

# Augmented reality -mobiilisovellus

LAHDEN  
AMMATTIKORKEAKOULU  
Tekniikan ala  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikka  
Opinnäytetyö  
Syksy 2018  
Teemu Juvonen

Lahden ammattikorkeakoulu  
Tietotekniikan koulutusohjelma

JUVONEN, TEEMU:

Augmented reality -mobiilisovellus

Ohjelmistotekniikka opinnäytetyö, 32 sivua

Syksy 2018

TIIVISTELMÄ

---

Opinnäytetyössä tarkastellaan Augmented reality -teknologiaa hyödyntävää mobiilisovellusta ja sen toimintaympäristöä. Sovelluksen taustatutkimuksen ja kehityksen Lahden ammattikorkeakoululta tilasi Laulumaa Huonekalut Oy. Laulumaa Huonekalut Oy on lahtelainen perheyriitys, joka suunnittelee ja valmistaa kotimaisia huonekaluja. Sovellus mahdollistaa käyttäjän tarkastella huonekaluja puhelimen takakameran kuvavirtaa vasten, tarjoten näin kuvan siitä, miltä tuote näyttää sen lopullisessa sijoitusympäristössä.

Projektissa tutkittiin Augmented reality -teknologian nykytilaa sekä vertailtiin erilaisia menetelmiä kehittää mobiilisovellus. Lisäksi paneuduttiin eroihin mobiilikäyttöjärjestelmien välillä sekä kerrotaan tarkemmin hybridimobiilikehityksessä käytetyistä teknologioista.

Projektissa kuvataan sovelluskehityksen vaiheita: sovelluksen toimintalogiikan suunnittelua, käyttöliittymäsuunnittelua sekä toteutusta. Lisäksi kuvataan sovelluksen rinnalla toimivan taustajärjestelmän toteutusta.

Lopputuloksena syntyi toimiva mobiilisovellus, joka vastasi asiakkaan asettamia vaatimuksia. Sovellusta ei julkaistu julkisesti, mutta se jäi sisäiseen käyttöön Laulumaa Huonekalut Oy:lle.

Asiasanat: Augmented reality, hybridisovelluskehitys, Ionic Framework

Lahti University of Applied Sciences  
Degree Programme in Information Technology

JUVONEN, TEEMU:

Augmented Reality mobile application

Bachelor's Thesis in Software Engineering, 32 pages

Autumn 2018

ABSTRACT

---

This thesis deals with a mobile application that utilizes Augmented Reality technology and its operating environment. The research and development of the application was commissioned from Lahti University of Applied Sciences by Laulumaa Huonekalut Ltd, a family business in Lahti, which designs and manufactures domestic furniture. The application would allow the user to look at the furniture against the photo stream of the phone's back camera, providing an image of what the product looks like in its final environment.

The theoretical part of the thesis consisted of the examination of the current state of Augmented Reality technology and comparison of various methods to develop a mobile application. In addition, the differences between mobile operating systems were addressed and the technologies used in hybrid mobile development were further explored.

The steps of the application project were: application logic design, user interface design and implementation. In addition, a background system that works alongside the application was implemented.

The end result was a working mobile application that met the customer's requirements. The application was not published publicly but it is in internal use at Laulumaa Huonekalut Ltd.

Key words: Augmented Reality, hybrid application development, Ionic Framework

## SISÄLLYS

1	JOHDANTO	1
2	TOIMINTAYMPÄRISTÖ	3
3	AUGMENTED REALITY	6
4	MOBIILISOVELLUSKEHITYS	9
5	TEKNOLOGIAT	14
5.1	NodeJS	14
5.2	npm	14
5.3	Apache Cordova	14
5.4	AngularJS	16
5.5	Ionic Framework	17
5.6	ezAR VideoOverlay	18
6	SOVELLUSKEHITYS	20
6.1	Kehitysympäristö	20
6.2	Kansiorakenne	21
6.3	Käyttöliittymäsuunnittelu	22
6.4	Backend	25
7	SOVELLUS	29
7.1	Sovelluksen toiminta	29
7.2	Backend	30
8	YHTEENVETO	32
	LÄHTEET	33

## TERMIT JA LYHENTEET

Augmented reality      Lisätty todellisuus. Tietokoneella tuotettua mediaa perinteisen näkymän päällä.

## 1 JOHDANTO

Tässä opinnäytetyössä tarkastellaan Augmented reality -teknologiaa hyödyntävää mobiilisovellusta. Augmented reality, AR tai lisätty todellisuus, viittaa näkymään, jossa tietokoneella tuotettua mediaa on yhdistetty todellisen näkymän kanssa. Nykyaikainen ihminen on sidoksissa teknologian kanssa tavalla, joka on meille uusi ja vielä kartoittamaton. Lähestulkoon jokaisella lapsesta vaariin on tätä nykyä käytössään älypuhelin, joka on merkittävä osa päivittäistä elämäämme. Elämme tällä hetkellä murroshetkessä teknologian kanssa. Teemme päivittäisiä askareitamme ja hyödynnämme teknologiaa, mutta nämä kaksi asiaa ovat usein erillisiä toisistaan. Tässä Augmented reality astuu kuvaan. Se mahdollistaa teknologian ja todellisuuden yhdistämisen yhdeksi, saumattomaksi käyttökokemukseksi. Augmented reality -teknologia tarjoaa miltein rajattomat käyttömahdollisuudet, joita kaikkia emme osaa vielä edes visioida. Miltä kuulostaisi auto, jossa kojelaudan sijaan mittaristo olisikin osa tuulilasia? Augmented reality -teknologian avulla se voi olla jo huomisen todellisuutta.

Opinnäytetyön tavoitteena on tuottaa Augmented reality -teknologiaa hyödyntävä proof of concept -mobiilisovellus Laulumaa Huonekalut Oy:lle. Laulumaa Huonekalut Oy on vuonna 1988 perustettu lahtelainen perheyrittäjä, jonka juuret juontavat Rauno Laulumaan vuonna 1956 perustamasta raahelaisesta puusepäntehtaasta. Nykyisin yritys kehittää, valmistaa ja tukkukauppaa kotimaisia, kestäviä ja korkealaatuisia huonekaluja. Laulumaa Huonekalut Oy:n liikevaihto vuonna 2017 oli 4,2 miljoonaa euroa ja yritys työllistää Lahden pääkonttorilla 10 - 20 henkilöä (Kauppalehti 2018). Mobiilisovelluksen käyttötarkoituksena oli mahdollistaa asiakkaalle tuotteiden selaaminen erilaisista kategorioista, tuotteiden tarkasteleminen niiden lopullista sijoitusympäristöä vasten kotoa käsin, sekä lisätä samalla tuotteiden näkyvyyttä ja kysyntää.

Tutkimusongelmana on ensisijaisesti selvittää Augmented reality -mobiilisovellusta varten käytettävät tekniikat sekä kehittää

esimerkkisovellus, joka hyödyntää Augmented reality -teknologiaa, sekä toimii mahdollisimman monella älypuhelinlustralalla.

Opinnäytetyö on Business Finlandin (ent. Tekes) Laulumaa Huonekalut Oy:lle myöntämän innovaatioasetelin käyttökohde. Innovaatioasetelillä halutaan kannustaa pieniä ja keskisuuria yrityksiä tutkimaan uusia innovaatioita ja niiden mahdollistamia käyttötarkoituksia. Laulumaa Huonekalut Oy tilasi tämän asiantuntijatyön Lahden ammattikorkeakoululta.

Luvussa kaksi käydään läpi mobiilisovelluksen toimintaperiaatetta, sen toimintaympäristöä sekä työn vaatimuksia. Luku kolme keskittyy Augmented reality -teknologiaan. Luku neljä käsittelee yleisesti mobiilisovelluskehitystä. Luvussa viisi tarkastellaan lähemmin sovelluskehityksessä käytettyjä teknologioita. Luku kuusi keskittyy syvemmin sovelluskehitykseen. Luvussa seitsemän on kuvattu sovelluksen toimintaa ja luvussa kahdeksan on koostettu yhteenveto projektin osalta.

## 2 TOIMINTAYMPÄRISTÖ

Augmented reality -teknologiaa haluttiin sovellettavan tässä projektissa siten, että älypuhelimien kameran tarjoaman todellisen kuvavirran päälle sijoitettaisiin tuotekuva huonekalusta. Tällä tavoin loppukäyttäjä voisi nähdä tuotteen sen lopullisessa sijoitusympäristössä suoraan puhelimen ruudulta. Laitteen ruudulla käyttäjän tulisi olla mahdollista siirtää, skaalata ja kiertää huonekalua kosketuksen avulla. Näin käyttäjä saa välittömästi kuvan siitä, miten kyseinen huonekalu istuu huoneeseen ja sopii muuhun sisustukseen. Sovellusprototyypin haluttiin hyödyntävän Augmented reality -teknologiaa sekä toimivan mahdollisimman monessa älypuhelinympäristössä ja eri hintaluokan mallissa.

Projektin aluksi tutkittiin Augmented reality -teknologiaa, sen nykytilaa ja tämänhetkisiä toteutuksia mobiiliteknologiassa. Vertailtaessa vaihtoehtoja Augmented reality -mobiilitoteutukselle nousivat kaksi vaihtoehtoa selvästi ylitse muiden. Nämä olivat Unity-pelimoottori sekä Apache Cordova -mobiilialustan liitännäinen ezAR. Unity-pelimoottori on Unity Technologiesin vuonna 2005 kehittämä monialustainen pelimoottori, joka soveltuu pelien tekemisen lisäksi myös interaktiivisen median toteutukseen (Unity Technologies 2018). Unity tarjoaa valmiita työkaluja Augmented reality -teknologian kehittämiseen mobiilialustaa varten, mutta pelimoottori on verrattain raskas, etenkin hintaluokan alapään älypuhelimilla. Mikäli kyseessä olisi ollut Augmented reality -teknologiaa hyödyntävä peli tai laajempi sovellus, olisi Unity-pelimoottori ollut varteenotettava vaihtoehto, mutta ainoastaan yhteen verrattain kevyeseen käyttötarkoitukseen sen valjastaminen tuntui ylilyönniltä. Sen sijaan ezAR, joka toimii mobiililaitteen kameraa ja verkkoselainta hyödyntäen, on vähemmän puhelimen resursseja vaativa vaihtoehto ja näin toimii liki kaikissa hintahaarukan päätelaitteissa.

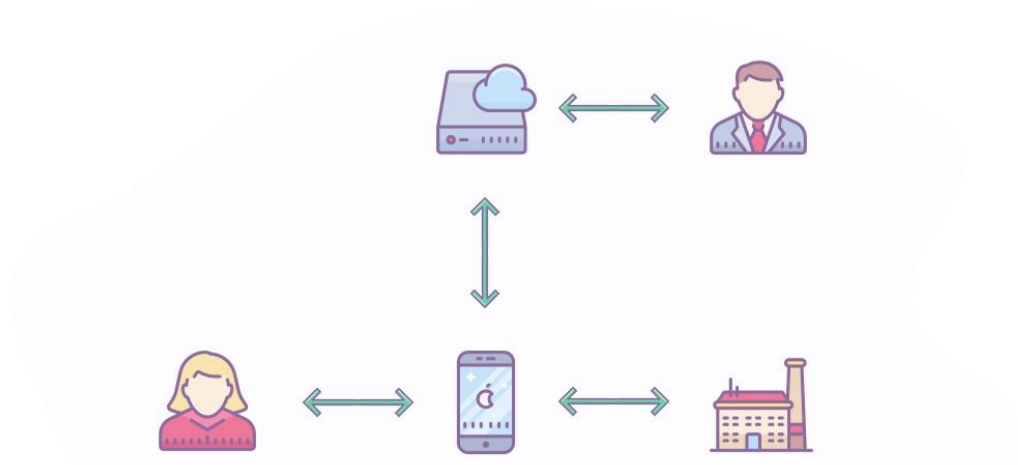
Sovelluksessa erilaisten tuotteiden toivottiin olevan listattuna kategorioittain, jolloin käyttäjän valitessa tuotekategorian sovellus näyttäisi ainoastaan tähän kategoriaan kuuluvat tuotteet. Jotta loppukäyttäjälle välittyisi todenmukainen kuva tuotteesta, oli tuotekuvien oltava syvättyjä.



Tämä edellytti asiakkaalta saatujen tuotekuvien manuaalista työstämistä. Tuotekuvista tuli poistaa tausta, jotta toteutuksen puhelimen kameranäkymän päälle luoma immersio olisi mahdollisimman eheä loppukäyttäjälle. Lisäksi kustakin tuotteesta tuli olla saatavilla lisätietoja, kuten sen mitat, valmistusmateriaali ja hinta. Lisätoiveena oli myös mahdollisuus avata aikaisemmin otettu kuva puhelimen galleriasta näkymän taustalle ja tarkastella tuotekuvaa tätä taustaa vasten. Edellä mainitut vaikuttivat osaltaan käyttöliittymäsuunnitteluun, jota käsitellään tarkemmin alaluvussa 6.3.

Lisäksi toiveena oli, että mobiilisovelluksen tuotevalikoimaan on helposti asiakkaan toimesta lisättävissä uusia tuotteita ja kategorioita. Tätä varten sovelluksen toimintaympäristöön lisättiin taustajärjestelmä eli niin sanottu backend, joka mahdollisti edellä mainitun toiminnallisuuden. Backend-osuudesta kerrotaan tarkemmin alaluvussa 6.4.

Seuraavassa kuviossa (kuvio 1) on esitetty sovelluksen toimintaympäristö.



KUVIO 1. Toimintaympäristö

Kuten kuvioista 1 ilmenee, sovellus mahdollistaa loppukäyttäjälle yrityksen tuotteiden selaamisen sovelluksen kautta sekä yritykselle väylän tuoda tuotteet käyttäjien tietoisuuteen lisäten näin niiden näkyvyyttä.

Yrityksellä on mahdollisuus vuorovaikuttaa pilvessä toimivan palvelimen kautta suoraan loppukäyttäjän sovelluskokemukseen.

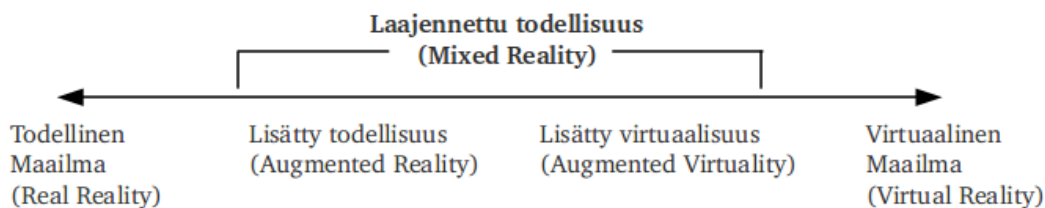
Näiltä osin mobiilisovelluksen oli täytettävä seuraavat vaatimukset:

- Augmented reality -teknologian hyödyntäminen siten, että tuotetta voidaan kosketuksen avulla liikutella, skaalata ja kiertää kameranäkymän päällä.
- Mahdollisimman monessa eri päätelaitteessa toimiva sovellus.
- Tuotteiden selaaminen kategorioittain ja tuotteen tarkempien tietojen tarkasteleminen.
- Kuvan avaaminen puhelimen galleriasta näkymän taustalle.
- Kuvan tallentaminen laitteen galleriaan huonekalun Augmented reality -sommittelusta.

Tämän lisäksi omana vaatimuksenaan toimi asiakkaan toive tuotteiden ja kategorioiden helppokäyttöisestä lisäämisestä, joka huomioitiin sovelluksen backend-osuuden toteutuksessa. Edellä mainitut vaatimukset ohjasivat osaltaan mobiilisovelluskehitystä, josta kerrotaan tarkemmin luvussa neljä.

### 3 AUGMENTED REALITY

Augmented realityllä tarkoitetaan lisättyä todellisuutta. Tämä tarkoittaa näkymää, jossa todellista näkymää on ehostettu digitaalisin elementein. Augmented reality on käsitteenä osa Paul Milgramin esittelemää todellisuus-virtuaalisuus-jatkumoa, joka on esitetty seuraavassa kuviossa (Milgram 1994).



KUVIO 2. Todellisuus-virtuaalisuus-jatkumo (Junkkari 2012)

Jatkumon ääripäinä toimivat täysin virtuaalisesti tuotettu näkymä, jossa kaikki elementit ovat tietokoneella luotuja. Toisena ääripäänä on perinteinen, todellinen näkymä, jossa ei ole mitään keinotekoisesti lisättyjä elementtejä. Jatkumon ääripäät jakaa toisistaan isompi käsite, Mixed reality, jonka osa-alue Augmented reality on. Mixed realityllä tarkoitetaan yleisesti todellisuutta ja virtuaalisuutta yhdistelevää näkymää. Toinen tunnettu Mixed realityn käsite on Augmented virtuality, jossa todellisuuden elementtejä on lisätty virtuaaliseen ympäristöön, kun Augmented reality:ssä virtuaalisia elementtejä on lisätty todelliseen ympäristöön.

Augmented reality -teknologian näyttöteknologiat voidaan jakaa karkeasti kolmeen luokkaan:

- päässä pidettäviin näyttöihin, kuten erilaiset kypärät ja silmien eteen sijoitettavat lasit
- projektoreihin, jotka yleisesti heijastavat informaation kiinteästä pisteestä ilman reaaliaikaista seuranta
- kädessä pidettäviin näyttöihin, kuten älypuhelimien ruudut.

Tässä opinnäytetyössä keskitytään näistä viimeisimpään, jossa Augmented reality -teknologia tuotetaan älypuhelimien näytölle hyödyntäen puhelimen kameraa.

Augmented reality -teknologia otti ensimmäiset askeleensa jo 1990-luvun alkupuolella Yhdysvaltain ilmavoimien laboratorioissa (Wikipedia 2018). Teknologia oli kuitenkin tuolloin niin varhaisissa askeleissa ja siihen vaadittava laitteisto oli siihen aikaan hintansa puolesta tavallisen kuluttajan ulottumattomissa, että todellisen läpimurtonsa Augmented Reality teki vasta 2010-luvun taitteessa. Tätä edesauttoivat teknologian kehittyminen vuosien varrella sekä päätelaitteiden laskentatehon radikaali nousu. Tehokkaiden mobiililaitteiden myötä valtaosalla väestöstä on tätä nykyä käytössään laite, joka mahdollistaa Augmented reality -teknologian valjastamisen.

Augmented reality -teknologialla on lähes lukematon määrä käyttökohteita, joista kaikkia emme osaa vielä edes kuvitella. Näistä esimerkkeinä ovat lääketiede, sotateollisuus, pelituollisuus ja opetuskäyttö, joiden saralla teknologiaa käytetään jo nyt aktiivisesti.

Peliteollisuudessa Augmented reality oli suuresti esillä valtamedioissa Niantic-peliyhtiön vuonna 2016 julkaiseman supersuosituksen Pokemon Go -pelin ansiosta. Pelissä virtuaalisia hahmoja esitetään puhelimen kamerakuvan päällä. Peli julkaistiin asteittain ympäri maailmaa kesällä 2016 ja jo saman vuoden syyskuuhun mennessä, peliä oli ladattu yhteensä yli 500 miljoonaa kertaa (TechCrunch 2016).

Augmented reality on valjastettu käyttöön myös Suomessa. Messukeskus lanseerasi vuonna 2017 käyttöön sovelluksen, joka auttaa kävijöitä navigoimaan messuhallissa. Sovellus piirtää laitteen kameranäkymän päälle polun, jota seuraamalla käyttäjä löytää halutuun messukohteeseen. (Messukeskus 2017.)

Augmented reality -teknologian markkinaosuudeksi on vuodelle 2017 laskettu 5,85 miljardia Yhdysvaltain dollaria ja vuoteen 2023 mennessä

markkinaosuuden arvellaan saavuttavan 70 miljardin dollarin rajan (Digital Journal 2018).

#### 4 MOBIILISOVELLUSKEHITYS

Koska sovelluksen tuli toimia useassa eri älypuhelinympäristössä, valittiin kehityssuunnaksi hybridimobiilisovelluskehitys. Hybridisovelluskehitys tarkoittaa sitä, että samalla koodipohjalla on käännettävissä eri päätelaitteissa toimivat versiot. Hybridisovelluskehityksen vastakohtana voidaan pitää niin sanottuja natiivisovelluksia, joilla on päätelaittekohteen mukaan räätälöity koodipohja. Usein päätelaitteet on koodattu erilaisilla ohjelmointikielillä, jolloin natiivisovellusten osalta tämä tarkoittaa sitä, että yhdelle alustalle koodi kirjoitetaan yhdellä koodikielellä ja toiselle alustalle toisella. Hybridisovelluskehitys poikkeaa tästä siten, että koodipohjia on ainoastaan yksi, eikä se välttämättä vastaa kummankaan alustan koodikieltä, mutta samasta koodipohjasta käännetään kullekin kohdealustalle tämän omalla koodikielellä toimiva versio.

Hybridisovelluskehityksen suurimpana etuna on se, että ylläpidettäviä koodipohjia on ainoastaan yksi. Sovelluskehitys usealla eri alustalle on tällöin nopeampaa, eikä tulevaisuudessa korjauksia, muutoksia tai lisäominaisuuksia tarvitse kirjoittaa erikseen alustakohtaisesti.

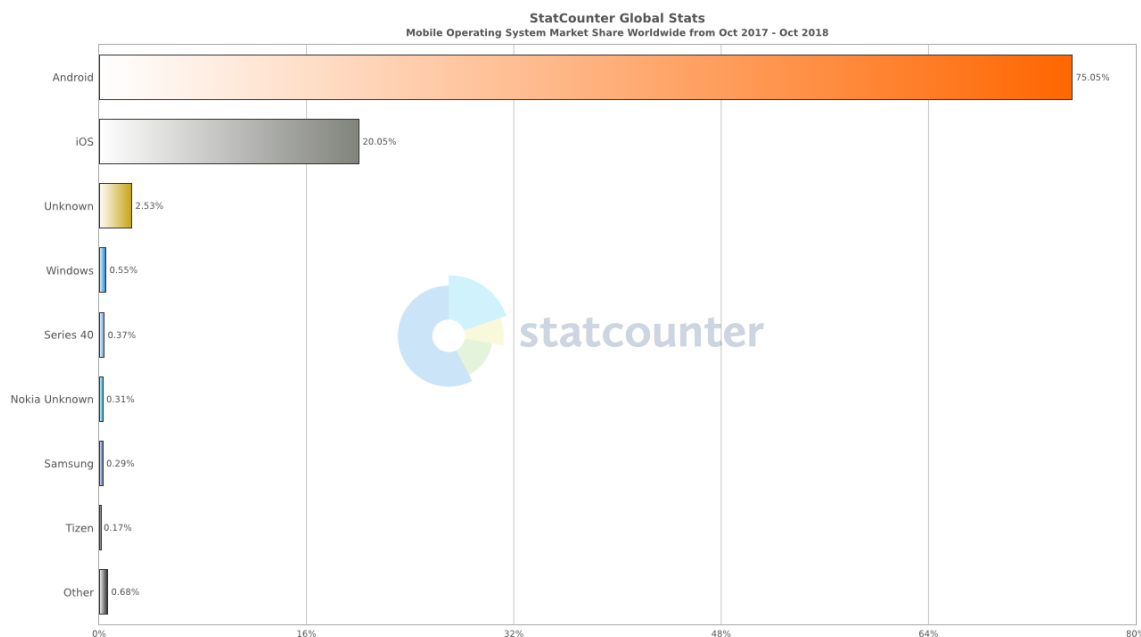
Hybridisovelluskehitys asettaa tosin myös rajoituksia, sillä aina sama koodi hybridikoodipohjassa ei käänny täysin samalla tavalla kaikille alustoille.

Myös suorituskyky on usein natiivisovelluksia heikompaa, sillä alustakohtaista optimointia on haastava tehdä hybridikoodipohjassa (Bitmovin 2018).

	Apple iOS	Google Android	Microsoft Windows Phone
Sovelluskauppa	App Store	Google Play	Windows Store
Gloaali markkinaosuus	20,05 %	75,05 %	0,55 %
Sovellusten ohjelmointikielät	Objective-C, Swift	Java, Kotlin, C++	C++, C#, JavaScript, Microsoft Visual Basic
Sovellusten määrä sovelluskaupassa	2 000 000	2 100 000	669 000
Kehitystyökalut	Xcode	Android Studio	Visual Studio

KUVIO 3. Tunnetuimmat mobiilikäyttöjärjestelmät

Kuviossa 3 on kuvattu tunnetuimpien mobiilikäyttöjärjestelmien eroja. Mobiilisovelluskehityksessä yleisimpiä kohdealustoja ovat Googlen Android-käyttöjärjestelmä sekä Applen iOS-käyttöjärjestelmä. Molemmilla käyttöjärjestelmillä on oma sovelluskauppa, jonka kautta sovellusten julkaisu ja jakaminen tapahtuu. Googlen Android-käyttöjärjestelmässä pääasiallisena ohjelmointikielenä sovelluksille toimivat Java, Kotlin tai C++ ja sovelluskauppa kantaa nimeä Google Play (Android 2018). Applen iOS-käyttöjärjestelmän sovellukset käyttävät ohjelmointikielenään Objective-C:tä tai Swiftiä ja sovelluskauppa on nimeltään App Store (VentureBeat 2014). Android-käyttöjärjestelmä on julkaistu avoimena lähdekoodina, minkä ansiosta käyttöjärjestelmä löytyykin vakiona tai muokattuna usean eri valmistajan puhelinmalleista. iOS-käyttöjärjestelmiä käyttäviä puhelimia valmistaa ainoastaan Apple, eikä käyttöjärjestelmän lähdekoodi ole julkisesti saatavilla.



**KUVIO 4. Mobiilikäyttöjärjestelmien globaalit markkinaosuudet (StatCounter 2018b)**

Kuten kuviosta 4 selviää, Androidin markkinaosuus mobiilikäyttöjärjestelmistä oli lokakuussa 2018 globaalisti noin kolme neljäsosaa ja iOS:n yksi neljäsosa. Suomessa Androidin markkinaosuus on vastaavasti noin kaksi kolmasosaa ja iOS:n yksi kolmasosa (StatCounter 2018a).

Mobiilikehityksessä natiivi-hybridit -jaottelu on hieman erilainen ja astetta haastavampi. Usein mobiilisovellukset jaotellaan natiivisovelluksiin, hybridisovelluksiin ja HTML5-sovelluksiin (W3 2014).

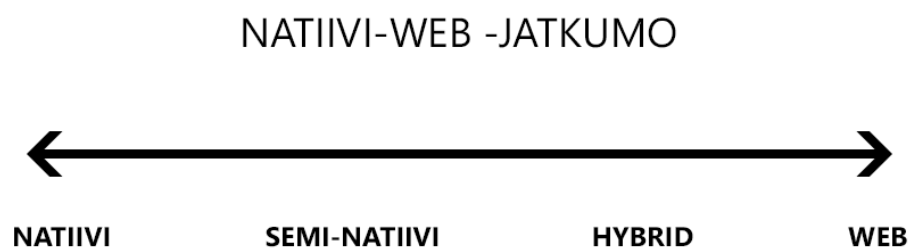


	Natiivi	Hybridi	Web
Kustannukset	Korkeat	Keskitasoa	Matalat
Kehitys	Hidasta	Keskitasoa	Nopeaa
Tukee useita alustoja	Ei	Kyllä	Kyllä
Tehokkuus	Korkea	Keskitasoa	Matala
Ominaisuudet	Laajat	Keskitasoa	Suppeat

KUVIO 5. Mobiilisovelluskehityksen eri menetelmiä

Kuviossa 5 on kuvattu jaottelu mobiilisovelluskehityksen menetelmistä. Tässä jaottelussa natiivimobiilisovellus vastaa pitkälti yleisen natiivisovelluksen kuvausta, eli sovellusta, joka on kehitetty vartavasten kyseistä alustaa varten tarjoten näin parhaan suorituskehon. Toinen ääripää, HTML-sovellus, tarkoittaa mobiilisovellusta, joka ajetaan älypuhelimien verkkoselainsovelluksen sisällä. Näitä kutsutaan usein myös web-sovelluksiksi. Nämä sovellukset toimivat käytännössä lähestulkoon kaikilla päätelaitteilla, jotka pystyvät näyttämään verkkosivuja. Hybridimobiilisovellus taas yhdistää näitä edellä mainittuja, käyttäen hyväkseen puhelimen natiiviominaisuuksia ja verkkoselainsovelluksen yhdenmukaista näkymää.

Uutena tulokkaana ovat niin sanotut semi-natiivit sovellukset, jotka hyödyntävät natiivikirjastoja, mutta hybridisovelluksen tavoin omaavat vain yhden koodipohjan. Esimerkkinä tästä ovat React Native –frameworkillä kehitetyt mobiilisovellukset. Nämä ns. semi-natiivit sovellukset vastaavat suorituskeholtaan natiivisovelluksia (Calderaio 2017).



KUVIO 6. Natiivi-web-jatkumo

Kuviossa 6 mobiilisovellusten kehityssuunnat ovat sijoitettuna natiivi-web-jatkumoon. Lähestyttäessä kuvaajan vasenta ääripäätä mobiilisovelluksen suoritusteho kasvaa. Lähestyttäessä taas kuvaajan oikeaa ääripäätä kasvaa päätelaitteiden määrä, jotka toimivat samalla koodipohjalla.

## 5 TEKNOLOGIAT

### 5.1 NodeJS

NodeJS on avoimen lähdekoodin runtime-ympäristö JavaScript-kielelle. Alun perin sen kehitti Ryan Dahl vuonna 2009. Runtime-ympäristöllä tarkoitetaan ympäristöä, jossa koodi ajetaan. Tyypillisesti JavaScript-kieltä käytetään verkkoselaimissa, jolloin koodi ajetaan käyttäjän laitteessa. NodeJS mahdollistaa JavaScript-koodin ajon palvelimen puolella sekä komentorivityökaluissa. Palvelimen päässä ajettava JavaScript-koodi mahdollistaa dynaamisten verkkosivujen toteutuksen, joissa sivun sisältö luodaan ennen sen lähettämistä käyttäjälle. NodeJS kehitettiin Googlen V8 JavaScript-moottorin pohjalta, joka kääntää JavaScript-koodin natiiville konekielelle (Laurens 2013). Tämä mahdollistaa koodin nopeamman ajon.

### 5.2 npm

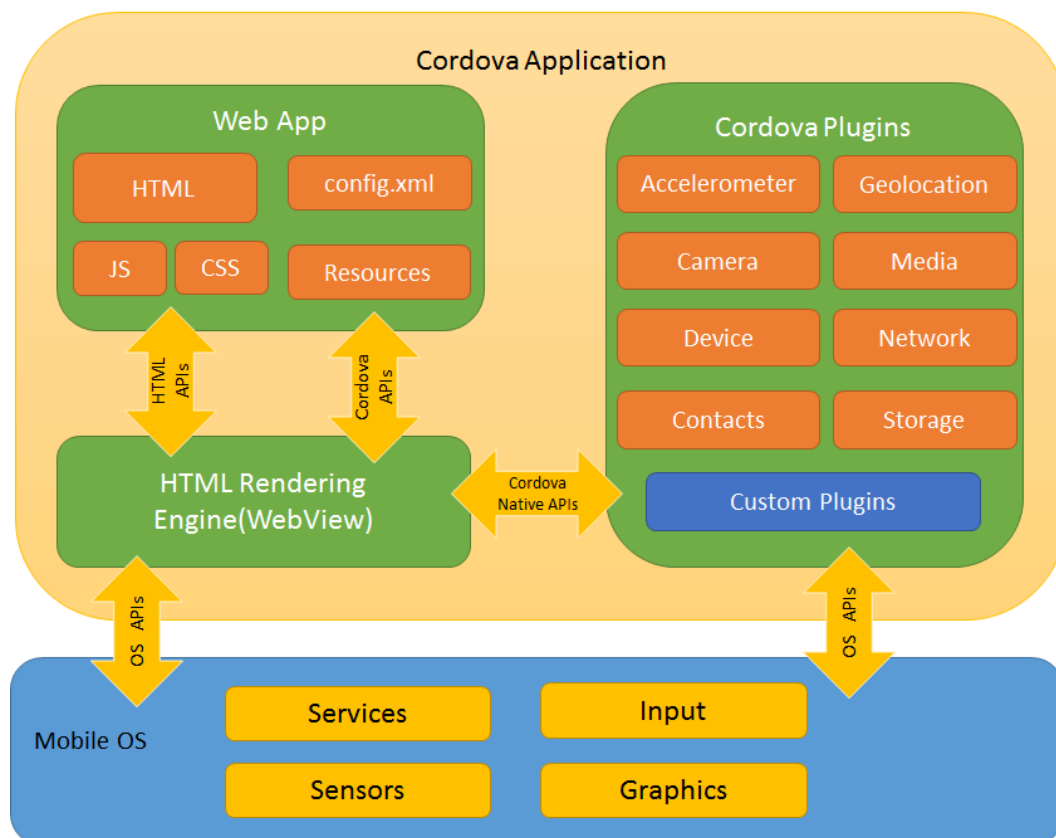
npm on JavaScript-kielen pakkaustyökalu eli niin sanottu package manager. Package manager on ohjelmistotyökalu, jonka avulla ohjelmistojen ja ohjelmistopakettien asentaminen, konfigurointi, päivittäminen ja poistaminen onnistuu automaattisesti, komentoriviltä annettavien komentojen avulla. Package managerin avulla koodin jakaminen ja käyttö onnistuu vaivatta jaettua rekisteriä hyödyntäen. npm on NodeJS-ympäristön oletus- package manager.

### 5.3 Apache Cordova

Apache Cordova (ent. PhoneGap) on avoimen lähdekoodin kehitysalusta hybridimobiilisovelluksille. PhoneGap julkaistiin vuonna 2009 Nitobi-yhtiön toimesta. Adobe osti yhtiön 2011, jolloin tuote uudelleennimettiin Apache Cordovaksi (Cnet 2011).

Alusta mahdollistaa hybridimobiilisovellusten kehittämisen eri alustoille ilman alustakohtaisia ohjelmistorajapintoja, kuten Android, iOS ja Windows Phone. Sen sijaan hybridisovelluksia kehitetään HTML-merkintäkielen,

CSS-tyyliohjeiden ja JavaScript-koodikielen avulla. Näin tuotettu koodi toimii usealla eri päätelaitteella sen verkkosivumaisten teknologioiden ansiosta. Tunnetuimpia Apache Cordovan tukemia alustoja ovat Android, iOS, Symbian ja Windows Phone.



KUVIO 7. Apache Cordovan arkkitehtuuri (Apache Cordova 2018)

Kuviossa 7 on kuvattu Apache Cordova -sovelluksen arkkitehtuuri. Web app -komponentti käsittää mobiilisovelluksen lähdekoodin, joka suoritetaan natiivissa mobiililaitteessa selaimessa toimivan WebView -komponentin sisällä. Web app -komponentin sisältämä sovellus toteutetaan toisin sanoen verkkosivuna WebView -komponentissa.

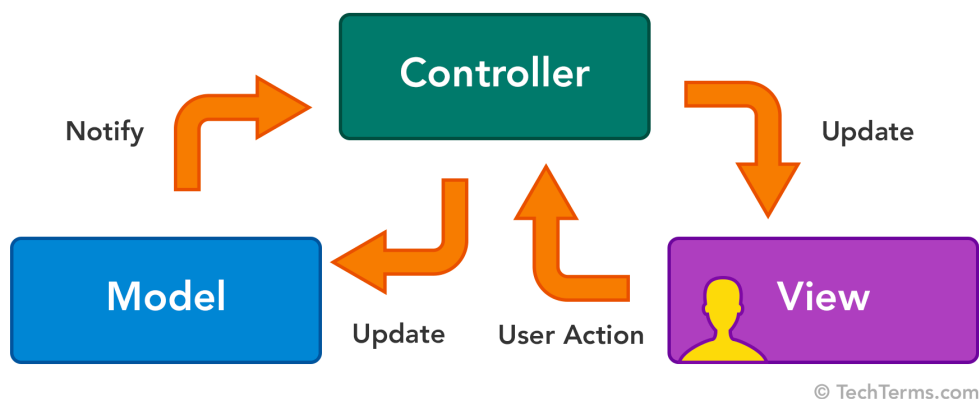
Plugins-komponentti on oleellinen osa Apache Cordova -sovelluksen toimintaa, sillä se tarjoaa rajapinnan sovelluksen ja laitteen natiivien komponenttien välille. Näin sovellus voi käyttää verkkonäkymän kautta esimerkiksi laitteen kameraa. Plugins-komponentti mahdollistaa myös kolmansien osapuolien kehittämien liitännäisten käytön.

## 5.4 AngularJS

AngularJS on avoimen lähdekoodin web-sovelluskehys JavaScript-kielelle. Sen kehitteli kaupalliseen käyttöön Miško Hevery vuonna 2009, mutta sittemmin sovelluskehys julkaistiin avoimen lähdekoodin lisenssin alla (Hevery 2009).

Kehys helpottaa yksisivuisten sovellusten kehittämistä. Yksisivuisella sovelluksella tarkoitetaan web-sovellusta, jossa sisältö ladataan vain kerran ja sitä päivitetään dynaamisesti eri näkymien välillä. AngularJS täydentää osaltaan Apache Cordovaa, lisäten sovelluskehyyksen model-view-controller (MVC) -arkkitehtuurin.

MVC-arkkitehtuuri jakaa sovelluksen kolmeen komponenttiin. Model-komponentti vastaa sovelluksen tietojen hausta ja ylläpidosta. Usein tietokantavuorovaikutus on keskeinen osa tämän komponentin toimintaa. Model-komponentti kommunikoi ainoastaan Controller-komponentin kanssa. View-komponentti vastaa siitä, mitä loppukäyttäjä näkee, kun tämä käyttää sovellusta. Käyttöliittymäelementit ovat keskeinen osa tätä komponenttia. View-komponentti kommunikoi ainoastaan Controller-komponentin kanssa. Controller-komponentti vastaa tietojen päivittämisestä View-komponentin ja Model-komponentin välillä. Sovelluksen näkymän muuttuessa, esimerkiksi käyttäjän painaessa painiketta ruudulla, Controller-komponentti käsittelee käyttäjän syötteen, hakee tarvittavat tiedot Model-komponentilta ja välittää ne edelleen View-komponentille. Voidaankin sanoa, että Controller-komponentti toimii eräänlaisena vuorovaikutustasona View- ja Model-komponenttien välillä.



KUVIO 8. MVC-arkkitehtuuri (TechTerms 2018)

Kuviossa 8 on havainnoitu MVC-arkkitehtuurin toimintaa. Kuvassa View-komponentti ja loppukäyttäjä on luettu yhdeksi komponentiksi, sillä loppukäyttäjän ja käyttöliittymän välistä vuorovaikutusta ei lasketa kuuluvan MVC-arkkitehtuurin piiriin.

AngularJS tarjoaa käyttöön myös direktiivit, jotka ovat laajennettuja HTML-attribuutteja. Direktiivit kertovat AngularJS:n HTML-kääntäjälle, liittävät direktiivin määräinen käyttäytyminen kyseiseen DOM-elementtiin. AngularJS sisältää valmiiksi määriteltyjä direktiivejä, joiden etuliite on ng, mutta direktiivejä voi luoda myös itse. Esimerkkinä tästä on näkyvässä käytettävä direktiivi ng-controller, joka sitoo Controller-luokan kyseiseen näkymään (AngularJS 2018a). AngularJS sisältää myös Servicet, joiden avulla tietoa on helppo järjestellä ja jakaa eri komponenttien kesken sovelluksen sisällä (AngularJS 2018b).

## 5.5 Ionic Framework

Ionic Framework on avoimen lähdekoodin ohjelmistokehityspakkaus, joka tarjoaa työkaluja ja palveluita web-teknologioita käyttävien hybridisovellusten kehittämiseen. Sen kehitti Drifty Co -yhtiö vuonna 2013.

Ionic on rakennettu AngularJS:n ja Apache Cordovan pohjalta. Ionic keskittyy parantamaan mobiilisovelluksen ulkonäköä ja käyttökokemusta, yksinkertaistamalla sovelluksen käyttöliittymän toteutusta. Tämä onnistuu

kehityspakkauksen muokattavien käyttöliittymäelementtien avulla, jotka nopeuttavat sovelluskehitystä, tarjoten kehittäjän käyttöön valmiita ja monikäytännöllisiä käyttöliittymäkomponentteja. Pakkauksen mukana tulee myös laaja kuvakekirjasto, joka osaltaan edesauttaa tarjoamalla kehittäjän käyttöön laajan valikoiman valikoissa ja käyttöliittymissä esiintyviä kuvakkeita.



KUVIO 9. Ionic Button-käyttöliittymäkomponentti

Kuviossa 9 on esimerkki Ionic-kehityspakkauksen Button-käyttöliittymäkomponentista ja sen muutamista valmiista tyyli variaatioista.

## 5.6 ezAR VideoOverlay

ezAR VideoOverlay on ezAR Technologies:n vuonna 2016 kehittämä Apache Cordova-liitännäinen, joka lisää Cordovan Webview-elementin ezAR-kameraelementin päälle (GitHub 2016). Kamerakomponentti näyttää laitteen kameran kuvavirran näkymän taustalla, jonka päällä näytettävä Webview-elementti mahdollistaa virtuaalisen sisällön lisäämisen todellisen näkymän päälle. Liitännäisen avulla otettu kuva ottaa kuvakaappauksen laitteen ruudulta, näin taltioiden kuvaan virtuaalisen sisällön todellista taustaa vasten. Komponentti mahdollistaa myös vaihdon laitteen etu- ja takakameroiden välillä.

<code>ezar.initializeVideoOverlay()</code>	Alustaa ezAR-komponentin, hakee tiedot laitteen kameroista ja zoomaus-toiminnoista.
<code>ezar.getBackCamera().start()</code>	Aloittaa laitteen takakameran videokaappauksen ja kuvavirran esittämisen.
<code>ezar.getBackCamera().stop()</code>	Lopettaa laitteen takakameran videokaappauksen ja kuvavirran esittämisen.
<code>ezar.snapshot()</code>	Ottaa kuvakaappauksen ruudusta ja tallentaa tämän laitteen kuvagalleriaan.

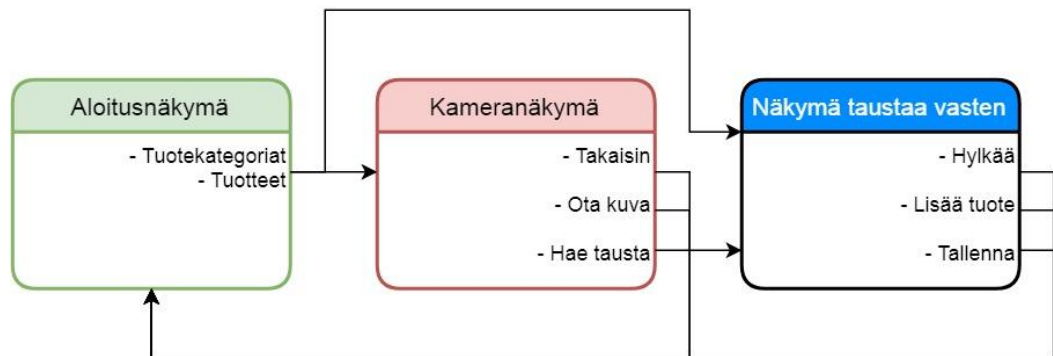
### KUVIO 10. ezAR-komponentin komentoja

Kuviossa 10 on listattu projektissa käytettyjä komponentin komentoja sekä kuvattu niiden toimintaa. Ensimmäisenä listattu komento on merkittävässä roolissa komponentin toiminnan osalta, sillä muut komennot vaativat toimiakseen sen, että komponenti on alustettu ja tiedot laitteen kameroista on taltioitu ennen niiden käyttöä.



## 6 SOVELLUSKEHITYS

Sovelluksen toimintalogiikkaa suunniteltaessa otettiin huomioon sovellukselle asetetut vaatimukset ja niiden pohjalta lähdettiin muovaamaan sovelluksen näkymiä ja niiden välistä navigointia. Sovellussuunnittelun lopputuloksena muodostui kuvion 11 mukainen vuokaavio sovelluksen toimintalogiikasta.



KUVIO 11. Vuokaavio sovelluksen toimintalogiikasta

Sovellus koostuu kolmesta näkymästä ja niiden luomasta toiminnallisuudesta. Ensimmäinen näistä on sovelluksen päänäkö, jossa kategoriat ja tuotteet on listattu. Sovellus on rakennettu siten, että se käyttää kategorioiden ja tuotteiden listaamiseen samaa sivua, vaihtaen ainoastaan sivun tietosäiliöiden sisällön, käyttäjän navigoidessa listauksien välillä. Toinen näkö kattaa sovelluksen Augmented reality -osuuden. Siinä laitteen kameran kuvavirta on ohjattu näkön taustaksi, jonka päällä esitetään käyttäjän valitsema tuote. Sovelluksessa on mahdollista valita, kameran kuvavirran sijaan, näkön taustalle kuva laitteen galleriasta. Kolmas näkö kattaa tämän toiminnallisuuden.

### 6.1 Kehitysympäristö

Kehitysympäristön käyttöjärjestelmänä toimi Microsoft Windows. Tämä on huomionarvoista, sillä vaikka Apache Cordova -kehitysalusta mahdollistaakin mobiilisovelluksen julkaisemisen periaatteessa usealle eri

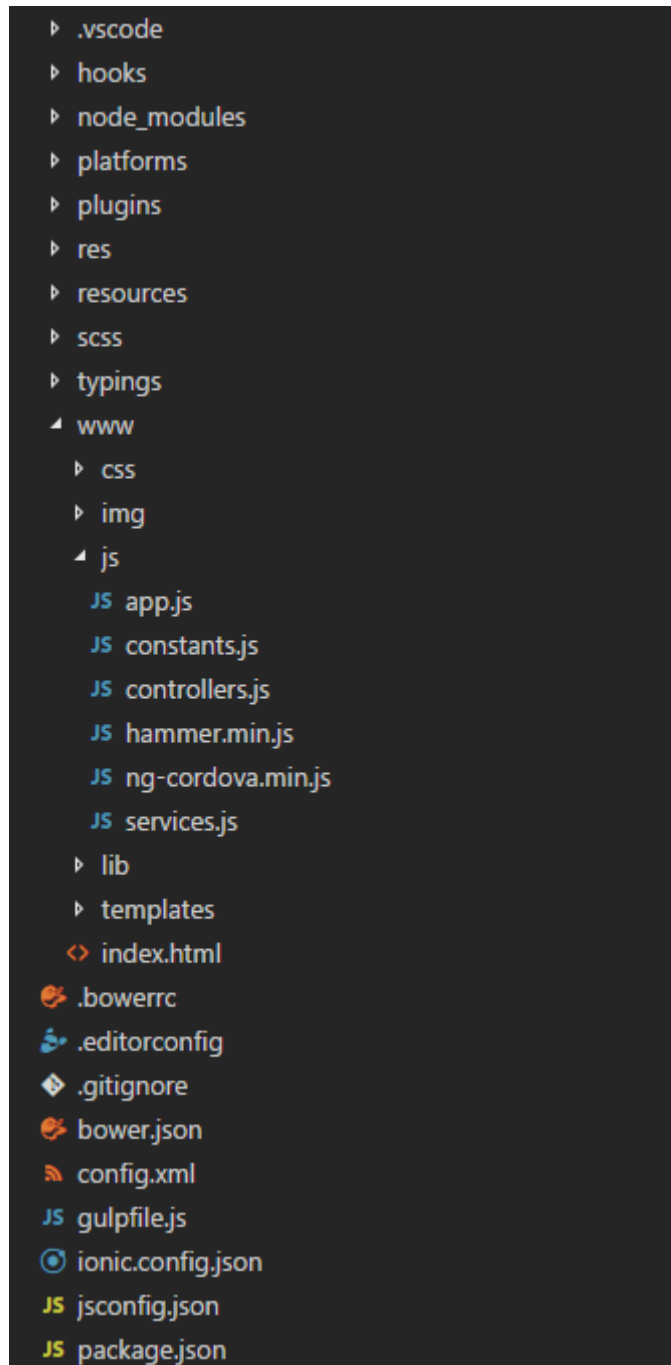
alustalle, vaatii Applen iOS -sovelluksen kääntäminen kehitysympäristöksi Applen macOS:n.

Sovelluskehityksessä koodieditorina käytettiin Microsoftin avoimen lähdekoodin alla kehittämää Visual Studio Code -editoria.

Sovelluskehityksen alkuvaiheessa sovellusta testattiin Googlen Chrome -selaimella, ennen sen kääntämistä Android-mobiilikäyttöjärjestelmälle laiteläheisempää testausta varten. Projektissa käytettiin Git-versionhallintaohjelmaa ja isännöintipalveluna GitHub-verkkototeutusta.

## 6.2 Kansiorakenne

Projektin kansiorakenne on pitkälti Ionic-projektille tyypillinen kansiorakenne. Siinä pääosa sovelluksen toteutuksesta sijoittuu www-kansion alle. Sovelluksen verkkosivumainen rakenne käy hyvin ilmi kansiorakenteesta. Näkymien ulkoasuun vaikuttavat CSS-tyylimääritteet löytyvät css-kansiosta ja sovelluksen käyttämät kuvat img-kansiosta. Sovelluksen toiminnalliset komponentit ovat js-kansiossa, joka nimellään viittaa JavaScript-koodikieleen. Sovelluksen näkymät löytyvät templates-kansiosta. Projektin kansiorakenne on visualisoitu kuviossa 12.

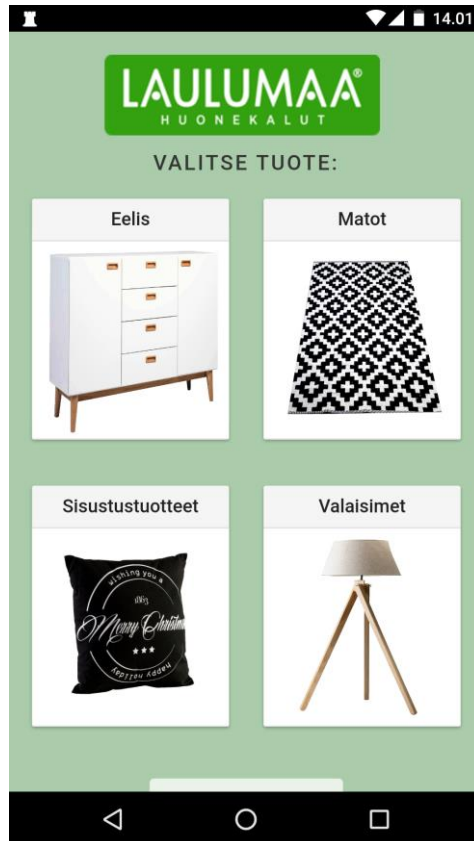


KUVIO 12. Kansiorakenne

### 6.3 Käyttöliittymäsuunnittelu

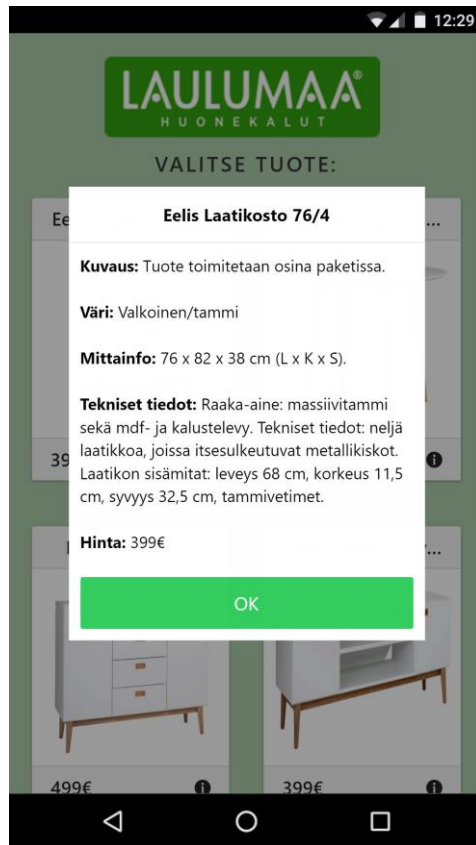
Työn vaatimukset ohjasivat osaltaan sovelluksen käyttöliittymäsuunnittelua. Jotta tuotteita olisi mahdollista selata kategorioittain, sekä tarkastella tarkemmin myös yksittäisenä tuotteena, oli sovelluksen näkymät rakennettava tukemaan tätä toiminnallisuutta.

Tuotehierarkia ohjasi osaltaan näkymien suunnittelua niin, että sovelluksen ensimmäinen näkymä listaa puhelimen ruudulle tuotekategoriat, joita painamalla edetään kyseisen kategorian tuotteisiin. Kuviossa 13 kyseinen näkymä sovelluksessa.



KUVIO 13. Sovelluksen tuotekategorialistaus

Listauksen jäsenet on esitetty eräänlaisina kortteina. Kortissa on kuvan lisäksi ominaista ylä- sekä alaotsikko. Sovellus rakennettiin siten, että siirryttäessä tuotehierarkiassa kategoriasta tuotteisiin, näkymän ulkoasu säilyisi yhtenäisenä. Sama korttimainen listaus esiintyykin myös kategorian tuotteiden listauksessa.



KUVIO 14. Tuotetiedot avattuina

Sovellus rakennettiin hyödyntämään samaa näkymää kussakin listauksessa, ainoastaan näkymien tietosäiliöiden sisällöt päivitettiin vastaamaan uutta tietoa. Tuotelistauksessa on tämän lisäksi mahdollista avata tuotteen tarkemmat tiedot erillisessä ikkunassa. Kuviossa 14 on esitetty tuotteen tarkemmat tiedot avattuna sovelluksessa.



KUVIO 15. Sovelluksen kameranäkymä

Kameranäkymässä (kuvio 15) oleellinen toiminnallisuus on tuotekuvan ruudulla siirtely, skaalaaminen ja kiertäminen kosketuksen avulla. Tämän lisäksi huomioitiin toive mahdollisuudesta avata puhelimen galleriasta kuva kameranäkymän taustalle ja tarkastella tuotekuvia tätä taustaa vasten. Sovelluksen ottaessa kuvan, se piilottaa näytöltä ensin kaikki käyttöliittymäelementit, jonka jälkeen ruutukaappaus toteutetaan ja kuva tallennetaan laitteen kuvagalleriaan. Ruutukaappauksen jälkeen käyttöliittymäelementit palautetaan takaisin ruudulle.

#### 6.4 Backend

Backend toimii rajapintana tietoja taltioivalle tietokannalle. Se tarjoaa väylän lukea ja kirjoittaa tietoja tietokantaan. Sovellus hyödyntää tätä rajapintaa sovelluksen sisäisillä rajapintakutsuilla hakeakseen tarvittavat tiedot tietokannasta. Toisaalta rajapinta tarjoaa myös mahdollisuuden tallentaa uusia tuotteita ja kategorioita tietokantaan. Tätä varten

rajapinnasta tehtiin käyttäjäystävällisempi toteutus hyödyntämällä verkkorajapintaa, jossa suorien rajapintakutsujen sijaan käytetään verkkoselaimen lomake-elementin komponentteja avuksi tietojen välityksessä. Backend-toteutus asennettiin Lahden ammattikorkeakoulun palvelimelle käyttäen PHP-koodikieltä.

Mobiilisovellus kutsuu koodissa rajapintaa, muodostaakseen ensin yhteyden tietokannan kanssa käyttäen esimääriteltyä käyttäjätunnusta ja salasanaa. Tämän jälkeen sovellus toteuttaa tietokantaan haun, saadakseen tietokannasta listauksen tuotteista tai tuotekategorioista.

Asiakkaan verkkorajapinta toimii samaa periaatetta käyttäen, mutta tietokantayhteyteen vaadittava käyttäjätunnus ja salasana poimitaan käyttäjän verkkoselaimen lomakekenttään syöttämistä tiedoista.

Tuotteiden ja tuotekategorioiden haku tapahtuu samalla periaatteella, joskin nyt lisänä on mahdollisuus syöttää uusi tuote tai tuotekategoria. Tämä tapahtuu syöttämällä tuotteen tai kategorian tiedot lomakekenttiin, sekä valitsemalla lähetettävä kuvatiedosto. Kuvatiedosto ladataan tuotteen lisäyksen yhteydessä palvelimelle, jonka jälkeen tämä liitetään uuden tuotteen tai kategorian tietokantaan lisättäviin tietoihin. Loppukäyttäjän älypuhelin hakee uuden tuotteen tai kategorian palvelimelta sovelluksen käynnistyksen yhteydessä.

Interaktiota suunniteltiin kehitettävän myös toiseen suuntaan. Tällöin loppukäyttäjä välittäisi ostoslistan tuotteista sovelluksen rajapinnan kautta yritykselle. Proof-of-concept -toteutuksessa ostoslistan välittäminen Laulumaa Huonekalut Oy:lle jätettiin kuitenkin toteuttamatta.

```

1 <?php
2 include_once("db.php");
3 $target_dir = "uploads/";
4 $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
5 $title=$_POST['product_name'];
6 $title = utf8_decode($title);
7 $category=$_POST['category'];
8 $category = utf8_decode($category);
9 $img=$_FILES["fileToUpload"]["name"];
10 $uploadOk = 1;
11 $imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
12
13 if(isset($_POST["submit"])) {
14     $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
15     if($check !== false) {
16         $uploadOk = 1;
17     } else {
18         echo "File is not an image.";
19         $uploadOk = 0;
20     }
21 }
22
23 if (file_exists($target_file)) {
24     echo "Sorry, file already exists.";
25     $uploadOk = 0;
26 }
27
28 if ($_FILES["fileToUpload"]["size"] > 5000000) {
29     echo "Sorry, your file is too large.";
30     $uploadOk = 0;
31 }
32
33 if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" && $imageFileType !=
"gif" ) {
34     echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
35     $uploadOk = 0;
36 }
37
38 if ($uploadOk == 0) {
39     echo "Sorry, your file was not uploaded.";
40 } else {
41     if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
42         $sql_list = "INSERT INTO laulumaa_products (name, category, img) VALUES ('$title', '$category',
'$img')";
43         $results = mysqli_query($db, $sql_list) or die(mysql_error());
44     } else {
45         echo "Sorry, there was an error uploading your file.";
46     }
47 }
48 ?>

```

## KUVIO 16. Backendin Lisää tuote -toteutus

Kuviossa 16 on esimerkkinä backend-koodi uuden tuotteen lisäämisestä. Rivit 1-11 ovat alustuksia, merkistökäännöksiä ja tietojen poimimista palvelimen vastaanottamista HTTP POST -kutsun avain-arvo-pareista. Tuotteen lisäyksessä avain-arvo-pareja ovat tuotteen nimi, kategoria ja kuvatiedosto. Seuraavaksi riveillä 13-21 koodi tarkastaa, että käyttäjän syöttämä tiedosto on kuva, määrittämällä sen leveys- ja korkeustiedot. Tämän jälkeen rivit 23-26 tarkastavat, löytyykö samannimistä tiedostoa jo ennestään palvelimelta. Rivit 28-31 tarkastavat, ettei syötetyn kuvatiedoston koko ylitä maksimirajaa, joka on määritetty viiteen megatavuun. Viimeinen tarkistus riveillä 33-36 tarkistaa, että



kuvatiedoston tiedostotyyppi on yleisesti tunnettu. Toteutuksen hyväksymät tiedostopäätteet ovat gif, jpg, jpeg ja png. Mikäli syötetty tiedosto läpäisee edellä mainitut tarkistukset, se ladataan palvelimelle ja tämän jälkeen lisätään syöte uudesta tuotteesta tuotetietokantaan. Virhetilanteessa backend lähettää käyttäjälle tiedon virheestä ja tämä virheviesti tulostetaan verkkosivulle.

## 7 SOVELLUS

Sovelluksen pääasiallisena tarkoituksena on testata Augmented reality -teknologiaa ja toimia proof-of-concept -toteutuksena tälle konseptille. Etenkin näkymä, jossa käyttäjä voi sijoitella, skaalata ja kiertää valitsemaansa tuotetta laitteen kameran kuvavirran päällä, on toteutuksen kannalta keskeisessä asemassa.

### 7.1 Sovelluksen toiminta

Sovellus käynnistyy aloitusnäkömään, jossa tuotekategoriat ovat listattuna. Kun käyttäjä valitsee tuotekategorian klikkaamalla kyseistä korttia, näkymän listaus päivittyy vastaamaan kyseisen kategorian tuotteita. Käyttäjän valittaessa tuote sitä klikkaamalla, avautuu sovelluksen kameranäkymä. Käyttäjä voi siirtää tuotetta laitteen näytöllä ”raahaamalla” sitä yhden sormen painalluksella. Tuotteen skaalaaminen onnistuu kahden sormen ”nipistys”-kosketuksella ja kiertäminen vastaavasti kiertämällä kahden sormen kosketusta. Kun käyttäjä on sommitellut tuotteen ruudulle haluamallaan tavalla, näkymän alareunassa on painike, jonka avulla sovellus ottaa ruutukaappauksen ja tallentaa kuvan laitteen kuvagalleriaan. Näkymän alareunan valikosta löytyy myös painike, jonka avulla voidaan laitteen galleriasta hakea kuva näkymän taustalle. Kun käyttäjä on valinnut kuvan näkymän taustalle, sovellus siirtyy erilliseen näkömään. Tämä näkymä muistuttaa kameranäkymää, mutta siinä tuotteita taustalla olevan kuvan päälle voi lisätä useita. Kun käyttäjä painaa Lisää tuote -painiketta, sovellus ottaa uuden ruutukaappauksen, korvaten tällä edellisen. Näin saadaan uusi tausta, jossa on mukana käyttäjän juuri lisäämä tuote. Samaan aikaan käyttäjä ohjataan valitsemaan uutta tuotetta. Valinnan jälkeen käyttäjä palaa ruudulle, jossa on taustalla edellisessä istunnossa tallennettu kuva. Tallenna-painikkeella sovellus ottaa ruutukaappauksen, ohjaamatta käyttäjää uudelleen samaan navigaatiopolkuun. Sen sijaan Tallenna-painike lopettaa polun ja sovellus palaa normaalissa tilassa aloitusnäytölle.

## 7.2 Backend

Verkkorajapinta, jonka kautta sovellukseen voi lisätä tuotteita ja tuotekategorioita, toimii verkkoselaimen kautta. Rajapintaan on ensin kirjaututtava ylläpitotunnuksin, jotta tietoja pääsee muuttamaan (kuvio 17).

**Kirjautuminen**

Käyttäjätunnus Salasana

Kirjaudu sisään

KUVIO 17. Backendiin kirjautuminen

Kirjautumisen jälkeen käyttäjä ohjataan sivulle, jossa kategoriat ja tuotteet on listattu. Syöttääkseen uuden kategorian tai tuotteen, käyttäjän tulee ensin täyttää tarvittavat tiedot sivun syöttökenttiin.

**Lisää uusi tuote:**

Id  
Eelis  
Valitse tiedosto Ei valittua tiedostoa

Lisää

310031	Eelis Laatikosto 76/4	Valkoinen/tammi	399	499	Raaka-aine: massiivitammi sekä mdf- ja kalustelevy. Tekniset tiedot: neljä laatikkoa, joissa itsesulkeutuvat
--------	-----------------------------	-----------------	-----	-----	--

KUVIO 18. Backendin Lisää tuote -toiminnallisuus

Kuviossa 18 on havainnoitu uuden tuotteen lisäämistä. Tätä varten lisättävästä tuotteesta tarvitsee syöttää tuotteen yksilövä id, kategoria johon tuote kuuluu sekä syväty tuotekuva.

## Lisää uusi kategoria:

1	Eelis	Eelis senkki 125.gif
2	Matot	Milo_160x230.gif
3	Sisustustuotteet	Kotyyny1.gif
4	Valaisimet	Skane Lattiavalaisin.gif

Nimi	
Valitse tiedosto	Ei valittua tiedostoa
<b>Lisää</b>	

KUVIO 19. Backendin Lisää kategoria -toiminnallisuus

Kuviosta 19 ilmenee vastaavasti backendin Lisää kategoria -toiminnallisuus. Kategoriaa lisättäessä vaatimuksina on ainoastaan kategorian nimi sekä havainnekuva, jonka ei tarvitse olla syväty.

Kun syöttökentät on täytetty, käyttäjä voi lisätä tuotteen tai kategorian Lisää-nappia painamalla. Tuotekuva ladataan ensin palvelimelle, jonka jälkeen uudesta syötteestä kirjoitetaan merkintä tietokantaan. Käyttäjä voi sulkea istonnon kirjautumalla ulos rajapinnasta Kirjaudu ulos -painikkeella.

## 8 YHTEENVETO

Työn tavoitteena oli testata Augmented reality -teknologiaa hyödyntävä mobiilisovellus, jonka avulla loppukäyttäjän on mahdollista tarkastella Laulumaa Huonekalut Oy:n tuotteita puhelimen takakameran tarjoamaa näkymää vasten.

Työn teoriaosuudessa tutkittiin Augmented reality -teknologian nykytilaa ja sen toteutusmahdollisuuksia älypuhelinlustralle. Teoriaosuudessa paneuduttiin myös erilaisiin tapoihin tuottaa mobiilisovellus ja vertailtiin niiden eroavaisuuksia. Tämän lisäksi avattiin hybridimobiilisovelluksen toteutuksessa käytettyjä tekniikoita.

Käytännön osuudessa käytiin läpi sovelluksen rakennetta ja toteutusta. Lisäksi avattiin myös backend-rajapintaa ja sen toiminnallisuutta.

Tuloksena oli toimiva mobiilisovellus, joka vastasi asiakkaan vaatimuksia. Laulumaa Huonekalut Oy:n toimitusjohtaja Aki Alenius oli erittäin tyytyväinen projektin tulokseen.

Versioon 2.0 parantelisin toteutusta niin, että sovellus hakisi tuotteet ja niiden tiedot suoraan Laulumaan Huonekalut Oy:n verkkokaupasta. Lisäksi jatkokehittäisin ominaisuuden, jossa loppukäyttäjä voisi sovelluksen avulla lähettää ostoslistan huonekaluista.

## LÄHTEET

Android 2018. Application Fundamentals [viitattu 23.10.2018]. Saatavissa: <https://developer.android.com/guide/components/fundamentals>

AngularJS 2018a. Directives [viitattu 23.10.2018]. Saatavissa: <https://docs.angularjs.org/guide/directive>

AngularJS 2018b. Services [viitattu 23.10.2018]. Saatavissa: <https://docs.angularjs.org/guide/services>

Apache Cordova 2018. Overview [viitattu 23.10.2018]. Saatavissa: <https://cordova.apache.org/docs/en/latest/guide/overview/>

Bitmovin 2018. There's an App for That? HTML5 vs Native for In-App Video Playback [viitattu 23.11.2018]. Saatavissa: <https://bitmovin.com/theres-app-html5-vs-native-app-video-playback/>

Calderaio, J. 2017. Comparing the Performance between Native iOS (Swift) and React-Native [viitattu 23.11.2018]. Saatavissa: <https://medium.com/the-react-native-log/comparing-the-performance-between-native-ios-swift-and-react-native-7b5490d363e2>

Cnet 2011. Adobe buys PhoneGap, TypeKit for better Web tools [viitattu 23.10.2018]. Saatavissa: <https://www.cnet.com/news/adobe-buys-phonegap-typekit-for-better-web-tools/>

Digital Journal 2018. Augmented Reality Market Booming at a CAGR of 51.24% by 2023 [viitattu 23.10.2018]. Saatavissa: <http://www.digitaljournal.com/pr/3903464>

GitHub 2016. ezartech/ezar-ionic [viitattu 23.10.2018]. Saatavissa: <https://github.com/ezartech/ezar-ionic>

Hevery, M. 2009. Hello World, <angular/> is here [viitattu 23.10.2018]. Saatavissa: <http://misko.hevery.com/2009/09/28/hello-world-angular-is-here/>

Junkkari, S. 2012. Milgramin jatkumo [viitattu 16.10.2018]. Saatavissa: [https://fi.wikipedia.org/wiki/Lis%C3%A4tty\\_todellisuus#/media/File:Laajennettu\\_todellisuus\\_5\\_milgram.png](https://fi.wikipedia.org/wiki/Lis%C3%A4tty_todellisuus#/media/File:Laajennettu_todellisuus_5_milgram.png)

Kauppalehti 2018. Laulumaa Huonekalut Oy yritystiedot [viitattu 4.12.2018]. Saatavissa: <https://www.kauppalehti.fi/yritykset/yritys/laulumaa+huonekalut+oy/27409439>

Laurens, T. 2013. How the V8 engine works? [viitattu 23.10.2018]. Saatavissa: <http://thibaultlaurens.github.io/javascript/2013/04/29/how-the-v8-engine-works/>

Messukeskus 2017. Augmented Reality eli lisätty todellisuus tekee messukäynnistä seikkailun! [viitattu 23.10.2018]. Saatavissa: <https://messukeskus.com/yrityksille/mainospaikat-jalisanakyvyys/augmented-reality-lisatty-todellisuus/>

Milgram, P. 1994. Augmented reality: a class of displays on the reality-virtuality continuum [viitattu 19.11.2018]. Saatavissa: [http://etclab.mie.utoronto.ca/people/paul\\_dir/SPIE94/SPIE94.full.html](http://etclab.mie.utoronto.ca/people/paul_dir/SPIE94/SPIE94.full.html)

StatCounter 2018b. Mobile Operating System Market Share Worldwide [viitattu 23.10.2018]. Saatavissa: <http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201710-201810-bar>

StatCounter 2018a. Mobile Operating System Market Share Finland [viitattu 23.10.2018]. Saatavissa: <http://gs.statcounter.com/os-market-share/mobile/finland#monthly-201710-201810-bar>

TechCrunch 2016. With 500M downloads, Pokémon Go is coming to the Apple Watch [viitattu 23.10.2018]. Saatavissa: <https://techcrunch.com/2016/09/07/pokemon-go-the-hottest-game-on-the-planet-is-coming-to-the-apple-watch/>

TechTerms 2018. MVC [viitattu 23.10.2018]. Saatavissa: <https://techterms.com/definition/mvc>

Unity Technologies 2018. Products [viitattu 19.11.2018]. Saatavissa:

<https://unity3d.com/unity>

VentureBeat 2014. Apple announces 'Swift,' a new programming language for OS X & iOS [viitattu 23.10.2018]. Saatavissa:

<https://venturebeat.com/2014/06/02/apple-introduces-a-new-programming-language-swift-objective-c-without-the-c/>

W3 2014. Kolme tapaa kehittää mobiilisovellus [viitattu 23.10.2018].

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

Wikipedia 2018. Augmented reality [viitattu 16.10.2018]. Saatavissa:

[https://en.wikipedia.org/wiki/Augmented\\_reality](https://en.wikipedia.org/wiki/Augmented_reality)



