedon kerääminen oli työlästä, sillä aihe oli laaja ja

siitä kertyi paljon lähteitä. Suurin aika työssä meni eri cross-platform menetelmien ymmärtämiseen ja lähteistä saadun tiedon oikein kääntämiseen. Tämän työn tekemisestä sai parempaa ymmärrystä mobiilisovellusten kehittamisestä, vaikka sen yhteydessä ei tehty mobiilisovellusta, ja se motivoi myös kokeilemaan hybridisovelluksen kehittämistä.

## LÄHTEET

About AppGyver. AppGyver. Viitattu 25.2.2016.

<https://www.appgyver.com/about/>

Account types, locations and fees. Microsoft. Viitattu 12.1.2016.

<https://msdn.microsoft.com/en-us/library/windows/apps/jj863494.aspx>

Andrews, M. 2014. Tutorial: How to make an offline HTML5 web app, FT style 11/2014. Viitattu 2.3.2016.

<http://labs.ft.com/2012/08/basic-offline-html5-web-app/>

Android Studio

Viitattu 6.1.2016.

<http://developer.android.com/sdk/index.html>

App submissions. Windows Dev Center. Microsoft. Viitattu 23.2.2016.

<https://msdn.microsoft.com/library/windows/apps/hh694062.aspx>

Beta testing and targeted distribution. Windows Dev Center. Microsoft. Viitattu 23.2.2016.

<https://msdn.microsoft.com/library/windows/apps/mt185377.aspx>

Bootstrap. Viitattu 25.1.2016.

<http://getbootstrap.com/>

Choosing a Membership. Compare Memberships. Apple. Viitattu 12.1.2016.

<https://developer.apple.com/support/compare-memberships/>

Compiler Options. Microsoft Developer Network. Viitattu 26.1.2016.

<https://msdn.microsoft.com/en-us/library/9s7c9wdw.aspx>

Composer 1. AppGyver. Viitattu 26.2.2016.

<http://www.appgyver.io/composer>

Composer 2. AppGyver. Viitattu 25.2.2016.

<https://www.appgyver.com/composer/>

Data Access. AppGyver. Viitattu 25.2.2016.

<https://www.appgyver.com/products/data-access/>

Distributing Apple Developer Enterprise Program Apps. iOS Developer Library. App Distribution Guide. Apple. Viitattu 4.2.2016.

[https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/DistributingEnterpriseProgramApps/DistributingEnterpriseProgramApps.html#//apple\\_ref/doc/uid/TP40012582-CH33-SW1](https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/DistributingEnterpriseProgramApps/DistributingEnterpriseProgramApps.html#//apple_ref/doc/uid/TP40012582-CH33-SW1)

Eclipse

Viitattu 6.1.2016. <https://www.eclipse.org/>

Editions and Pricing. Mendix. Viitattu 5.2.2016. <https://www.mendix.com/editions/#theFAQs>

Forum moderator Gajotres. Is the jQuery mobile the best UI framework to integrate with PhoneGap or Ionic? 7/2015. Viitattu 1.3.2016. <https://forum.ionicframework.com/t/is-the-jquery-mobile-the-best-ui-framework-to-integrate-with-phonegap-or-ionic/29305>

Galloway, M. 2013. iOS Assembly Tutorial: Understanding ARM 6/2013. Viitattu 21.1.2016. <http://www.raywenderlich.com/37181/ios-assembly-tutorial>

Gartner tiedotus 2013. Gartner Says by 2016, More Than 50 Percent of Mobile Apps Deployed Will be Hybrid 2/2013. Viitattu 6.1.2016. <http://www.gartner.com/newsroom/id/2324917>

Get Started with Publishing. Android. Viitattu 4.2.2016. <http://developer.android.com/distribute/googleplay/start.html#related-resources>

Getting Started with Ionic. Ionic framework. Viitattu 25.1.2016 <http://ionicframework.com/getting-started/>

Getting Started with Sencha Touch. Sencha. Viitattu 26.1.2016. [http://docs.sencha.com/touch/2.4/getting\\_started/getting\\_started.html](http://docs.sencha.com/touch/2.4/getting_started/getting_started.html)

Google Play –kehittäjien jakelusopimus 2015. Google Play. Viitattu 4.2.2016. <https://play.google.com/about/developer-distribution-agreement.html>

Google Playn kehittäjäkonsolin käyttö. Android Developer. Google Play. Viitattu 12.1.2016. <https://support.google.com/googleplay/android-developer/answer/6112435?hl=fi>

Grisogono, G 2014. 5 Best Mobile Web App Frameworks: jQuery Mobile + Backbone 3/2014. Viitattu 18.2.2016. <http://moduscreate.com/5-best-mobile-web-app-frameworks-jquery-mobile-backbone-part4/>

Hytönen H-K. Mobiilipalveluiden kehittäminen kaikki vastaukset. Viitattu 17.2.2016.

In-App Purchase for Developers. In App Purchase. Apple Viitattu 12.1.2016 <https://developer.apple.com/in-app-purchase/>

Introduction to Mobile Development. Getting Started. Xamarin. Viitattu 17.1.2016

[https://developer.xamarin.com/guides/cross-platform/getting\\_started/introduction\\_to\\_mobile\\_development/](https://developer.xamarin.com/guides/cross-platform/getting_started/introduction_to_mobile_development/)

jQuery Mobile. Viitattu 6.2.2016.  
<https://jquerymobile.com/>

Khalaf, S. 2014. Apps Solidify Leadership Six Years into the Mobile Revolution –blogi. Julkaistu 1.4.2014. Viitattu 2.3.2016.  
<http://flurrymobile.tumblr.com/post/115191864580/apps-solidify-leadership-six-years-into-the-mobile>

Kiran, V. 2013. Native vs Mobile Web vs Hybrid applications 12/2013. Viitattu 6.1.2016.  
<http://www.javacodegeeks.com/2013/12/native-vs-mobile-web-vs-hybrid-applications.html>

Li, S. 2014. Most People Can't Be Bothered to Download Apps 8/2014. Viitattu 11.3.2016  
<http://www.thewire.com/technology/2014/08/most-people-cant-be-bothered-to-download-apps/378989/>

Mendix. Viitattu 5.2.2016.  
<https://www.mendix.com/>

Mobile Angular UI. Mobile Angular UI. Viitattu 25.1.2016.  
<http://mobileangularui.com/>

Montecucollo, M. 2014. Native or Web-Based? Selecting the Right Approach for Your Mobile App 1/2014. Viitattu 6.1.2016.  
<https://uxmag.com/articles/native-or-web-based-selecting-the-right-approach-for-your-mobile-app>

OutSystems. Viitattu 1.2.2016.  
[www.outsystems.com](http://www.outsystems.com)

OutSystems Evaluation Guide a. Mobile. How does OutSystems Platform support mobile development? Viitattu 1.2.2016.  
[https://success.outsystems.com/Evaluation/Mobile/00\\_How\\_does\\_OutSystems\\_Platform\\_support\\_mobile\\_development](https://success.outsystems.com/Evaluation/Mobile/00_How_does_OutSystems_Platform_support_mobile_development)

OutSystem Evaluation Guide b. Mobile. What native capabilities does OutSystems Platform support? Viitattu 1.2.2016.  
[https://success.outsystems.com/Evaluation/Mobile/02\\_What\\_native\\_capabilities\\_does\\_OutSystems\\_Platform\\_support](https://success.outsystems.com/Evaluation/Mobile/02_What_native_capabilities_does_OutSystems_Platform_support)

OutSystems Evaluation Guide c. Mobile. Is there an enterprise app store for OutSystems applications? Viitattu 1.2.2016.  
[https://success.outsystems.com/Evaluation/Mobile/03\\_Is\\_there\\_an\\_enterprise\\_app\\_store\\_for\\_OutSystems\\_applications](https://success.outsystems.com/Evaluation/Mobile/03_Is_there_an_enterprise_app_store_for_OutSystems_applications)

OutSystems Evaluation Guide d. Mobile. Deploying OutSystems Platform on-premises. Viitattu 1.2.2016. [https://success.outsystems.com/Evaluation/Deploying-OutSystems-Cloud-On-Premises-or-Hybrid/01\\_Deploying-OutSystems-Platform-on-premise](https://success.outsystems.com/Evaluation/Deploying-OutSystems-Cloud-On-Premises-or-Hybrid/01_Deploying-OutSystems-Platform-on-premise)

OutSystems Now. Viitattu 1.2.2016. <https://labs.outsystems.net/OutSystemsNowDocs/Web-Site.OutSystemsNow.aspx>

OutSystems Platform Subscriptions. OutSystems. Viitattu 2.2016. <http://www.outsystems.com/pricing-and-licensing/>

Perez, S. 2015. App Submissions On Google Play Now Reviewed By Staff, Will Include Age-Based Ratings 3/2015. Viitattu 4.2.2016. <http://techcrunch.com/2015/03/17/app-submissions-on-google-play-now-reviewed-by-staff-will-include-age-based-ratings/>

Platform Overview. Mendix. Viitattu 5.2.2016. <https://www.mendix.com/application-platform-as-a-service/>

Premayogan, M. 2015. Why LinkedIn & Facebook dumped HTML5 and went native for its mobile apps 10/2015. Viitattu 6.1.2016. <https://www.linkedin.com/pulse/why-linkedin-facebook-dumped-html5-went-native-its-mobile-m>

Pricing. AppGyver. Viitattu 25.2.2016. <https://www.appgyver.com/pricing/>

Publishing Overview. Android. Viitattu 4.2.2016. [http://developer.android.com/tools/publishing/publishing\\_overview.html](http://developer.android.com/tools/publishing/publishing_overview.html)

Publish Windows apps. Windows Dev Center. Microsoft. Viitattu 4.2.2016. <https://dev.windows.com/en-us/publish>

Raj, J. 2014. The Top 7 Hybrid mobile App Frameworks 11/2014. Viitattu 21.1.2016. <http://www.sitepoint.com/top-7-hybrid-mobile-app-frameworks/>

Reynolds, C. 2014. Ten of the Best Cross-Platform Mobile Development Tools for Enterprises –blogi. Julkaistu 16.10.2014. Viitattu 14.1.2016 <http://appindex.com/blog/ten-best-cross-platform-development-mobile-enterprises/>

Rouse, M. 2007. rapid application development (RAD) 2/2007. Viitattu 1.2.2016. <http://searchsoftwarequality.techtarget.com/definition/rapid-application-development>

Salmaan, A. 2015. Windows Bridge for iOS: Let's open this up -blogi. Julkaistu 6.8.2015. Viitattu 26.1.2016.

<https://blogs.windows.com/buildingapps/2015/08/06/windows-bridge-for-ios-lets-open-this-up/>

Setup Mendix UI Framework with Koala. Docs & API. Mendix. Viitattu 5.2.2016.

<https://world.mendix.com/display/public/howto6/Setup+Mendix+UI+Framework+with+Koala>

Simple and transparent pricing. Xamarin.

Viitattu 28.1.2016

<https://store.xamarin.com/>

Smart Forms. AppGyver. Viitattu 25.2.2016. <https://www.appgyver.com/products/smart-forms/>

Smus, B. 2011. Multi-touch Web Development 8/2011. Viitattu 6.2.2016.

<http://www.html5rocks.com/en/mobile/touch/>

Submitting Your App to the Store. iOS Developer Library. Apple. Viitattu 4.2.2016.

<https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/SubmittingYourApp/SubmittingYourApp.html>

Steroids. AppGyver. Viitattu 26.2.2016.

<http://www.appgyver.io/steroids>

Subscriber Benefits. Visual Studio. Viitattu 26.1.2016

<https://www.visualstudio.com/products/subscriber-benefits-vs>

Supersonic. AppGyver. Viitattu 26.2.2016.

<http://www.appgyver.io/supersonic>

Tello, D. 2015. Native or Not? The Untapped Power of Web Apps 1/2015.

Viitattu 2.3.2016.

<https://www.viget.com/articles/native-or-not-the-untapped-power-of-web-apps>

The app certification process. Windows Dev Center. Microsoft. Viitattu 4.2.2016.

<https://msdn.microsoft.com/library/windows/apps/mt148554.aspx>

Tietoa HAMKista. HAMK. Viitattu 11.3.2016.

<http://www.hamk.fi/tietoa-hamkista/Sivut/default.aspx>

Tietohallinto, Mobiilistrategia -työryhmä. HAMKin mobiilipalveluiden kehittämisen toimintasuunnitelma 2016. Julkaistu 11.12.2015. Viitattu 17.2.2016.

Tietohallinto, Rintala, M. HAMK mobiilistrategia. Julkaistu 11.1.2016 Viitattu 17.2.2016.

Trice, A. 2012. PhoneGap Explained Visually 3/2012. Viitattu 24.1.2016.  
<http://phonegap.com/2012/05/02/phone-gap-explained-visually/>

Upload app packages. Windows Dev Center. Microsoft. Viitattu 23.2.2016.  
<https://msdn.microsoft.com/library/windows/apps/mt148542.aspx>

Usage Statistics and market share of JQuery for Websites. Viitattu 4.2.2016.  
<http://w3techs.com/technologies/details/js-jquery/all/all>

Validating Your App. iOS Developer Library. App Distribution Guide. Apple. Viitattu 4.2.2016.  
[https://developer.apple.com/library/ios/recipes/xcode\\_help-archives\\_organizer/articles/ValidatingYourApp.html](https://developer.apple.com/library/ios/recipes/xcode_help-archives_organizer/articles/ValidatingYourApp.html)

Visual Studio Purchasing Options. Visual Studio. Viitattu 26.1.2016.  
<https://www.visualstudio.com/products/how-to-buy-vs>

Vuorinen, C. 2014. Kolme tapaa kehittää mobiilisovellus –blogi. Julkaistu 21.10.2014. Viitattu 6.1.2016.  
<http://w3.fi/kolme-tapaa-kehittaa-mobiilisovellus/>

Web browser engine. Wikipedia. Viitattu 7.1.2016.  
[https://en.wikipedia.org/wiki/Web\\_browser\\_engine](https://en.wikipedia.org/wiki/Web_browser_engine)

Windows Bridges for Android. Viitattu 26.1.2016.  
<https://dev.windows.com/en-us/bridges/android>

Windows Bridges for iOS. Viitattu 26.1.2016.  
<https://dev.windows.com/en-us/bridges/ios>

Workflow. AppGyver. Viitattu 25.2.2016.  
<https://www.appgyver.com/products/workflow/>

### HAASTATTELUT

Kiviranta, P. 2016. Director. Outsystems. Haastattelu 29.2.2016.

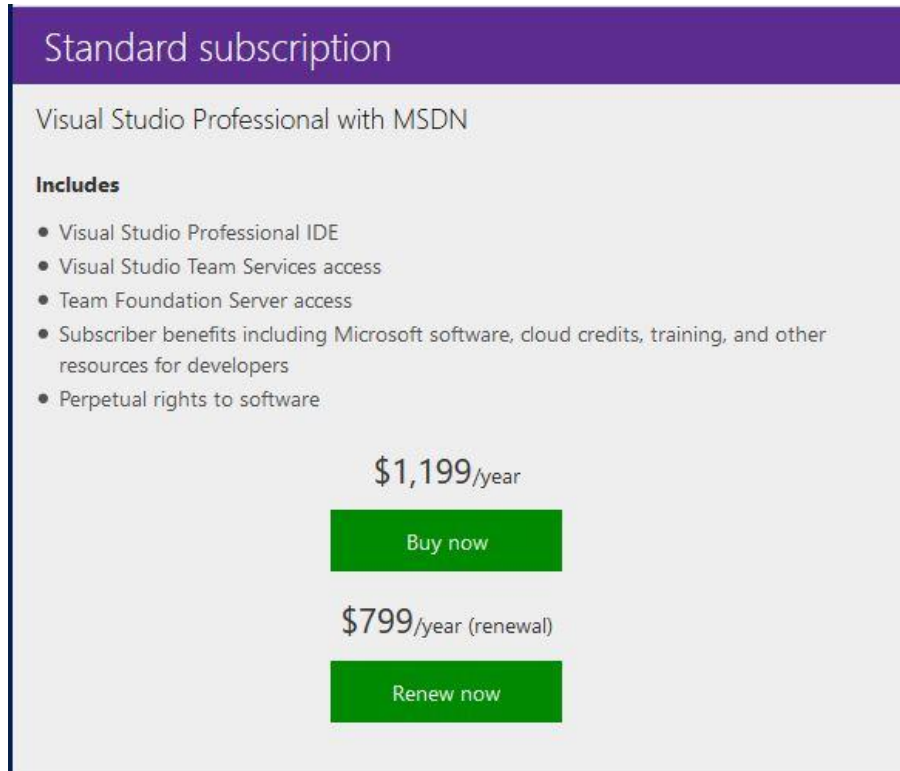
Rintala, M. 2016. Projektipäällikkö. HAMK. Haastattelu 17.2.2016.

### YOUTUBE

outsystems. 2015. Enterprise App Store – OutSystems Now [YouTube-video.] Julkaistu 8.4.2015. Viitattu 2.2.2016.  
<https://www.youtube.com/watch?v=wU8gL1R9Zb4>



VISUAL STUDIOON MAKSULLISET LISENSIT



Standard subscription

Visual Studio Professional with MSDN

**Includes**

- Visual Studio Professional IDE
- Visual Studio Team Services access
- Team Foundation Server access
- Subscriber benefits including Microsoft software, cloud credits, training, and other resources for developers
- Perpetual rights to software

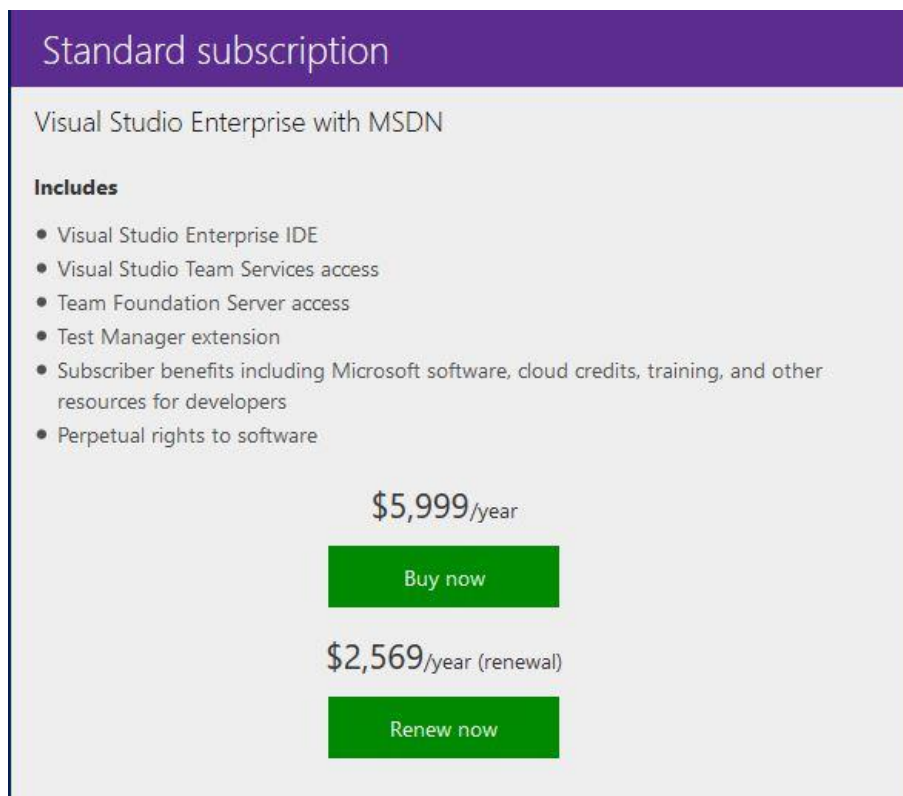
\$1,199/year

Buy now

\$799/year (renewal)

Renew now

<https://www.visualstudio.com/products/visual-studio-professional-with-msdn-vs>



Standard subscription

Visual Studio Enterprise with MSDN

**Includes**

- Visual Studio Enterprise IDE
- Visual Studio Team Services access
- Team Foundation Server access
- Test Manager extension
- Subscriber benefits including Microsoft software, cloud credits, training, and other resources for developers
- Perpetual rights to software

\$5,999/year

Buy now

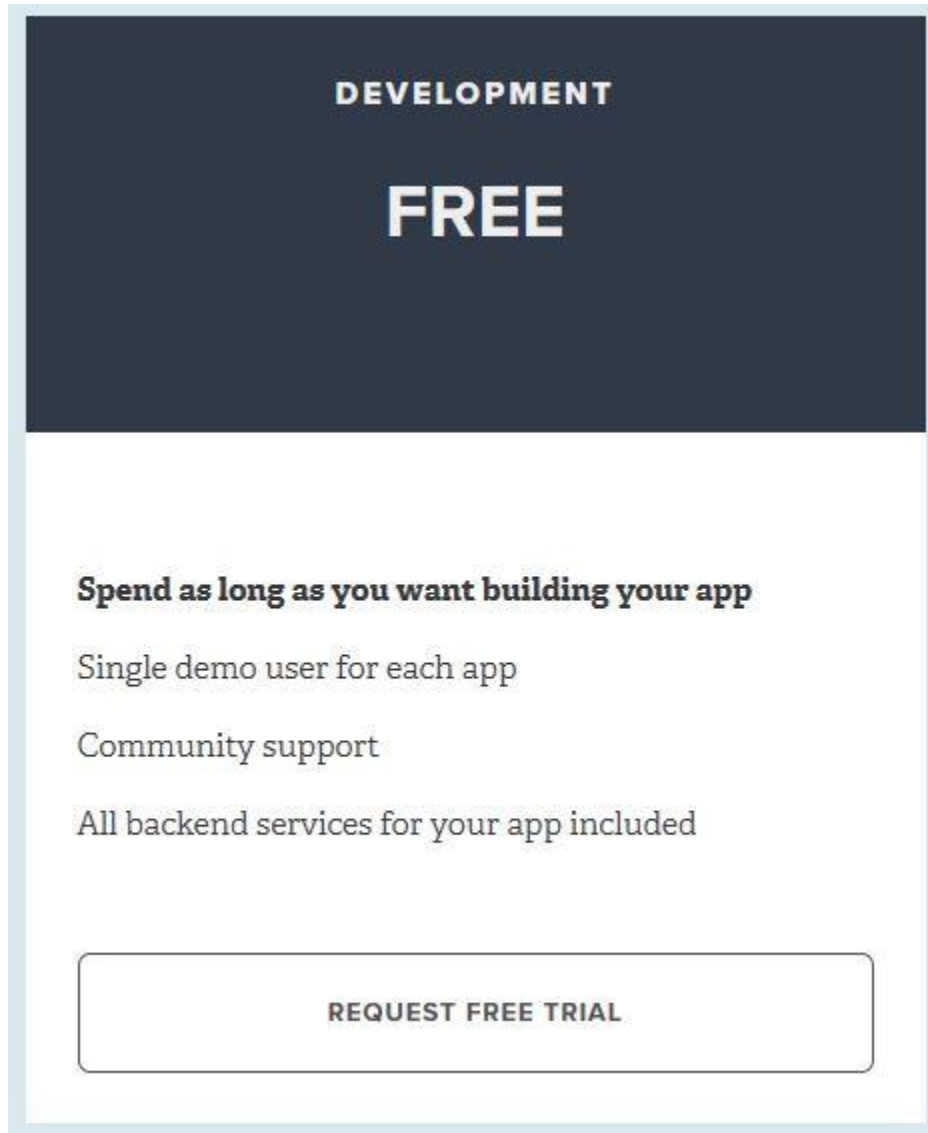
\$2,569/year (renewal)

Renew now

<https://www.visualstudio.com/products/visual-studio-enterprise-vs>



## APPGYVERIN LISENSSIT



The image is a screenshot of a pricing card for the 'DEVELOPMENT' plan on the Appgyver website. The card has a dark blue header with the word 'DEVELOPMENT' in white, uppercase letters. Below the header, the word 'FREE' is written in large, bold, white, uppercase letters. The main body of the card is white and contains the following text: 'Spend as long as you want building your app' in bold, followed by 'Single demo user for each app', 'Community support', and 'All backend services for your app included'. At the bottom of the card, there is a rounded rectangular button with a thin black border containing the text 'REQUEST FREE TRIAL' in uppercase.

**DEVELOPMENT**

**FREE**

**Spend as long as you want building your app**

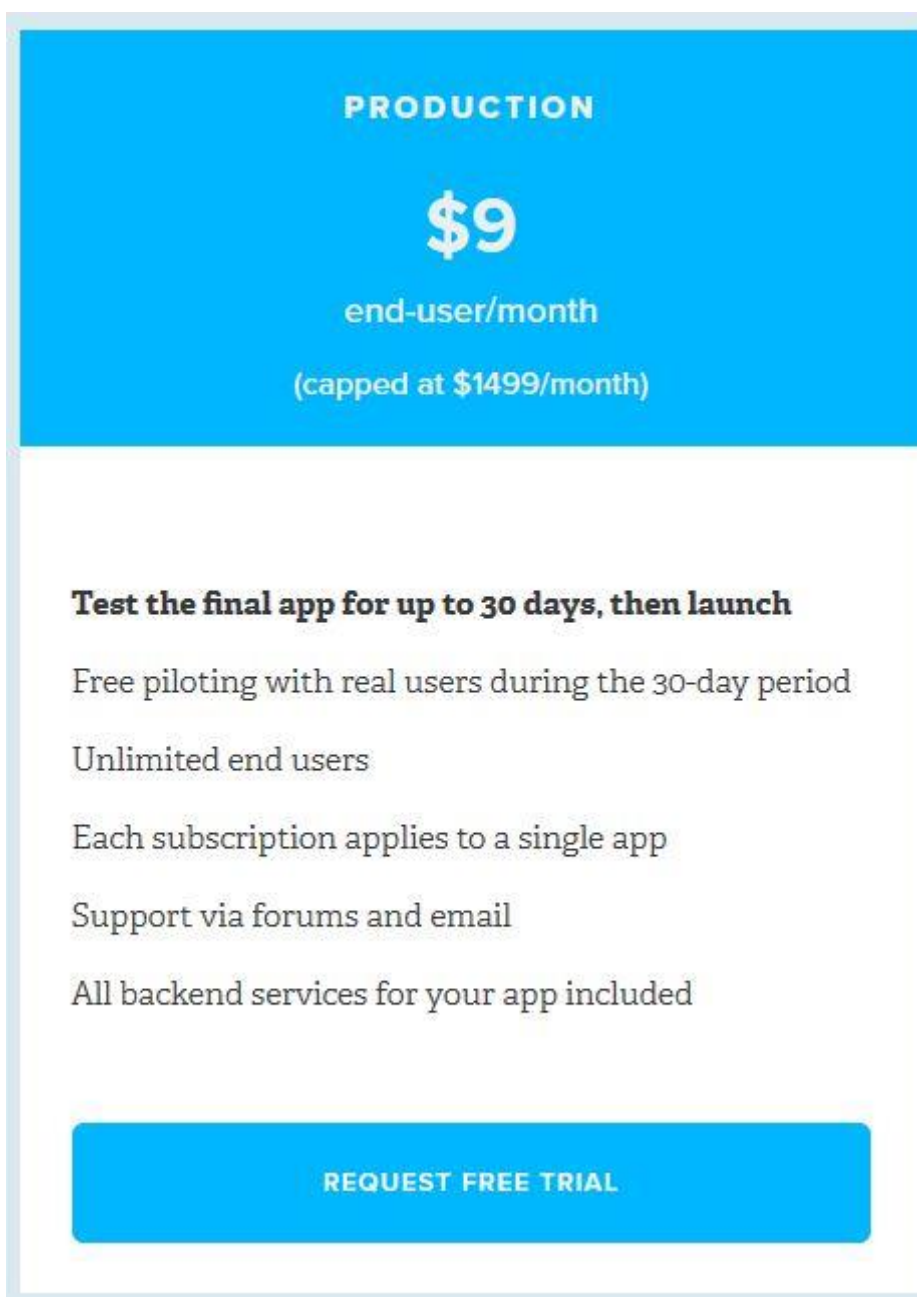
Single demo user for each app

Community support

All backend services for your app included

**REQUEST FREE TRIAL**

<https://www.appgyver.com/pricing>



**PRODUCTION**

**\$9**

end-user/month

(capped at \$1499/month)

**Test the final app for up to 30 days, then launch**

Free piloting with real users during the 30-day period

Unlimited end users

Each subscription applies to a single app

Support via forums and email

All backend services for your app included

**REQUEST FREE TRIAL**

<https://www.appgyver.com/pricing>

## ENTERPRISE

Let's talk

### **Get white glove service, training and private installation**

Unlimited end-users

Online and on-site training

Co-development and consulting

Turn-key app development

White labeling

Private cloud installations

Data virtualization

Dedicated support

**CONTACT OUR SALES**

<https://www.appgyver.com/pricing>

MENDIX PLATFORMIN LISENSSIT

EDITION COMPARISON	FREE	PRO	ENTERPRISE
	<p><b>€0</b></p> <p><i>Start instantly</i></p> <p><b>SIGN UP</b></p>	<p><b>STARTS AT €1.250</b></p> <p><i>per month (billed annually)</i></p> <p><b>GET QUOTE</b></p>	<p><b>STARTS AT €1.675</b></p> <p><i>per month (billed annually)</i></p> <p><b>GET QUOTE</b></p>
<b>Users and Apps</b>	Unlimited Apps with up to 10 Users	Choose Number of Apps and Users	Choose Number of Apps and Users
Users	10 Internal Users per App	Starts with 50 Internal Users or 500 External Users	Starts with 50 Internal Users or 500 External Users
Developers and Project Team Members Included	✓	✓	✓
No Application Limitations (# of Objects, Pages, Flows and API Calls)	✓	✓	✓
Anonymous Users	✓	✓	✓
Always On		✓	✓
<p><b>Social Collaboration</b> Enterprise-wide social collaboration incl. agile project management, requirements management, activity streams, user feedback, and chat.</p>			
Social Collaboration	✓	✓	✓
Requirements Management	✓	✓	✓
Agile Project Management	✓	✓	✓
User Feedback Management	✓	✓	✓

<https://www.mendix.com/editions/comparison-chart/>

## Mobiilisovellustyypit ja niillä kehittämisen vaihtoehdot

	FREE	PRO	ENTERPRISE
<b>EDITION COMPARISON</b>	<b>€0</b> <i>Start instantly</i> <b>SIGN UP</b>	<b>STARTS AT €1.250</b> <i>per month (billed annually)</i> <b>GET QUOTE</b>	<b>STARTS AT €1.675</b> <i>per month (billed annually)</i> <b>GET QUOTE</b>
<b>Development</b> Develop multi-channel apps using Mendix's signature Model-Driven Development approach to build domain models, multi-device UIs, business rules, process logic and microflows.			
Model-driven App Development Environment (Mendix Business Modeler)	✓	✓	✓
Release Management, Shared Modeling and Version Repository for Teams (Mendix Team Server)	✓	✓	✓
Java and Eclipse Integration	✓	✓	✓
Mobile Apps	✓	✓	✓
Multi-tenancy	✓	✓	✓
<b>App Environments</b> Choose from different app staging and production and pre-production environment options.			
Complete Deployment Stack Including Database and Runtime Server	✓	✓	✓
Max # of TAP (Test, Acceptance, Production) Environments)	1	3	Unlimited

<https://www.mendix.com/editions/comparison-chart/>

## Mobiilisovellustyypit ja niillä kehittämisen vaihtoehdot

	FREE	PRO	ENTERPRISE
<b>EDITION COMPARISON</b>	<b>€0</b> <i>Start instantly</i> <b>SIGN UP</b>	<b>STARTS AT €1.250</b> <i>per month (billed annually)</i> <b>GET QUOTE</b>	<b>STARTS AT €1.675</b> <i>per month (billed annually)</i> <b>GET QUOTE</b>
<b>App Resource Management</b> Scale and manage app resources as needed.			
Number of App Containers Included	1	2	3
Choose Different / Add Additional App Containers		✓	✓
Multi App Resource Management		✓	✓
<b>Enterprise Integration</b> Multiple integration options for enterprise needs.			
App Integration APIs (Java, web services, etc.)	✓	✓	✓
Platform APIs	✓	✓	✓
Third Party Identity Management Integration		✓	✓

<https://www.mendix.com/editions/comparison-chart/>

## Mobiilisovellustyypit ja niillä kehittämisen vaihtoehdot

	FREE	PRO	ENTERPRISE
<b>EDITION COMPARISON</b>	<b>€0</b> <i>Start instantly</i> <b>SIGN UP</b>	<b>STARTS AT €1.250</b> <i>per month (billed annually)</i> <b>GET QUOTE</b>	<b>STARTS AT €1.675</b> <i>per month (billed annually)</i> <b>GET QUOTE</b>
<b>App Operations and Backup</b> Manage and monitor the ongoing operation of your apps.			
Automated Backups	Weekly *	Daily	Daily
App Monitoring	Basic *	Advanced	Advanced
Fallback Environment with Data Replication			Optional
High Availability			Optional
Horizontal Scaling			Optional
<b>Multi-App Administration</b> Manage all apps in one place.			
Enterprise Multi-App Admin		✓	✓
Private Enterprise App Store		✓	✓
<b>Support</b> All platform updates are automatically available to all users. Platform users also have multiple support options.			
Community Support	✓	✓	✓
Platform Support		✓	
Gold Support		Optional	
Platinum Support			✓

<https://www.mendix.com/editions/comparison-chart/>



XAMARININ LISENSSIT

Prices below are **per developer, per device platform** with no royalties or hidden fees.

	<b>INDIE</b> <b>\$25 / month</b> paid monthly or annually	<b>BUSINESS</b> <b>\$999 / year</b> paid annually	<b>ENTERPRISE</b> <b>\$1899 / year</b> paid annually
Permitted Use	Individual	Organization	Organization
Subscription Type	Monthly	Annual	Annual
Deploy to Device	✓	✓	✓
Deploy to App Stores	✓	✓	✓
Xamarin Studio	✓	✓	✓
Unlimited App Size	✓	✓	✓
Xamarin.Forms	✓	✓	✓
Xamarin Test Cloud Access	✓	✓	✓
Visual Studio Integration		✓	✓
Business Features		✓	✓
Email Support		✓	✓
One Business Day SLA			✓
Hotfixes			✓
Technical Kick-off Session			✓
Account Manager			✓
Embedded Assemblies			✓
Code Troubleshooting		At Extra Cost	At Extra Cost
	Subscribe	Subscribe	Subscribe

<https://store.xamarin.com/>