



Timurhan Jari Calisiyor

## Verkkotilaussovellus ravintolalle

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

17.5.2021

## Tiivistelmä

Tekijä: Timurhan Calisiyor  
Otsikko: Verkkotilaussovellus ravintolalle  
Sivumäärä: 39 sivua  
Aika: 17.5.2021

Tutkinto: Insinööri (AMK)  
Tutkinto-ohjelma: Tieto- ja viestintätekniikka  
Ammatillinen pääaine: Mobile Solutions  
Ohjaaja: Lehtori Ilkka Kylmäniemi

---

Insinööriyössä perehdyttiin erilaisiin verkkotilaussovelluksiin ja verkkotilaussovellusten toteutustapoihin. Työn tarkoituksena oli luoda ravintolalle konsepti omasta natiivista tilaussovelluksesta, jota voitaisiin käyttää välityspalvelujen kanssa tai niiden sijasta tarvittaessa. Kehitettävän sovelluksen avulla tulisi pystyä lisäämään ostoskoriin kaikki ravintolan ruokalistan annokset. Sovelluksen avulla pitäisi pystyä myös lähettämään tilaus niin, että se voitaisiin ravintolassa ottaa vastaan ja toteuttaa.

Suosittujen verkkotilaussovellusten etuna on niiden helppokäyttöisyys. Välityspalveluiden sovelluksista löytyy tuhansia ravintoloita, mikä on asiakkaiden kannalta kätevää. Välityspalvelujen kautta tilattu ruoka on usein myös saman hintaista kuin ravintolassa. Välityspalvelut tulevat komissionen monelle pienravintolayrittäjälle liian kalliiksi.

Ennen oman ravintolan verkkotilaussovelluksen kehityksen aloittamista on hyvä perehtyä erilaisiin toteutustapoihin. Mobiililaitteiden suosio tarkoittaa kuitenkin sitä, että ravintolan kannalta natiivin sovelluskaupoista ladattavan sovelluksen kehittäminen on järkevin vaihtoehto. Natiivi Android- ja iOS-sovelluskehitys voi olla monelle yrittäjälle liian kallista, minkä takia kannattaa valita kehitykseen ohjelmistokehitys, kuten React Native. Sovelluksessa on tärkeä olla toimintoja, joilla voi houkuttaa käyttäjiä vaihtamaan välityspalvelujen sovellukset omaan sovellukseen.

Insinööriyön kehityksen aikana käytettiin tutkimuksissa kerättyä tietoa ja luotiin React Nativella Android-sovellus, jolla tilausten tekeminen on mahdollista. Sovelluksen kehitystyö jatkuu siihen asti, että sovellus saadaan sovelluskauppoihin.

Avainsanat: mobiilisovellus, React Native. Redux, Node

## Abstract

Author: Timurhan Calisiyor  
Title: Online ordering application for a restaurant  
Number of Pages: 39 pages  
Date: 17 May 2021

Degree: Bachelor of Engineering  
Degree Programme: Information and Communications Technology  
Professional Major: Mobile Solutions  
Instructor: Ilkka Kylmäniemi, Senior Lecturer

---

This thesis focuses on getting to know different kinds of online ordering applications and implementations on online ordering applications. The target of the mobile application that was developed for this thesis is to be a proof of concept of a restaurant specific online ordering system that can be used instead of, or with online food delivery companies. The application should have functionalities for users to be able to add any items from the restaurant's menu to a shopping cart and send the order to the restaurant.

The advantage of most popular online food delivery companies is that they are easy to use. Online food delivery companies' applications host thousands of restaurants, which is very convenient to the users. In addition, the cost of ordering from online food delivery companies is often priced like the restaurant's own menu. Some restaurant owners think the commission costs are too high for the service provided by online food delivery companies.

Before a restaurant starts the development process for their own online ordering application, it is a good idea to explore the different ways to develop an application. The rise in popularity of smartphones means that the best solution for development might often be a native mobile application that is available to customers in app stores. Fully native Android or iOS development might be too costly to many small restaurants in which case the best option might be to develop the app with a framework such as React Native. React Native allows for native Android and iOS applications with one and the same code base. If a restaurant chooses to go down the road of application development, it is important to have some sort of a customer loyalty program in the application to make it more convenient to use instead of the applications of online food delivery companies.

The data collected in the research part of the thesis was used in the development work with React Native. The purpose was to create a native mobile application that has a functionality for creating and sending food orders. The development of the application will continue after the thesis until the application is available in the app stores.

Keywords: Mobile, React Native, Redux, Node

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Verkkotilaussovellukset	3
2.1	Tilauspalveluiden historia ja kehitys	4
2.2	Kilpailijoiden analysointi	6
2.3	Verkkotilaussovelluksien kehitystyö	12
3	Ravintolakohtaisen sovelluksen suunnittelu	16
3.1	Käyttöliittymä ja navigointi	16
3.2	Sovelluksen vaatima toiminnallisuus	19
3.3	Työkalujen valinta	20
4	Ravintolan tilausjärjestelmä	21
4.1	React ja React Native	21
4.2	Redux-kirjasto	22
4.3	Front endin kehitys	23
4.4	Nodemailer-moduuli	28
4.5	Back endin kehitys	29
4.6	Tulokset ja testaus	30
4.7	Mobiilisovelluksen tulevaisuus	32
5	Yhteenveto	34
	Lähteet	36

## Lyhenteet

- MaRa: Matkailu- ja Ravintolapalvelut MaRa on matkailu-, majoitus-, ravintola- ja vapaa-ajanpalvelualan sekä niihin liittyvän hyvinvointipalvelualan yrittäjien ja työnantajien edunvalvoja ja työmarkkinajärjestö.
- PWA: Progressiivinen verkkosovellus (Progressive Web Application). Mobiililaitteen verkkoselaimessa toimiva sovellus, joka pyrkii yhdistelemään natiivin mobiilisovelluksen ja reagoivan verkkosivun parhaat puolet.
- API: API eli ohjelmointirajapinta on määritelmä, jonka mukaan eri ohjelmat voivat lähettää ja vastaanottaa tietoja toisiltaan.
- MVP: Pienin toimiva tuote (Minimum Viable Product), on uusi markkinoille tuotava tuote. MVP:ssä on sellaiset ominaisuudet, jotka riittävät käyttäjiltä palautteen keräämiseen tuotekehitystä varten.
- DOM: Dokumenttioliomalli (Document Object Model) on tapa, jolla kuvataan dokumentin rakennetta puuna, jonka eri olioita voi hakea. DOM:n tarkoitus on määrittää, miten elementit välittävät tietoa keskenään.

# 1 Johdanto

Melkein jokaisen taskusta löytyy nykypäivänä mobiililaitte, ja mobiilisovelluksia käytetään enemmän kuin koskaan. Mobiilisovellusten suosion kasvaessa on erilaisten sovellusten käyttö ruoan tilaamiseen myös yleistynyt. Mobiilisovellus voi myös vaikuttaa suoraan ravintolan tuloihin. Esimerkiksi Hesburger kasvatti myyntiään 25 prosenttia mobiilisovelluksen julkaisun myötä. (1.)

Suosittuja Suomessa käytettyjä sovelluksia ruoan tilaamiseen ovat Foodora ja Wolt. Ne ovat välityspalveluita, jotka tarjoavat alustansa ravintoloiden ja yritysten käyttöön. Woltin välityspalvelu toimii siten, että Woltin oma lähetti tulee noutamaan ravintolaan Woltin kautta tulleen tilauksen ja vie sen tilaajalle. Lähettipalvelu on osa myös Foodoran toimintamallia, mutta Foodoraa voi käyttää myös vain tilausten välitykseen, mikäli ravintolalla on oma kuljetuspalvelu. Välityspalvelut veloittavat alustansa kautta tulleista tilauksista komission. Alustojen kaltaista hinnasta huolimatta niiden tuoma näkyvyys ja kauppa on elinehto monelle pienravintolalle.

Edellä mainituista syistä moni pienravintolayrittäjä saattaa haluta kokeilla oman sovelluksen menestymismahdollisuutta. Monella ravintolalla on olemassa omat verkkosivut, mutta verkkosivut eivät ole yhtä käytännöllisiä pikaruoan tilauksen tekemiseen. Mobiilisovelluksen käyttö on asiakkaalle helpompaa ja nopeampaa. Mobiilisovelluksen edut ovat kuitenkin suuremmat kuin vain asiakaskokemuksen parantaminen. Mobiilisovellus on näkyvillä mobiililaitteissa aina, kun sitä käytetään. Omaan sovellukseen investoiminen kohentaa ravintolan brändiä osoittamalla asiakkaille, että ravintola haluaa tarjota parhaan mahdollisen kokemuksen.

Insinöörityössä selvitetään, millaiseen kehitystyöhön ravintolan kannattaa keskittyä, kun se päättää investoida omaan mobiilisovellukseen. Samalla tutkitaan, miten oma sovellus saadaan erottumaan välityspalveluista.

Insinööriyön tarkoituksena on tutkia menestyvien ruoantilauspalveluiden toimintatapoja ja selvittää niiden ongelmat ja edut. Lisäksi selvitetään, minkälainen kehitystyö sopii pienravintolan mobiilisovelluksen luomiseen ja minkälaisilla toiminoilla voidaan edistää asiakkaiden siirtymää omaan sovellukseen. Taustatyön tekemisen jälkeen kehitetään ravintolalle oma mobiilisovellus. Sovelluksella tilausten tekeminen ja lähettäminen on mahdollista. Mobiilisovelluksen kehitystyöhön käytetään React Nativea, jolla keskitytään toteutukseen Android-käyttöjärjestelmälle. Työn teettävä ravintola toivoo sovelluksen avulla saavansa asiakkaiden tilaamiskäytöksessä aikaan muutoksia.

## 2 Verkkotilaussovellukset

Ruoan verkkotilauspalvelut ovat olleet jo muutamien vuosien ajan kovemmassa käytössä kuin koskaan. Kun kasvavan trendin lisäksi koronapandemia alkoi levitä maailmanlaajuisesti vuoden 2020 alussa, ruoan kuljetuspalveluiden suosio lähti räjähdysmäiseen kasvuun. Yhä useampi tekee töitä kotona ja korvaa lounaan kollegojen kanssa tilausruoalla. Myös vapaa-aika vietetään useammin kotona, mikä lisää ruoan kotiin tilaamista myös vapaa-ajalla. Matkailu- ja Ravintolapalveluiden (MaRa) teettämän trenditutkimuksen mukaan 13 prosenttia vastaajista aikoo jatkaa ravintolaruoan kuljetuspalveluiden käyttöä enemmän myös tulevaisuudessa. (2.) Se tarkoittaa, että yritysten kannattavuus sijoittaa kuljetus- ja verkkotilauspalveluihin on nousussa.

Helpoin ratkaisu monelle yritykselle on ottaa käyttöön jo olemassa oleva palvelu, johon kuuluu sekä verkkotilauspalvelu että kuljetuspalvelu. Tällaisia välityspalveluita ovat esimerkiksi Wolt ja Foodora (entinen Pizza Online). Woltilla on Suomessa yli 2 000 ravintolaa asiakkainaan. Foodoralla asiakasravintoloita on lähes 3 000. Wolt ja Foodora ovatkin saaneet osansa mainitusta räjähdysmäisestä kasvusta. (3.) Wolt ja Foodora tuovat ravintolalle omat etunsa ja myös haittansa. Jotkut ravintolat eivät halua taipua Woltin ja Foodoran ehtoihin, joten vaihtoehdoksi jää oman kuljetus- ja verkkotilauspalvelun luominen. Monilla pika-ruokaravintoloilla oman kuljetuspalvelun järjestäminen on ollut tavallinen käytäntö jo vuosia. (2.)

Insinööriyössä on tarkoitus luoda pitseriaravintolalle natiivi verkkotilaussovellus Androidille ja myöhemmin iOS:lle React Nativea käyttäen. Työn teettävä ravintola haluaa nähdä, voiko mobiilisovelluksen tuoma helppous lisätä tilauksia omien tilausjärjestelmien kautta ravintolan käytössä olevan Foodoran sijasta. Foodoran lisäksi ravintolalla on oma kuljetuspalvelu ja verkkoselaimella toimiva verkkotilausjärjestelmä. Insinööriyönä tehtävän sovelluksen on tarkoitus tulla ravintolan vakituiseen käyttöön.



## 2.1 Tilauspalveluiden historia ja kehitys

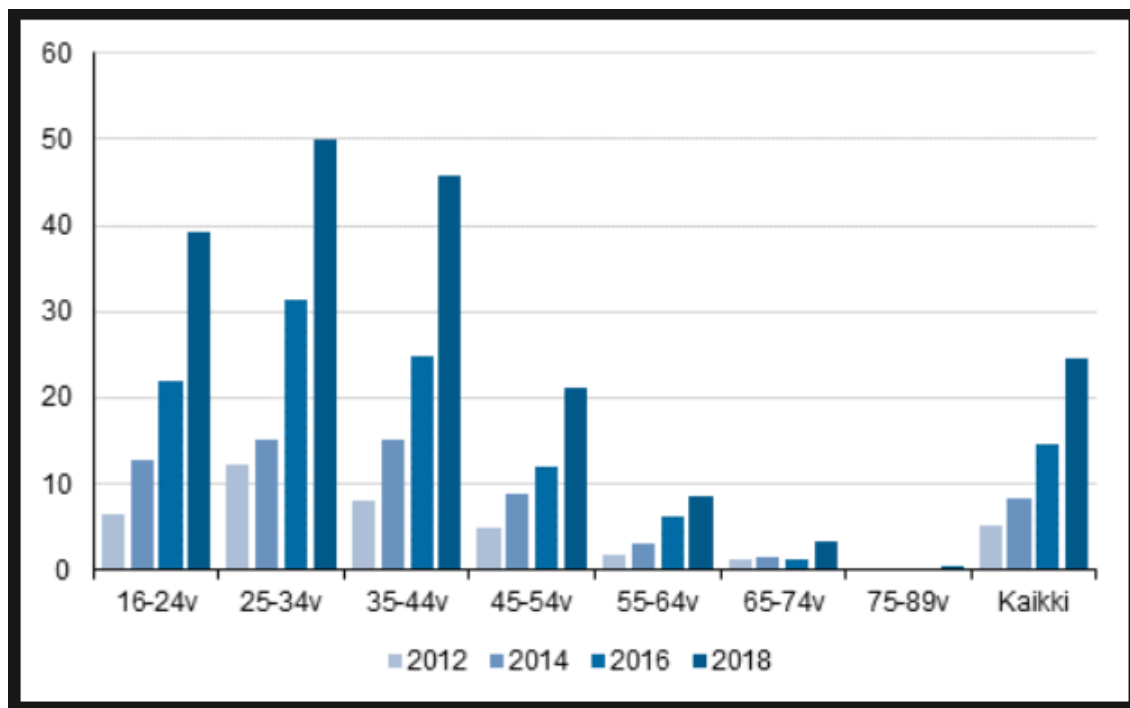
Ruoan kuljetuspalvelut ovat olleet olemassa jo vuosikymmenien ajan. Trendin aloittajia olivat pitsaravintolat ja muut pikaruokaravintolat. Tilaukset tehtiin ennen puhelinsoitolla suoraan ravintolaan. Tämä normalisoitui tavaksi tilata ruokaa kotiin.

Nykyisin tilaaminen puhelinsoitolla voidaan kokea vaivalloisena. Puhelintilausten huonona puolena on, että vastassa voi olla kielimuuri. Kielimuuri puhelintilauksessa voi usein johtaa virheeseen tilauksessa, jolloin tilaaja joutuu odottelemaan uutta ruokaansa. Myös ravintolan täytyy käyttää aikaa virheellisen tilauksen korjaamiseen. Puhelintilaus on nykyään myös usein hidasta ja hankalaa, kun verrataan verkkotilausjärjestelmien helppouteen. Tilauksen tekeminen on usein vain muutaman painalluksen päässä. Asiakkaan itse täyttämässä tilauslomakkeessa on usein pienempi mahdollisuus virheiden ilmenemiseen. Jos tilauksessa ilmenee virhe, se ei ole ravintolan vastuulla, sillä asiakas on itse täyttänyt tilauksensa. Moni nykyihminen kokee myös niin sanotusti puhelinahdistusta eli puhelinsoiton pelkoa tai jännitystä. Verkkosivulla teetetyn kyselyn mukaan vain 6 prosenttia vastaajista ei koskaan koe ahdistusta puhelimella soittamisesta. (4.) Voidaan siis olettaa, että joillekin asiakkaille puhelinsoiton välityksellä tilaaminen on epämiellyttävä kokemus, jota saatetaan vältellä viimeiseen asti.

Ensimmäiset verkossa käytettävät tilausjärjestelmät olivat ravintoloiden luomat omat verkkosivut, joilta ruoan tilaus oli mahdollista. Tämä vaati sen, että yrittäjä hoiti itse kuljetuksen järjestämisen. Nykyihmiset voivat kokea verkkosivujen kautta tilaamisen hankalaksi. Joko pitäisi avata tietokone, jolla navigoisi ravintolan verkkosivuille, tai avata verkkoselain puhelimella ja sitä kautta tehdä tilaus. Kun tilausprosessista poistetaan vielä lisää askelia mobiilisovellusten avulla, eivät ihmiset viitsi vaivautua käyttämään ravintoloiden verkkosivuja tilauksen tekemiseen.

Tilastokeskuksen tekemän tutkimuksen mukaan 16–89-vuotiaista suomalaisista vuonna 2018 käytti internetiä 89 prosenttia ja 76 prosenttia käytti internetiä

monta kertaa päivässä. Yleisin laite internetin käyttöön oli matkapuhelin, jota käytti kolme neljäsosaa väestöstä. Samana vuonna älypuhelin löytyi 80 prosentilta väestöstä ja 16–64-vuotiailta jopa yli 92 prosentilta. (5.) Älypuhelimien yleisyydessä myös niillä tehdyt ostokset ovat olleet kasvussa (kuva 1).



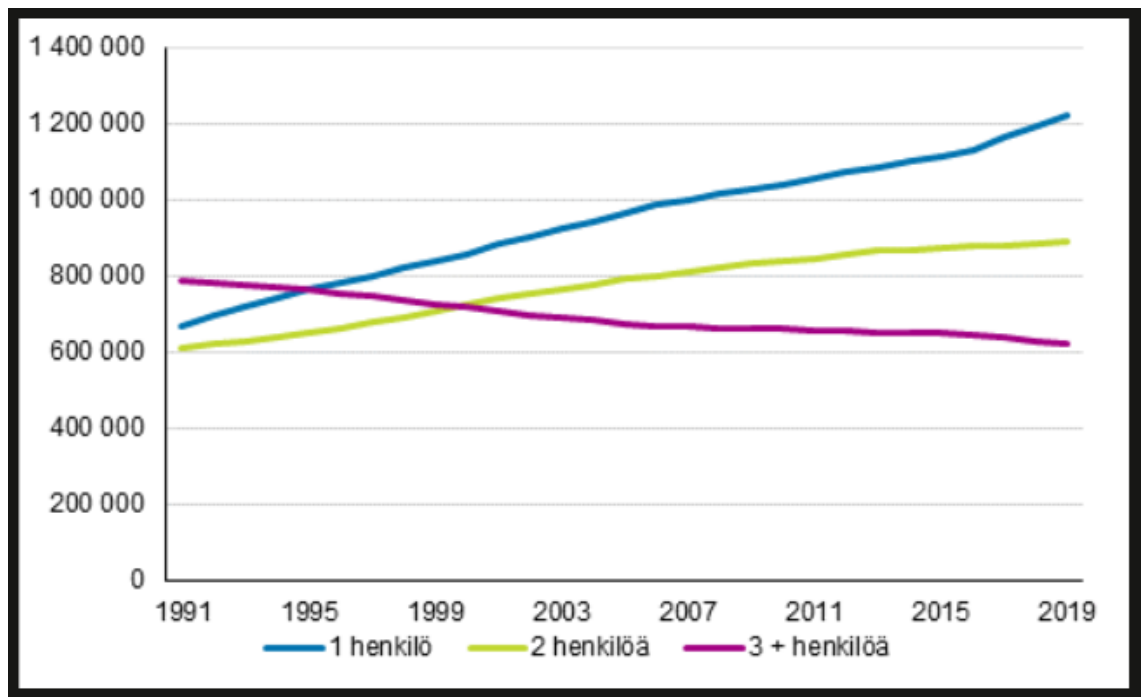
Kuva 1. Matkapuhelimella tehdyjen ostojen prosenttiosuus väestöstä (6).

Älypuhelimien ja niillä tehdyjen ostosten kasvun myötä ovat mobiilisovellukset yleistyneet tavaksi tilata ruokaa kotiin. Tilanteesta täyden hyödyn Suomessa ovat saaneet välityspalvelut, kuten Foodora ja Wolt.

Foodora ja Wolt eroavat hieman toisistaan toimintatavoiltaan, mutta molemmat luokitellaan ruokatilausten välityspalveluiksi. Woltin palveluun kuuluu myös kuljetuspalvelu ja omat kuriirit, kun taas Foodoralla kuljetuspalvelu on vapaavalintainen, jos asiakasyrityksellä on oma kuljetuspalvelu. Vuonna 2002 aloittanut Gastronomatti yritti samantapaista ruoan välityspalvelun liikeideaa. (7.) Gastronomatin listoilta löytyi hyvä valikoima ravintoloita, mutta palvelu ei kuitenkaan lähtenyt samoin tavoin nousuun kuin nykypäivän vastaavat palvelut. Gastronomatti epäonnistui aikanaan ja syitä voi arvella, mutta huomattava asia on,

että suomalainen yhteiskunta oli hyvin erilainen. Lisäksi ei ollut maailmanlaajuisia pandemioita vauhdittamassa ihmisiä etäruokailuun. MaRa:n vuonna 2020 teettämässä ravintolaruokailun trenditutkimuksessa tuli esille, että pandemian aikana ravintoloissa paikan päällä ruokailleiden osuus on laskenut, kun taas kokonaan syötävä ravintolaruoka on ollut nousussa. (8.)

Suurin osa ravintolapalvelujen käyttäjistä on yksin tai kaksin eläviä kaupunkilaisia. (9.) Kun katsotaan tilastokeskuksen tutkimuksien tuloksia yksin asuvien määrän kasvusta (kuva 2), saa mahdollista viitettä siihen, miksi Wolt ja Foodora menestyvät Gastronauttia paremmin.



Kuva 2. Asuntokunnat henkilöluvun mukaan (10).

## 2.2 Kilpailijoiden analysointi

Suurimpia kilpailijoita välityspalveluiden alalla Suomessa ovat Wolt ja Foodora, ja niiden markkinajohtajan asema on aiheuttanut monille ravintolayrittäjille ongelmia. Jotkut ravintolaketjut ovatkin taistelleet markkinajohtajia vastaan omilla verkkotilausjärjestelmillään. Esimerkiksi Kotipizza ei käytä Foodoraa tai Woltia

ollenkaan. Se tarjoaa asiakkaidensa käyttöön omaa PWA-sovellusta, joka on asennettavissa Kotipizzan verkkosivuilta. haamu.) Joillakin suurilla ketjuyrityksillä on omia natiiveja mobiilisovelluksia, esimerkiksi Hesburgerilla, McDonald'silla ja Subwaylla. Kaikki edellä mainitut ravintolaketjut tarjoavat jonkinlaisen bonuspisteiden keräysmahdollisuuden sovellusta käyttämällä. Yllättävästi vain Hesburgerin sovelluksesta pystyy tekemään noutotilauksen ravintolaan. Play-kaupassa Hesburgerin Android-sovelluksella latauskertoja on yli miljoona, mikä on enemmän kuin Foodoran 500 000 latauskertaa, mutta huomattavasti vähemmän kuin Woltin 5 miljoonaa latausta. (12.)

Muualla maailmassa on esimerkiksi käytössä hyvin suosittu UberEats, joka toimii hyvin samantapaisesti kuin Wolt. UberEatsin Play-kaupan latauskerrat ovat yli 100 miljoonaa. Luvuista voi nähdä, kuinka suosituksi trendiksi ruoan tilaaminen välityspalvelujen kautta on tullut ympäri maailmaa. (12.)

## Edut

Wolt ja/tai Foodora ovat nykyään pakolliset monelle ravintolayrittäjälle. Suosionsa ansiosta palvelut tuovat monelle yritykselle näkyvyyttä täysin eri tavalla kuin yrityksillä olisi ilman niitä. Välityspalveluiden markkinointibudjetti on moninkertainen pienravintolayrittäjiin verrattuna. Ilman alustojen tarjoamaa näkyvyyttä jää yritys helposti muiden jalkoihin. Sovelluksien yhteistyökumppaniksi voi liittyä melkein mikä tahansa ravintolayritys. Wolt tarjoaa myös muita palveluita, kuten tavarakaupoista tilaamista, mikä laajentaa asiakaskuntaa pelkistä ruuantilaajista myös muihin kategorioihin. Esimerkkejä Woltista löytyvistä kaupoista ovat Alepa, Marimekko, Clas Ohlson ja Musti ja Mirri. (13.)

Laajan tarjonnan ansiosta Wolt on ladattuna erittäin monen suomalaisen puheliimeen. Woltin tuomasta näkyvyydestä on pienyrittäjien vaikea kieltäytyä. Alusta mahdollistaa myös pienyritysten kilpailun isompien ketjujen kanssa. Foodora tarjoaa samankaltaista näkyvyyttä, vaikka hieman pienemmällä mittakaavalla, ravintolakeskeisyytensä ansiosta. Foodora kertoo omilla sivuillaan missiostaan hyvin selvästi:

Palvelumme tarkoituksena on koota Suomen ravintolat yhden osoitteen alle. Palvelussa ravintolalla on mahdollisuus tarjota tuotteitaan kuluttajille oman verkkokauppansa kautta. (14.)

Suomalaiset tilaavat esimerkiksi valtaosan noutopitsoistaan Foodoran kautta. (15.)

Foodora on ottanut tilanteesta täyden taloudellisen hyödyn irti ja tarjoaa asiakasravintoloilleen Premium-lisäpalvelua, joka tuo ravintolalle yhä enemmän näkyvyyttä sovelluksessa. (15.) Ravintola nousee sovelluksen sivuilla ensimmäisenä esille käyttäjän avatessa sovelluksen (kuva 3).

All restaurants

<p><b>Pizza Hut Matinkylä</b> ★4/5 (160)            €€€. pizza, italian, american, gluten free            0,90 € delivery fee</p>	<p><b>Ravintola Kuningas (ent...</b> ★4.1/5 (1k+)            €€€. pizza, burgers, grilli, falafel, kebab, vöner            Free delivery</p>	<p><b>Gangnam Kitchen - Mankkaa</b> ★5/5 (3)            €€€. korean, burgers            7,90 € delivery fee</p>
<p><b>Arnolds Iso Omena</b> ★4/5 (11)            €€€. cakes &amp; bakery, sandwiches, desserts, sm...</p>	<p><b>Lucy's Sushi</b> ★4.7/5 (5)            €€€. sushi, seafood, japanese, asian</p>	<p><b>Bunit</b> ★3.4/5 (12)            €€€. asian, salaatti, gluten free</p>

Kuva 3. Esimerkki Foodoran tarjoamasta Premium-ravintolan näkyvyyden nostamisesta (14).

Ravintoloille, joilla ei ole omaa tilaus-/kuljetuspalveluaan, välityspalvelut avaavat aivan uudenlaisia markkinoita helposti ilman pääoman ja ajan investoimista infrastruktuurin rakentamiseen. Ravintola voi teoriassa myös palvella useampaa

asiakasta samanaikaisesti kuin ennen, sillä asiakkaiden määrä ei rajoitu ravintolan ruokasalin koon mukaan. (16.)

Ruoan välityspalveluiden toiminta ja suosio ovat myös luoneet mahdollisuuden uudenlaisien ravintoloiden synnylle niin sanotusti ”haamuravintoloille”. Haamuravintoloilla ei ole asiakaspaikkoja laisinkaan, vaan ne toimivat välityspalveluiden generoimien tilausten varassa. Toiset välityspalvelut ovat perustaneet omia haamuravintoloitaan, joita on sitten listattu palveluun. Haamuravintoloiden toiminta voi salakavalasti viedä asiakaskuntaa muilta ravintoloilta. (17.)

Kuvan 4 malli osoittaa, kuinka teoriassa välityspalvelut, kuten UberEats, ovat hyödyllisiä yrityksille, kun ne tuovat myös uusia asiakkaita mukanaan. Mallissa on laskettu 35 prosentin provisio ja 15 prosentin myynninlisäys palvelun kautta. Mallissa on myös oletettu, että olemassa oleva asiakaskunta ei siirry välityspalvelun käyttäjäksi, vaan 10 prosenttia myynnistä tapahtuisi välityspalvelun kautta (kuva 4). (16.)

Profit & Loss Restaurant Example For the year ended 30 June 2018			Assumptions	
	Base Case	Adjusted		
Revenue	\$700,000	\$805,000	Additional sales volume	15.00%
Direct Variable Costs	\$259,000	\$326,025	% mix of Sales via UberEats	10.00%
Gross Profit	\$441,000	\$478,975	Direct Variable Cost %	37.00%
Margin	63.00%	59.50%	UberEats Commission	35.00%
Less Fixed Operating Expenses	\$420,000	\$420,000	Direct Cost After UberEats	72.00%
Net Profit	\$21,000	\$58,975	Direct Costs (no UberEats)	\$288,065
Margin	3.00%	7.33%	Direct Costs (via UberEats)	\$57,960
Net Profit Improvement by Percentage		\$37,975 181%	Total Direct Costs	\$326,025

Kuva 4. Malli UberEatsin generoimasta 15 prosentin asiakaslisäyksen vaikutuksesta yrityksen tuloihin (16).

## Ongelmat

Kuluttajien kannalta Foodora ja Wolt ovat erittäin käytännöllisiä, samoin myös kaupoille ja ravintoloille, jotka eivät etätilausta ja kuljetusta itsekseen tarjoa. Häviöjäksi jäävät ravintolat, jotka tarjoavat omaa kuljetuspalveluaan ja oman tilausalustan. Oman tilauspalvelun kautta saataisiin 100 prosenttia tilauksen tuloista, kun taas nykyään suosiotaan kasvattavat ruoan välityspalvelut ottavat noin yhden kolmasosan tilauksen summasta omaksi palkkiokseen. (18.)

Yritysten olisi tärkeää panostaa omiin verkkotilauspalveluihin ja markkinointiin eikä vain luottaa välityspalvelujen tuomaan liikenteeseen. Yksittäisten ravintoloiden on hyvin vaikea tuoda välityspalveluista asiakkaita omaan piiriinsä, jolloin ravintolalle ei synny omaa kanta-asiakaskuntaa. Oma kanta-asiakaskunta ja tilauskanava antavat turvaa ja neuvotteluvoimaa, kun ravintola päättää liittyä välittäjäpalvelun asiakasyritykseksi. (19.) Laadukkaaseen omaan verkkotilauskanavaan investoiminen voi olla monelle pienyritykselle liian kallista, jolloin yritykset ovat puristuksissa välistyspalvelujen ehtojen tai mahdollisen konkurssin välissä. Useat yrittäjät kokevatkin olevansa lähes pakotetussa suhteessa välityspalveluihin.

Esimerkiksi entinen Pizza-Online, nykyinen Foodora on saanut joiltakin ravintolayrityksiltä syytteitä riistosta. Foodora perii jokaisesta sen palvelun kautta tilatun annoksen hinnasta 16–24 prosentin osuuden. Tämän lisäksi sovitaan myös perusvälityspalkkio, joka on 12–18 prosenttia tilauksen hinnasta. Foodoran markkinajohtajan aseman vuoksi pienravintolan vaihtoehdot ovat vähäiset. Lisäksi Foodoran kanssa tehdyssä sopimuksessa on hinnoitteluehdot, jotka määräävät, että ravintolan tulee pitää tuotteita samalla hinnalla alustalla kuin ravintolassa tai verkkosivuilla. Näin välittäjäpalvelu saadaan kuluttajan kannalta ongelmattomaksi, sillä lisämaksu jää yrittäjien maksettavaksi. (15.) Välittäjäpalvelut yrittävät tällä taktiikalla myös minimoida mahdollisen asiakaskadon omalta alustalta suoraan yrityksestä tilaamiseen siksi, että se olisi halvempaa.

Suurien komissioiden lisäksi yrityksillä on entistä enemmän kilpailua. Ennen ravintola saattoi kilpailla samassa ostoskeskuksessa olevien ravintoloiden kanssa, kun taas nykyään kilpailu kohdistuu välityspalveluiden muuta valikoi-  
maa vastaan. Paras tapa päihittää kilpailijat Foodorassa onkin maksaa premiumjäsenyydestä, jonka hinta on noin 400–600 euroa kuukaudessa. (15.)

Välityspalvelut nostavat yrityksen tuloja vain, jos palvelun kautta saadaan myös uusia asiakkaita. Vaikutus on päinvastainen, jos uusia asiakkaita ei saada ja olemassa olevaa asiakaskuntaa siirtyy käyttämään välityspalvelua (kuva 5). Tällöin yritys tekisi kyseisen asiakkaan kohdalla noin 30 prosenttia tappiota riippuen sopimusehdoista. (16.)

Profit & Loss Restaurant Example For the year ended 30 June 2018			Assumptions	
	Base Case	Adjusted		
Revenue	\$700,000	\$700,000	Additional sales volume	0.00%
Direct Variable Costs	\$259,000	\$283,500	% mix of Sales via UberEats	10.00%
Gross Profit	\$441,000	\$416,500	Direct Variable Cost %	37.00%
Margin	63.00%	59.50%	UberEats Commission	35.00%
Less Fixed Operating Expenses	\$420,000	\$420,000	Direct Cost After UberEats	72.00%
Net Profit	\$21,000	-\$3,500	Direct Costs (no UberEats)	\$233,100
Margin	3.00%	-0.50%	Direct Costs (via UberEats)	\$50,400
Net Profit Improvement by Percentage		-\$24,500 -117%	Total Direct Costs	\$283,500

Kuva 5. Malli UberEatsin generoimasta 0 prosentin asiakaslisäyksen vaikutuksesta yrityksen tuloihin (16).

Monet yrittäjät ovat yrittäneet irtaantua Foodorasta mahdollisimman paljon ohjaamalla kanta- ja uusia asiakkaitaan omille verkkotilaussivuilleen. Tästä huolimatta ravintolat ovat huomanneet, että asiakkaat pitävät enemmän Foodoran mobiilisovelluksesta, joka on tuttu ja helpokäyttöinen.

Ravintolayrityksiä voi myös mietittyä imagon ja laadun ylläpidon tärkeys, ja sen takia osalle yrittäjistä annoksen kuljetettavaksi antaminen tuntemattomalle



Wolt- tai Foodora-kuriirille voi kuulostaa turhalta riskiltä. Viekö kuriiri ruoan tilaajalle lämpimänä ja toimituskelpoisena? Vaikka vastuu jääkin kuriirille eikä ravintolalle, voi asiakas yhdistää kylmänä tuodun ruoan ravintolaan eikä kuriiriin. (20.) Imagon ja laadun tärkeys on myös näkynyt MaRa ry:n trendikyselyssä nousseen yhdeksi tärkeimmistä valintaperusteista ruokaa tilattaessa, joten yrittäjien huoli ei ole turhaa (19).

Lisäksi epävarmuutta ruokavälityspalveluihin luottamisessa lisää se, että alalla kilpailu tällä hetkellä on todella kovaa. On hyvinkin mahdollista, että esimerkiksi joku alan suurempi toimija ostaisi Woltin markkinoilta pois. Alan kilpailu on niin kovaa, että suurin osa välityspalveluista tekee suurta tappiota ja niiden ainoaksi vaihtoehdoksi lähitulevaisuudessa jää joko kasvaa nopeasti tai yhdistyä suuremman palvelun kanssa. Esimerkiksi Wolt teki vuonna 2020 tappiota 38 miljoonaa euroa. Tästä huolimatta pääomasijoittajat uskovat liikemallin olevan tuotava tulevaisuudessa, sillä Wolt on kerännyt rahoitusta noin 700 miljoonaa euroa. Kuluttajien kilpailutus kuitenkin johtaa siihen, että yritykset joutuvat tekemään yhä enemmän tappiota. (21.)

### 2.3 Verkkotilaussovelluksien kehitystyö

Ennen kuin yritys alkaa oman sovelluksen kehitystyöhön, on mietittävä sovelluksen toteutustapa. Toteutustapaa miettiessä kannattaa ottaa huomioon kehitystyön hinta, sovelluksen lopullinen tarkoitus ja eri toteutustapojen edut.

#### Kehitystyön toteutuksen vaihtoehdot

Mobiililaitteille kehitystyötä tehtäessä vaihtoehtoina on kehittää natiivi mobiilisovellus tai PWA-sovellus, joka on optimisoitu mobiililaitteille. Natiivi mobiilisovellus kehitetään käyttöjärjestelmäkohtaisesti, mutta joillakin viitekehityksillä on mahdollista kehittää natiivi mobiilisovellus monelle käyttöjärjestelmälle samanaikaisesti.

Natiivit sovellukset voidaan kehittää suoraan valitulle käyttöjärjestelmälle, esimerkiksi yleensä Androidille tai iOS:lle. Suurin ongelma puhtaasti natiivissa kehitystyössä on se, että jos sovellus halutaan molemmille alustoille, se on kehitettävä kahteen kertaan. Etuna puhtaasti natiiveissa sovelluksissa on, että ne ovat optimoituja käyttöjärjestelmän mukaan ja voivat käyttää käyttöjärjestelmän ja puhelimen viimeisempiä ominaisuuksia. (22.) Jos natiiviin sovelluskehitykseen päädytään, kannattaa myös ottaa huomioon eri käyttöjärjestelmien suosio. Suomessa käytettävistä mobiililaitteista noin 66 prosenttia käyttää Android-käyttöjärjestelmää ja noin 33 prosenttia laitteista käyttää iOS-käyttöjärjestelmää. (23.) Silloin kun kehitystyönä tehdään sovellusta kaupalliseen tarkoitukseen, tulisi keskittyä suurimpiin käyttöjärjestelmiin.

PWA-sovellukset ovat melko uusi ilmiö, mutta niiden suosio kasvaa niiden huomattavien etujen takia. PWA-sovellukset eivät varsinaisesti ole sovelluksia vaan verkkosivuja, jotka on optimoitu mobiililaitteille. Niitä ei asenneta laitteeseen, vaan laitteen kotinäytölle lisätään vain pikakuvake verkkosivulle navigointiin. PWA-sovellusten kehitys on nopeaa, halpaa ja helppoa. Niiden lataamiseen ei tarvita sovelluskauppoja, ja käyttäjät voivat löytää sovelluksen esimerkiksi Google-haulla. Tämä voi toisaalta olla ongelmallista, koska monelle ihmiselle sovelluskaupoista lataaminen voi tuntua tutulta ja turvalliselta vaihtoehdolta. Ongelmana PWA-sovelluksissa on myös se, että jotkin selaimet ja käyttöjärjestelmät eivät tue niitä, jolloin ne eivät aina toimi tarkoitettusti. (24.) PWA-sovellukset eivät myöskään ole paras ratkaisu paljon resursseja käyttäville sovelluksille tai sovelluksille, joiden toiminta vaatii mobiililaitteen ominaisuuksia, esimerkiksi Bluetoothia (25).

Natiiveja mobiilisovelluksia voidaan kehittää myös erilaisten ohjelmistokehysten avulla. Esimerkkejä tämmöisistä viitekehyksistä ovat Facebookin vuonna 2016 julkaisema React Native ja Googlen Flutter, joka julkaistiin vuonna 2017. Esimerkiksi React Nativen avulla sovelluskehitys voidaan toteuttaa Javascript-komponenteilla, jotka rakentuvat natiiveiksi komponenteiksi käyttöjärjestelmän mukaan. Tämä mahdollistaa sen, että yhdellä koodikannalla voidaan kehittää sovellus sekä Android- että iOS-käyttöjärjestelmille. React Nativella ja Flutterilla

luodut sovellukset voivat hyödyntää laitteen ominaisuuksia, kuten sijaintipalveluja tai push-ilmoituksia. (26.)

#### Natiivien mobiilisovelluksien edut

Yksi suurimpia etuja siinä, että yrityksellä on oma mobiilisovellus, on sen vaikutus yrityksen brändiin asiakkaiden silmissä. Omaan sovellukseen investoiminen osoittaa, että yritys välittää asiakkaistaan ja yrittää jatkuvasti kehittää asiakaskokemuksia parempaan suuntaan. (27.) Pelkästään jo sovelluskaupoissa oleminen voi luoda vahvempaa mielikuvaa yrityksen brändille.

Vuonna 2019 Yhdysvalloissa tehdyn tutkimuksen mukaan 90 prosenttia mobiililaitteilla ja 77 prosenttia tableteilla internetissä käytetystä ajasta tapahtuu sovellusten kautta (28). Varsinkin nuoret kohderyhmänä ovat nykyään teknisesti tietävämpiä ja taitavampia. Moni nuori tietää, että verkkosivujen käyttö mobiililaitteilla voi olla turhauttavaa, jos sivut ovat huonosti optimoituja. Tämä voi johtaa siihen, että palvelua ei haluta käyttää. Mobiilisovellukset taas ovat trendikkäitä, helppokäyttöisiä ja nopeita. (29.)

Mobiililaitteiden käyttö on lisääntynyt tasaisesti kaikenikäisillä, ja tutkimusten mukaan suomalaiset käyttävät mobiililaitteita lähes kolme ja puoli tuntia päivässä (30). Ei siis kannata ohittaa mahdollisuutta sijoittaa omaa palveluaan suoraan kuluttajan taskuun. Natiivien mobiilisovelluksien etuna on, että kun sellaisen asentaa laitteelle, se tallentuu laitteelle logoineen. Tätä voi ajatella tavallaan ilmaisena mainostilana. Sovelluksen lataus mobiililaitteelle tarkoittaa, että yritys on kuluttajalle esillä kaiken aikaa ja missä vain. (31.) Mobiilisovelluksen luominen on hyvä tapa lisätä yrityksen tuottoja kasvattamalla kosketusta asiakkaisiin.

Investoiminen omaan mobiilisovellukseen aikaisin auttaa erottumaan joukosta, varsinkin kun on kyse pienyrityksistä. Alueilla, joissa pienyrittäjät kilpailevat toisiaan vastaan, mobiilisovelluksen vaikutusta ei voi liioitella. (32.)

## Kuinka erottua välityspalveluista

Ravintolan toteuttaessa oman verkkotilaussovelluksen on tärkeää miettiä, miten erotetaan välityspalvelun sovelluksesta. Toisin sanoen, miksi kuluttaja haluaisi käyttää ravintolan omaa sovellusta tilaamiseen välityspalvelun sijasta?

Helpoin tapa poistaa tilaukset välittäjäpalveluista on irtautua niistä kokonaan oman sovelluksen tullessa sovelluskauppoihin. Näin ravintolasta tilaaminen saadaan yksinomaiseksi vain omasta sovelluksesta. Tämä tuskin kuitenkaan on paras tapa toteuttaa asiakkaiden houkuttelu omaan sovellukseen. Monet välityspalveluja käyttävät saattavat vain siirtyä tilaamaan seuraavasta ravintolasta välityspalvelun listalla.

Hesburger julkaisi kesäkuussa 2016 oman mobiilisovelluksensa, ja se nousi käyttäjien suosioon hetkessä. Tietysti osansa tässä on Hesburger-ketjun suuruudella, mutta Hesburger on myös esimerkiksi välittäjäpalvelu Woltissa. Miksi siis yli miljoona ihmistä latasi Hesburgerin oman sovelluksen? Hesburger tarjoaa sovelluksessaan yksinomaisia asiakasetuja, joita ei Woltin kautta voi keräyttää. Sovellukseen kehitettiin moderni bonusklubi ja mahdollisuus maksaa ostokset. Sovelluksen julkaisun jälkeen asiakkaiden keskiostosten määrä kasvoi 25 prosenttia. (1.)

Eksklusiivisten asiakasetujen lisäksi kannattaa asiakkaille tiedottaa sovelluksesta ja sen eduista. Monet yrittäjät lähettävät välityspalveluiden lähettien kuljetaman ruoka-annoksen mukana tiedotteen siitä, että ravintolalta suoraan tilaaminen on toivottua. (33.)

### 3 Ravintolakohtaisen sovelluksen suunnittelu

Insinööriyönä kehitettiin pienin julkaisukelpoinen tuote, eli MVP, natiivi mobiili-sovellus paikalliselle pitsaravintolalle, joka on ollut samassa sijainnissa vuodesta 1998. Ravintola kuului aikoinaan yhteen Suomen isoimmista pitseriaketteista, mutta ei ole osana sitä enää. Ravintolalla on ennestään kanta-asiakaskuntaa, mikä tekee ravintolasta hyvän kohteen ravintolakohtaisen tilaussovelluksen käyttökokeiluun.

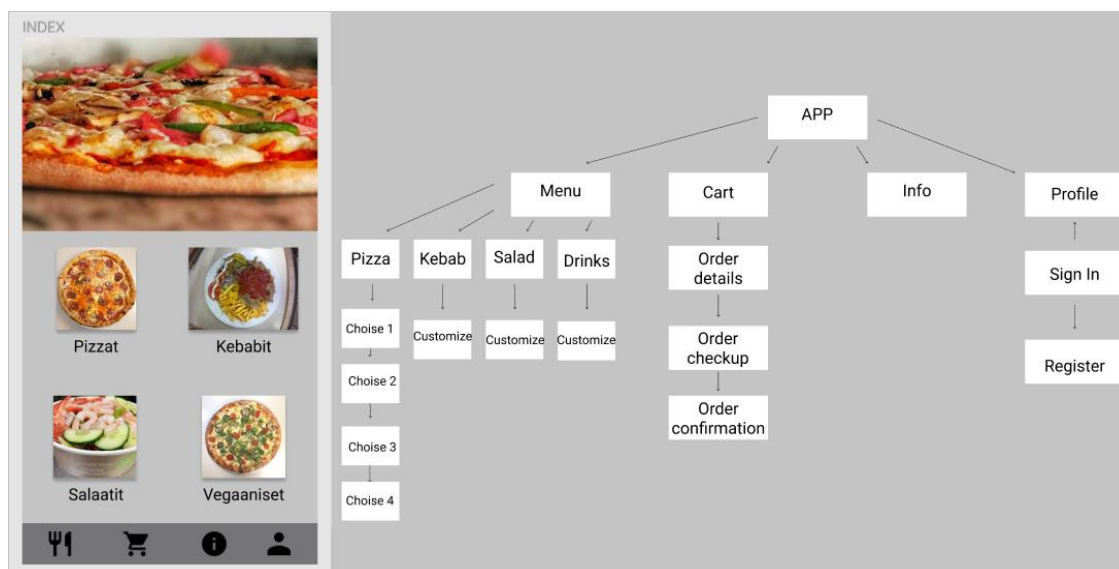
Sovellus oli tarkoitus kehittää siten, että sen pohjalta olisi mahdollista luoda myös muille pitseriaravintoloille oma tilaussovellus. Tässä tuli ottaa huomioon se, että jokaisella ravintolalla on erilainen tarjonta annoksia ja juomia. Hyvä on huomioida myös esimerkiksi se, että jokaisella ravintolalla on omanlaiset toimintusehdot ja aukioloajat. Sovelluksen ulkoasun tuli myös olla neutraali esimerkiksi pitsaravintolan logoja lukuun ottamatta, jotta se olisi helppo tulevaisuudessa muokata ravintolan kriteerien mukaan.

Sovelluksen päätoimintona olisi ruokatilausten tekeminen ja lähettäminen. Sovelluksessa oli siis oltava jonkinlainen ruokalista ruoka-annoksille ja mahdollisuus muuttaa esimerkiksi niiden lisukkeita tai kastikkeita. Toinen erittäin tärkeä elementti kaikissa verkkokaupoissa on ostoskori. Ostoskorissa on hyvä nähdä tietoja sinne lisätyistä tuotteista. Ostoskorin tulisi esimerkiksi näyttää, miten annosta on muutettu ja mikä yksittäisten tuotteiden hinta on. Ostoskorista tulisi pystyä poistamaan kaikki tuotteet kerrallaan tai yksi kerrallaan. Lisäksi koko ostoskorin yhteissumma olisi hyvä laskea ja näyttää.

#### 3.1 Käyttöliittymä ja navigointi

Suunnittelu alkoi sovelluksen rakenteen, navigoinnin ja ulkonäön ratkaisuvoista. Luonnokset päätin tehdä Figma-sovelluksella. Ensimmäinen piirros navigoinnin toteutuksesta rakentui hampurilaisvalikon ympärille. Lopuksi päädyttiin kuitenkin käyttämään rakennetta, jossa on neljä päävälilehteä sovelluksen alapalkissa. Välilehtirakenne tuntui käyttäjätavallisemmältä vaihtoehdolta, sillä

sovelluksen sivut tuntuivat olevan paremmin esillä. Sovelluksen ensimmäinen välilehti on ravintolan ruokavalikoima, josta käyttäjä pääsee ruokalistalle. Ruokalistan päänäkymä jaoteltiin kategorioiden mukaan: esimerkiksi yhdestä kuvakkeesta painaessa pääsee näkemään pizzat, kun taas toisesta kebab-annokset (kuva 6).



Kuva 6. Luonnos sovelluksen navigoinnista ja pääsivusta, jolla selataan tilattavia ruoka-annoksia.

Sovelluksen seuraava välilehti on ostoskori. Ostoskorin luonnostelu jäi hyvin vähäiseksi. Ostoskori on rakenteeltaan lista, josta sinne lisätyt tuotteet löytyvät. Sovellus haluttiin pitää mahdollisimman helppokäyttöisenä, joten mallia otettiin jo olemassa olevista verkkokaupoista. Ostoskori on myös reitti tilauksen viimeistelyyn, sillä se on usein tie "kassalle" muissakin verkkokaupoissa. Tilausprosessista haluttiin selkeä, ja siinä halutaan näyttää käyttäjälle oleelliset tiedot ennen tilauksen lähettämistä. Ostoskorin kuvakkeen päällä on merkki, joka osoittaa ostoskorin sisällä olevien tuotteiden määrän. Käyttäjä voi kuvakkeen avulla tarkistaa helposti ostoskorissa olevien tuotteiden lukumäärän.

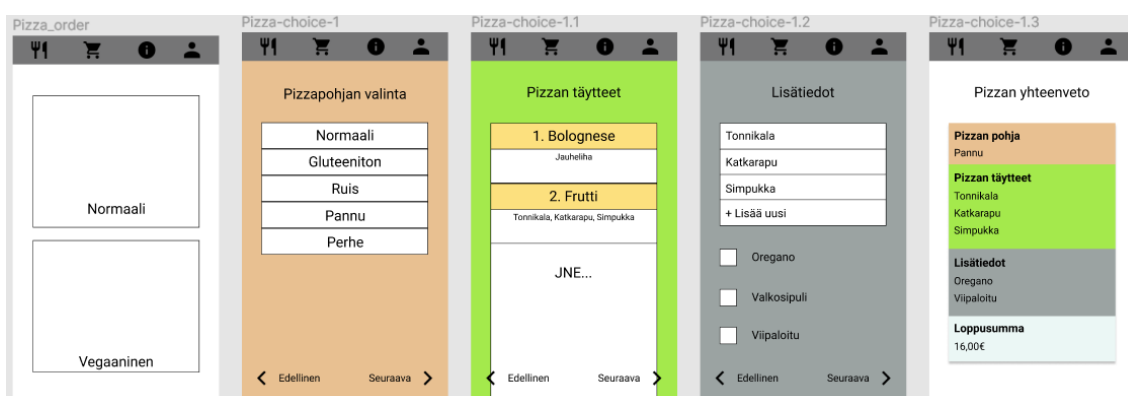
Sovelluksen kolmas välilehti suunnitelmassa navigoisi tietosivulle, josta löytyy ravintolan tietoja. Sivulta löytyvät esimerkiksi yhteystiedot, kuljetusehdot ja aukioloajat.

Viimeinen välilehti oikealla navigoi käyttäjän omaan profiiliin tai sisäänkirjautumiseen ja rekisteröintiin, jos käyttäjä ei ole kirjautunut aikaisemmin. Rekisteröintisivun tulisi vaatia samoja tietoja kuin verkkosivu, jotta ne olisivat yhteensopivat olemassa olevan SQL-tietokannan kanssa.

Sovelluksen navigoinnin suunnittelun jälkeen alkoi pääsivun eli valikkonäkymän suunnittelu. Ravintola toivoi tästä osa-alueesta hyvin selkeää ja nopeakäyttöistä. Sovelluksen kategorioita varten ruokalista jaoteltiin kymmeneen eri osaan, pitsat, kebabit ja kanakebabit, salaatit, rullat, vönerit, zonet, falafelit, kana ja kala, sides and dips ja juomat.

Ravintolan tarjoamista tuotteista moni oli rakenteeltaan hyvin samankaltainen. Monissa annoksissa on yksi pääaines, jonka sai valittua monen eri lisukkeen kanssa. Esimerkiksi voidaan ottaa kebab- ja kanakebabannokset. Vaihtoehtona on jompikumpi pääaines, kebab tai kanakebab, ja yksi kahdeksasta eri lisukevaihtoehdosta. Valintalomakkeen rakenteessa tuli olla kahden pääaineen ja usean lisukkeen valinta.

Ainoa suuresti muista eroava tuote on pitsat, sillä pitsalista sisältää yli kolmekymmentä vaihtoehtoa. Lisäksi pitsaan pitää pystyä valitsemaan yksi viidestä eri pohjavaihtoehdosta ja muokkaamaan täytteitä tai tekemään kokonaan oma-valintaisen pitsan. Ravintola ehdotti, että pitsan valinta voitaisiin jakaa eri vaiheisiin, samaan tapaan kuin Subway-ravintoloissa ruokaa tilattaessa (kuva 7).



Kuva 7. Luonnos viidestä sivusta, joiden avulla pizza rakennetaan sovelluksessa.

## 3.2 Sovelluksen vaatima toiminnallisuus

Sovelluksen toiminnallisuutta suunniteltaessa aloitettiin ostoskorin ratkaisutavoista. Kun ostoskorin suunniteltiin sovellukselle, ensimmäisenä ideana oli käyttää mobiililaitteen työmuistia, johon ostoskoriin lisätyt annokset tallennettaisiin listaan. Valikkosivulla olevien kuvakkeiden kautta aukeavien sivujen päätoiminto on lisätä ostoskoriin käyttäjän valitsema tuote. Annos voitaisiin luoda objektiksi, jolla olisi attribuutteina ainakin yksilöivä tunnus ja tyyppi. Näiden attribuuttien lisäksi objektilla olisi kaikki tieto siitä, mitä käyttäjä muuttanut annoksessa.

Mobiilisovelluksella piti myös pystyä lähettämään tilaus johonkin, mistä ravintola voi ottaa sen vastaan. Ensimmäinen vaihtoehto ollut rakentaa back end, joka vastaanottaisi käyttäjien luomat tilaukset ja näyttäisi ne esimerkiksi verkkosivulla. Verkkosivulta ravintolan työntekijät voisivat tulostaa tilauksen paperimuotoon.

Sovelluksessa olisi hyvä olla mahdollisuus kirjautua sisään omalle käyttäjätilille tai luoda sellainen. Käyttäjätunnusten olemassaolo lisäisi käyttäjäystävällisyyttä, sillä käyttäjän tietoja voitaisiin käyttää tilaustietojen automaattiseen täyttöön. Tämä helpottaisi ja nopeuttaisi tilausten tekemistä sovelluksella huomattavasti. Käyttäjän kirjautuessa sovellus voisi tehdä kutsun tietokannalle tunnistetta varten, joka tallennettaisiin sovelluksen työmuistiin. Kirjautuminen tapahtuisi automaattisesti seuraavilla kerroilla tunnisteen avulla. Tietokannasta haettaisiin tilaukseen tarvittavat tiedot, jos sovellukseen on kirjaututtu sisään. Back endin tulisi myös pystyä käsittelemään sähköposteja, jotta käyttäjälle voitaisiin lähettää tilausvarmistus tilauksen onnistuessa.

Back endin ja sovelluksen pitäisi myös tarkistaa, mitkä tilaukset saavat mennä läpi ja mitkä eivät. Esimerkiksi jos kuljetustilaus yritetään tilata ravintolan kuljetusalueen ulkopuolelta, pitäisi itse sovelluksen tai viimeistään tilausvahvistuksen estää se. Tämän voisi toteuttaa käyttämällä jotakin API:a, joka tarkistaa kuluuko käyttäjän antama osoite kuljetusalueeseen. Samaa API:a voisi myös hyödyntää osoitteen todentamiseen, sillä pilan päiten tehtyjä tilauksia tapahtuu



ajoittain. Väärään kellonaikaan aikaan tehty tilaus ei saisi mennä läpi järjestelmästä. Sovelluksen tulisi pystyä lähettämään tilaus vain ravintolan määrittelemien aikojen puitteissa.

### 3.3 Työkalujen valinta

Sovelluksen kehitystyö päätettiin tehdä React Nativella, sillä sovellus olisi tarkoitus saada sekä Android- että iOS-alustoille. Kehitykseen valittiin React Native, sillä työn tekijällä oli ennestään kokemusta React-kirjaston käyttämisestä.

React Nativella voidaan luoda yhden koodikannan kautta sovellus molemmille alustoille. Ensimmäinen versio tulisi vain Androidille, mutta kehitystyötä olisi tarkoitus jatkaa, jotta sovellus saadaan myös iOS-käyttöjärjestelmälle.

Paketinhallintatyökaluna käytettiin npm:ää, ja sovelluksessa käytettiin useita erilaisia npm-paketteja.

## 4 Ravintolan tilausjärjestelmä

### 4.1 React ja React Native

#### React

React on Facebookin vuonna 2013 julkaisema Javascript-kirjasto, jota käytetään interaktiivisten käyttöliittymien ja komponenttien kehittämiseen. Reactin päätoiminnallisuutena on tilanhallinta ja DOM-objektin muuttaminen. React-komponenttien tilaa voidaan hyödyntää tietojen tallentamiseen ja päivittämiseen komponentissa. Kätevä tapa komponentin tilan muuttamiseen on käyttää Reactin hook-funktioita. Hook-funktiot ovat kehittäjälle tapa päästä käsiksi komponentin tilaan ja muutella sitä. Projektissa on käytetty "useState"- ja "useEffect"-funktioita tilan manipulointiin. Komponentin tilan oletusarvo voidaan määrittää komponentin alussa "useState"-funktiolla. Lisäksi määritetään kahva, jolla tilaa päästään muuttamaan tarvittaessa "setState"-funktiolla (kuva 8). (34.)

```
import React, { useState } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Kuva 8. React-komponentti ja sen useState-tila "count".

Toista projektissa käytettyä hook-funktiota, eli "useEffect"-koukkuja käytetään operaatioihin, joita ei voida tehdä samalla, kun DOM-objekti renderöidään.

"useEffect"-funktion avulla voidaan esimerkiksi hakea http-kutsulla dataa, joka halutaan näyttää komponentissa. (35.)

Komponenttien avulla koodi voidaan jakaa itsenäisiin osiin, joita voidaan käyttää tarvittaessa uudelleen. Komponenteille voidaan antaa dataa "props"-argumentin avulla, jota komponentti voi hyödyntää renderöidessä. "Props"-argumentit mahdollistavat komponenttien välisen vuorovaikutuksen. Reactin tukena kehityksessä käytetään usein lisäksi muita kirjastoja. (36.)

## React Native

React Native on Facebookin vuonna 2015 julkaisema ohjelmistokehys, joka käyttää Reactin komponentteja Android-, iOS- ja muiden sovellusten rakentamiseen (38). React Native käyttää JSX:ää eli JavaScript XML:ää, joka on eräänlainen merkintäkieli, joka vastaa HTML:ää ja CSS:ää. JSX rakentuu natiivien alustojen vastaavaksi merkintäkieleksi, mikä mahdollistaa JSX:n käytön iOS- ja Android-käyttöjärjestelmillä. Esimerkiksi React Nativen tekstikomponentti rakentuu iOS:lla UILabeliksi ja Androidilla TextView'ksi. (37.)

## 4.2 Redux-kirjasto

Redux on Javascript-kirjasto, jota käytetään käyttöliittymien tilanhallinnan yksinkertaistamisessa. Reduxin tilanhallinta eroaa Reactin tilanhallinnasta, sillä Reduxin tilaan päästään käsiksi mistä tahansa komponentista. Redux on käytännöllinen, kun kehitettävä sovellus vaatii sovelluksen yhteisen tilan, johon mikä vain komponentti pääsee käsiksi. Reduxin avulla voidaan esimerkiksi lisätä sovelluksen yhteiseen ostoskoriin tuotteita eri tuotteiden komponenteista. (38.)

## Redux Persist

Redux Persist -kirjaston avulla Reduxin hallitsema tila voidaan tallentaa verkkosovelluksissa selaimen tai mobiililaitteen työmuistiin. Ilman Redux Persistiä Reduxin tila ei pysy tallennettuna, kun sovellus käynnistetään uudelleen. Redux

Persistin avulla sovelluksen tilaan tallennetut tiedot eivät pyyhkiydy, ellei sitä manuaalisesti pyyhitä. Redux Persist on kätevä työkalu esimerkiksi käyttäjätietojen tallentamiseen. (39.)

### 4.3 Front endin kehitys

Työn kehitys alkoi itse mobiilisovelluksen kehittämisestä, minkä jälkeen kehitettiin back end tilausten käsittelyä varten.

#### Pääsivu ja ruokavalikko

Front endin kehitys aloitettiin pääsivun eli ruokalistanäkymän luomisesta. Ruokalistasivu tulisi olemaan suurin ja tärkein osa sovellusta. Valikkonäkymään luotiin kymmenen kuvaketta, joista navigoitaisiin valitun tuotealueen räätälöintiin. Kuvakkeilla on "tyyppi"-attribuutti, joka hyödyntää React Propsia datan eteenpäin lähettämiseksi. Napin painatuksella lähetetään dataa, jonka avulla seuraava sivu tietää, mikä komponentti renderöidään.

Ravintolan kaikille ruokakategorioille tehtiin esimerkkikoodin 1 tapainen kuvake, jolla on oma tyyppi-attribuutti. Kaikki kuvakkeet vievät samalle "Serving"-sivulle, joka käyttää switch...case-rakennetta painetun kuvakkeen tyyppin tarkastukseen.

```
<TouchableOpacity
  style={styles.innerContainer}
  onPress={() => navigation.push('Serving', {type: "Kebab"})}>
  <Image
    style={styles.image}
    source={require('../assets/category/kebab.png')} />
  <Text style={styles.title} >Kebab ja kanakebab</Text>
</TouchableOpacity>
```

Esimerkkikoodi 1. Painettava kuvake, joka ohjaa "Serving"-sivulle tyyppi-attribuutin kanssa.

Jokainen "Serving"-sivun komponenteista noutaa esimerkkikoodin 2 näköisen listan ruokavaihtoehdoista ennen komponentin renderöintiä. Esimerkkikoodin 2 tapauksessa komponentilla on kaksi päävaihtoehtoa, kebab tai kanakebab,

joista käyttäjä valitsee yhden. Tämän jälkeen käyttäjä valitsee lisukkeen annokselleen.

```
export const kebabAnnokset = [
  {
    key: '0',
    id: 'K',
    name: 'Kebab',
    options: kebabLisukkeet,
    description: 'Kaikkiin kebab-annokseen sisältyy kebabliha,
chili, salaatti, kebabkastiketta sekä lisukevaihtoehto.'
  },
  {
    key: '1',
    id: 'KK',
    name: 'Kanakebab',
    options: kebabLisukkeet,
    description: 'Kaikkiin kanakebab-annokseen sisältyy kanakebab-
liha, chili, salaatti, currykastiketta sekä lisukevaihtoehto.'
  }
]
```

Esimerkkikoodi 2. “Kebabannokset”-kategorian vaihtoehtoiset oliot.

Käyttäjä valitsee vaihtoehdoista yhden ja siirtää ostoskoriin objektin, jolla options-attribuutti on yksi listan vaihtoehdoista. Kaikilla ruokaolioilla on myös key-, id-, name- ja description-attribuutit, joita ei voi muokata.

Kategorioiden komponentit käyttävät Reactin useState-funktiota näkymän päivittämiseen käyttäjän muokkailujen mukaan. Ostoskoriin lisättäessä luodaan oliolle uusi nimi, id ja hinta valittujen vaihtoehtojen mukaan.

## Profiilisivu

Profiilisivujen oli sovellusta suunniteltaessa tarkoitus antaa käyttäjän kirjautua sisään tai rekisteröidä uusi tili. Ajan puitteissa kuitenkin päädyttiin luomaan vaihtoehto, joka ei vaadi tietokantaa tai sisään kirjautumista. Profiilisivu koostuu viidestä tekstikentästä, etunimi, sukunimi, katuosoite, puhelinnumero ja sähköpostiosoite. Käyttäjä täyttää tekstikentät ja tallentaa tiedot napista. Käyttäjätiedot tallennetaan Redux Persist -kirjaston avulla, jolloin tiedot säilyvät puhelimen

muistitilassa, vaikka sovellus käynnistettäisiin uudelleen. Redux Persistillä työmuistiin tallennettavan tilan luonti onnistuu esimerkkikoodin 3 tavalla. Käyttäjä voi myös muokata tietoja profiilisivulta koska vain.

```
const persistConfig = {
  key: "root",
  storage: AsyncStorage
}

const persistedReducer = persistReducer(persistConfig, userReducer);
```

**Esimerkkikoodi 3.** Redux Persistin avulla luodaan pysyvä muistitila käyttäjätiedoille.

Ratkaisu osoittautui hyväksi, sillä kun käyttäjä on kerran tallentanut tietonsa, voidaan tietoja käyttää tilausta täytettäessä tekstikenttien automaattiseen täyttöön. Tämä teki tilausprosessista paljon nopeamman ja käyttäjäystävällisemmän.

### Ostoskori ja tilauksien lähetys

Sovellus hyödyntää Redux-kirjastoa myös, kun käyttäjä lisää ostoskoriin tuotteita. Ostoskärryn tietoja ei kuitenkaan tallenneta puhelimen välimuistiin samalla tavalla kuin käyttäjätietoja, mikä tarkoittaa, että kun sovellus suljetaan, ostoskorin tiedot pyyhkiytyvät.

Ostoskorin tilanhallintaan tarvittiin toiminnot, joilla objektien lisääminen ja poistaminen olisi mahdollista. Tämä onnistui Reduxin avulla esimerkkikoodin 4 tapaan.

```

const cartItems = (state = initialState, action) => {
  switch(action.type) {
    case 'ADD_TO_CART':
      return {...state, productsInCart: [action.payload,
...state.productsInCart]}
    case 'REMOVE_FROM_CART':
      const index = state.productsInCart.indexOf(action.pay-
load);
      if (index > -1) {
        state.productsInCart.splice(index, 1);
      };
      return {...state, productsInCart: [...state.prod-
uctsInCart]};
    case 'CLEAR_CART':
      return {...state, productsInCart: []};
    default:
      return state;
  }
}

```

Esimerkkikoodi 4. Ostoskorin tilaa hallinnoiva metodi, joka päättää toiminnon tyyppin perusteella.

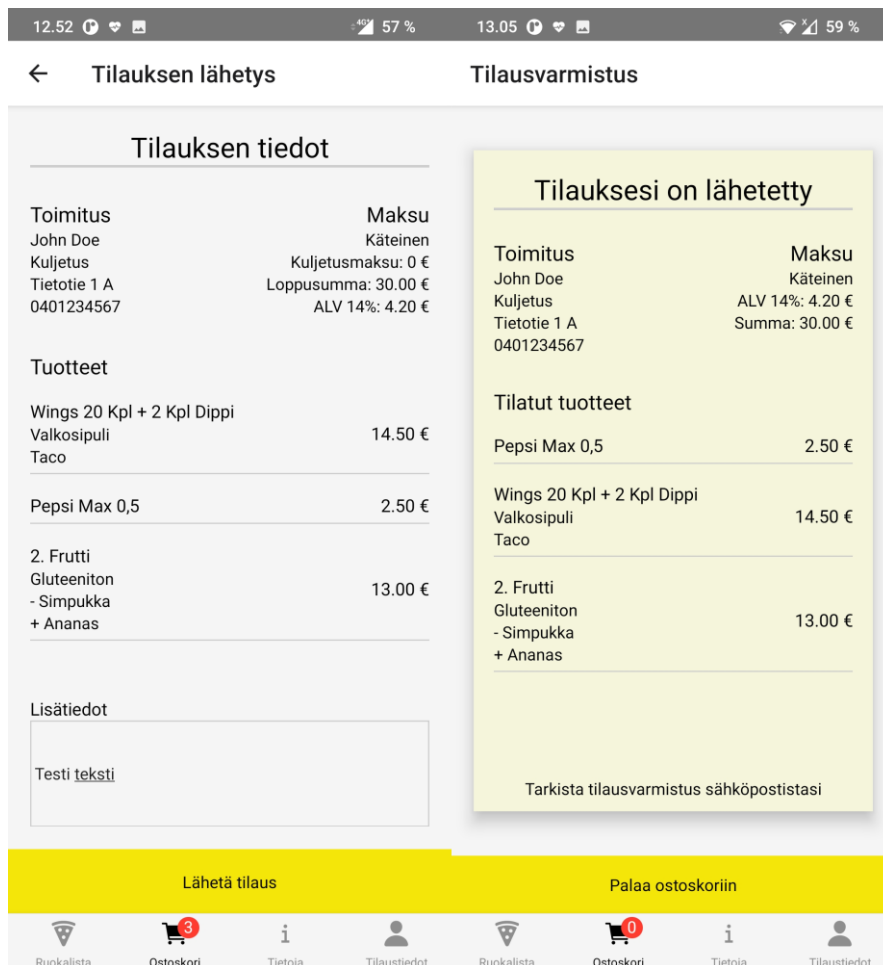
Ostoskorin tilan päivittäminen onnistuu Reduxin "dispatch"-funktiolla, jolla lähetetään parametrinä toiminnon tyyppi ja tarvittava data. Dispatch-funktio päivittää ostoskorin tilan tiedot riippuen funktiolle annetuista parametreistä. Esimerkkikoodissa 5 ostoskoriin lisätään tuote "cartObj()". Ostoskoriin lisättäessä lista päivitetään lisäämällä uusi olio listan loppuun. Ostoskorista poistettaessa käytetään splice-metodia objektin poistamiseksi listasta.

```
dispatch({type: 'ADD_TO_CART', payload: cartObj()})
```

Esimerkkikoodi 5. Ostoskoriin lisättäessä käytettävä metodi, joka määrittää tapahtuman tyyppin ja lisättävän olion.

Kun käyttäjä on tyytyväinen ostoskoriin lisäämiinsä tuotteisiin, siirrytään sivulle, jossa käyttäjältä pyydetään tilaukseen tarvittavia tietoja. Tilauksen jatkamiseen tarvitaan ensimmäisenä toimitustapa ja tieto, halutaanko tilaukseen myös kuljetus vai noudetaanko se paikan päältä. Jos tilaus halutaan kuljetettuna, sovellus pyytää lisäksi kuljetusosoitetta. Mikäli tilaus halutaan noutaa itse, pyydetään käyttäjältä tieto siitä, koska tilaus noudetaan. Tämän lisäksi tilaajalta vaaditaan myös puhelinnumero.

Tilauksen tietojen täyttämisen jälkeen edetään tilauksen tietojen tarkastus sivulle. Tarkastelusivulla näkyvät tilattavat tuotteet, niihin tehdyt muutokset ja niiden hinnat. Sivulla näkee myös edellisellä sivulla täytetyt tiedot ja tietoja tilauksen maksusta. Maksuosiosta näkee tilauksen kokonaissumman ja sen, tuleeko kuljetukseen kuljetusmaksua (kuva 9). Tilaukseen voidaan myös kirjoittaa lisätietoja, esimerkiksi talon ovikoodi. "Lähetä tilaus" -nappia painettaessa sovellus tekee https-post-pyyntöä back endille. Pyyntöissä lähetetään tilauksen tiedot sekä ostoskorin tiedot yhdessä objektissa.



Kuva 9. Tilaustietojen tarkistussivu ja tilauksen onnistuessa näytettävä tilausvarmistus.

Tilauksen onnistuessa sovellus näyttää tilausvarmistukseksi kuitin ja ohjaa käyttäjän tarkastamaan sähköpostin.



## Navigoinnin toteutus

Sovelluksen navigointi toteutettiin React Navigationin avulla. Toteutuksessa käytettiin yhdessä välilehtinavigointia ja pinonavigointia siten, että jokaisella välilehdellä oli oma pinonsa sivuja navigoitavaksi.

Esimerkkikoodi 6 on sovelluksen ostoskori välilehden navigaatiopino. Välilehteä painettaessa sovellus ohjaa pinon ensimmäiselle sivulle, josta voidaan nopein navigoida pinon muille sivuille.

```
const CartStackScreen = () => (
  <CartStack.Navigator>
  <CartStack.Screen name="Cart" component={Cart} options={{title: "Ostoskori"}} />
  <CartStack.Screen name="OrderDetails" component={OrderDetails} options={{title: "Toimitustiedot"}} />
  <CartStack.Screen name="OrderSend" component={OrderSend} options={{title: "Tilauksen lähetys"}} />
  <CartStack.Screen name="OrderConfirmed" component={OrderConfirmed} options={{title: "Tilausvarmistus", headerLeft: null}} />
</CartStack.Navigator>
)
```

Esimerkkikoodi 6. Ostoskori välilehden sivupino.

## 4.4 Nodemailer-moduuli

Nodemailer on Node.js-moduuli, jonka avulla voidaan lähettää sähköposteja SMTP-protokollan avulla. Nodemailerilla luodaan esimerkkikoodin 7 mukaisesti kuljettaja-objekti, jolla on lähettävän sähköpostin osoite ja salasana.

```
let transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: process.env.SENDER_USER,
    pass: process.env.SENDER_PASS
  }
});
```

Esimerkkikoodi 7. Nodemailer moduulin transporter, joka lähettää viestin gmail-osoitteesta muihin osoitteisiin.

Lisäksi Nodemailer tarvitsee sähköpostiosoitteen, johon viesti lähetetään, aiheen ja sisällön viestille. (40.)

#### 4.5 Back endin kehitys

Sovelluksen tilaustoiminnallisuutta varten tarvittiin back end, joka käsittelee käyttäjien sovelluksesta lähettämät tilaukset. Back end tehtiin Node js:ää ja sen moduuleja hyödyntäen.

Tilausjärjestelmän tilausten lähetys ja vastaanotto päätettiin toteuttaa samalla tavalla, kuin ravintolan oman sivun järjestelmä toimi eli sähköpostein. Back endin päätoiminto on vastaanottaa mobiilisovelluksen käyttäjän lähettämän tilauksen tiedot ja lähettää tiedoin varustettu sähköposti ravintolalle. Tilauksen lähdettyä ravintolaan saa käyttäjä omaan sähköpostiinsa tilausvarmistuksen.

Ennen kuin viesti lähetetään, lisätään viestin vastaanottaja, aihe ja viestin sisältö. Tilauksen tiedot lähetetään parametrina Nodemailerin käyttöön. Tilauksen tiedot järjestellään sopivaan muotoon, jotta ravintolalle saapuva sähköposti olisi mahdollisimman helppolukuinen ja sopiva tulostettavaksi. Ravintolalle lähetetään viesti aiheella "Uusi tilaus", ja kun lähetys on onnistunut, lähetetään tilaajalle viesti aiheella "Tilausvarmistus". Viestien lähetystapa on nähtävissä esimerkkikoodissa 8. Aihetta lukuun ottamatta viestien sisältö on samankaltainen.

```

transporter.sendMail(mailOptionsPT, (err, data) => {
  if(err) {
    console.log(err)
    return "Error"
  } else {
    console.log(data)
    transporter.sendMail(mailOptionsCustomer, (err, data)
=> {
      if(err) {
        console.log(err)
      } else {
        console.log(data)
        return "Success"
      }
    })
  })
})

```

Esimerkkikoodi 8. Nodemailer-transporterin sendMail-funktio, joka lähettää viestin.

Ennen kuin viesti lähetetään eteenpäin, tarkistetaan tilaajan antaman katuosoitteen olemassaolo. Annettu katuosoite lähetetään GET-pyyntöillä Hsl:n digitransit API:iin. Jos osoite on olemassa, API lähettää vastauksena osoitteen, postinumeron ja kaupunkialueen nimen. Vastauksen tietoja käytetään myös tilaustietojen täydentämiseen tarvittaessa.

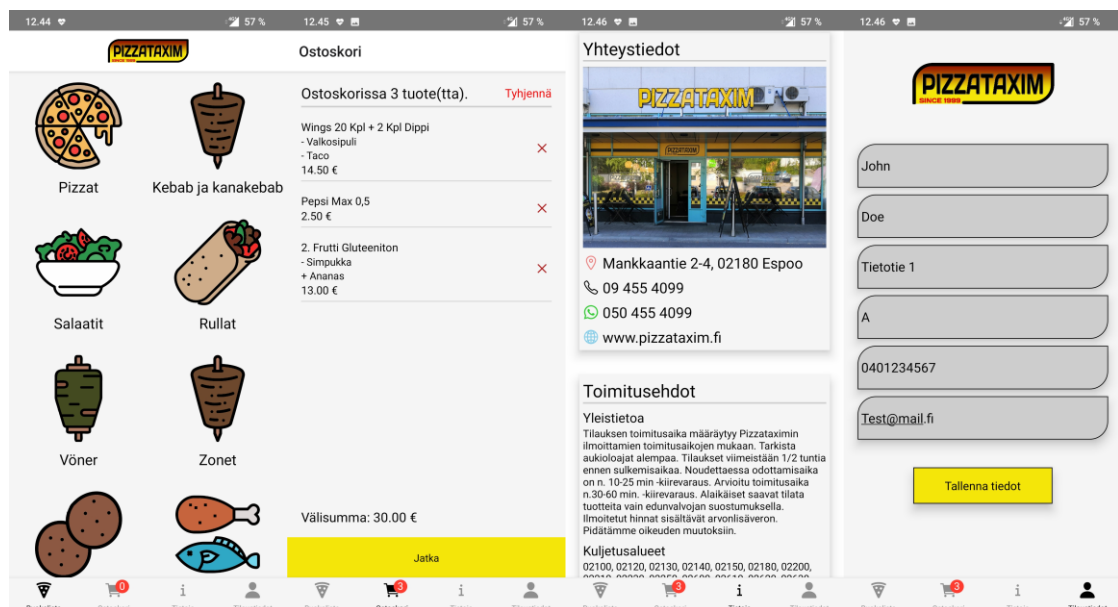
#### 4.6 Tulokset ja testaus

Tilausjärjestelmää kehitettiin noin kahden kuukauden ajan, ja vaikka sovellusta ei saatu vielä käyttäjien käsiin sovelluskauppojen kautta, saatiin projektista MVP-tilaussovellus. Ravintola, jolle sovellus tehtiin, oli tyytyväinen työn tuloksiin ja halusi jatkaa sen kehitystä pidemmälle tulevaisuudessa.

Sovellus saatiin vaiheeseen, jossa on vaaditut päätoiminnallisuudet. Sovelluksesta löytyy ravintolan koko ruokalista. Kaikkia ravintolan ruokalistan annoksia voi muokata mielensä mukaan joko lisukkeiden, kastikkeiden tai dippien osalta. Kaikki annokset voidaan lisätä ostoskoriin, josta voidaan lähettää joko noutotilaus tai kuljetustilaus. Kun tilaus lähetetään, ravintola ja tilaaja saavat sähköpostiviestin vahvistukseksi.

Infosivulla on kaikki oleellinen tieto ravintolasta: sieltä löytyvät ravintolan yhteystiedot, toimitusehdot ja aukioloajat. Tilaustiedot sivuksi muutettuun profiilisivuun voidaan lisätä omat tiedot ja tallentaa ne. Tilausta tehdessä sovellus hyödyntää tallennettuja tietoja kenttien täydentämiseen. Sovellusta ei kuitenkaan vielä voitu laittaa sovelluskauppoihin, sillä järjestelmä tarvitsee lisää testausta ja hienosäätöä.

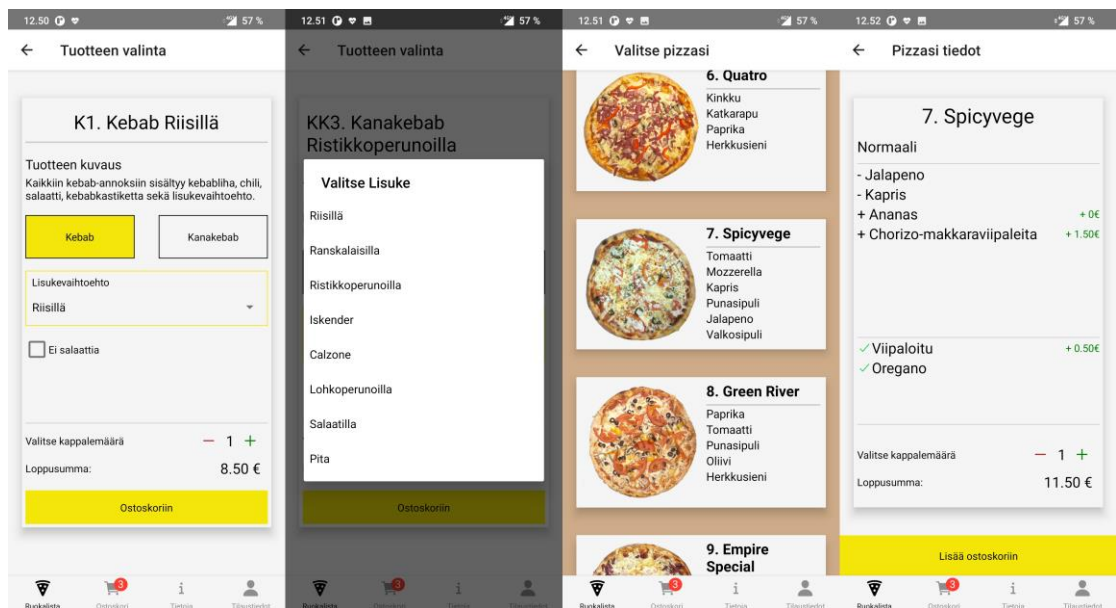
Sovellusta käyttäjien kanssa testattaessa selvisi, että käyttöliittymästä oli tullut toivottu eli helppokäyttöinen (kuva 10). Käyttöliittymää testattiin muutamien kymmenien ihmisten kanssa.



Kuva 10. Sovelluksen välilehtien pääsivut.

Sovelluksen käyttö sujui testaajilta usein ilman ohjeita. Testaajille annettiin tehtäväksi tilata jokin tietty annos. Välillä kategorioita tutkittiin hieman ennen oikean löytymistä. Sovelluksen teettäjä uskoo silti kompaktin kategorianäkymän olevan parempi kuin liian jaoteltu näkymä. Osa testaajista viesti siitä, että pitsan valintaprosessi oli turhan monimutkainen, sillä se oli jaoteltu turhan moneen osaan (kuva 11). Pitsan valintaprosessi onkin osa, jota tulevaisuudessa muutetaan siten, että pohja valitaan pudotusvalikosta sen jälkeen, kun täytteet on valittu.

Eräs testaaja ehdotti, että kun käyttäjä valitsee jonkin ruoka-annoksen, pitäisi sovelluksen heti seuraavassa näkymässä tarjota mukaan esimerkiksi juotavaa tai vaikka perunoita. Testaajan mainitsema toiminto toimii mielestäni paremmin juuri välityspalvelujen sovelluksessa, sillä se voi alkaa ärsyttää käyttäjiä, jos se toistuu liian usein. Variaatio tästä voisi olla yrittää tarjota käyttäjälle jotain ravintolan tarjouskupongeista. Esimerkiksi, kun ostoskoriin lisätään pitsa, voisi sovellus ehdottaa tarjouskupongeja, joihin sisältyy pitsa.



Kuva 11. Vasemmalla sivu, jolla annosta muokataan, ja oikealla kaksi näkymää pitsan valinnasta.

Ravintolalla oli pitsan täytteiden muokkaamista varten sellainen toiminnallinen pyyntö, että kun käyttäjä muokkaa pitsaansa, saa yhden täytteen vaihtaa ilmaiseksi. Ravintolan mukaan tämä olisi asiakkaille tärkeä toiminto, jonka vaikutuksen voi nähdä kuvan 11 oikeassa laidassa.

#### 4.7 Mobiilisovelluksen tulevaisuus

Sovelluksen kehitystyö jatkuu, sillä tarkoitus on, että se tulisi ihan oikeaan käyttöön ravintolalle. Puuttumaan jäivät käyttöliittymän animaatiot ja mukauttaminen eri laitteiden näyttökokoon. MVP-ajattelu ja sovellukseen tähän asti käytetty

aika tarkoittavat myös sitä, että jotkin osat, kuten sovelluksen infosivun merkkijonot, on kovakoodattu sovellukseen. Kaikki sovelluksessa näytettävä data olisi tulevaisuudessa tarkoitus hakea samalta back endiltä, johon tilaukset lähetetään. Back endiä varten pitäisi myös löytää sopiva hostauspalvelu, sillä kehitystyön aikana tilaustoimintoa on testattu vain lähiverkossa. Siitä, miten back end selviytyy joukosta samaan aikaan lähetettyjä tilauksia, ei ole vielä tietoa.

Järjestelmästä puuttuu vielä myös tilaamista rajoittavat ominaisuudet. Sovellus tarvitsee toiminnallisuuden, joka tarkistaa, että tilaajan ilmoittama katuosoite kuulu ravintolan kuljetusalueeseen. Tarvitaan myös toiminto, joka tarkistaa tilausajan. Kuljetustilaus pitää pystyä tekemään vain ravintolan ilmoittamien kuljetusajankohtien välissä, kun taas noutotilaus vain ilmoitetun aukiolon aikana. Myös juhla- ja erikoispäivät, jolloin ravintola on suljettu, täytyy ottaa huomioon sovelluksen toiminnassa.

Ravintolalta on tullut uusia ideoita ja ehdotuksia projektin edetessä. Yksi niistä on se, että sovellus laskisi matkan tilaajan ja ravintolan välillä ja asettaisi kuljetusmaksun sen mukaan. Toiminnon myötä voisi toimintoon käyttää tilaajan puhelimen sijaintidataa, jota voitaisiin myös hyödyntää tilauksen tekemisessä. Vartenotettava kehitysmahdollisuus on tehdä ravintolan verkkosivuilla olevan bonuspistejärjestelmän kaltainen mekanismi myös sovellukseen. Sovelluksesta olisi myös hyvä pystyä hyödyntämään ravintolan tarjouskupongeja.

Puuttuvien toiminnallisuuksien valmistuessa sovellus on vielä saatava yhteensopivaksi iOS-käyttöjärjestelmän kanssa sekä molempiin sovelluskaappoihin. Ravintolalla on jo ennen kehitystyön alkua ollut kyselyitä omasta mobiilisovelluksesta, mikä lupaa sen, että halukkaita käyttäjiä sovellukselle on jo olemassa. Jos sovellus menestyy odotetusti, tarkoituksena olisi tarjota myös muille pienravintolayrittäjille mahdollisuutta oman mobiilisovelluksen julkaisuun.

## 5 Yhteenveto

Insinööriyössä tutkittiin erilaisten verkkotilausjärjestelmien suosiota sekä trendien muuttumista ajan edetessä. Mobiililaitteiden yleistyminen ja ihmisten eristäytyminen koteihin koronapandemian vuoksi on nostanut noutopikaruokaravintoloiden suosiota. Tilanteesta suurimman hyödyn ovat saaneet välityspalvelut.

Välityspalvelut, kuten Wolt ja Foodora, ovat hyödyllisiä alustoja monille yrityksille, sillä ne laajentavat alustalla olevien ravintoloiden asiakaskuntaa merkittävästi. Palveluiden tarjoama näkyvyys on välttämätön monille pienille ja uusille ravintoloille, jotka eivät pärjäisi ilman alustojen tuomaa kauppaa. Välityspalvelut ovat joidenkin yrittäjien mielestä liian kalliita ja käyttävät liian vahvasti hyväksi niiden markkinajohtajan asemaa. Välityspalveluiden ongelmien takia moni pienyrittäjä saattaa pohtia oman sovelluksen kehittämismahdollisuuksia.

Välityspalvelujättejä ja niiden miljoonabudjetteja vastaan taistelemisen ei ole helppoa, ja jos urakkaan lähdetään, on tärkeää luoda oikeanlainen sovellus. Monet ihmiset suosivat mobiilisovelluksia niiden helppouden ja nopeuden takia. Sovellukseen saadaan asiakkaita helpommin, jos sovelluksen käyttäjille tarjotaan yksinomaisia asiakasetuja.

Työssä luotiin React Nativella Android-sovellus. React Native valikoitui työkaluksi sen etujen takia. Kehitystyöhön käytettiin noin kaksi kuukautta, ja sinä aikana saatiin sovellukseen toivotut päätoiminnot valmiiksi. Kehitystyön alkaessa päämääräksi asetettiin sovellus, jolla voitaisiin tehdä tilaus samalla tavalla kuin ravintolan verkkosivulla. Sovellus luo tilauksen käyttäjän ostoskorin ja tilaustietojen perusteella, minkä jälkeen se lähetetään sähköpostina sekä tilaajalle että ravintolalle. Insinööriyönä luotua sovellusta voidaan soveltaa tulevaisuudessa myös muiden ravintoloiden tarpeisiin.

Ravintolakohtaisen mobiilisovelluksen luominen on mahdollista, mutta se vaatii paljon huomioitavaa. Itse en suosittelisi omaa sovellusta ravintoloille, joilla ei ole tarpeeksi suurta asiakaskuntaa, joka sovellusta käyttäisi. Lisäksi laadukkaan sovelluksen kehitys vaatii aikaa ja rahaa. Prosessissa ei kannata yrittää oikoa

mutkia, sillä huonosti toteutettu sovellus voi jopa vahingoittaa ravintolan brändiä. Jos sovellus kehitetään ravintolalle, joka pärjää myös ilman välityspalveluja, uskon sovelluksen menestyvän, kun sitä markkinoidaan oikein. Paras tulos saadaan sovelluksen alkuvaiheessa siten, että sitä käytetään yhdessä välityspalveluiden kanssa. Näin asiakkaita voidaan yrittää siirtää välityspalveluista omalle alustalle.



## Lähteet

- 1 Hesburgerin mobiilisovellus kasvatti myyntiä 25 %. Verkkoaineisto. Vincit. <<https://www.vincit.fi/fi/asiakkaat/hesburger/>>. Luettu 17.4.2021.
- 2 Vuorela, Arvo. 2021. Ravintolaruoan kotiin tilaaminen hurjassa nosteessa, eikä ilmiö selity pelkällä koronalla. Verkkoaineisto. Yle. <<https://yle.fi/uutiset/3-11719615>>. 10.1.2021. Luettu 15.3.2021.
- 3 Ellilä, Tiina. 2021. Poikkeuksellisen moni tilaa ruokaa nyt kotiin – Ravintolat ja lähettipalvelut kertovat: pienin tilaus voi olla jäätelö, jotkut päivät ovat ”ruoan tykitystä pihalle”. Verkkoaineisto. Aamulehti. <<https://www.aamulehti.fi/uutiset/art-2000007853404.html>> 14.3.2021. Luettu 16.3.2021.
- 4 Kärstitkö puhelimessa puhumisen pelosta? 2016. Näin siitä pääsee eroon. Verkkoaineisto. Mtv uutiset. <<https://www.mtvuutiset.fi/artikkeli/karsitko-puhelimessa-puhumisen-pelosta-nain-siita-paasee-eroon/5986584>>. 18.7.2016. Luettu 17.4.2021.
- 5 Internetiä käytetään yhä yleisemmin matkapuhelimella – myös ostosten tekemiseen. 2018. Verkkoaineisto. Tilastokeskus. <[https://tilastokeskus.fi/til/sutivi/2018/sutivi\\_2018\\_2018-12-04\\_tie\\_001\\_fi.html?ad=notify](https://tilastokeskus.fi/til/sutivi/2018/sutivi_2018_2018-12-04_tie_001_fi.html?ad=notify)>. 4.12.2018. Luettu 17.3.2021.
- 6 Matkapuhelimella ostaminen. Verkkoaineisto. Tilastokeskus. <[https://tilastokeskus.fi/til/sutivi/2018/sutivi\\_2018\\_2018-12-04\\_kat\\_003\\_fi.html](https://tilastokeskus.fi/til/sutivi/2018/sutivi_2018_2018-12-04_kat_003_fi.html)>. Luettu 23.3.2021.
- 7 Gastronautti tulee. 2002. Verkkoaineisto. City. <<https://www.city.fi/opas/gastronautti+tulee/678>>. 20.12.2002. Luettu 28.3.2021.
- 8 Ravintolaruokailun trenditutkimus 2020: Ravintolaruokailu palaa ennalleen koronapandemian jälkeen. 2020. Verkkoaineisto. MaRa. <<https://www.mara.fi/ajankohtaista/tiedotteet/ravintolaruokailun-trenditutkimus-2020-ravintolaruokailu-palaa-ennalleen-koronapandemian-jalkeen.html>>. 9.12.2020. Luettu 17.3.2021.
- 9 Ranta, Elina. 2016. Mikä lama? Ravintolaruuan kotitilauksilla menee lujaa. Verkkoaineisto. Iltasanomat. <<https://www.is.fi/taloussanomat/art-2000001903820.html>>. 12.2.2016. Luettu 18.3.2021.
- 10 Yksinasuminen yleistynyt etenkin nuorten ja iäkkäiden ikäryhmissä 2019. Verkkoaineisto. Tilastokeskus. <[https://tilastokeskus.fi/til/asas/2019/asas\\_2019\\_2020-05-20\\_tie\\_001\\_fi.html](https://tilastokeskus.fi/til/asas/2019/asas_2019_2020-05-20_tie_001_fi.html)>. Luettu 23.3.2021.
- 11 Kotipizza-Äppi. Verkkoaineisto. Kotipizza. <<https://www.kotipizza.fi/app>>. Luettu 25.3.2021.

- 12 Google Play. Verkkoaineisto. Google. <<https://play.google.com/store?hl=en&gl=FI>>. Luettu 25.3.2021.
- 13 Wolt. Verkkoaineisto. Wolt. <<https://wolt.com/en>>. Luettu 25.3.2021.
- 14 Mikä on Foodora? Verkkoaineisto. Foodora. <<https://www.foodora.fi/en/contents/tietoa-palvelusta?r=1>>. Luettu 23.3.2021.
- 15 Toivonen, Janne& Björkstén, Tuomo. 2019. Yle tutki: Pizzayrittäjät syyttävät Pizza-onlinea riistosta – Näin nettipalvelusta kasvoi jättiläinen, joka vie pizzasta enemmän kuin verottaja. Verkkoaineisto. Yle. <<https://yle.fi/uutiset/3-10839203>>. 4.7.2019. Luettu 23.3.2021.
- 16 Andrew, Jason. 2019. Should your business use UberEats? All the pros, cons and costs explained. Verkkoaineisto. SmartCompany. <<https://www.smartcompany.com.au/industries/hospitality/ubereats-delivery-pros-cons/>>. 29.3.2019. Luettu 2.4.2021.
- 17 Lavas, Ilkka. 2019. 1000 uutta haamuravintolaa - ravintola-alan uusi salakavala kilpailija. Verkkoaineisto, City. <<https://www.city.fi/blogit/lavas/1000+uutta+haamuravintolaa+ravintola-alan+uusi+salakavala+kilpailija/135025>>. 16.6.2019. Luettu 3.4.2021.
- 18 Puljujärvi, Ismo. 2020. MTV Uutiset selvitti: Jopa 30 prosenttia! Näin muhkean siivun ruoanvälityspalvelut Foodora ja Wolt veloittavat ravintoloilta. Verkkoaineisto. Mtv uutiset. <<https://www.mtvuutiset.fi/artikkeli/mtv-uutiset-selvitti-jopa-30-prosenttia-nain-muhkean-siivun-ruoanvalityspalvelut-foodora-ja-wolt-veloittavat-ravintoloilta/7691324#gs.xlzxww>>. 15.1.2020. Luettu 4.4.2021.
- 19 Tolvanen, Perttu. 2019. Case Pizza-online – miksi omaan webbikanavaan kannattaa panostaa. Verkkoaineisto. Vierityspalkki.fi. <<https://vierityspalkki.fi/2019/07/15/case-pizza-online-miksi-omaan-webbikanavaan-kannattaa-panostaa/>>. 17.7.2019. Luettu 4.4.2021.
- 20 Calisiyor, Ercan. 2021. Ravintolayrittäjä. Pizzataxim, Espoo. Keskustelu 5.4.2021.
- 21 Rinta-Jouppi, Anton. 2021. Ennätysrahoituksen keränneen Woltin kilpailijat vaihtelevat miljardien kolosseista pienempiin alabrändeihin, mutta kilpailu alalla on kovaa – ”Ihmisten kulutuksen muutos on isoin kasvun ajuri”. Verkkoaineisto. Kauppalehti. <<https://www.kauppalehti.fi/uutiset/ennatysrahoituksen-keranneen-woltin-kilpailijat-vaihtelevat-miljardien-kolosseista-pienempiin-alabrandeihin-mutta-kilpailu-alalla-on-kovaa-ihmisten-kulutuksen-muutos-on-isoin-kasvun-ajuri/8c65623f-292a-4f5a-b761-94b6147f05be>>. 13.2.2021. Luettu 7.4.2021.
- 22 Riippi, Juha. 2019. Overview of Mobile Development Frameworks in 2019. Verkkoaineisto. Vincit. <<https://www.vincit.fi/fi/overview-of-mobile-development-frameworks-in-2019/>>. 4.3.2019. Luettu 10.4.2021.

- 23 Mobile Operating System Market Share Finland. 2021. Verkkoaineisto. Statcounter. <<https://gs.statcounter.com/os-market-share/mobile/finland>>. 1.3.2021. Luettu 7.4.2021.
- 24 Poot, Arthur. 2020. The state of PWA support on mobile and desktop in 2020. Verkkoaineisto. Simplabs. <<https://simplabs.com/blog/2020/06/10/the-state-of-pwa-support-on-mobile-and-desktop-in-2020/>>. Luettu 12.4.2021.
- 25 Stribos, Tim. 2019. Advantages and Disadvantages of Progressive Web Apps. Verkkoaineisto. Moqod. <<https://moqod.com/advantages-and-disadvantages-of-pwa-progressive-web-app/>>. 26.10.2019. Luettu 12.4.2021.
- 26 React Native – Tehokkaampaa mobiilikehitystä. 2018. Verkkoaineisto. Hurja. <<https://www.hurja.fi/blogi/react-native-tehokkaampaa-mobiilikehitysta/>>. 20.6.2018. Luettu 12.4.2021.
- 27 Blair, Ian. Verkkoaineisto. Buildfire. <<https://buildfire.com/ways-business-benefit-having-mobile-app/#>>. Luettu 13.4.2021.
- 28 Wurmser, Yoram. 2019. US Time Spent with Mobile 2019. Verkkoaineisto. eMarketer. <<https://www.emarketer.com/content/us-time-spent-with-mobile-2019>>. 30.5.2019. Luettu 13.4.2021.
- 29 Mobile app vs mobile website statistics. Verkkoaineisto. Jmango. <<https://jmango360.com/mobile-app-vs-mobile-website-statistics/>>. Luettu 15.4.2021.
- 30 Riippi, Juha. 2020. Mobiilisovelluksen markkinointi – älä unohda tehdä! Verkkoaineisto. Vincit. <<https://www.vincit.fi/fi/mobiilisovelluksen-markkinointi/>>. 27.1.2020. Luettu 15.4.2021.
- 31 Bhatt, Tuhin. 2020. What Are the Benefits of Having a Mobile App for Your Business. Verkkoaineisto. Intelvita. <<https://www.intelvita.com/blog/benefit-having-mobile-app-business/>>. 7.7.2020. Luettu 16.4.2021.
- 32 Gazdecki, Andrew. 2016. 10 Ways Your Business Can Grow with a Mobile App. Verkkoaineisto. Small Business Trends. <<https://smallbiz-trends.com/2016/08/benefits-of-having-an-app.html>>. 29.8.2016. Luettu 16.4.2021.
- 33 Are DoorDash, UberEats Good For Restaurants? 2021. Verkkoaineisto. CNBC. <[https://youtu.be/-KwtJX\\_Tcjo](https://youtu.be/-KwtJX_Tcjo)>. 30.3.2021. Luettu. 17.4.2021.
- 34 Hooks at a Glance. Verkkoaineisto. Reactjs. <<https://reactjs.org/docs/hooks-overview.html>>. Luettu 5.5.2021.
- 35 React. Verkkoaineisto. React JS. <<https://reactjs.org/>>. Luettu 2.4.2021.

- 36 Lobera, Alex. 2018. A Beginner's Guide to React: How does React work? Verkkoaineisto. Medium. <<https://medium.com/leanjs/introduction-to-react-3000e9cbcd26>>. 5.6.2018. Luettu 5.5.2021.
- 37 React Native. Verkkoaineisto. React Native. <<https://reactnative.dev/>>. Luettu 5.5.2021.
- 38 Wryat, Amaury. 2019. Redux-Persist: How it Works and How to Change the Structure of Your Persisted Store. Verkkoaineisto. Bam. <<https://blog.bam.tech/developer-news/redux-persist-how-it-works-and-how-to-change-the-structure-of-your-persisted-store>>. Luettu 5.5.2021.
- 39 Redux Persist. 2021. Verkkoaineisto. Github. <<https://github.com/rt2zz/redux-persist>>. Luettu 5.5.2021.
- 40 Nodemailer. Verkkoaineisto. Nodemailer. <<https://nodemailer.com/about/>>. Luettu 5.5.2021.