



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Jussi Saarela

# OHJELMISTON KÄYTETTÄVYYS JA KÄYTTÖLIITTYMÄSUUNNITTELU

Case Mealston

Liiketalous ja matkailu  
2014

## TIIVISTELMÄ

Tekijä	Jussi Saarela
Opinnäytetyön nimi	Ohjelmiston käytettävyys ja käyttöliittymäsuunnittelu
Vuosi	2014
Kieli	suomi
Sivumäärä	37
Ohjaaja	Klaus Salonen

---

Suomalainen aloittava startup-yritys kehittää ravintoloille tilausjärjestelmää ja tarvitsee sille käyttöliittymän. Olen itse mukana kehittämässä järjestelmää ja minun osa työstä on käyttöliittymän kehitys. Käyttöliittymästä piti saada mahdollisimman helppokäyttöinen, koska käyttäjäryhmien ikä voi vaihdella suuresti ja se on käytössä muun muassa ravintoloiden keittiössä, jossa ei ole aikaa tuhlattavaksi hitaasti ja vaikeasti toimivan järjestelmän kanssa.

Opinnäytetyössä kerron käytettävyyden teoriasta, käyttöliittymäsuunnittelun psykologisesta näkökulmasta ja käyn läpi käyttöliittymän suunnittelun eri prosesseja.

## ABSTRACT

Author	Jussi Saarela
Title	Usability and User Interface Design
Year	2014
Language	Finnish
Pages	37
Name of Supervisor	Klaus Salonen

---

A Finnish startup-company is developing a new ordering software for restaurants and needed a user interface for it. I am a member of the developing team and my job is to design the user interface. The interface had to be as easy to use as possible because the range of ages among the users varies greatly and the software is being used in, for example, restaurant kitchens where there is no time to spend on software that is slow and difficult to use.

This thesis examined theory of usability, the psychological basis for user interface design and the different stages of designing a user interface.

# SISÄLLYS

## TIIVISTELMÄ

## ABSTRACT

1	JOHDANTO .....	4
2	TOIMEKSIANTAJA JA TAVOITTEET .....	5
2.1	Toimeksiantajan tavoitteet .....	5
2.2	Omat tavoitteeni .....	5
3	SOVELLUKSEN ESITTELY .....	6
3.1	Staff-sovellus.....	6
3.2	Customer-sovellus.....	6
3.2.1	Customer-sovelluksen verkko- ja mobiilisovellukset.....	6
4	KÄYTTÖLIITTYMÄT.....	8
4.1	Mobiili- ja tietokonesovelluksien käyttöliittymien erot.....	8
5	KÄYTTÖLIITTYMÄSUUNNITTELU .....	9
5.1	Käyttöliittymäsuunnittelun vaiheet .....	9
5.2	Sisällön suunnittelu .....	9
5.3	Mockup .....	10
5.3.1	Mockup-kuvien suunnitteluohjelmistot .....	10
5.4	Graafinen suunnittelu .....	11
5.4.1	Layout-kuvat .....	11
6	KÄYTETTÄVYYS.....	12
6.1	Käyttäjäkeskeinen suunnittelu .....	15
6.2	Käyttäjien toiminta.....	15
6.2.1	Käyttätarina Mealston-sovellukselle.....	16
6.3	Käytettävyyden psykologia.....	16
6.3.1	Biologisia vaikuttajia .....	17
6.4	Käytettävyyden testaus.....	18
7	MEALSTON-SOVELLUKSEN KÄYTTÖLIITTYMÄ .....	20
7.1	Staff-puolen käyttöliittymä .....	20
7.1.1	Navigointi .....	20
7.1.2	Ruokalistat .....	21
7.1.3	Ruoat .....	23

7.1.4	Tilaukset.....	25
7.1.5	Tilin asetukset.....	26
7.2	Customer-sovellus.....	27
8	TAVOITTEET JA NIIDEN TOTEUTUMINEN .....	30
9	TULEVAISUUS .....	31
10	YHTEENVETO.....	32
11	LÄHTEET .....	33

**KUVIO- JA TAULUKKOLUETTELO**

<b>Kuvio 1</b> .....	10
<b>Kuvio 2</b> .....	13
<b>Kuvio 3</b> .....	14
<b>Kuvio 4</b> .....	14
<b>Kuvio 5</b> .....	21
<b>Kuvio 6</b> .....	21
<b>Kuvio 7</b> .....	22
<b>Kuvio 8</b> .....	22
<b>Kuvio 9</b> .....	23
<b>Kuvio 10</b> .....	24
<b>Kuvio 11</b> .....	24
<b>Kuvio 12</b> .....	25
<b>Kuvio 13</b> .....	26
<b>Kuvio 14</b> .....	27
<b>Kuvio 15</b> .....	29
<b>Taulukko 1</b> .....	18

# 1 JOHDANTO

Tämä opinnäytetyö kertoo ohjelmiston käyttöliittymäsuunnittelusta ja käytettävyydestä. Käytettävyys on tärkeä osa sovelluksen suunnittelu- ja toteutusprosessia, koska ilman hyvää käytettävyys- tai käyttöliittymäsuunnitelmaa sovelluksesta voi tulla vaikeakäyttöinen, ja tämän takia se voi menettää tai sen on vaikea saada uusia käyttäjiä. Työ on tehty paikalliselle startup-yritykselle, jonka sovellus on tarkoitus lanseerata syksyllä 2014.

Opinnäytetyössä kerron käyttöliittymäsuunnittelun teoriasta ja sen eri suunnittelu- vaiheista ennen varsinaista koodaamisen alkua. Kerron myös, kuinka olen itse toteuttanut Mealston-sovellukselle käyttöliittymän ja mitä työkaluja tähän käytin.

## **2 TOIMEKSIANTAJA JA TAVOITTEET**

Toimeksiantaja on Vaasaan perustettava uusi startup-yritys. Yritys tuottaa erilaisia mobiili- ja verkkosovellusratkaisuja kuluttajille ja yrityksille sekä palvelualueille. Tässä opinnäytetyössä kerron yrityksen ensimmäisestä tuotteesta, Mealstonista. Yrityksellä on jo ensimmäinen asiakas, joka ottaa Mealston-ohjelmiston käyttöön alkusyksyn 2014 aikana.

### **2.1 Toimeksiantajan tavoitteet**

Toimeksiantajan tavoitteena on saada ohjelmistostaan asiakkailleen helposti käyttöön otettava ilman, että siihen tarvitsee erillistä it-tukea. Ohjelmiston toinen tärkeä kriteeri on käytettävyys; koska ohjelmiston toista osaa tulee käyttämään hyvin laaja ikäryhmä, se pitää rakentaa niin, että se on helposti opittavissa ja sitä on helppo käyttää, vaikka ei välttämättä olisikaan kovin tuttu mobiililaitteiden kanssa.

### **2.2 Omat tavoitteeni**

Ominä tavoitteenani on saada työstä mahdollisimman hyvä referenssi mahdollista tulevaisuuden uraa varten. Haluan todistaa taitoni sekä tietotaitoni käyttöliittymän suunnittelusta ja käytettävydestä. Toinen tavoite on myös saada toimeksiantajan ohjelmistosta toimiva kokonaisuus, koska olen itse mukana kyseisessä startup-yrityksessä. Kolmantena tavoitteena on oppia työn aikana enemmän käytettävydestä sekä käyttöliittymän suunnittelusta ja sen erilaisista menetelmistä ja työkaluista.



### **3 SOVELLUKSEN ESITTELY**

Ohjelmisto koostuu kahdesta eri sovelluksesta: Ravintoloille tulee käyttöön verkossa selaimella käytettävä sovellus (staff-sovellus) ja loppukäyttäjille eli ravintolan asiakkaille on käytössä ruokalistasovellus (customer-sovellus), joka toimii sekä verkossa selaimella että mobiilisovelluksena.

#### **3.1 Staff-sovellus**

Verkkosovellus on tarkoitettu ravintolan hallinnolle. Sovelluksella hallitaan ruokalistasovellusta sekä vastaanotetaan tilaukset.

Staff-sovellus on toteutettu verkkoteknologioilla, kuten HTML5, PHP ja jQuery. Sovellus on rakennettu MVC-arkkitehtuuriin, joka mahdollistaa sovelluksen käyttöliittymän erottamisen sovelluksen back end:stä.

#### **3.2 Customer-sovellus**

Ruokalista-sovelluksella selataan ruokalistaa ja tehdään tilaukset. Tilaukset, jotka ovat toteutettu tämän sovelluksen kautta, menevät suoraan staff-sovelluksen tilauslistaukseen.

Customer-sovellus on joko verkossa toimiva etätilausruokalista tai ravintolan asiakkaille jaettava tablet-pc -laitteella toimiva natiivi mobiilisovellus. On täysin ravintolan päätettävissä, minkälaista ratkaisua he haluavat käyttää.

##### **3.2.1 Customer-sovelluksen verkko- ja mobiilisovellukset**

Etätilaussovellus toimii Mealstonin omalla palvelimella. Kun ravintola tekee etätilausruokalistan, se saa linkin ruokalistan osoitteeseen. Tämä linkki sisältää valmiiksi ravintolan ID:n, joka identifioi ruokalistan.

Etätilaussovellus on toteutettu yleisillä verkkoteknologioilla, mutta tietokantayhteydet eivät toteudu PHP:n kautta vaan ajaxilla. Tämä oli ratkaisu, johon päädyttiin siksi, että ei tarvitse koodata kahta ohjelmistoa samasta sovelluksesta. Mobiilisovellus käännetään natiiviksi Android-sovellukseksi käyttäen Phonegap-

palvelua, joka tarkoittaa sitä, että siinä ei voi käyttää PHP:tä, koska Phonegap ei tue sitä. Täten kaikki tietokantayhteydet tuli hoitaa jQueryn avulla.

## 4 KÄYTTÖLIITTYMÄT

Käyttöliittymä on ohjelmiston tai laitteen osa, jolla käyttäjä käyttää tuotetta. Käyttöliittymä voi olla tietokoneohjelman graafinen käyttöliittymä kuin myös pyykinpesukoneen käyttöpaneeli. Käyttöliittymä sisältää asioita, joilla käyttäjä on vuorovaikutuksessa tuotteen kanssa, kuten pesukoneen painikkeet ja tietokoneohjelman valikot tai –tekstikentät.

### 4.1 Mobiili- ja tietokonesovelluksien käyttöliittymien erot

Tietokonesovelluksien ja mobiilisovelluksien käyttöliittymillä on suurikin ero. Siinä missä tietokonesovelluksia ohjataan hiirellä ja näppäimistöllä, mobiilisovelluksia käytetään useimmiten sormella. Tällöin mobiililaitteiden käyttöliittymissä täytyy ottaa huomioon eri elementtien suunnittelu; jos painikkeet tekee samoin, kun ne on tarkoitettu painettaviksi hiirellä, ne voi olla liian pieniä tai liian lähekkäin ja niitä sormella painettaessa voi painaa vahingossa kahta kerralla. Tämä oli muun muassa Applen myyntiargumentti ensimmäisen iPhoneen kanssa; puhelimen sanottiin pyörittävän Applen OS X –tietokonekäyttöjärjestelmää, mutta se oli optimoitu käytettäväksi sormella eikä esimerkiksi Stylus-kynällä, koska kukaan ei halua käyttää sellaisia (Apple 2007). Käyttöliittymän kuvakkeet olivat tarpeeksi selkeitä ja käyttöliittymän kaikki elementit oli optimoitu käytettäväksi sormella. Tämän lisäksi mobiililaitteiden suunnitteluun vaikuttavat seikat, kuten käyttö ulkona, jolloin näyttöön heijastuu enemmän valoa, niiden käyttö tilanteissa, joissa sitä täytyy pystyä käyttämään nopeasti yhdellä kädellä tai ilman sen kummempaa huomion kiinnittämistä itse näyttöön, kuten autossa (jossa tosin ajaessa mobiililaitteiden käyttö on kielletty lailla) ja näytön koko. Tämä pätee nykypäivänä jokaiseen uuteen mobiililaitteeseen. Nämä mainitut seikat eivät päde tietokoneisiin, ja vaikka tablet-pc –laitteet ovatkin teoriassa mobiililaitteita, ne voidaan rinnastaa useimmissa tapauksissa kannettaviin tietokoneisiin, joskin käyttöliittymä pysyy kosketusnäytön takia kosketukselle suunniteltuna.

## 5 KÄYTTÖLIITTYMÄSUUNNITTELU

Käyttöliittymäsuunnittelu on ohjelmiston toteuttamisen kannalta yhtä tärkeä vaihe kuin itse ohjelmiston koodaus. Huonosti suunniteltu käyttöliittymä on vaikeakäyttöinen, ja jos käyttäjät eivät ole tyytyväisiä ohjelmiston toimintaan, he voivat helposti jättää käyttämättä sitä. Käyttöliittymän suunnittelu voi olla pelkkien mockup-kuvien tekoa (kappale 5.3) tai sitten graafista suunnittelua (kappale 5.4).

### 5.1 Käyttöliittymäsuunnittelun vaiheet

Kun suunnitellaan käyttöliittymää, sitä voi tehdä yksi tai useampi työntekijä. Aluksi suunnitellaan sovelluksen sisältö. Tämän jälkeen sisällön pohjalta käyttöliittymästä tehdään niin sanottu mockup-versio. Mockup-kuvista graafikko yleensä tekee vielä graafisen layout-kuvan.

### 5.2 Sisällön suunnittelu

Sovelluksen kehityksen alussa pitää miettiä, mitä sovelluksella voi tehdä, mikä on sen käyttäjäryhmä ja minkälaisissa ympäristöissä sitä käytetään. Ominaisuuksista kannatta karsia alkuun pois kaikki sellaiset, joita ei välttämättä tarvitse heti ensimmäiseen sovelluksen versioon. Näin saadaan vähennettyä resursseja, eikä niitä mene hukkaan, mikäli sovellus ei jostain syystä menesty. Näin teimme myös Mealstonin kohdalla. Kannattaa myös kirjoittaa ylös heti alkuun, mitä missäkin sivulla voi tehdä; esimerkiksi etusivulla voi olla promootio-teksti ja navigointi, josta pääsee sovelluksen kirjautumissivulle. Tämän voi suunnitella esimerkiksi mindmapillä (**Kuvio 1**).



**Kuvio 1**

Käyttäjärühmän identifiointi suunnittelun aikana on hyvin tärkeää käyttöliittymän kannalta. Sovelluksella voi olla esimerkiksi iäkkäämpiä ihmisiä käyttäjinä ja suunnittelun aikana täytyy pitää mielessä, että tämä käyttäjärühmä ei ole välttämättä ikinä käyttänyt tietokonetta tai muuta päätelaitetta.

### 5.3 Mockup

Mockup-kuvat tai ”wireframe” ovat käyttöliittymän raakaversioita. Niitä voidaan ajatella sovelluksen pohjapiirroksina. Mockupit sisältävät suunniteltavan ohjelmiston rakenteen, sisällön, toiminnallisuuden ja sovelluksen käyttäytymisen. Rakenteessa näkyy, kuinka sovelluksen osat on rakennettu. Sisältö suunnitellaan niin, että missä ja miten sisältö esitetään. Sovelluksen toiminnallisuus näyttää, kuinka eri elementit vaikuttaa sovelluksen käyttöliittymään ja kuinka käyttöliittymä käyttäytyy käyttäjän kanssa (Bank 2014).

#### 5.3.1 Mockup-kuvien suunnitteluohjelmistot

Mockup-kuvien suunnitteluun on lukuisia eri sovelluksia, esimerkiksi UXPin, Balsamiq ja Proto.io. Nämä ohjelmistot antavat työkalut suunnitella käyttöliittymiä sekä tietokoneelle että verkkoon. UXPin muun muassa mahdollistaa myös Google Glass -käyttöliittymien suunnittelun. Opinnäytetyön tekemiseen käytin

Balsamiq-ohjelmistoa, koska se oli jo ennestään tuttu ja siitä oli olemassa ilmainen demo-versio sekä työpöytäsovelluksena Mac-käyttöjärjestelmälle että verkkosovelluksena.

Ohjelmistot yleensä antavat sekä valmiita elementtejä, kuten input-kenttiä ja navigointipalkkeja, että tavallisia muototyökaluja, joilla voi rakentaa omia elementtejä. Näiden työkalujen avulla suunnittelija voi rakentaa käyttöliittymän mockupin helposti ja nopeasti.

## **5.4 Graafinen suunnittelu**

Käyttöliittymien graafinen suunnittelu tarkoittaa käyttöliittymän näköisversion suunnittelua. Suunnitteluun voi käyttää erilaisia työkaluja, mutta Adobe antaa suunnittelijoille sekä kehittäjille ehkä suurimman kokoelman erilaisia työkaluja. Mockup-kuville adobella on Reflow, grafiikoille Photoshop, Illustrator ja Fireworks ja koodaukseen löytyy Edge Code ja Dreamweaver.

### **5.4.1 Layout-kuvat**

Layoutit tehdään edellä mainituilla grafiikkaohjelmistoilla. Omassa työssäni käytin Adobe Photoshop -ohjelmaa, koska olen työskennellyt sen parissa jo monta vuotta, joten sen käyttö on helppoa ja se antaa työkalut jokaiseen tarpeeseeni.

Layouteihin määritellään tarkkaan elementtien värit, fonttikoot ynnä muut. Nämä kuvat ovat yleensä lopullisia sovelluksen näköiskuvia, joiden mukaan koodaaja rakentaa ohjelmiston. On tärkeää tehdä pikselintarkkaa työtä, koska koodauksessa täytyy asettaa useimmiten elementeille niiden paikka ja koko tarkasti pikseleinä, eikä koodaaja välttämättä ole graafisen suunnittelun ammattilainen, joten hän ei voi arvata näitä. Värit pitää myös määrittää. Useasti koodaaja saa esimerkiksi verkkosivusta PSD (Photoshop) -tiedoston, josta hän pystyy ottamaan koot, värit ja elementtien paikat ja kääntää ne koodiksi.

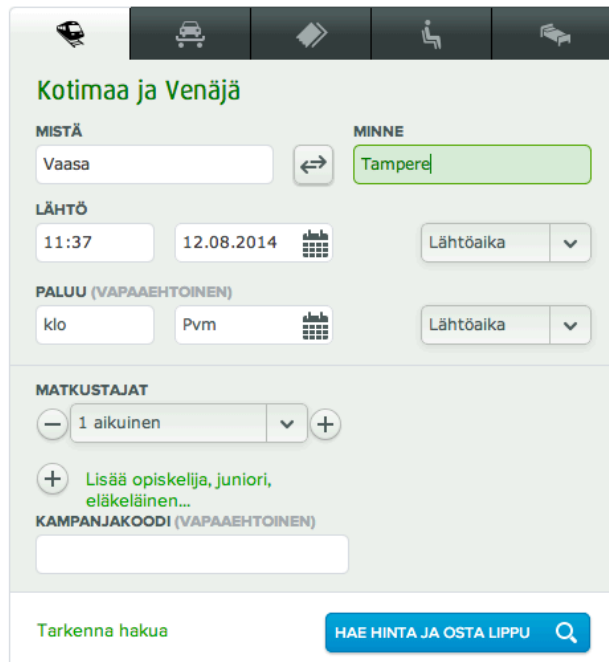
## 6 KÄYTETTÄVYYS

Käytettävyys on tärkeä osa jokaista käyttöliittymää, sekä tietokonesovelluksen että kaikkien laitteiden, esimerkiksi pyykinpesukoneen hallintapaneelin käyttöliittymää. Hyvin usein käytettävyyteen kiinnitetään liian vähän huomiota, jolloin syntyy sovelluksia, jossa on paljon ominaisuuksia, mutta on hyvin hankala käyttää ja tapahtumat tehdään monen vaiheen kautta.

Esimerkkinä voin käyttää VR:n verkkokauppaa, jonka hakulomakkeet ovat mielestäni liian epäselviä ja jossa on liikaa vaiheita. Lipun ostoon Vaasasta Tampereelle on yhteensä kuusi vaihetta (jos haluaa valita itse paikkansa), joihin lisätään vielä verkkopankin omat maksuvaiheet, mikä esimerkiksi Danske Bankilla sisältää kaksi vaihetta. Tämä tekee yhteensä kahdeksan eri vaihetta ennen kuin lipun saa tulostettua. Vaiheet ovat: hakukenttien täyttö, junavuoron valinta, junavuoron tarkastelu ja palveluiden valinta, paikan valinta, lippujen toimitustapa ja maksutapa, Danske Bankin tunnistautuminen, maksun hyväksyminen, (uudelleenohjaus VR:n sivuille), ilmoitus onnistuneesta ostotapahtumisesta ja lipun tulostus. Eri vaiheiden lomakkeetkin on suunniteltu hämmentävästi, ja käyttäjä joutuu tekemään ylimääräisiä klikkauksia saadakseen oikean vaihtoehdon (**Kuvio 3**).

Esimerkkinä vaikeasti luettavasta lomakkeesta on heti VR:n etusivulla matkustajatyypin valinta (**Kuvio 2**). Oletuksena se näyttää dropdown-valikon, jossa on yksi aikuinen. Jos käyttäjä haluaa esimerkiksi opiskelijalipun, hän joutuu tyhjentämään ensimmäisen dropdown-valikon ja sen jälkeen lisäämään matkustajatyypin, sitten hän valitsee uudesta dropdown-valikosta ”opiskelija” –matkustajatyypin ja lippujen määrän (**Kuvio 3**). Tämänkin saisi helpommaksi, jos esimerkiksi oletuksena näkymässä näkyy dropdown, jossa on matkustajatyypin ja sen vieressä lippujen määrä toisessa dropdown-valikossa (**Kuvio 4**). Tästä käyttäjä voisi suoraan vaihtaa oletuksen (aikuinen) opiskelijaksi ja lisätä määrää tarvittaessa. Tarvittaessa käyttäjä voisi lisätä uuden rivin, jossa on samat valinnat (lipputyypin ja määrä). Näin saisi vähennettyä klikkausten määrää. Hakulomakkeessa on myös kenttiä, joita käyttäjä ei välttämättä tarvitse lainkaan, kuten paluu-aika ja kampanjakoodi. Nämä voisi piilottaa ja tuoda esille vain tarvittaessa, kuten **Kuvio 3**:ssa olen esit-

tänyt, koska käyttäjä voi hämmentyä suuresta määrästä kenttiä ja luettuaan ne huomaakin niiden olevan epäolennaisia. Näin saataisiin helposti luettava ja ymmärrettävä lomake.



The image shows a web form for booking a train ticket. At the top, there is a navigation bar with icons for different transport modes: a train, a car, a bicycle, a wheelchair, and a bed. Below this, the title "Kotimaa ja Venäjä" is displayed in green. The form is divided into several sections:

- MISTÄ** (From): A text input field containing "Vaasa".
- MINNE** (To): A text input field containing "Tampere", highlighted with a green border.
- LÄHTÖ** (Departure): A time input field with "11:37", a date input field with "12.08.2014", and a "Lähtöaika" (Departure time) dropdown menu.
- PALUU (VAPAAEHTOINEN)** (Return - optional): A time input field with "klo", a date input field with "Pvm", and a "Lähtöaika" dropdown menu.
- MATKUSTAJAT** (Passengers): A section with a minus button, a text input field containing "1 aikuinen", a dropdown menu, and a plus button. Below this is a plus button and the text "Lisää opiskelija, juniori, eläkeläinen...".
- KAMPANJAKOODI (VAPAAEHTOINEN)** (Campaign code - optional): An empty text input field.

At the bottom left, there is a link "Tarkenna hakua" (Refine search). At the bottom right, there is a blue button with the text "HAE HINTA JA OSTA LIPPU" (Search price and buy ticket) and a magnifying glass icon.

**Kuvio 2**



**Kotimaa ja Venäjä**

**MISTÄ** Vaasa **MINNE** Tampere

**LÄHTÖ** 11:37 12.08.2014 Lähtöaika ▼

**PALUU (VAPAAEHTOINEN)** klo Pvm Lähtöaika ▼

**MATKUSTAJAT**

- 0 aikuista
- 0 junioria (6-16 v.)
- 1 opiskelija
- 0 eläkeläistä
- 0 siviilipalvelusmiestä

**KAMPANJAKOODI (VAPAAEHTOINEN)**

Tarkenna hakua **HAE HINTA JA OSTA LIPPU**

Kuvio 3

**Kotimaa ja venäjä**

**Mistä**  **Minne**

**Lähtö** 12:45 13.08.2014 Lähtöaika ▼

Meno-paluu

**Matkustajat**

Aikuinen 1

**Lisää matkustaja**

Kampanjakoodi

[Tarkenna hakua](#) **Hae hinta ja osta lippu**

Kuvio 4

## 6.1 Käyttäjakeskeinen suunnittelu

Kun käyttöliittymän suunnittelun lähtökohtana käyttää järjestelmän lopullista käyttäjää, voidaan varmistaa se, että käytettävyys on huomioitu ja järjestelmästä tulee helppokäyttöinen. Suunnittelun alkuvaiheessa voi tehdä esimerkiksi käyttäjätutkimusta ja tutkia, miten esimerkiksi muita samankaltaisia järjestelmiä on toteutettu ja mitä käyttäjät ovat niistä pitäneet.

Mealstonia suunnitellessa etsimme samankaltaisia sovelluksia ympäri maailman. Kovin montaa samanlaista ei löytynyt, mutta samankaltaisia sovelluksia oli josain maailmalla käytössä. Yhdestäkin sovelluksesta oli tehty video, jossa haasteltiin sovelluksen käyttäjiä eli ravintolan asiakkaita. He antoivat kaikki hyvää palautetta ohjelmistosta ja sanoivat, että ruokailukokemus muuttui paljon hauskemmaksi. Näiden benchmarkkauksien eli vertailuanalyysien perusteella saimme hyvää dataa ja varmuutta siihen, että sovellus voi muuttaa ruokailukokemusta ja itse ravintolan imagoa paremmaksi.

## 6.2 Käyttäjien toiminta

Käyttäjän toiminnan tutkiminen on tärkeä osa tuotekehitysprojektia. Se antaa palautetta siitä, kuinka erilaiset käyttäjät käyttävät järjestelmää, ja näiden tietojen pohjalta sitä voidaan kehittää.

Käyttäjien toimintaa ohjaa monta eri tekijää: koulutus, ikä, toiminnan rajoitteet ja tila, jossa järjestelmää käytetään (Sinkkonen, Kuoppala, Parkkinen & Vastamäki 2006, 29). Suunnittelun aikana kannattaa kiinnittää tämänlaisiin tekijöihin huomiota. Itse suunniteluun voi lisätä näiden käyttäjäryhmien näkökulmasta käyttökenaarioita eli käyttötarinoita. Käyttötarina eli ”use case” kertoo, kuinka tiettyyn käyttäjäryhmään kuuluva käyttäjä käyttää järjestelmää, ja se avaa suunnittelijalle itsellekin ideoita siitä, miten järjestelmä kannattaisi loppujen lopuksi toteuttaa, jotta saataisiin järjestelmä, joka on helppokäyttöinen kaikille käyttäjäryhmille. Seuraavassa kappaleessa on esimerkki käyttötarinasta, jonka pohjalta Mealston on rakennettu.

### 6.2.1 Käyttätarina Mealston-sovellukselle

Ravintolaan saapuu asiakas, joka menee suoraan pöytään istumaan. Hänelle tuodaan tablet-pc, jossa on ravintolan ruokalista. Asiakas selaa listaa pyyhkäisemällä näyttöä ylös tai alas. Hän valitsee mieleisensä ruoat painamalla jokaisen ruoan kohdalla tilaa-painiketta. Tämän jälkeen asiakas tarkastelee tilauksiaan painamalla tilaukset-painiketta sovelluksen vasemmasta ala-nurkasta. Hän haluaa huomauttaa, että on tietylle ruoka-aineelle allerginen, joten hän painaa tietyn ruoan kohdalla painiketta, jossa lukee ”lisätiedot”. Sovellus näyttää tekstikentän, johon käyttäjä voi kirjoittaa toiveensa. Kun toiveet ovat valmiit, hän painaa ”valmis” ja tämän jälkeen lähettää tilauksen painamalla ”lähetä”. Asiakas saa ilmoituksen onnistuneesta tilauksesta.

Tilaus etenee ravintolan järjestelmään. Tämän järjestelmän tilaus-näkymään tulee uusi tilaus. Ravintolan kokki kuittaa tilauksen aktiiviseksi. Kun ruoat ovat valmiit, kokki kuittaa ne tilausnäkyvästä, jolloin ne siirtyvät tarjoilijan näkymään. Tarjoilijan näkyvästä tarjoilija huomaa, mikä tilaus on valmis toimitettavaksi. Tarjoilija noutaa sen keittiöltä ja toimittaa oikeaan pöytään. Tämän jälkeen tarjoilija kuittaa ruoan toimitetuksi, jolloin tilaus jää odottamaan maksua näkymään. Kun ravintolan asiakas on valmis, hän painaa ruokalista-sovelluksestaan ”tilaa lasku” – painiketta. Tästä lähtee ilmoitus tarjoilijan tilaus-näkymään. Kun tarjoilija toimittaa laskun ja se on maksettu, hän kuittaa tilauksen maksetuksi. Tällöin tilaus häviää näkyvästä ja tallentuu tietokantaan muistiin.

### 6.3 Käytettävyyden psykologia

Käyttöliittymien suunnitteluun on ollut tietynlaisia sääntöjä tai pikemmin ohjeita niin kauan, kun tietoteknisiä käyttöliittymiä on suunniteltu. Näille ohjeille on olemassa hyvät perusteet: psykologia. Ihmisen käyttäytymisen havainnointi ja biologiset perusteet ovat suuria vaikuttajia siihen, kuinka hyvin ihminen käyttää käyttöliittymää.

### 6.3.1 Biologisia vaikuttajia

Biologisia vaikuttajia voi olla esimerkiksi ihmisen silmä. Siinä, missä kamera kuvauttaessaan ottaa koko sensorin täydeltä tarkkaa kuvaa, ihmisen silmän tarkkuus kohdentuu vain pienelle alueelle, eli verkkokalvon keskikuoppaan. Sitä ympäröivä alue on nimeltään periferia. Verkkokalvon keskikuopassa on noin 158000 tappisolua, kun sitä ympäröivässä periferiassa on vain noin 9000 tappisolua. Tämä tarkoittaa sitä, että vain se alue, johon ihminen kohdistaa katseen, on tarkkaa ja muu kuva on epätarkkaa (Johnson 2014, 49).

Yllä olevaa esimerkkiä ajatellen yhtenä esimerkkinä asia, kuten varoitustekstien sijoittelu, on hyvinkin tärkeä suunnitella tarkasti. Jos käyttäjä on esimerkiksi kirjautumassa verkkopalveluun ja antaa vahingossa väärän salasanan tai muuten kirjautumistiedoissa on vikaa, pitäisi varoitustekstien tulla näkyviin paikkaan, johon käyttäjä juuri kirjautuessaan tarkentaa katseensa eli kirjautumispainikkeen ympärille. Jos varoitusteksti ilmestyy esimerkiksi kauas sivun ylälaitaan, käyttäjä ei heti huomaa tekstiä ja voi hämmentyä. Jos silti halutaan, että varoitusteksti tulee sivun yläreunaan, sille voidaan tehdä esimerkiksi lyhyt mutta huomattava animaatio, joka kiinnittää käyttäjän huomion (Johnson 2014, 55-57). Tämä taas johtuu siitä, että silmän ääreisnäkö (eli verkkokalvon keskikuopan ympäröivän alueen, periferian näkö) on hyvä havaitsemaan liikettä, joka johtaa silmän vaistomaiseen tarkentamiseen sille alueelle, jossa liikettä tapahtui (Johnson 2014, 60).

Näillä kahdella sekä lukuisilla muilla biologisilla vaikuttajilla, kuten ihmisen muisti ja värinäkö, on hyvin iso osa helppokäyttöisen ja helposti opittavan käyttöliittymän toteuttamisessa. Kun suunnittelija ottaa huomioon mahdollisimman tarkasti, kuinka ihminen käyttäytyisi tietyn käyttöliittymän elementin kanssa, käyttöliittymästä saadaan varmuudella sellainen, joka on helppo omaksua ja jota on helppo ja luonnollista käyttää. Sovellusta, joka ei vaadi sen kummempaa opettelusta, käytetään jatkossakin joten käyttäjien tyytyväisyys on taattu.

## 6.4 Käytettävyyden testaus

Käytettävyydestaus on menetelmä, jossa arvioidaan järjestelmän käytettävyyttä (Koskinen 2005). Kappaleessa 5.4.1 tein pienimuotoisen käytettävyyden testauksen VR:n lippujärjestelmälle. Käytin lomaketta ja sen jälkeen raportoin, mitä eri vaiheita siinä on. Lopuksi tein ehdotuksen käytettävyyden parantamiseksi oman raporttini pohjalta.

Tavoitteena testauksessa on löytää järjestelmän käytössä ongelmakohtia, joita voidaan korjata jatkokehityksessä. Testauksen aikana testaaja kirjoittaa raporttia testatuista asioista ja samalla kerää yhteen järjestelmän hyvät ja huonot puolet käytettävyyden kannalta. Testauksen lopputulokset käydään läpi kehitystiimin kanssa, jotta käytettävyydsiantuntija voi selittää, minkälaisia ongelmia hän järjestelmässä havaitsi ja keskustella siitä, kuinka niitä voitaisiin mahdollisesti parantaa. Ongelmakohdat kannattaa aina perustella, miksi se on käytettävyyden kannalta huono ratkaisu. Alla käytettävyyden testaustaulukko, jonka laadin eräästä sovelluksesta.

Taulukko 1

Toiminto	Huomautukset	Korjauskehotus
<b>”Varastoryhmä”-taulukko</b>	Varastoryhmien painikkeet vaikeasti painettavia mobiililaitteilla.	Ikonit erotellaan selkeiksi napeiksi tai ne erotetaan suuremmalla marginaalilla.
	Varastoryhmien muokauspainikkeiden toiminnot epäselviä.	Painikkeille ohjeteksti.
<b>”Näytä kaikki tuotteet”-painike</b>	”Uusi ryhmä” -painike vaihtaa paikkaa seuraavassa näkymässä, hämentää käyttäjän.	”Uusi ryhmä” -painike pitää paikkansa, eikä vaihda keskenään paikkaa alemman painikkeen kanssa.

<b>Uusi tapahtuma</b>	Rakennuksen alkupäivämäärän kalenteri alkaa kuluvasta päivästä.	Kalenteri hyppää suoraan käyttäjän aiemmin syöttämään tapahtuman alkupäivämäärään, joka on kalenterissa korostettu.
	Kaikki ”loppupäivämäärät” avaavat kalenterin kuluvan päivän ja kuukauden kohdalta.	Kalenteri avautuu suoraan alkupäivämäärän kohdalta, jos esimerkiksi käyttäjä kirjaa tapahtuman alkamaan kuukauden päähän, loppupäivämäärien kalenteri hyppää suoraan sinne.

## 7 MEALSTON-SOVELLUKSEN KÄYTTÖLIITTYMÄ

Mealstonin suunnittelun alkuvaiheessa suunniteltiin ensin, että mitä kaikkea sovelluksella voi tehdä. Tämän jälkeen alkoi hahmottua sovelluksen kokonaisuus ja sisältö. Tuli selväksi, että kokonaisuus muodostuu kahdesta eri sovelluksesta (verkko- ja mobiilisovellukset). Näiden pohjalta aloin suunnittelemaan käyttöliittymää. Suunnittelun aloitin mockupeista. Mockup-kuvien jälkeen tein niistä layoutit Adobe Photoshop -ohjelmalla. Näiden layout-kuvien pohjalta sovellus koodattiin.

### 7.1 Staff-puolen käyttöliittymä

Koska staff-puolta käytetään pääasiassa tietokoneella, sen suunnittelun lähtökohta oli tehdä siitä mahdollisimman yksinkertainen ja sellainen, jossa ei tarvitse tehdä asioita monen klikkauksen kautta, mikä vie aikaa.

Aloitin suunnitteluprosessin etsimällä inspiraatiota erilaisista verkkosovelluksista. Pinterest on tähän hyvä palvelu. Sieltä löytää kuvia lähes asiasta kun asiasta. Hain esimerkiksi hakusanoilla ”web application”, jolloin sain paljon erilaisia verkkosovelluksia laidasta laitaan. Aloin jo hahmottaa konseptia siitä, minkälaisen sovelluksen tulen suunnittelemaan.

#### 7.1.1 Navigointi

Käyttöliittymän suunnittelun aloitin navigaatiosta. Sovelluksen navigaatiopalkki koostuu kahdesta elementistä: header- ja itse navigaatio-palkki (**Kuvio 5**). Header on aina näkyvä sivun yläreunassa ja siinä on sovelluksen logo ja niin sanottu ”hampurilaispainike”. Hampurilaispainike on saman näköinen, kun mobiilisovelluksissa on tullut tutuksi. Tämä painike tuo esiin menun. Se on yleistynyt myös verkkosovelluksissa ja tavallisilla verkkosivuilla, joten päätin itsekkin käyttää sellaista. Painiketta painamalla esiin tulee headerin alle sovelluksen navigointipalkki. Navigoinnissa on seuraavat linkit: etusivu, ruokalistat, tilaukset, tili sekä uloskirjaus. Linkit ovat ympyröitä, joiden sisällä on sekä sivun nimi että kuvake. Sivun päänavigointi eli etusivu, ruokalistat ja tilaukset sijoittelin palkin vasempaan reu-

naan, koska ne ovat tärkeimmät ominaisuudet. Tilin asetukset sekä uloskirjauksen sijoitin palkin oikeaan reunaan. Uloskirjauksen linkin erotin eri värillä. Myös aktiivisen sivun linkki eli sen sivun, joka on juuri auki, on aktiivinen. Aktiivisuuden erotin sinisellä reunuksella ja täysin valkoisella tekstillä ja ikonilla. Passiivisilla linkeillä ei ole reunusta ja niiden tekstillä ja ikonilla on hieman läpinäkyvyyttä.

Navigointipalkki pysyy auki niin kauan, kunnes käyttäjä vie hiiren sen päältä pois. Jos käyttäjä vahingossa vie hiiren pois sen päältä, se ei mene kiinni, koska siinä on ajastin. Tällä eliminoidaan käyttäjän turhautuminen huonosti toimiviin elementteihin.



**Kuvio 5**



**Kuvio 6**

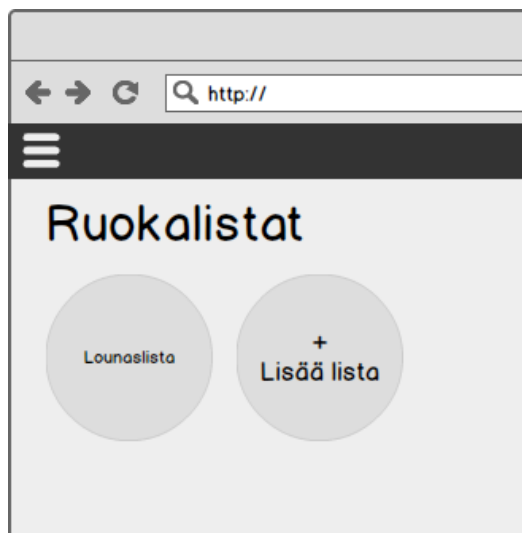
### 7.1.2 Ruokalistat

Navigoinnin jälkeen aloin miettimään sisältöä. Sisältönä tässä sovelluksessa on käyttäjän luomat ruokalistat ja niiden sisältö.

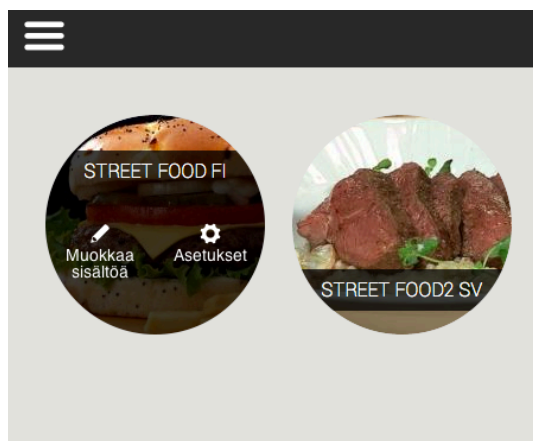
”Ruokalistat” –sivulla käyttäjä näkee ensimmäisenä tyhjän sivun, jossa on painike uuden ruokalistan lisäämiseen (mikäli ei ole aiemmin luotuja ruokalistoja). Ruokalistat näkyvät ympyröinä, joiden sisällä on sekä ruokalistan kansikuva että nimi. Kun käyttäjä vie hiiren tämän ympyrän päälle, siihen nousee tummalla taustalla kaksi linkkiä: ”asetukset” sekä ”muokkaa sisältöä” (**Kuvio 8**). Jos käyttäjä painaa ”asetukset” –linkkiä, näyttöön aukeaa modal-ikkuna, jossa on ruokalistan asetukset. Näistä asetuksista käyttäjä voi muokata ruokalistan nimeä, määrittää onko se



etätilauslista vai käytetäänkö sitä ravintolan sisällä tablet-pc:llä, vaihtaa kansikuvaa ja määrittää kellonajat, jolloin ruokalista on käytössä. Ruokalistan luonnissa asetetut asetukset ovat valmiina asetuksissa. Tämä sama asetusikkuna tulee esiin, kun käyttäjä luo uutta ruokalistaa. Tällöin asetukset ovat tyhjiä.



**Kuvio 7**

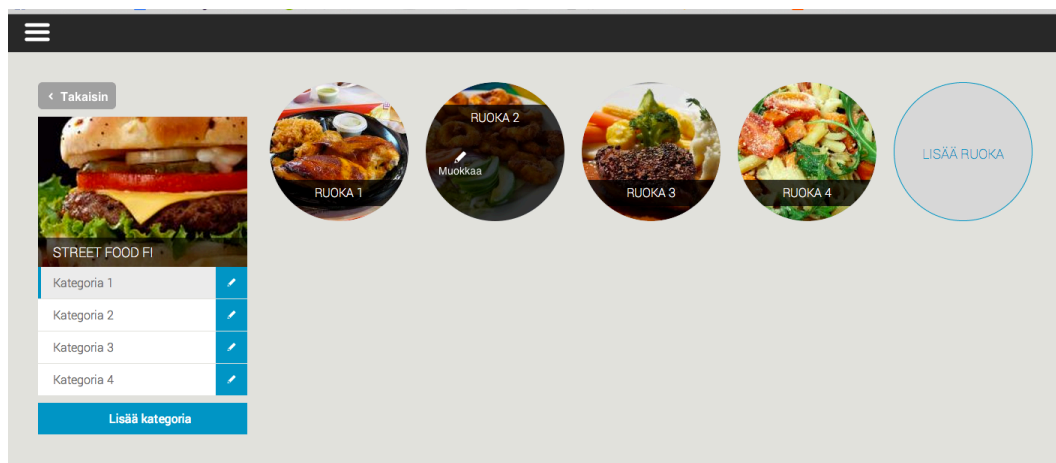


**Kuvio 8**

**Kuvio 9**

### 7.1.3 Ruokat

Kun käyttäjä valitsee ruokalistan kohdalta ”muokkaa sisältöä”, sovellus ohjautuu uudelle sivulle. Tällä sivulla näkyy valitun ruokalistan sisältö (**Kuvio 10**). Ruokat on jaoteltu kategorioittain. Jos ruokalista on tyhjä, käyttäjä tekee ensin kategorian. Tämän jälkeen kategorian sisälle lisätään ruokia. Ruokien ulkonäkö ja lisäys on samanlainen kun ruokalistoilla (**Kuvio 11**). Ruokien luontipainike näkyy tyhjässä kategoriassa aina ensin ja viimeisenä silloin, kun siellä on jo sisältöä. Ruokaa lisätessä käyttäjä painaa painiketta ”Lisää ruoka”. Tämä tuo näyttöön modal-ikkunan (**Kuvio 12**), jossa on ruoan asetukset, samalla tavalla kuin ruokalistan luonnissa. Modal-ikkunan lomakkeeseen käyttäjä lisää ruoan nimen, selityksen, kuvan ja hinnan. Näitä asetuksia pääsee muokkaamaan myöhemmin ja ruokat sekä itse kategoriat on myös mahdollisuus poistaa.

**Kuvio 10****Kuvio 11**

**Kuvio 12**

#### 7.1.4 Tilaukset

Ruokalistasivun jälkeen siirryin tilauslistaus –sivulle. Tämän sivun suunnittelussa tuli ottaa huomioon se, että se on näkyvässä ravintolan keittiössä. Tämän sivun käyttöön ravintolaa neuvotaankin käyttämään kosketusnäyttöistä tietokonetta, koska se on keittiössä käytännöllisempi kuin hiiri ja näppäimistö. Täten myös elementit piti suunnitella käytettäväksi kosketuksella. Piti ottaa myös huomioon se, että näyttö ei välttämättä ole aina lähellä kokkia, joten elementtien ja tekstien pitää olla mahdollisimman selkeitä lukea. Näkymiä sivulla on kaksi: kokkien ja tarjoilijoiden näkymä. Kokeille näkymään tulee kaikki uudet ruokatilaukset, kun taas tarjoilijoille tulee tarjoiluvalmiit ruoat ja ruoat, jotka on merkattu tarjoilijan valmistettavaksi.

Yhdessä tilauksessa näkyy tilauksen otsikko, tilatut ruoat ja painike kokeille. Otsikkopalkki on selkeästi erotettu tummalla taustalla, ja siinä lukee joko tilaustunniste tai pöydän numero, josta tilaus on tehty. Näistä ensimmäinen on tarkoitettu etätilausruokalistalta tehtyihin tilauksiin. Otsikkopalkin alle lisäsimme kehitysvaiheessa tilauksen tilausajan, jotta ravintola näkee, kuinka kauan tilaus on odottanut valmistumista. Otsikkopalkin ja ajan alla on luettelo tilatuista ruoista selkeästi isommalla kirjasinkoolla ja rivivälillä. Näiden alla on painike. Tämä painike

on oletuksena sininen, kun uusi tilaus tulee. Painikkeessa lukee ”valmista”, joka tarkoittaa sitä, että tilaus odottaa valmistusta. Kun kokki painaa painiketta, ruoille tulee jokaiselle oma ”check” –painike (**Kuvio 13**). Tällä painikkeella on kaksi tarkoitusta: osoittaa, että tilaus on valmistuksen alla, jotta muut kokit eivät tee samoja ruokia sekä valmiiden ruokien valmiiksi merkkäminen. Kun kokki merkkää yhden ruoan valmiiksi, se siirtyy kokkien näkymästä tarjoilijoiden näkymään. Kun kaikki ruoat on merkattu valmiiksi, tilaus häviää näkymästä. Kokki voi myös halutessaan merkata kaikki ruoat kerralla valmiiksi isolla painikkeella elementin pohjalla. Tarjoilijan näkymässä tilaukset näkyvät niin kauan, kunnes tarjoilija on merkannut ne toimitetuksi.

TILAUS 200	
Tilattu: 2014-08-10 13:26:58	
RUOKA 2	✓
RUOKA 2	✓
RUOKA 1	✓
RUOKA 2	✓
RUOKA 2	✓

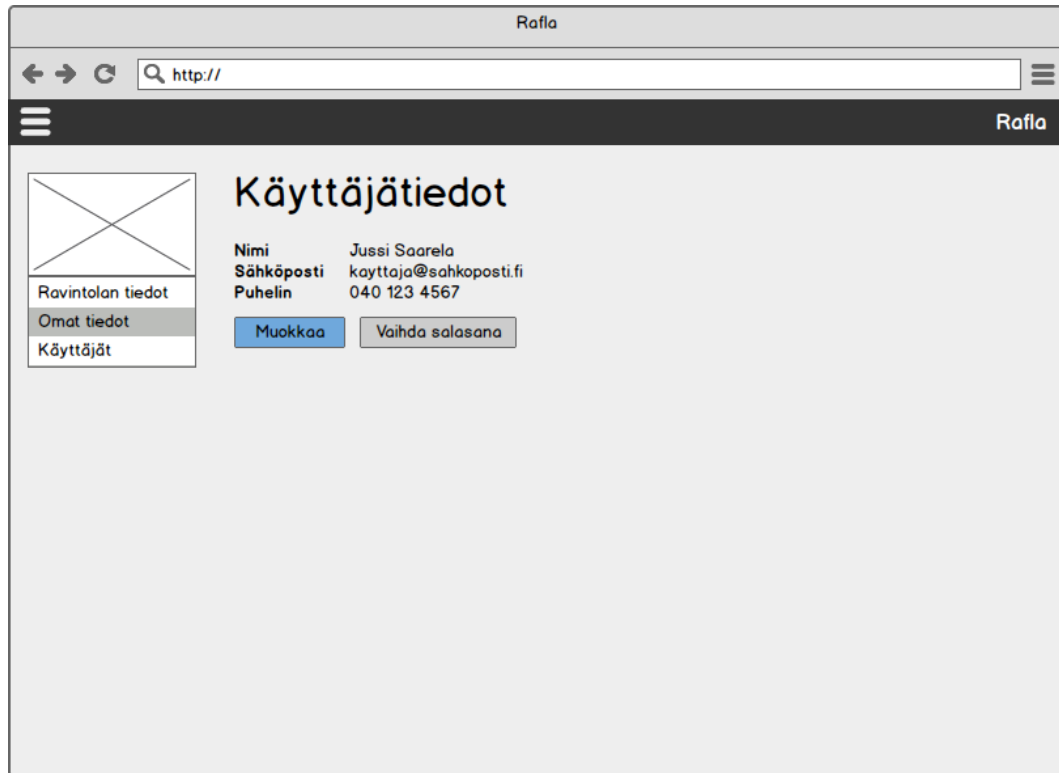
**Kuvio 13**

### 7.1.5 Tilin asetukset

Tilin asetukset –sivulla (**Kuvio 14**) käyttäjä voi tarkastella ja muokata tilin sekä ravintolan tietoja. Tiedot on jaettu kahteen osaan: tilin tiedot ja ravintolan tiedot. Eri välilehtien välillä navigoidaan samanlaisella elementillä kuin ruokalista-sivulla kategoriat on jaettu. Tietojen alla on muokkaus-painike, josta tietoja voi muokata.

Tällä sivulla on myös uusien käyttäjien lisääminen. Kun ravintola rekisteröityy, tämä käyttäjätunnus on automaattisesti admin-käyttäjä. Admin-käyttäjä ainoastaan voi lisätä uusia käyttäjiä. Nämä uudet käyttäjät lisätään samalle ravintolan ID:lle, joilla se on luotukin. Uudelle käyttäjälle annetaan nimi, sähköpostiosoite (jota

käytetään sisäänkirjauksessa) ja valitaan, onko käyttäjä tarjoilija vai kokki. Tämä viimeinen valinta vaikuttaa siihen, mitä käyttäjä näkee tilaus-näkymässä.



**Kuvio 14**

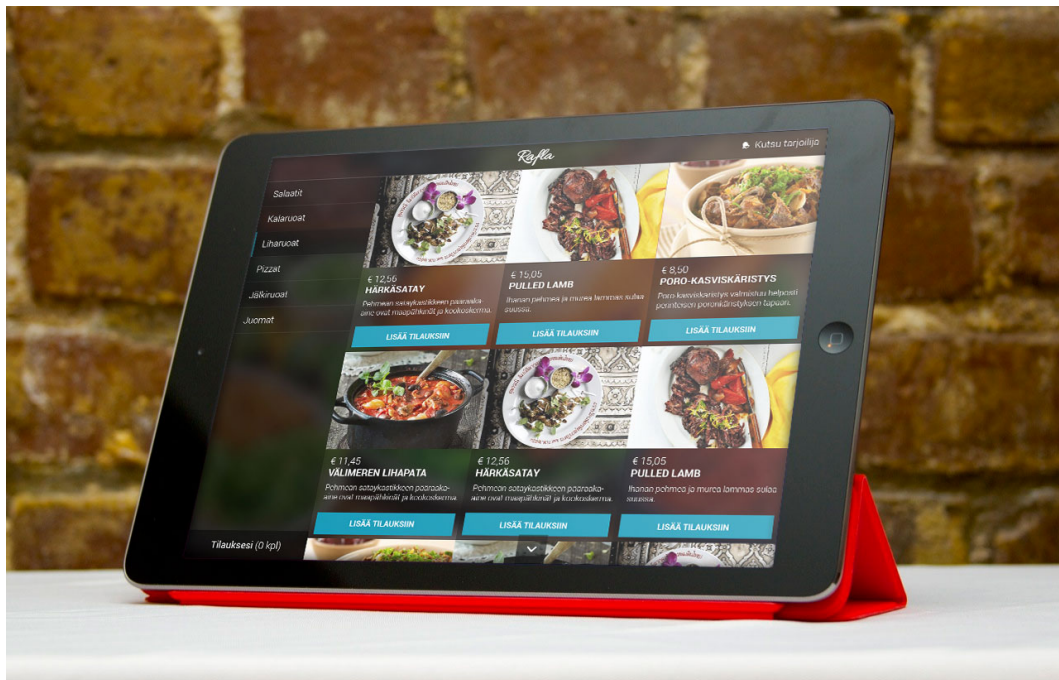
## 7.2 Customer-sovellus

Ruokalistasovelluksen suunnittelussa tuli ottaa huomioon se seikka, että se tulee käyttöön sekä verkkoon että tablet-pc:lle ravintolan saliin. Verkkotilaajat eivät välttämättä ikäryhmältään ole kovin vanhoja, mutta ravintolan asiakkaina voi olla hyvinkin laaja ikäryhmä. Vanhemmat käyttäjät eivät välttämättä ole koskaan käyttäneet tablet-laitteita joten käytettävyydeltään tämän sovelluksen käyttöliittymä piti olla mahdollisimman yksinkertainen ja helppo oppia.

Näkymiä tässä sovelluksessa on vain kolme: kielen valinta, ruokalista ja tilauksen vahvistus. Kielen valintanäkymässä näkyy vain saatavilla olevat kielet. Nämä kielet määrittävät, millä kielellä sovellus näyttää painikkeiden ja elementtien tekstit ja ruokalistan sisällön. Kielen valinta aloittaa myös uuden istunnon. Kun kieli on

valittu, näkymään tulee koko ruokalista (**Kuvio 15**). Vasemmalla näytössä näkyvät ravintolan logo, ruokalistan kategoriat sekä painike tilausten tarkastelemiseen. Näiden vieressä, isommalla palstalla näkyy itse ruoat. Jokaisella ruoalla on kuva, otsikko, hinta, selite ja osto-painike. Kaikki kategoriat näkyvät allekkain. Painettaessa kategorialinkkiä vasemmalta, sovellus vierittää näkymän valitun kategorian kohdalle ja näyttää sen ruoat. Ruokien välissä ovat kategorioiden otsikot, jotka jaottelevat ne omiin kategorioihinsa. Kun kaikki ruoat näkyvät samalla sivulla, minimoidaan tarve hakea dataa tietokannasta, jolloin sovelluksen käyttö on nopeampaa. Tablet-sovelluksessa on myös painike tarjoilijan kutsumiselle.

Tilaukset-näkymä on ostoskori. Täältä käyttäjä näkee, mitä hän on lisännyt ostoskoriin ja voi tarvittaessa kirjoittaa tietylle ruoalle lisätietoja kuten jonkun ruoka-aineen pois jättäminen. Ruoat voi myös poistaa ostoskorista. Näkymässä näkyy myös tilausten yhteissumma. Jos tilaukset tehdään verkossa, näkymässä on myös kenttä käyttäjän sähköpostiosoitteelle sekä maksutavan valinta. Sähköpostiosoitetta tarvitsee tilauksen vahvistamiseen ja tilaustunnuksen lähettämiseen, ja maksutavan valinta on sitä varten, että käyttäjä voi valita, maksaako hän heti verkossa vai paikan päällä. Tablet-sovellus-versiossa maksutavan valintaa tai sähköpostivahvistusta ei tarvita. Tietenkin näkymässä näkyy aina myös tilauksen lähetyspainike. Tilauksen lähdettyä, näkymään tulee ilmoitus tilauksen onnistumisesta, ja sovellus sulkee istunnon ja palaa takaisin kielivalintaan.



Kuvio 15



## 8 TAVOITTEET JA NIIDEN TOTEUTUMINEN

Ensimmäisenä tavoitteena oli luoda sovellus, jonka ravintola voi ottaa helposti ja itsenäisesti käyttöön ilman it-tukea. Ohjelmiston kokonaisuudesta saimme rakennettua sellaisen, että sinne on helppo rekisteröityä ja tarvittavat mobiili-sovellukset on helppo ladata Google Play –sovelluskaupasta ja asentaa tabletpc:lle. Ruokalistan näkyviin saamiseen sovellukseen ei tarvitse muuta tehdä, kuin kirjautua sisään, ja sovellus hakee automaattisesti oikean ruokalistan.

Toinen tavoite oli käytettävyys. Staff-sovelluksesta suunnittelin mahdollisimman yksinkertaisen ja vähän työtä vaativan järjestelmän. Sain minimoitua vaiheiden ja klikkausten määrän, ja lomakkeet ovat selkeitä ja helposti ymmärrettävissä. Customer-sovelluksesta suunnittelin myös sellaisen, jossa on mahdollisimman vähän vaiheita. Tein sovelluksesta selkeän ja erotin tärkeät elementit, kuten tilausnapit taustasta ja muusta sisällöstä helposti nähtäviksi.

Lopputuloksena sain rakennettua sovelluksesta hyvinkin helppokäyttöisen ja sellaisen, jolla ei mene turhaa aikaa hukkaan sen hitauden tai monimutkaisuuden takia. Ensimmäisen asiakkaan kommentti järjestelmästä oli ”Vau, todella yksinkertainen!” ja ”Paljon yksinkertaisempi, kuin mitä odotin!”. Tämä pelkästään kertoo sen, että onnistuin käyttöliittymän suunnittelussa ja rakentamisessa.

## 9 TULEVAISUUS

Mealstoniin on tulevaisuudessa tarkoitus lisätä uusia ominaisuuksia. Alkuun oli tärkeä saada perusominaisuudet kuntoon, jotta ohjelmiston voi julkaista. Jos alkuun käyttää turhan paljon resursseja siihen, että ohjelmistoon lisää kaikki ominaisuudet joita on mielessä, niin se voi koitua omaksi harmiksi, jos sovellus ei menesty. Jos sovellukseen kehittää vain ominaisuudet, jotka ovat sille olennaisia ja julkaisee niin sanotun ”perusversion”, siitä on helpompi jatkaa kehittämistä, kunhan sovellus alkaa menestymään. Tämä vie vähemmän resursseja ja, jos sovellus ei menesty, häviö ei ole niin suuri. Tässä opinnäytetyössä en kerro, minkälaisia ominaisuuksia Mealston tulee tulevaisuudessa sisältämään suojatakseni mahdollista kopiointia vastaan.

## 10 YHTEENVETO

Mealstonin käyttöliittymästä tein lopulta yksinkertaisen ja helpon käyttää. Sain eliminoitua ylimääräiset klikkaukset ja elementit on selkeitä ja helposti luettavia. Koodausvaiheessa käyttöliittymään ei juurikaan tullut muutoksia. Joitain ominaisuuksia tuli lisättyä jälkikäteen, jota ei tullut mietittyä alussa. Tämä johti siihen, että kesken koodauksen tiettyä ominaisuutta joutui miettimään hyvinkin kauan, ennen kuin sille sai miellyttävän lopputuloksen. Tämä opetti sen, että ennen koodausta ja käyttöliittymän suunnittelua kaikki ominaisuudet pitäisi käydä läpi henkilön/henkilöiden kanssa, jotka ohjelmistoa koodaavat. Jos käyttöliittymäsuunnittelija suunnittelee järjestelmän omasta mielestään toimivaksi, mutta ilman koodaajan näkökantaa, voi koodauksen aikana syntyä konflikteja sen suhteen, miten se on mahdollista toteuttaa. Tämän takia koodaajankin on hyvä olla alusta alkaen suunnittelemassa järjestelmää ja sen käyttöliittymää, jotta saadaan mahdollisimman helppo työprosessi eikä synny ylimääräisiä konflikteja ja kompromisseja.

## 11 LÄHTEET

Apple. 2007. Keynote.

Johnson, J. 2014. Design with the Mind in Mind. Watlham, MA. Morgan Kaufmann

Sinkkonen, I. Kuoppala, H. Parkkinen, J & Vastamäki, R. 2006. Käytettävyyden Psykologia. Helsinki. Edita Publishing Oy.

Koskinen, J. 2005. Käytettävyydestaus. Tampere. Tampereen yliopisto. Viitattu 22.9.2014. <http://www.cs.uta.fi/usabsem/luvut/13-Koskinen.pdf>

Bank, C. 2014. The Guide to Wireframing. Viitattu 22.9.2014. <http://www.uxpin.com/guide-to-wireframing.html>