

Emilia Ryhänen

## **MOBIILISOVELLUKSEN KÄYTTÖLIITTYMÄN SUUNNITTELU**

# **MOBIILISOVELLUKSEN KÄYTTÖLIITTYMÄN SUUNNITTELU**

Emilia Ryhänen  
Opinnäytetyö  
Kevät 2019  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma, laite- ja tuotesuunnittelu

---

Tekijä: Emilia Ryhänen

Opinnäytetyön nimi suomeksi: Mobiilisovelluksen käyttöliittymän suunnittelu

Opinnäytetyön nimi englanniksi: User Interface Design for a Mobile Application

Työn ohjaaja: Eino Niemi

Työn valmistumislukukausi ja -vuosi: Kevät 2019

Sivumäärä: 29

---

Tämän opinnäytetyön aiheena oli mobiilisovelluksen käyttöliittymäsuunnittelu ottaen huomioon kohdekäyttäjät ja mobiilisovelluksen käytettävyys. Tavoitteena oli perehtyä mobiilisovelluksen suunnitteluun, käytettävyyteen ja käytettävyydestäukseen yleisesti sekä käytettävyyteen mobiilisovelluksissa ja suunnitella mobiilisovelluksen käyttöliittymä. Tällä työllä ei ollut toimeksiantajaa, vaan se perustui omaan mobiilisovellusideaan.

Opinnäytetyö on toteutettu etsimällä aiheeseen liittyvää tietoa internetistä sekä lukemalla alan kirjallisuutta. Ensimmäiseksi opinnäytetyössä käydään läpi erilaisia käyttöliittymävaihtoehtoja, mobiilisovelluksen käyttöliittymäsuunnittelua sekä käyttöliittymäsuunnittelun vaiheita. Tämän jälkeen opinnäytetyössä käsitellään käytettävyyttä, käytettävyydestäusta sekä mobiililaitteen käytettävyydestäusta. Lopuksi opinnäytetyössä käydään läpi työn toteutuksen eri vaiheita käyttäjävuokaavion, luonnosten sekä mock-up-kuvien kautta. Mock-up-kuvalla tarkoitetaan visuaalista mallinnusta sovelluksesta.

Työn tuloksena valmistui mobiilisovelluksen käyttöliittymän visuaalinen suunnitelma. Itse mobiilisovellusidea on salassa pidettävää tietoa, joten mobiilisovelluksesta sekä sen suunnittelusta kerrotaan hyvin rajallisesti. Kirjoittajalla oli aikaisempaa kokemusta käytettävyydestäuksesta sekä käytettävyydestä, mutta ei käyttöliittymäsuunnittelusta. Työn on tehty käyttäen Colour Hunt -verkkosivua sekä Lucidchart- ja SketchApp-tietokoneohjelmistoja. Työn tekeminen oli hyvin opettavaista ja palkitsevaa, koska opinnäytetyö opetti paljon uusia asioita mobiilisovelluksen käyttöliittymäsuunnittelusta.

---

Asiasanat: käyttöliittymät, mobiilisovellukset, käytettävyys, käytettävyydestäusta

# ABSTRACT

Oulu University of Applied Sciences  
Information Technology, Option of Equipment and Product design

---

Author: Emilia Ryhänen  
Title of thesis: User interface Design for a Mobile Application  
Supervisor: Eino Niemi  
Term and year when the thesis was submitted: Spring 2019  
Pages: 29

---

The aim of this thesis was to design user interface for mobile application considering usability and usability testing. This thesis was based on authors own mobile application idea and the work doesn't have a client. The mobile application itself is confidential information, so there is very limited information about the mobile application and its design.

The main sources of this thesis were the internet and literature of the field. At the beginning this thesis examines the fundamentals of mobile application user interface design. After that the thesis examines usability, usability testing and mobile device usability testing. Last part of this thesis was to design the authors own mobile application idea.

The result of this work was the visual design of the mobile application interface. The author had previous knowledge of usability and usability testing, but user interface design was unfamiliar. The work was very educational and rewarding. The work was done by using Lucidchart, Colorhunt, and SketchApp applications.

---

Keywords: usability, usability testing, user interfaces, mobile application design

# SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
1 JOHDANTO	6
2 KÄYTTÖLIITTYMÄ	7
2.1 Mobiilinettisivu vai mobiilisovellus	8
2.2 Mobiilisovelluksen käyttöliittymäsuunnittelu	9
2.2.1 Kognitiivisen kuorman minimointi	9
2.2.2 Käyttäjän huomiointi	11
2.2.3 Sisällön optimointi pienelle näytölle ja kosketukselle	11
2.2.4 Ilmoitukset	12
2.3 Käyttöliittymäsuunnittelun vaiheet	13
3 KÄYTETTÄVYYS	14
3.1 Käytettävyystestaus	14
3.1.1 Testin tavoitteet ja testisuunnitelmat	15
3.1.2 Testikäyttäjä	16
3.1.3 Testitehtävät	16
3.1.4 Testin eteneminen	17
3.1.5 Testausmenetelmät	18
3.1.6 Käytettävyyslaboratoriot	21
3.2 Mobiililaitteen käytettävyystestaus	22
4 TYÖN TOTEUTUS	23
4.1 Käyttäjävuokaavio	23
4.2 Raakapiirros sekä rakenteen, kuvioiden ja värien valinta	24
4.3 Mock-up-kuvat	25
5 YHTEENVETO	27
LÄHTEET	28

# 1 JOHDANTO

Tämä opinnäytetyö on tehty Oulun ammattikorkeakoulussa osana tietotekniikan tutkinto-ohjelmaa. Idea opinnäytetyön aiheeseen lähti omasta kiinnostuksesta lisätä ihmisten terveyttä ja hyvinvointia. Tällä opinnäytetyöllä ei ole toimeksiantajaa, vaan se perustuu omaan mobiilisovellusideaan.

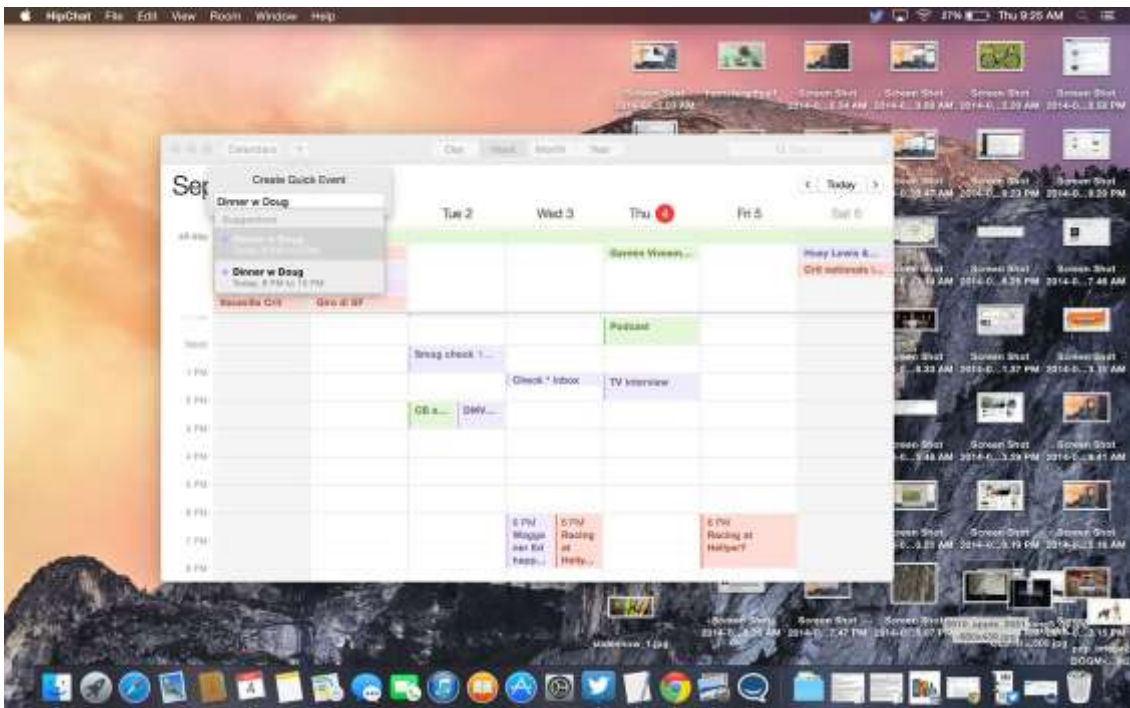
Tämän opinnäytetyön tarkoituksena on perehtyä mobiilisovelluksen käyttöliittymäsuunnitteluun sekä käytettävyyteen teoriassa ja suunnitella mobiilisovelluksen käyttöliittymä. Työssä on otettu huomioon käytettävyys yleisesti ja käytettävyys mobiilisovelluksen käyttöliittymäsuunnittelussa. Itse mobiilisovellusidea on salassa pidettävää tietoa, joten mobiilisovelluksesta sekä sen suunnittelusta kerrotaan hyvin rajallisesti.

Mobiilisovelluksen suunnittelussa on otettava huomioon monia asioita, joista mielestäni tärkein on käytettävyys. Jos mobiilisovelluksen käyttö on vaikeaa ja hidasta, tulee käyttäjä poistamaan sovelluksen ja etsimään sovelluskaupasta uuden vaihtoehtoisen sovelluksen ja kokeilee, onko sen käytettävyys parempi kuin aikaisempi sovellus. Käytän päivän aikaan useita eri mobiilisovelluksia ja törmään ajoittain käytettävyysongelmiin. Yleensä käytettävyysongelmia on uusissa sovelluksissa, joiden käytettävyyteen ei ole panostettu tarpeeksi. Käytettävyyteen vaikuttavat myös käyttäjä ja käyttäjän tarpeet, jotka on hyvä ottaa huomioon mobiilisovellusta suunnitellessa.

## 2 KÄYTTÖLIITTYMÄ

Käyttöliittymä on laitteen, ohjelmiston tai minkä tahansa muun tuotteen osa, jonka kautta käyttäjä käyttää laitetta. Esimerkiksi mobiililaitteessa sillä tarkoitetaan sitä osaa, jonka kautta käyttäjä käyttää sovellusta tai laitetta. Jotta käyttäjä voisi toimia yhteistyössä laitteen kanssa, hänen on paitsi nähtävä käyttöliittymä mutta myös osattava tulkita sen sanallisia, kuvallisia ja äänellisiä viestejä. (1.)

Käyttöliittymiä on kahdenlaisia: graafisia käyttöliittymiä sekä komentoliittymiä. Graafinen käyttöliittymä (kuva 1) koostuu käyttöliittymäelementeistä, joita käytetään näppäimistön lisäksi osoitinlaitteella tai kosketusnäytöllä. Käyttöliittymäelementeillä tarkoitetaan ikkunoita, valikoita, valintanappeja, valintaruutuja sekä kuvakkeita. (2.)



KUVA 1. Graafinen käyttöliittymä (4)

Komentoliittymällä (kuva 2) tarkoitetaan käyttöliittymää, jossa käyttäjä kirjoittaa komentoriville esimerkiksi käynnistettävän ohjelman nimen mahdollisine parametreineen ja painaa syöttöpainiketta, jolloin komentotulkki käsittelee käskyn ja tulostaa vastineen näytölle. (3.)

```

mars@marsmain ~ $ pwd
/home/mars
mars@marsmain ~ $ cd /usr/portage/app-shells/bash
mars@marsmain /usr/portage/app-shells/bash $ ls -al
total 130
drwxr-xr-x  3 portage portage 1024 Jul 25 10:06 .
drwxr-xr-x 33 portage portage 1024 Aug  7 22:39 ..
-rw-r--r--  1 root  root   35808 Jul 25 10:06 ChangeLog
-rw-r--r--  1 root  root   27002 Jul 25 10:06 Manifest
-rw-r--r--  1 portage portage  4645 Mar 23 21:37 bash-3.1_p17.ebuild
-rw-r--r--  1 portage portage  5977 Mar 23 21:37 bash-3.2_p39.ebuild
-rw-r--r--  1 portage portage  6151 Apr  5 14:37 bash-3.2_p48-r1.ebuild
-rw-r--r--  1 portage portage  5988 Mar 23 21:37 bash-3.2_p48.ebuild
-rw-r--r--  1 portage portage  5643 Apr  5 14:37 bash-4.0_p10-r1.ebuild
-rw-r--r--  1 portage portage  6230 Apr  5 14:37 bash-4.0_p10.ebuild
-rw-r--r--  1 portage portage  5648 Apr 14 05:52 bash-4.0_p17-r1.ebuild
-rw-r--r--  1 portage portage  5532 Apr  8 10:21 bash-4.0_p17.ebuild
-rw-r--r--  1 portage portage  5660 May 30 03:35 bash-4.0_p24.ebuild
-rw-r--r--  1 root  root   5660 Jul 25 09:43 bash-4.0_p28.ebuild
drwxr-xr-x  2 portage portage  2048 May 30 03:35 files
-rw-r--r--  1 portage portage   468 Feb  9 04:35 metadata.xml
mars@marsmain /usr/portage/app-shells/bash $ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
<herd>base-system</herd>
<use>
  <flag name='bashlogger'>Log ALL commands typed into bash; should ONLY be
    used in restricted environments such as honeypots</flag>
  <flag name='net'>Enable /dev/tcp/host/port redirection</flag>
  <flag name='plugins'>Add support for loading builtins at runtime via
    'enable'</flag>
</use>
</pkgmetadata>
mars@marsmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
Password:
* status: started

```

KUVA 2. Komentoliittymä (3)

## 2.1 Mobiilinettisivu vai mobiilisovellus

Ihmiset käyttävät puhelinta koko ajan enemmän ja hakevat sillä koko ajan enemmän tietoa netistä. Tästä syystä verkkosivut kannattaa suunnitella siten, että niiden käyttö on helppoa ja selkeää myös mobiililaitteella. Mobiilinettisivujen hyötyjä ovat kustannustehokkuus, laajempi käyttäjien tavoitettavuus ja parempi yrityksen näkyvyys. Mobiilinettisivut ovat kustannustehokkaampia, koska ne toimivat kaikilla mobiililaitteilla, kun taas Android- ja iOS-laitteille pitää koodata molemmille omat sovellukset. Mobiilisovelluksessa voidaan panostaa parempaan toiminnallisuuteen, joka parantaa myös käyttäjäkokemusta. Mobiilisovelluksessa voidaan hyödyntää laitteen ominaisuuksia sekä sovellusta on mahdollista käyttää offline-tilassa. Asiakkaat voidaan sitouttaa omaan sovellukseen. (5.)

Liiketoiminnan tavoitteet ja mitä halutaan saavuttaa vaikuttaa siihen, tehdäänkö mobiilisovellus vai mobiilinettisivut. Mikäli tarkoituksena on tarjota mobiililaitteelle tarkoitettua sisältöä laajalle asiakaskunnalle kannattaa valita mobiilinettisivut. Jos



halutaan tuottaa käyttäjäystävällisempi ympäristö, jossa on yksilöityä sisältöä, kannattaa valita mobiilisovellus. (5.)

## **2.2 Mobiilisovelluksen käyttöliittymäsuunnittelu**

Mobiilisovelluksen käyttöliittymäsuunnittelussa täytyy ottaa huomioon käyttäjäkokemus. Jos sovellus on hidas sekä sen käyttö on vaikeaa ja hankala oppia, käyttäjä tuskastuu eikä halua käyttää sitä enää. Kun käyttäjäkokemus on miellyttävä, tulee käyttäjä todennäköisemmin käyttämään sovellusta uudelleen, jolloin sovelluksesta voi tulla menestyvä. Hyvä sovellus on nopea ja helppo käyttää sekä siinä on selkeä käyttöympäristö. (6.)

Mobiilisovelluksen käyttöliittymäsuunnittelu on pitkä prosessi eikä se lopu, vaikka sovellus julkaistaisiin. Mobiilisovellusta kannattaa kohdella kuin koko ajan kehitettävää projektia. Täydellistä mobiilisovellusta ei saa kerralla suunnitelluksi eikä sitä tulisi edes tavoitella. Kun mobiilisovelluksen ensimmäinen versio on valmis, kannattaa sen jatkokehittämisessä käyttää hyödyksi testikäyttäjiä sekä käyttäjäpalautetta. (6.)

### **2.2.1 Kognitiivisen kuorman minimointi**

Sovellusta suunniteltaessa kannattaa huomioida kognitiivisen kuorman määrä. Kognitiivisella kuormalla tarkoitetaan sitä, kuinka paljon sovelluksen käyttö vaatii aivotyöskentelyä. Pieneltä näytöltä lukeminen nostattaa kognitiivisen kuorman määrää aivoissa niin, että mobiililaitteelta luettu teksti on puolet vaikeampi ymmärtää kuin tietokoneen näytöltä luettu teksti. (7, s. 26.) Ihmisaivolla on vain rajoitettu määrä käyttötehoa, liian paljon tietoa kerralla voi saada käyttäjän hämmentymään, jolloin käyttäjä lopettaa sovelluksen käytön (6).

Mikäli sovelluksen käyttö vaatii käyttäjältä monta vaihetta ja toimenpidettä, on ne hyvä purkaa pieniin palasiin. Tämä voi myös helpottaa mobiilisovelluksen kehittäjää yhdistämään kaksi toimintoa, esimerkiksi selaaminen ja osto. Turhat toiminnot kannattaa poistaa käyttäjältä, jotta sovelluksen käyttö on mahdollisimman selkeää. (6.)

Jotta käyttöliittymää olisi helppo käyttää, kannattaa siinä keskittyä sisältöön, joka on tärkeä käyttäjälle. Siitä kannattaa poistaa ylimääräiset painikkeet, ja yhdellä

näkymällä kannattaisi olla yksi pääominaisuus. Mobiilisovelluksessa kannattaa myös käyttää progressiivista paljastusta. (6.) Progressiivisella paljastuksella tarkoitetaan sitä, että käyttäjälle tai asiakkaalle annetaan ensiksi vain ne tärkeimmät tiedot ja vaihtoehdot, joita hän tarvitsee. Jos käyttäjä haluaa lisätietoja, hän saa niitä klikkaamalla. Aloitteleville käyttäjille tämä on tärkeä siksi, että he käyttävät aikaa vain siihen, mikä on todennäköisesti heille tärkeää. Kokeneille käyttäjille tämä menetelmä säästää aikaa, koska heidän ei tarvitse käydä läpi kaikkea tietoa, mitä he eivät edes tarvitse. Progressiivinen paljastus parantaa opittavuutta, tehokkuutta, sekä virheiden määrää. (7, s. 60.)

Mobiililaitteella kirjoittaminen on hidasta, sekä kirjoitusvirheille altista (7, s. 51). Mikäli käyttäjän täytyy syöttää tietoja sovellukseen, kannattaa häneltä kysyä vain aivan pakolliset kysymykset. Käyttäjää kannattaa helpottaa tietojen syötössä, näyttämällä kuinka monta merkkiä häneltä vaaditaan. Esimerkiksi luottokortin tietoja syöttäessä, luottokorttinumero jaetaan neljän numeron sarjoihin, jolloin sen tarkistaminen helpottuu käyttäjälle. Automaattinen täydennys auttaa käyttäjää täyttämään vaadittavat tiedot, koska sovellus voi täyttää automaattisesti jo tiedossa olevat tiedot, kuten nimi ja osoitetiedot, jolloin käyttäjän ei tarvitse kirjoittaa mobiililaitteella niin paljon. (7, s. 77–78.)

Suunnittelun täytyy olla johdonmukaista, jolloin käyttäjän tekemien virheiden määrä laskee. Visuaalisesti johdonmukaisesti suunnitellussa mobiilisovelluksessa fontti, painikkeet ja merkit näyttävät samalta kaikissa sovelluksen osissa. Toiminnallisesti johdonmukaisesti suunnitellussa mobiilisovelluksessa, toiminnallisuudet toimivat samalla tavalla koko sovelluksessa. (6.) Esimerkiksi takaisin-painikkeella ei saa olla monta eri toiminnallisuutta. Takaisin-painike toimii yleensä siten, että se vie käyttäjän takaisin edelliseen näkymään tai kumoaa äsken tehdyn valinnan. (7, s. 65.)

Tärkeimpiä ominaisuuksia kannattaa korostaa visuaalisesti esimerkiksi väreillä ja muodoilla (6). Mobiilisovellusta suunniteltaessa kannattaa välttää ammattikieltä ja käyttää kohdeyleisölle kohdennettuja sanoja. ”Puhu käyttäjän kieltä” on ollut käytettävyyden ohjenuorana jo yli 20 vuoden ajan. Turhaa tekstiä kannattaa karsia ja panostaa vain tekstiin, joka on tärkeää. (7, s. 111.)

### 2.2.2 Käyttäjän huomiointi

Mobiilisovelluksen käyttäjän tarpeet tulee huomioida jo mobiilisovelluksen suunnitteluvaiheessa. Käyttäjällä voi olla jokin sairaus tai vamma, jonka voi ottaa huomioon mobiilisovellusta suunniteltaessa. 4,5 % ihmisistä kärsii värisokeudesta, joista punavihersokeus on yleisin. Tämä kannattaa ottaa huomioon, kun sovelluksessa ilmoitetaan onnistumiset ja virheet. Tekstin perään voi lisätä symbolin, joka kuvastaa tilannetta (kuva 3). Käyttäjä voi kärsiä myös matkapahoinvoinnista, jolloin animaatioista kannattaa tehdä valinnaisia. (6.)



KUVA 3. Värisokeuden huomioon ottaminen (6)

Virheilmoituksista kannattaa tehdä tarkoituksen mukaisia ja tarjota käyttäjälle lisää tietoa virheestä, mikä meni vikaan ja miksi. Käyttäjälle kannattaa myös antaa ohjeet, kuinka korjata tilanne ja kuinka välttää tilanteen uusiutuminen. Käyttäjää ei kannata johdattaa harhaan esimerkiksi suunnittelemalla visuaalisia elementtejä, jotka muistuttavat painiketta, mutta eivät toimi painikkeen tavoin. (6.)

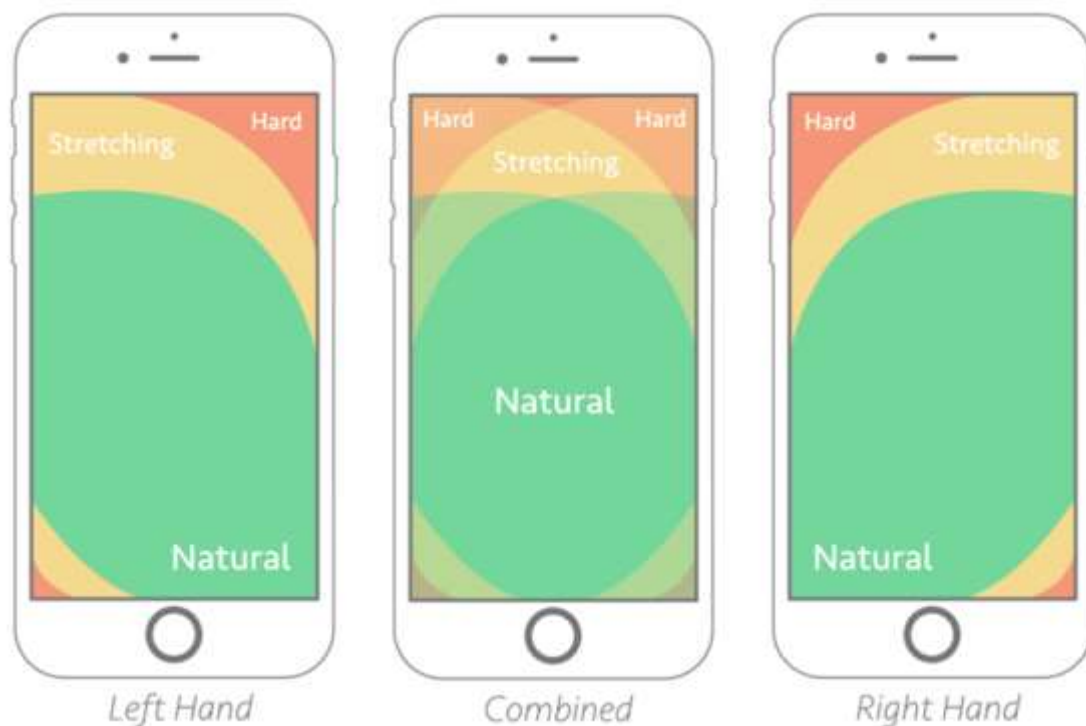
Helpota käyttäjää navigoimaan sovelluksessa käyttämällä tuttuja navigointikomponentteja. Navigointia suunniteltaessa kannattaa priorisoida navigointivaihtoehdot, sekä tehdä navigoinnista näkyvää. Käyttäjälle kannattaa kertoa nykyinen sijainti sovelluksen sisällä eli se, millä välilehdellä käyttäjä on. (6.)

### 2.2.3 Sisällön optimointi pienelle näytölle ja kosketukselle

Jotta mobiilisovelluksesta tulisi mahdollisimman menestynyt, kannattaa suunnittelussa ottaa huomioon näytön pieni koko sekä sormi, jolla sovellusta käytetään

(7, s. 52). Tekstistä tulisi tehdä helposti luettavaa ja selkeää. Tähän vaikuttavat fontin koko, fontin teema sekä kontrasti. Tekstin selkeyteen vaikuttaa myös tekstin pituus, joka on hyvä pitää 30–40 merkissä/rivi. Riviväli täytyy määrittää juuri sopivaksi, ettei teksti ole liian sumpussa eikä liian harvassa. (6.)

Mobiilisovellusta suunniteltaessa tulisi huomioida se, että sovellusta käytetään sormella eikä kursorilla. Tämä kannattaa ottaa huomioon erityisesti painikkeiden suunnittelussa, tekemällä niistä tarpeeksi isoja sormelle. (7, s. 69.) Sormenpään koko on yleisesti 8–10 mm, joten sopiva koko painikkeelle on noin 10 x 10 mm. Sovelluksen suunnittelussa kannattaa huomioida peukalon alue, (kuva 4) ja sijoittaa tärkeät painikkeet sen mukaan. (6.)



*KUVA 4. Peukalon alue (6)*

#### **2.2.4 Ilmoitukset**

Mobiilisovelluksista tulee käyttäjälle ilmoituksia (notifications), jotka muistuttavat käyttäjää sovelluksen olemassaolosta ja sovelluksen tapahtumista. Nämä ilmoitukset voivat ärsyttää käyttäjää, jos niitä tulee liian paljon liian usein. Tämä voi johtaa siihen, että käyttäjä poistaa sovelluksen. (6.)

Käyttäjälle kannattaa ilmoittaa vain tärkeimmistä asioista, semmoisista asioista, joita käyttäjän voisi olettaa haluavan nähdä. Ajoita ilmoitusten lähettäminen käyttäjäystävälliseen aikaan, älä lähetä ilmoituksia esimerkiksi yöllä tai yleisiin työaikoihin. Kannattaa ottaa huomioon myös eri tavat, joilla ilmoituksia voi lähettää. Tällaisia tapoja ovat esimerkiksi push-ilmoitukset, jotka ponnahtavat näytölle, sähköposti-ilmoitus sekä sovelluksen sisäinen ilmoitus. (6.)

### **2.3 Käyttöliittymäsuunnittelun vaiheet**

Käyttöliittymäsuunnittelussa on hyvä luoda käyttäjävuokaavio kutakin näyttöä varten, tehdä raakapiirros sovelluksesta, valita rakenne, kuviot ja värit, tehdä mock-up-kuvat sovelluksesta, luoda animoitu prototyyppi sekä tehdä käyttäjä testauksia. Mock-up-kuvat ovat kuin menisit tulevaisuuteen ottamaan näyttökuvia omasta sovelluksestasi. Niiden pitäisi näyttää aidoilta ja juuri sellaisilta, kuin haluat sovelluksesi näyttävän. (8.)

Käyttäjävuokaaviossa kuvainnollistetaan käyttäjän toiminnot sovelluksessa. Tätä vaihetta ennen kannattaa määritellä ominaisuudet, joita sovellukselta haluaa. Vuokaaviossa on yleensä kolme eri kuviota: neliö, joka kertoo millä näytöllä ollaan, ruutu, joka kertoo mikä valinta tehdään sekä nuoli, joka yhdistää näytöt ja valinnat yhteen. (8.)

Raakapiirros on luonnos sovelluksesta, joka näyttää, miltä haluat sovelluksen näyttävän. Luonnoksessa käy ilmi, mihin kuvat teksti ja muu sisältö sijoitetaan. Tämän jälkeen valitaan rakenteet, kuviot ja värit, joita käytetään sovelluksessasi. Tämän jälkeen voidaan alkaa tekemään mock-up-kuvia. Kun mock-up-kuvat ovat tehty, voi niitä jakaa kavereille ja he voivat kertoa mielipiteensä. (8.)

Tämän jälkeen tehdään animoitu prototyypin sovelluksesta. Ihmiset voivat testata prototyyppiä ja antaa siitä palautetta. Palautteen perusteella tehdään tarvittavia muutoksia sovellukseen. Kun muutokset ovat valmiit ja sovellukseen ollaan tyytyväisiä, sitä voidaan alkaa koodaamaan. (8.)

### 3 KÄYTETTÄVYYS

Käytettävyydellä tarkoitetaan menetelmä- ja teoriakenttää, jonka kautta käyttäjän ja laitteen yhteistoimintaa pyritään saamaan tehokkaammaksi ja käyttäjäystävällisemmäksi. Käytettävyydessä käytetään hyväksi kognitiivisen psykologian sekä ihmisen ja koneen vuorovaikutuksen tutkimusta. (9, s. 17.) Jacob Nielsen määrittelee käytettävyyden opittavuuden, tehokkuuden, muistettavuuden, virheiden ja tyytyväisyyden perusteella (10, s. 26).

Opittavuudella Nielsen tarkoittaa sitä, että käytettävä laite tai järjestelmä on helppo oppia. Näin käyttäjä pääsee nopeasti suorittamaan sillä haluttuja töitä ja tehtäviä. Tehokkuudella hän tarkoittaa sitä, että laitteen tai järjestelmän käyttö on tehokasta. Kun käyttäjä on oppinut käyttämään laitetta tai järjestelmää, hän voi tehdä sillä tehokkaasti töitä. Muistettavuudella hän tarkoittaa sitä, että laite tai järjestelmä on helppo muistaa, jotta käyttäjän on helppo palata käyttämään laitetta, vaikka ei olisi sitä vähään aikaan käyttänytkään. Virheillä hän tarkoittaa sitä, että järjestelmän tai laitteen käytön aikana pitäisi tapahtua mahdollisimman vähän virheitä ja virheet on helppo korjata. Katastrofaalisia virheitä ei tulisi tapahtua. (10, s. 26.)

Käytettävyys on tärkeä osa jokaista laitetta ja järjestelmää. Jos jokin Nielsenin mainitsemista määritelmistä ei täyty, voi käyttäjä etsiä vaihtoehdoisen laitteen tai järjestelmän suorittaakseen tehtävänsä. Kun käytettävyys on sujuvaa ja ongelmattonta, käyttäjä todennäköisesti jatkaa tuotteen käyttöä jatkossa.

#### 3.1 Käytettävyystestaus

Käytettävyystestauksella tutkitaan tuotteen, laitteen tai järjestelmän käytettävyyttä. Käytettävyystestaus on tärkeä osa mobiilisovelluksen kehitysprosessia, ja sillä voidaan karsia pois turhia virheitä ja ongelmia, joita käyttäjä voi kohdata sovellusta käytettäessä.

Käytettävyyden testaaminen on joustava menetelmä, ja sitä muuntelemalla saadaan tuotteen käytöstä monenlaista tietoa. Sitä voidaan käyttää todella erilaisten tuotteiden testaamiseen. (10, s. 275.)

Käytettävyyttä mitataan yleensä useilla testihenkilöillä, jotka suorittavat ennalta määriteltäviä tehtäviä. Käytettävyydestejä voidaan myös tehdä oikeissa työskentelytilanteissa, jolloin käyttäjä antaa palautteen suorittamastaan tehtävästä ja sen sujuvuudesta. Molemmissa tapauksissa on tärkeää, että käytettävyys mitataan suhteessa tiettyihin käyttäjiin ja tiettyihin tehtäviin. (10, s. 27.) Käytettävyydestaus oikeilla käyttäjillä on tärkein perusteellinen käytettävyydestausmenetelmä, se on tietyssä mielessä korvaamaton, koska se tarjoaa suoraa tietoa, kuinka käyttäjät käyttävät laitetta ja mitkä heidän tarkat ongelmansa ovat laitteen käytön aikana. (10, s. 165.)

### **3.1.1 Testin tavoitteet ja testisuunnitelmat**

Ennen käytettävyydestin suorittamista kannattaa määritellä testin tarkoitus ja tavoite, jotta testissä käytetään oikeanlaisia metodeja ja niistä saa tarvittavat tiedot tuotteen kehittämiseen (10, s. 170). Tärkein valinta on, käyttääkö testissä laadullista vai määrällistä arviointia. Laadullisen (kvalitatiivisen) arvioinnin päätavoite on kertoa, mitkä osat käyttöliittymässä ovat toimivia ja mitkä eivät sekä kuinka käyttöliittymää voi parantaa. Tyypillisin menetelmä laadullisen arvioinnin käytössä on ääneen ajattelu testin aikana. Se voi auttaa kehittämään käyttöliittymää myös osana jatkuvaa kehittämisprosessia. Määrällisen (kvantitatiivisen) arvioinnin tavoitteena on arvioida käyttöliittymän laatua, esimerkiksi verrattuna annettuihin käytettävyydestavoitteisiin tai verrattuna kilpailijan tuotteeseen. Tyypillisin menetelmä määrällisen arvioinnin käytössä on mittaustesti. (9, s.281).

Käytettävyydestin tavoitteita voivat olla esimerkiksi yleinen käytettävyys, sopivuus kokeneille käyttäjille, sopivuus epäsäännölliseen käyttöön, opittavuus, tukitarpeiden minimointi ja virheensieto (9, s. 282).

Testisuunnitelma kannattaa kirjoittaa ennen testien aloittamista. Sen olisi hyvä käsitellä seuraavanlaisia asioita: testin tavoite, käytettävyydestin ajankohta, käytettävyydestin arvioitu kesto, käytettävyydestissä käytettävä laitteistot sekä ohjelmistot, testikäyttäjät ja niiden määrä, testitehtävien määrittely, datan kerääminen ja analysointi, sekä testin budjetti. (10, s. 170–171.)

Ennen virallista käytettävyydestä kannattaa suorittaa pilottitesti. Pilottitestin suorittaa yleensä sellainen henkilö, joka on helposti tavoiteltavissa mutta ei ole testattavan ympäristön kanssa päivittäin tekemisissä. Pilottitestissä havaitaan yleensä mahdolliset ongelmatilanteet testin aikana. Lisäksi havaitaan kysymykset, jotka ovat huonosti määritellyt. Samalla testataan laitteiston toimivuus. Testissä suoritettavat tehtävät voivat myös olla liian vaikeita tai helppoja. Tällaisissa tilanteissa voidaan arvioida uudelleen testikysymyksiä sekä mahdollisesti pidentää käytettävissä olevaa aikaa. (10, s. 174.)

### **3.1.2 Testikäyttäjä**

Testikäyttäjä on henkilö, joka suorittaa käytettävyydestin testattavalle ohjelmistolle tai laitteelle. Testikäyttäjien tärkein kriteeri on se, että heillä on samanlainen käyttötottumus kuin oikealla käyttäjällä. Jos käytettävyydestissä halutaan käyttää mahdollisimman vähän testikäyttäjiä, kannattaa testikäyttäjiksi valita henkilöitä, jotka eivät ole erityisen hyviä eivätkä huonoja laitteen tai ohjelmiston käyttäjiä, vaan keskivertotasolla. Mikäli käytettävyydestissä on mahdollista käyttää paljon testikäyttäjiä, kannattaa heitä valita mahdollisimman laajasti käyttötottumusten mukaan. (10, s. 175.)

Melkein jokainen käyttöliittymä kannattaa testata kokemattomalla käyttäjällä ja monet käyttöliittymät kannattaa testata kokeneilla käyttäjillä. Yleensä kyseisille käyttäjäryhmille kannattaa suunnitella omanlaiset käytettävyydestit ja kysyä joi-takin samoja asioita. Testikäyttäjät saattavat tarvita välillä ohjeistusta ja avustusta käytettävyydestin aikana. Tästä syystä käytettävyydestissä täytyy olla aina yksi testattavan laitteen tunteva henkilö testikäyttäjän lisäksi. (10, s. 177.)

### **3.1.3 Testitehtävät**

Testitehtäviä suunnitellessa kannattaa miettiä, kuinka testattavaa laitetta tai ohjelmistoa käytetään todellisuudessa ja minkälaisia tehtäviä sillä suoritetaan. Testitehtävien tulisikin kuvastaa mahdollisimman hyvin oikeasti työssä suoritettavaa tehtävää, jota laitteella tai ohjelmistolla halutaan tehdä. Testitehtävien tulisi myös kattaa tärkeimmät osat testattavasta käyttöliittymästä. (10, s. 185.)



Testitehtävien tulisi olla sopivan mittaisia, ei liian pitkiä eikä liian lyhyitä. Jos testitehtävät ovat liian lyhyitä, voi niistä tulla pinnallisia, jolloin niistä ei saa tarvittavasti tietoa tuotteen tai käyttöliittymän kehitykseen. Jos testitehtävät taas ovat liian pitkiä, voivat ne viedä liikaa aikaa eikä testikäyttäjä kerkeä tekemään niitä. (10, s. 186.)

Testitehtävistä kannattaa tehdä asiallisia, eikä niissä kannata käyttää humoristista tai loukkaavaa sisältöä, kuten esimerkiksi pyytää käyttäjää piirtämään viikset presidentille piirustussovellusta testatessa. Kaikki eivät välttämättä miellä vitsiksi tarkoitettua tehtävää humoristisena, vaan käyttäjä voi loukkaantua siitä, ja humoristiset asiat voivat myös häiritä testikäyttäjää ohjelmiston testauksessa. Mikäli testattava tuote on tarkoitettu viihdekäyttöön tai muuhun sellaiseen, voi siihen sisällyttää ”viihdettä” hyvän maun rajoissa. (10, s. 186.)

Testitehtävillä voi myös kohottaa testikäyttäjän itsetuntoa. Tämän takia ensimmäisestä testitehtävästä kannattaa tehdä yksinkertainen. Viimeiseksi testitehtäväksi kannattaa valita sellainen tehtävä, jossa käyttäjä oikeasti suorittaa jonkin tehtävän loppuun. Tehtävät kannattaa antaa käyttäjälle yksi kerrallaan, jotta käytettävyydesti voidaan keskeyttää milloin vain ilman, että testikäyttäjä tuntee itsensä epäonnistuneeksi. (10, s. 187.)

### **3.1.4 Testin eteneminen**

Käytettävyytsteissä on yleensä neljä vaihetta: valmistelu-, johdanto-, testaus- sekä raportointivaihe. Valmisteluvaiheessa käytettävyytstestin valvoja varmistaa, että testissä käytettävä huone on valmiina, käytettävät laitteistot toimivat oikein sekä ne ovat valmiina käyttöä varten. Valvoja varmistaa myös, että tarvittavat ohjeet, materiaalit sekä kysymyslomakkeet ovat saatavilla. (10, s. 187.)

Johdantovaiheessa käytettävyytstestin valvoja toivottaa testikäyttäjän tervetulleeksi ja käy hänen kanssaan lyhyesti läpi testin tarkoituksen ja testituloksien käyttötarkoituksen (käyttöliittymän kehittäminen). Lisäksi keskustellaan siitä, että käytettävyytstesti on luottamuksellinen molemmiin puolin, käytettävyytstesti on vapaaehtoinen ja sen voi keskeyttää koska vain. Testikäyttäjä saa myös esittää kysymyksiä ennen testiä ja koska tahansa testin aikana. Tarkempana ohjeistuksena voi sanoa esimerkiksi ääneen ajattelun. (10, s. 188–189.)

Testikäyttäjiltä kannattaa ottaa kirjallinen suostumus, mikäli käytettävyystestitilaisuus nauhoitetaan tai testin tuloksia näytetään muille. Suostumuskaavake kannattaa pitää lyhyenä ja selkokielisenä, erityisesti kannattaa vältellä lakitermejä. (10, s. 189.)

Testausvaiheessa valvoja ei saa ilmaista missään vaiheessa testikäyttäjälle omia mielipiteitään siitä, eteneekö testikäyttäjä testissä hyvin vai huonosti. Valvojan ei myöskään pidä auttaa testikäyttäjää, vaikka testitehtävä olisi kuinka vaikea. Testikäyttäjää voi auttaa silloin, kun käyttäjä on selvästi jumissa tehtävän kanssa ja alkaa tuskastua liikaa. Mikäli testiä valvoo useampi henkilö, tulisi etukäteen määrittää, kuka puhuu testikäyttäjän kanssa, ja muiden tulisi olla hiljaa käytettävyystestin aikana, ettei testikäyttäjä saa sekavia ohjeita. (10, s. 190.)

Kun testausvaihe on ohi, alkaa raportointivaihe. Testikäyttäjän tulee yleensä täyttää heti käytettävyystestin jälkeen kyselykaavake, jossa pyydetään esimerkiksi arvioimaan käyttöliittymän käytettävyys ja kommentoimaan muuten testattua ohjelmaa. Kaavakkeessa voidaan kysyä myös, onko testikäyttäjällä mahdollisesti mitään parannusehdotuksia käyttöliittymään liittyen. Kun testikäyttäjä on poistunut käytettävyystestitilaisuudesta, testauksen valvojan tulee tarkistaa, että kaikki materiaalit ovat tallennettuna tietokoneelle. Valvojan kannattaa myös kirjoittaa lyhyt raportti heti käytettävyystestin loputtua siitä, miten testi on sujunut ja oliko sen aikana mahdollisesti ongelmia. (10, s. 191.)

### **3.1.5 Testausmenetelmät**

Seuraavaksi tuon esille käytettävyystestauksessa käytettäviä testausmenetelmiä.

#### **Ääneen ajattelu**

Ääneen ajattelu on yksi hyödyllisimmistä käytettävyystestauksen metodeista. Ääneen ajattelulla tarkoitetaan sitä, että käyttäjä ajattelee ääneen kaikki tekemänsä ratkaisut ja ajatukset käytettävyystestin aikana. Ääneen ajattelussa testikäyttäjä ilmaisee, miten hän näkee testattavan ohjelmiston. Ääneen ajattelu helpottaa huomaamaan myös mahdollisia väärinymmärryksiä testikäyttäjän ja testattavan ohjelmiston välillä. (10, s. 195.)

Kaikille ihmisille ääneen ajattelu ei kuitenkaan ole helppoa, ja testin valvoja saattaa joutua kysymään testikäyttäjältä kysymyksiä kuten ”Mitä sinä ajattelen nyt?” tai ”Mitä luulet tämän viestin tarkoittavana?”. Jotkut testikäyttäjät saattavat edetä testitehtävissä hyvinkin nopeasti, eivätkä he välttämättä ehdi ajatella paljon testin aikana. Tällöin ääneen ajattelu voi olla turhaa. Ääneen ajattelua tulisi välttää etenkin, jos käytettävyydestä halutaan nähdä, kuinka kauan jonkin tehtävän suorittaminen vie aikaa. (10, s. 196–197.)

Ääneen ajattelu nostaa myös kognitiivisen kuorman määrää, joka voi tehdä puhumisesta vielä vaikeampaa. Ääneen ajattelun menetelmää käytettäessä testit yleensä taltioidaan videolle sekä testikäyttäjää haastatellaan lopuksi. (9, s. 286.)

### **Paritesti**

Paritestillä tarkoitetaan testiä, jossa kaksi testikäyttäjää käyttää yhtä järjestelmää samanaikaisesti ja samalla he keskustelevat tuotteesta keskenään. Tämä menetelmä on periaatteessa sama kuin ääneen ajattelu, mutta tässä keskustellaan toisen testihenkilön kanssa. Tämä menetelmä ei sovi kaikkiin tuotteisiin, vaikka onkin ääneen ajattelua helpompi testimenetelmä testikäyttäjille. Käyttäjäparien valinnassa tulee ottaa huomioon se, että molempien henkilöiden mielipiteet tulevat esille eikä toinen dominoi koko testin ajan. Kyseinen testaustilanne taltioidaan yleensä videolle, sekä käyttäjiä haastatellaan lopuksi. (9, s. 286.)

### **Yhteisläpikäynti**

Yhteisläpikäynnissä testinvalvoja sekä testikäyttäjä etenevät testissä keskustelun tuotteesta. Testinvalvoja istuu testikäyttäjän vieressä ja kyselee aktiivisesti testikäyttäjän ymmärrystä testattavasta kohteesta. Testivalvoja voi kysellä käyttäjältä miksi hän toimii, niin kuin hän toimii. Tämä testimenetelmä antaa tietoa mentaalisen mallin muodostumisesta, mutta liian aktiivinen kysely kesken testin voi myös häiritä testikäyttäjän keskittymistä. Tätä menetelmää kannattaa käyttää kehitystesteissä, jos testattavassa järjestelmässä on paljon toiminnallisia puutteita. Testivalvojalta vaaditaan taitoa ja herkkyyttä, ettei hän vääristä testitulosta johdattamalla testikäyttäjää. Tämäkin testimenetelmä voidaan taltioida videolle, mutta se ei ole pakollista. (9, s. 286.)

## **Jälkikäteen haastattelu**

Jälkikäteen haastattelussa testikäyttäjä tekee tehtävät itsekseen ja lopuksi testikäyttäjät haastatellaan tai he täyttävät kyselylomakkeen. Perustestaustapa on tässä menetelmässä sama kuin määrällisessä arvioinnissa. Samalla voidaan selvittää kognitiivista kuormitusta testin aikana, testikäyttäjän tyytyväisyyttä tuotteeseen ja mahdollisia käyttövirheitä. Mikäli testissä mitataan aikaa tai ei tehdä mitauksia ollenkaan, ei testaustilannetta tarvitse videoida. Virheiden poiminta vaatii kuitenkin videoinnin. Testivalvojan ei ole pakko olla paikalla, mikäli testikäyttäjät ovat kiinnostuneita pitämään kirjaa testin eri tilanteista ja täyttämään kyselylomakkeen. (9, s. 286.)

## **Jälkeenpäin kommentointi**

Jälkeenpäin kommentoinnissa testikäyttäjä tekee tehtävät itsenäisesti. Testin jälkeen testikäyttäjä ja valvojat katsovat nauhan ja käyttäjä kommentoi tilanteita nauhallalla. Tämä testimenetelmä sopii hyvin määrälliseen testaustapaan, kun selvitetään käytön ongelmakohtia. Tämä testimenetelmä on hyvä videoida. Käyttäjä ei yleensä muista kaikkia asioita testin jälkeen ja saattaa sensuroida joitain asioita. (9, s. 286–287.)

## **Ryhmäläpikäynti**

Ryhmäläpikäynti sopii prototyyppitestaukseen, jolloin toiminnallisuutta ei ole. Testiin osallistuu käyttäjä, testinvalvoja ja suunnittelija. He käyvät yhdessä läpi testitehtävät käyttäen käyttöliittymän kuvia (piirroskuvia, valokuvia tai näyttökopioita), kynää ja paperia. Testitapa on hyvä, koska silloin saadaan käyttäjä ja suunnittelija kasvotusten tuotteen käyttöliittymän ääreen. Käyttäjät uskaltavat myös kommentoida kuvia enemmän kuin valmista tuotetta. Videotallentointia tai sanelunauhuria kannattaa käyttää. (9, s. 287.)

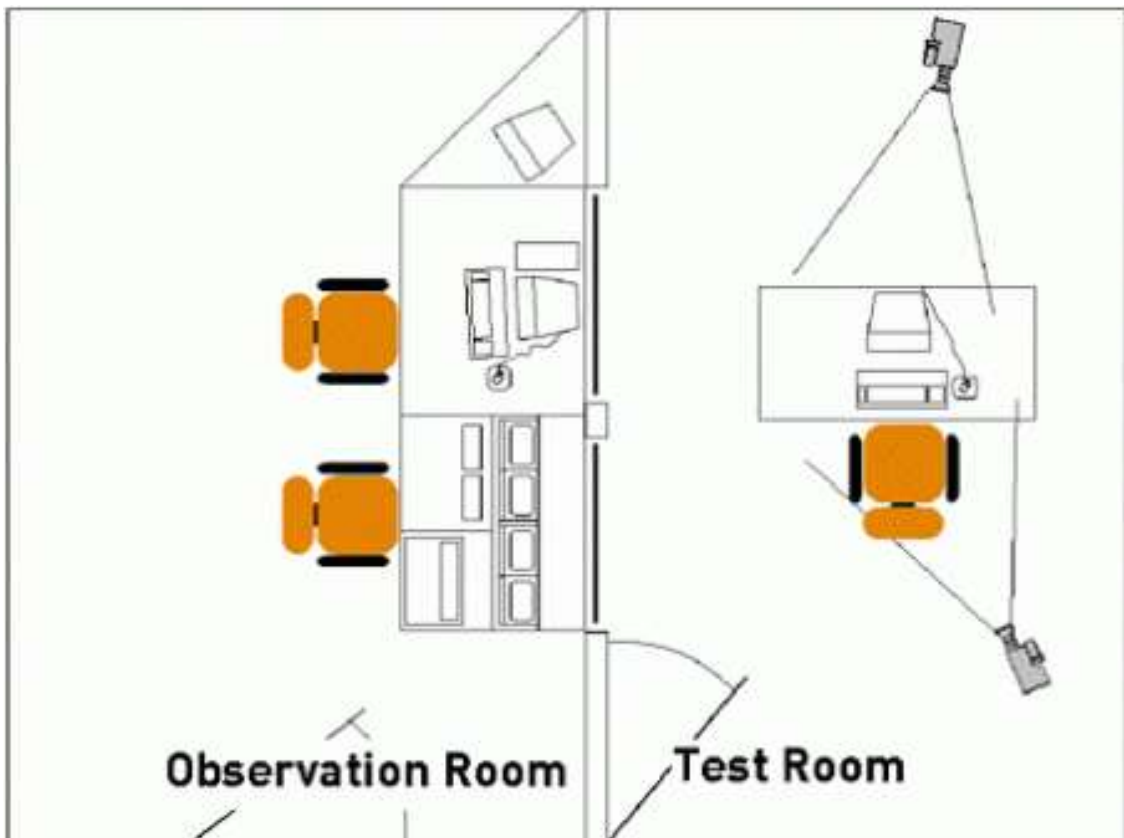
## **Vapaa läpikäynti**

Vapaa läpikäynti sopii valmiin tai lähes valmiin tuotteen testaukseen. Testikäyttäjä testaa testattavaa järjestelmää rauhassa, eikä testinvalvoja puutu testin kulkuun kuin vain silloin, jos käyttäjä tarvitsee apua. Tämän menetelmän vahvuutena on se, että testikäyttäjä löytää juuri ne ominaisuudet, joita järjestelmä tarjoaa

tai joita testikäyttäjä osaa etsiä. Testattavan järjestelmän on oltava sellainen, että testikäyttäjällä on ennestään samanlaisesta järjestelmästä kokemusta. Videoinnista on hyötyä testimenetelmässä. (9, s. 287.)

### 3.1.6 Käytettävyysslaboratoriot

Moni käytettävyyss testi pidetään käytettävyyss testauslaboratoriossa (kuva 5). Käytettävyyss testilaboratoriot ovat käytännöllisiä, mutta eivät pakollisia, sillä käytettävyyss testejä voidaan pitää myös normaaleissa toimistotiloissa. Käytettävyyss testilaboratorio on normaalisti äänieristetty ja siellä on yksisuuntainen peili tarkkailuhuoneesta testihuoneeseen. Tämä mahdollistaa testien tekijöiden ja muiden käytettävyyss asiantuntijoiden kommunikoinnin testin aikana ilman, että he häiritsevät testihenkilöä. Käytettävyyss testilaboratorioissa on myös yksi tai useampi videokamera tallentamassa käytettävyyss testin. (10, s. 200, 202–203.)



KUVA 5. Esimerkki käytettävyyss testilaboratorion pohjaratkaisusta (11)

### 3.2 Mobiililaitteen käytettävyydestaustaus

Mobiililaitteen käytettävyydestaustaus tapahtuu hieman eri tavalla kuin tietokoneella suoritettava käytettävyydestaustaus. Tietokoneella suoritettavassa käytettävyydestaustauksessa voidaan tallentaa näytöllä tapahtuvat asiat, kuten hiiren liike, klikkaukset sekä näppäimistön käyttö. Mobiililaitteella tämä on vaikeampaa, koska näytöllä tapahtuvat klikkaukset sekä sormen liikkeet eivät tallennu näyttötallennuksessa. Tästä syystä mobiililaitteella tehdyt käytettävyydestaustaukset kannattaa kuvata esimerkiksi dokumenttikameralla (kuva 6) tai muulla digikameralla. Dokumenttikameraa käyttäessä käytettävyydestaustauksen tarkkailija voi katsella käyttäjän tekemiä toimintoja tietokoneen näytöltä sekä samalla tallentaa omaa näyttökuvaa, jolloin käyttäjän puhelimella tekemät toiminnot myös tallentuvat. (12.)

Mobiililaitteen käytettävyydestaustaus, kuten minkä tahansa laitteen käytettävyydestaustaus, kannattaa suunnitella tarkasti etukäteen. Suunnittelussa kannattaa huomioida seuraavanlaisia asioita: aikataulu, budjetti, testiympäristö, testausten tallentaminen, kohdeyleisö, testattava laite, testin jäsentely, testien määrä sekä testaustilan verkkoyhteys. (12.)



KUVA 6. Mobiilisovelluksen käytettävyydestaustaus (13)

## 4 TYÖN TOTEUTUS

Työn tarkoituksena oli suunnitella mobiilisovellus ottaen huomioon käyttäjät sekä käytettävyys. Aloitin työn tekemällä käyttäjävuokaavion mobiilisovelluksesta. Käyttäjävuokaavion tarkoitus on kuvata mobiilisovelluksen toimintaa. Käyttäjävuokaavion jälkeen piirsin raakapiirroksia mobiilisovelluksesta. Raakapiirroksia tein käsin paperille, tämän jälkeen käytin Marvel – Design and build Apps -mobiilisovellusta. Kyseisellä sovelluksella pystyi demoamaan sovelluksessa tapahtuvat siirtymät eri näkymien välillä.

Seuraavaksi valitsin rakenteita, kuvioita sekä värejä sovellukseeni. Värien valinnassa käytin Color Hunt -verkkosivustoa, jolta löytyi paljon hyviä väripaletteja. Tämän jälkeen aloin tekemään mock-up-kuvia sovelluksestani. Mock-up-kuvalla tarkoitetaan sovelluksen visuaalisen ulkoasun mallinnusta, visuaalisen suunnittelun Sketch-tietokonesovelluksessa. Mock-up-kuva on kuin menisi tulevaisuuteen ottamaan näyttökuvan sovelluksestasi.

### 4.1 Käyttäjävuokaavio

Käyttäjävuokaavion tekemiseen käytettiin web-pohjaista Lucidchart-ohjelmistoa. Käyttäjävuokaaviossa kuvataan sovelluksen toiminta vaihevaiheelta. Käyttäjävuokaaviossa neliö kuvastaa sitä, millä näytöllä ollaan, ruutu kertoo, mikä valinta tehdään ja nuoli yhdistää näytöt ja valinnat yhteen. Kuvassa 7 näkyy esimerkki vuokaaviosta. Virallista käyttäjävuokaaviota ei ole esitelty tässä dokumentissa. Ennen vuokaavion tekoa minulla oli vahva näkemys siitä, mitä toimintoja halusin sovellukseeni ja kuinka halusin sovelluksen toimivan.



Kuva 7. Esimerkki vuokaaviosta

## 4.2 Raakapiirros sekä rakenteen, kuvioiden ja värien valinta

Raakapiirrosten tekemiseen käytin A4-paperia sekä lyijykynää. Raakapiirroksissa (kuva 8 ja 9) suunnittelin, millaisia elementtejä, tapahtumia ja siirtymiä missäkin näkymässä on. Tämän jälkeen havainnollistin eri siirtymiä Marvel – Design and Build Apps -mobiilisovelluksella. Sovelluksella otettiin kuvat raakapiirroksista ja sen jälkeen valittiin alue, joka toimii painikkeena, ja näkymä, johon painike vie. Kuvassa 8 esitellään raakapiirros sovelluksen kirjautumisikkunasta sekä kuvassa 9 esitellään raakapiirros rekisteröitymisikkunasta.



KUVA 8. Kirjautu sisään



KUVA 9. Rekisteröityminen



Raakapiirrosten tarkoitus on kuvata, mitä toimintoja milläkin ikkunalla on. Raakapiirrosten ei tarvitse olla tarkkoja piirroksia sovelluksesta, ja raakapiirroksissa näkyvät elementit voivat muuttua, kun sovellusta piirretään tietokoneella.

Erilaisiin rakenteisiin ja kuvioihin tutustuin Mobile Patterns- ja Pptrns-sivustoilla, mutta jo raakapiirrosten tekovaiheessa minulla oli vahva näkemys siitä, millaisia rakenteita ja kuvioita haluan käyttää sovelluksessani.

Värien valinta oli mielenkiintoinen vaihe, jossa apuna käytin Color Hunt -verkkosivua. Color Hunt -verkkosivulta valitsin useita eri väripaletteja, joista tein Sketch-tietokoneohjelmistolla visuaalisia suunnitteluja sovelluksesta. Lopulliseen suunnitelmaani valitsin kuvassa 10 näkyvän väripaletin.

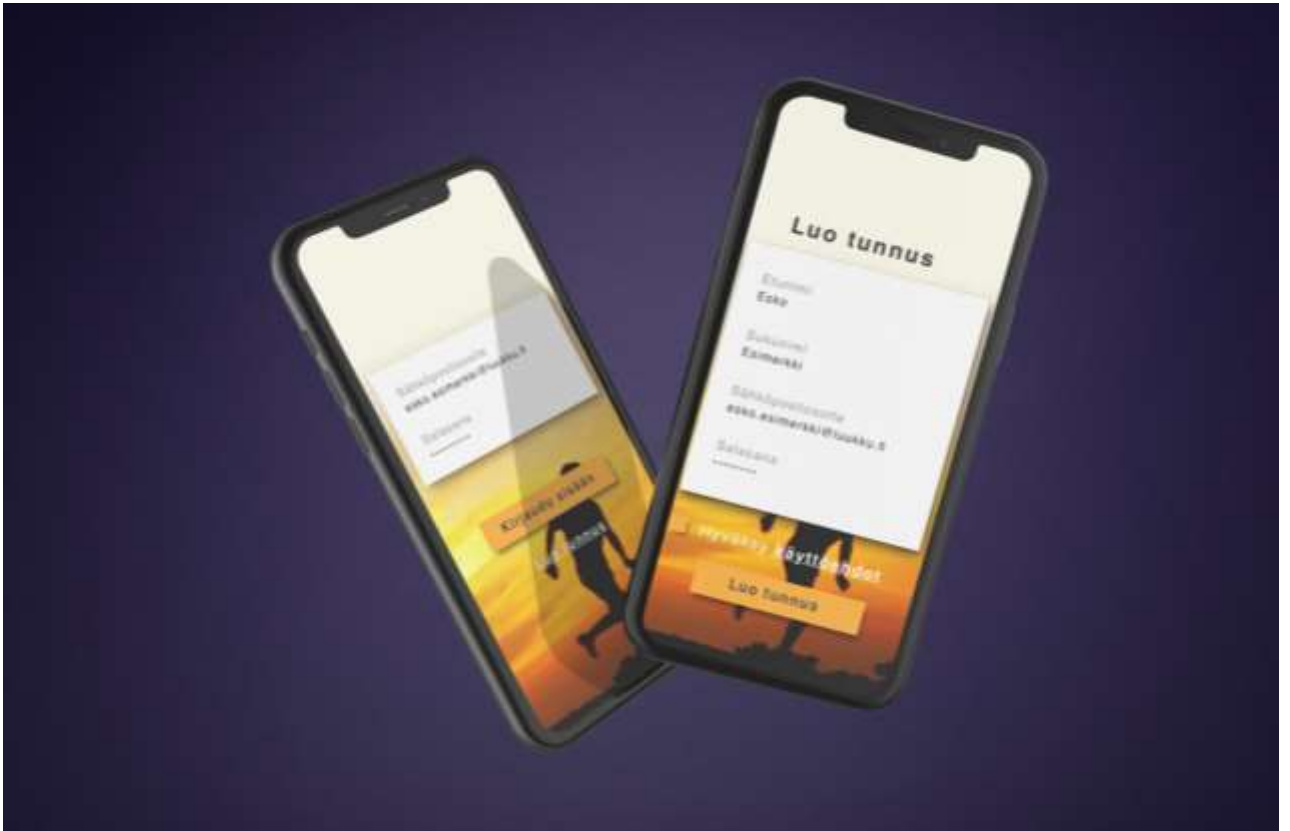


*KUVA 10. Väripaletti*

### **4.3 Mock-up-kuvat**

Mock-up-kuvilla tarkoitetaan mallikuvaa sovelluksesta. Se voi olla joko oikeassa koossa tai sen mittakaavaa voidaan muuttaa pienemmäksi tai suuremmaksi. Mock-up-kuvien tekoa varten kokeilin useita eri sovelluksia, joista päädyin Sketch-tietokoneohjelmistoon. Minulla ei ollut aikaisempaa kokemusta Sketch-sovelluksesta, mutta opin sen käytön suhteellisen nopeasti netin avustuksella.

Tein useita eri versioita mock-up-kuvista ja esittelin niitä kavereilleni. He antoivat työstä palautetta, jonka perusteella kehitin sovellusta. Kun sain mock-up-kuvat valmiiksi, vein kuvat Placeit-verkkosivustolle, jossa pystyin viimeistelemään visuaalisen ilmeen. Kuvassa 11 esitellään mock-up-kuvat Kirjautu sisään sekä Luo tunnus -ikkunasta.



*KUVA 11. Kirjaudu sisään ja Luo tunnus -näkymät.*

## 5 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli suunnitella mobiilisovelluksen käyttöliittymä ottaen huomioon mobiililaitteen käytettävyyden. Työn teoriaosassa perehdytään mobiilisovelluksen käyttöliittymäsuunnitteluun, käytettävyyteen sekä käytettävyydestäukseen. Itse mobiilisovelluksen suunnitteluvaihetta tai mobiilisovellusta ei ole opinnäytetyön julkisessa osassa esitelty.

Mobiilisovelluksen käyttöliittymäsuunnittelussa täytyy ottaa huomioon käyttäjäryhmä, jolle tuotetta suunnitellaan. Käyttäjäryhmän määrittely helpottaa valitsemaan sovellukseen käyttäjäryhmälle olennaista tietoa ja tekemään sovelluksen käytöstä miellyttävää. Sovelluksen sisältö kannattaa pitää lyhyenä ja selkeänä. Yksinkertainen sovellus voi olla monipuolinen ja runsas sisällöltään.

Käytettävyydestä kannattaa ottaa mukaan mobiilisovelluksen käyttöliittymäsuunnitteluun varhaisessa vaiheessa. Tällöin mahdolliset käytettävyydevirheet ovat nopea ja helppo korjata. Käytettävyydestäuksessa testikäyttäjinä voi käyttää esimerkiksi omia kavereita, jotka antavat palautetta suunnitellusta sovelluksesta. Pienet käytettävyydestit mobiilisovelluksen suunnittelu- ja tekovaiheessa voivat olla edullisempia ja nopeampia järjestää kuin yhden ison käytettävyydestin järjestäminen valmiille sovellukselle.

Minulla oli aikaisempaa kokemusta käytettävyydestäuksesta ja käytettävyydestä, mutta ei mobiilisovelluksen käyttöliittymän suunnittelusta. Työn tekeminen oli ajoittain haasteellista mutta erittäin opettavaista ja palkitsevaa. Opin paljon uusia asioita käyttöliittymäsuunnittelusta, siitä, mitä ottaa huomioon ja kuinka käyttöliittymäsuunnittelussa kannattaa edetä. Opin myös useita uusia asioita käytettävyydestä, käytettävyydestäuksesta ja mobiilisovelluksen käyttöliittymäsuunnittelusta. Samalla opin käyttämään useita uusia sovelluksia ja verkkosivustoja kuten Sketch ja Placeit.

Mobiilisovellusidean jatkokehittämiseksi tulisi tehdä oikea sovellus. Sen jälkeen mobiilisovellukselle voi tehdä aikaisemmin työssä mainittuja käytettävyydestestejä. Tämän hetkistä visuaalista suunnitelmaa on esitelty kavereille, ja heidän palautteiden pohjalta lähdän kehittämään sovellusta.

## LÄHTEET

1. Käyttöliittymä. 2017. Wikipedia. Saatavissa: <https://fi.wikipedia.org/wiki/Käyttöliittymä>. Haettu 8.11.2018.
2. Graafinen käyttöliittymä. 2018. Wikipedia. Saatavissa: [https://fi.wikipedia.org/wiki/Graafinen\\_käyttöliittymä](https://fi.wikipedia.org/wiki/Graafinen_käyttöliittymä). Haettu 8.11.2018.
3. Komentoliittymä. 2017. Wikipedia. Saatavissa: <https://fi.wikipedia.org/wiki/Komentoliittymä>. Haettu 8.11.2018.
4. Bonnington, Christina 2014. OS X Yosemite: See What's New in Apple's Calendar App. Wired. Saatavissa: <https://www.wired.com/2014/09/os-x-yosemite-calendar/>. Haettu 8.11.2018.
5. Armour, Britt 2017. Mobile App Vs. Mobile Website: Which Is Better Option? Clearbridge Mobile. Saatavissa: <https://clearbridgemobile.com/mobile-app-vs-mobile-website-which-is-the-better-option/>. Haettu 7.1.2019
6. Babich, Nick 2018. A Comprehensive Guide To Mobile App Design. Smashing Magazine. Saatavissa: <https://www.smashingmagazine.com/2018/02/comprehensive-guide-to-mobile-app-design/>. Haettu 24.9.2018.
7. Nielsen, Jakob – Budiu, Raluca 2013. Mobile Usability. Berkeley, California: New Riders.
8. Arora, Harshita 2018. Designing beautiful mobile apps from scratch. FreeCodeCamp. Saatavissa: <https://medium.freecodecamp.org/designing-beautiful-mobile-apps-from-scratch-1a3441ebd604>. Haettu 24.9.2018.
9. Sinkkonen, Irmeli – Kuoppala, Hannu – Parkkinen, Jarmo – Vastamäki, Raino 2006. Käytettävyyden psykologia. Helsinki: Edita Prima Oy.
10. Nielsen, Jakob 1993. Usability Engineering. San Diego, California: Academic Press.

11. Käytettävyytestilaboratorion pohjaratkaisu -kuva. Saatavissa: <https://i.stack.imgur.com/ZKBjt.gif>. Haettu 27.12.2018.
12. Budiu, Raluca 2014. Usability Testing for Mobile Is Easy. Nielsen Norman Group. Saatavissa: <https://www.nngroup.com/articles/mobile-usability-testing/>. Haettu 27.11.2018.
13. Mobile Usability Testing Demonstration. 2010. Flickr. Saatavissa: <https://www.flickr.com/photos/cinteractionlab/4557104495>. Haettu 14.1.2019.