
**Käyttöliittymän suunnittelu kosketusnäytöisille
mobiililaitteille**



Ammattikorkeakoulun opinnäytetyö

Mediatekniikan koulutusohjelma

Riihimäki, syksy 2014

Jan-Erik Bergman



RIIHIMÄEN TOIMIPISTE
Mediatekniikan koulutusohjelma
Ohjelmistotekniikka

Tekijä	Jan-Erik Bergman	Vuosi 2014
Työn nimi	Käyttöliittymän suunnittelu kosketusnäytöisille mobiililaitteille	

TIIVISTELMÄ

Opinnäytetyön aiheena on käyttöliittymän toteuttaminen kosketusnäytöisille mobiililaitteille. Työllä ei ole toimeksiantajaa, vaan tein opinnäytetyön oman kiinnostukseni pohjalta. Opinnäytetyössä tutkitaan kuinka hyvä käyttöliittymä kosketusnäytöisille mobiililaitteille toteutetaan. Työssä on teoriaa käyttöliittymäsuunnittelusta ja käyttöliittymän toteuttaminen. Opinnäytetyön tarkoituksena oli selvittää miten käyttöliittymäsuunnittelu mobiililaitteille eroaa käyttöliittymäsuunnittelusta työpöytäympäristöille. Samalla myös selvitettiin, miten toteutetaan hyvän käytettävyyden omaava käyttöliittymä mobiililaitteille.

Työssä perehdyttiin aluksi mobiililaitteiden käyttöliittymäsuunnittelun teoriaan ja perusteisiin. Mobiililaitteilla käytettyjä suunnittelukäytäntöjä verrattiin pöytäkoneiden käytäntöihin ja perehdyttiin näiden kahden eroihin. Tämä vaihe toimi pohjana opinnäytetyön käytännön osuudessa toteutettavan mobiilisovelluksen käyttöliittymän toteutukselle.

Käyttöliittymän toteutuksessa keskityttiin helppokäyttöisen ja käyttäjäystävällisen käyttöliittymään suunnitteluun ja toteuttamiseen. Käyttöliittymän suunnittelu- ja toteutusvaiheet dokumentoitiin, minkä ohella käyttöliittymä myös suunniteltiin ja toteutettiin esiteltyjen käytäntöjen pohjalta.

Lopputuloksena syntyi yleispätevä katsaus mobiililaitteiden käyttöliittymäsuunnittelun perusteisiin. Opinnäytetyön pohjalta pystyy tulevaisuudessa jatkamaan käyttöliittymäsuunnitteluun perehtymistä esimerkiksi tutustumalla aiheeseen jonkin tietyn mobiililaitteen näkökulmasta.

Avainsanat käyttöliittymä, käyttöliittymäsuunnittelu, mobiililaitte, kosketusnäyttö

Riihimäki
Degree Programme in Media Technology
Software Engineering

Author	Jan-Erik Bergman	Year 2014
Subject of Bachelor's thesis	User interface design for mobile Touchscreen	

ABSTRACT

The topic of this Bachelor's thesis is user interface design for mobile devices with touchscreens. This thesis was developed out of personal interest of the author and therefore has no commissioner. In this thesis it is examined how to design a good user interface for mobile devices such as smartphones. The thesis covers theory of user interface design and the implementation of the user interface. The purpose of the thesis was to find out how user interface design for mobile devices differs from the desktop environment, and how to execute a user interface with good usability for mobile devices.

Initially the theory and basics of user interface for mobile devices is described. The design practices that are used on mobile devices are compared to the desktop computer design practices to highlight the differences of the practices. This phase serves as the basis for the user interface that is implemented in the practical phase of the thesis.

In the implementation of the user interface, the focus is on the design and implementation of a user interface that is user friendly and easy to use. The design and implementation phases are documented and the user interface has been designed and implemented on the basis of the presented practices.

The result is a universal survey on the basics of user interface design for mobile devices. With the help of this thesis it is possible to continue the study of user interface design, for example, by looking at user interface design for a particular mobile device.

Keywords user interface, user interface design, mobile device, touchscreen

Pages 30 p.

SISÄLLYS

1	JOHDANTO.....	1
2	KÄYTTÖLIITTYMÄN SUUNNITTELEMISEN MOBIILILAITTEELLE.....	1
2.1	Käyttöliittymän suunnittelemisen periaatteet mobiililaitteilla.....	2
2.2	Rajoitteet ja vahvuudet.....	10
3	KÄYTTÖLIITTYMÄN SUUNNITTELEMISEN EROAVAISSUUDET MOBIILILAITTEIDEN JA TYÖPÖYTÄLAITTEIDEN VÄLILLÄ.....	11
3.1	Käyttöliittymän suunnittelu pienille näytöille.....	12
3.2	Käyttöliittymän suunnittelu kosketusnäytölle.....	13
4	OMA PROJEKTI.....	15
4.1	Mahdolliset ympäristöt.....	15
4.2	Android-sovellus.....	16
4.2.1	Kehitysympäristö.....	16
4.2.2	Suunnittelu.....	17
4.2.3	Käyttöliittymä.....	17
4.2.4	Toiminnallisuus.....	20
4.2.5	Toteutus.....	22
5	YHTEENVETO.....	28
5.1	Lopputulos.....	28
5.2	Ongelmat.....	29
5.3	Havainnot.....	29
5.4	Pohdinta.....	29

1 JOHDANTO

Käytettävyys on mielestäni ollut aina yksi sovelluksen tärkeimmistä ominaisuuksista. Olen usein törmännyt sovelluksiin, joiden käyttäminen on ollut turhauttavaa. Monesti olen etsinyt tilalle vastaavan sovelluksen, jonka käytettävyyteen on panostettu enemmän. Tuomio hyvän ja huonon käytettävyyden välillä on syntynyt omalla kohdallani todella nopeasti.

Muutama vuosi sitten ostin ensimmäisen älypuhelimeni. Tämän seurauksena huomasin, että myös älypuhelimien sovelluksissa on havaittavissa käytettävyysongelmia. Ne erosivat kuitenkin suuresti työpöytäsovellusten ongelmista.

Mikä sitten on hyvää käytettävyyttä mobiililaitteilla? Kiinnostuin tästä kysymyksestä sovelluskaupan valikoimaa tutkiessani. Tämän opinnäytetyön tarkoituksena on vastata tähän kysymykseen teorian ja käytännön tasolla.

2 KÄYTTÖLIITTYMÄN SUUNNITTELEMINEN MOBIILILAITTEELLE

Jakob Nielsenin vuonna 2009 tekemää tutkimusta mobiilialustojen käyttäjäkokemuksesta voidaan verrata vuoden 1998 työpöytäympäristön käyttäjäkokemukseen. Tutkimuksessa käytettiin internetsivuja mobiililaitteilla. Haasteita aiheuttivat muun muassa huono onnistumisprosentti, hitaat latausajat, vierityksen aiheuttamat ongelmat, liian täydet sivut, Javascript, video ja alustalle sopimattomat käyttöliittymät.

Huonolla onnistumisprosentilla tarkoitetaan tilannetta, jossa käyttäjä epäonnistuu useammin kuin onnistuu yrittäessään suorittaa haluamaansa toimintoa. Onnistumisprosentti oli jatkuva ongelma koko Nielsenin tutkimuksen ajan.

Tutkimuksessa huomattiin useimpien internetsivujen latautuvan aivan liian hitaasti. Tästä seurasi, että latausajat hallitsivat käyttäjäkokemusta eivätkä käyttäjät halunneet ladata ylimääräisiä sivuja vaan mieluummin luovuttivat. Myös vieritys aiheutti suuria käytettävyysongelmia, koska sivustoja ei ollut optimoitu mobiililaitteille. Mobiililaitteiden pieni näyttö ja tarve vierittää ikkunaa saivat käyttäjän kadottamaan sijaintinsa sivulla.

Monet sivut oli suunniteltu suurilla pöytäkoneiden näyttöjä varten, jolloin pienillä mobiililaitteiden näytöillä sivustot olivat liian täysiä. Käyttäjät eksyivät täysillä sivuilla ja samalla viitekehys katosi. Lisäksi Javascript aiheutti selaimen kaatumisia. Myös video ja muut mediat aiheuttivat käytettävyysongelmia.

Käyttöliittymien sopimattomuus mobiililaitteille johtui siitä, että sivustot oli suunniteltu pöytäkoneille. Tästä syystä sivustojen käyttö mobiililaitteella oli vaikeaa. (Nielsen & Budiu 2013, 9–11.)

Käytettävyys on sovelluksen tärkein ominaisuus. Sovelluksen muilla ominaisuuksilla ei ole väliä, jos käyttäjä ei löydä tai osaa käyttää niitä. Mobiilialustojen sovelluskaupoissa on aina vaihtoehtoja. Jos sovellus ei ole toimiva, käyttäjä poistaa sovelluksen armottomasti.

Kun käyttäjä käyttää sovellusta ensimmäistä kertaa, hän luo sen toiminnallisuudesta ajatusmallin. Tämän ajatusmallin avulla käyttäjä simuloi sovelluksen toimintaa mielessään ennustaakseen mitä tapahtuu ja milloin. Käyttöliittymän tulee tukea käyttäjän ajatusmallia, koska sovelluksen pitää olla johdonmukainen ja ennalta arvattava.

Sovellus on helppokäyttöinen, jos sen toiminta vastaa johdonmukaisesti käyttäjän ajatusmallia ja toisaalta intuitiivinen, jos käyttäjän on helppo luoda ajatusmalli. Jokainen ongelma, jonka käyttäjä kohtaa, johtuu epäjohdonmukaisuudesta käyttäjän ajatusmallin ja sovelluksen toiminnallisuuden välillä. (Lehtimäki 2012, 5.)

Käyttäjät tulevat useista eri kulttuureista ja useista eri taustoista. Jokainen käyttäjä luo ajatusmallin eri tavalla ja odottaa täten eri asioita. Heillä on myös eri kokemuksia ohjelmistojen ja ohjelmistoalustojen kanssa. Esimerkiksi: koko elämänsä Mac-tietokoneita käyttänyt henkilö odottaa käyttöliittymän toimivan eri tavalla kuin Windows-ympäristöä käyttänyt henkilö. (Lehtimäki 2012, 7.)

Vuosi vuodelta Jakob Nielsenin tutkimukset vahvistavat ensimmäisen tutkimuksen tulokset. Käyttökokemus mobiililaitteilla kuitenkin paranee hitaasti sen ansiosta, että yhä useampi sivusto tarjoaa mobiilioptimoitun version sivustostaan ja suunnittelijat ymmärtävät pienen näytön asettamat rajoitteet paremmin.

Vastaavasti myös käyttäjien onnistumisprosentti kasvaa hitaasti. Tämä pitää paikkansa etenkin silloin, kun käyttäjä käyttää internetsivujen sijaan mobiilisovellusta. Javascript aiheuttaa vielä nykyäänkin kaatumisia, mutta kehittäjät panostavat enemmän mobiisisivustojen ja sovelluksien testaamiseen, jotta ne toimisivat paremmin. Toisaalta latausajat ovat edelleen ongelma. Normaalien sivustojen käyttäminen mobiililaitteilla on edelleen hankalaa. (Nielsen & Budiu 2013, 14.)

2.1 Käyttöliittymän suunnittelemisen periaatteet mobiililaitteilla

Johdonmukaisuus on tärkeää käyttöliittymäsuunnittelussa. Tämän takia käyttöliittymäelementtejä ei tulisi sijoitella umpimähkään sivulle, vaan tulisi noudattaa selkeää ohjenuoraa ja käyttöjärjestelmän esimerkkiä. Johdonmukaisesti käytettävän ja brändätyn kokemuksen saavuttamiseksi tulee suunnitella yleinen asetelma ja hierarkia koko sovellukselle. Tämän asetelman pohjalta voidaan toteuttaa yksittäisien sivujen ulkonäkö. (Hoover & Berkman 2012, 2.)

Käyttäjä käyttää ja näkee sovelluksen käyttöliittymän kautta. Kaikki alla piilevä heijastuu käyttöliittymään. Sovelluksen toimintojen tulee olla intuitiivisesti toteutettu, koska muuten niiden toteuttamiseen käytetty aika on valunut hukkaan, kun käyttäjä ei edes löydä niitä. (Lehtimäki 2012, 3.) Käyttäjät haluavat käyttää sovelluksia, jotka ovat helppoja käyttää. Toimintojen tulisi olla helposti löydettävissä ja sopivan kokoisia. Näin ollen ne olisivat helppolukuisia ja täten helposti käytettäviä. (McWherter 2009, 93.)

Käyttöliittymäsuunnittelu mobiililaitteille on vaikeaa. Yleisimmät käytettävyysongelmat mobiililaitteilla syntyvät näytön pienestä koosta, hankalasta syötteestä, hitaista latausajoista ja huonosti suunnitelluista sivuista. Pienien ruutujen vuoksi käyttäjä kuormittaa lyhytkestoista muistiaan rakentaessaan käsitystä ympäröivästä informaatiosta. Tilanpuute vaikeuttaa myös monien käyttöliittymäratkaisujen käyttöä. Lisäksi mobiililaitteilla kirjoittaminen on hankalaa ja virhealtista, minkä ohella myös navigointi on hiireen verrattuna hidasta. Hitaiden latausaikojen takia uusien sivujen lataaminen kestää kauan. Sivujen käyttäminen on myös usein haastavaa, koska ne on optimoitu työpöytäympäristöä varten. (Nielsen & Budiu 2013, 49–51.)

Projektin alussa tulee miettiä mitä toimintoja sovellus sisältää ja mitä ei. Sovellus, joka tukee täydellisesti pienempää määrää käyttäjän tavoitteita, on parempi kuin sovellus, joka tekee kaikkea huonosti. (Lehtimäki 2012, 13.) Mobiilialustoille suunniteltaessa tulee leikata turhia mobiiliympäristöön sopimattomia ominaisuuksia pois ja vähentää tekstien sanamäärää. Myös toissijainen informaatio tulee jakaa eri sivuille. Käyttöliittymäelementtejä tulee suurentaa, jotta navigointi sivuilla kosketuksen avulla olisi helpompaa. (Nielsen & Budiu 2013, 20.)

Mobiililaitteen ruudun pinta-ala on arvokasta. Tämän takia paljon tilaa vievien bannereiden, kuvien ja grafiikoiden käyttöä tulisi välttää, jos niiden käytölle ei ole hyvää syytä. Hooper ja Berkman suosittelevat asettelemaan käyttöliittymäelementit suunnitteluhierarkian mukaan. Tämä hierakia on seuraavanlainen: sijainti, koko, hahmo, kontrasti, väri, muoto. Toisin sanoen tärkeimmät elementit ovat korkeammalla, suurempia ja niin edelleen. Myös hahmopsykologia tulisi ottaa huomioon käyttöliittymää suunniteltaessa.

Navigaatioelementtien tulisi olla selkeitä ja johdonmukaisia. Koska ihmisen lyhytkestoinen muisti on rajallinen, ihminen kiinnittää huomiota vain tärkeään informaatioon ja suodattaa pois usein toistuvan ja epäolennaisen informaation.

Hooperin ja Berkmanin mukaan sovelluksissa navigoiminen tapahtuu saman logiikan mukaan kuin oikeassa maailmassa. Ympäristöpsykologi Kevin Lynch kartoitti viisi navigoimisen osa-aluetta, jotka ovat polut, reunat, yhtymäkohdat, maamerkit ja piirit. Myös digitaalisen sisällön navigoiminen tapahtuu näiden samojen osa-alueiden avulla. Esimerkiksi

sivunumerot, otsikot, ylätunnisteet ja alatunnisteet, välilehdet ja linkit ovat lähes kokonaisuudessaan lainattu interaktiiviseen suunnitteluun.

Yleensä käyttäjät etsivät tärkeää informaatiota sisältöalueen vasemmasta yläkulmasta. Tämä tulee ottaa huomioon sivun sisältöä suunniteltaessa. Pitkät tekstirivit ovat vaikeampia lukea. Keskimäärin 60–65 kirjainta on maksimipituus, jota kannattaa käyttää. Tekstirivien pituutta voi rajoittaa käyttämällä monisarakkeista tekstiä. Otsikot kuvaavat sivujen sisältöä, joten niitä tulisi käyttää johdonmukaisesti. Teksti kohdistetaan oletuksena vasemmalle. Oikealta vasemmalle luettavien kielten teksti kohdistetaan oletuksena oikealle. (Hooper & Berkman 2012, 3–4.)

Jokainen tärkeä elementti tulee otsikoida, jotta asiayhteys olisi selvä. Otsikot ovat käyttöjärjestelmien, sovellusten ja verkkostandardien keskeisiä osia mutta niiden sisällyttäminen tarkoituksenmukaisesti suunnitelmaan on suunnittelijan harteilla. Myös sivujen ja elementtien tai sisältöalueiden tulisi lähes aina olla otsikoituja. Ponnahdusikkunoilla ja muilla kelluvilla ikkunoilla tulisi olla vastaavanlaiset otsikot kuin sivulla. (Hooper & Berkman 2012, 26.)

Mobiliisovelluksia on olemassa kolmenlaisia. Nämä ovat natiivisovellus, verkkosovellus ja hybridisovellus. Natiivisovellus on kehitetty yhtä ainoaa alustaa varten ja yleensä niiden käyttöliittymät mukailevat alustan käytäntöjä. Käyttäjät käynnistävät natiivisovelluksen ikonista laitteensa aloitusnäytöstä. (Nielsen & Budiu 2013, 40.)

Verkkosovellus pyörii selaimessa, joten se ei ole niin alustariippuvainen (Lehtimäki 2012, 37). Käyttäjät käynnistävät sovelluksen menemällä internetsivulle. Verkkosovelluksen voi asentaa puhelimeen luomalla kirjanmerkin sivulle. Sovellukset näyttävät natiivisovelluksilta, mutta ovat rajoittuneempia sen suhteen mihin puhelimen ominaisuuksiin ne pääsevät käsiksi. (Nielsen & Budiu 2013, 40.) Koska iOS ja Android käyttävät samaan WebKit-selainmoottoriin perustuvia selaimia, on mahdollista toteuttaa verkkosovellus, joka toimii molemmilla alustoilla. (Lehtimäki 2012, 37.)

Hybridisovellukset ovat natiivisovelluksia, jotka pitävät sisällään selaimen. Hybridisovellukset asennetaan puhelimeen samalla tavalla kuin natiivisovellukset, mutta osa sovelluksesta toistaa internetsivuja. Hybridisovelluksia luodaan yleensä toistamaan jo olemassa olevia internetsivuja. (Nielsen & Budiu 2013, 40.)

Mutta milloin toteuttaa natiivisovellus ja milloin verkkosovellus? Lehtimäen mukaan sovellus, jonka käyttö on epäsäännöllistä, tulisi toteuttaa verkkosovelluksena. Sovellus, jota käytetään usein, tulisi taas toteuttaa natiivisovelluksena. Myös tekniset seikat tulee ottaa huomioon päätöstä tehtäessä. Verkkosovellus ei pääse käsiksi kaikkiin alustan ohjelmistorajapintoihin ja vaatii jatkuvan verkkoyhteyden toimiakseen täysin. (Lehtimäki 2012, 37.)

Nielsen toteaa, että käyttöliittymäsuunnittelussa käyttöliittymää ei tulisi kääntää suoraan toiselle alustalle. Käytännössä tämä tarkoittaa esimerkiksi sitä, että ei ole suositeltavaa lisätä kosketuskontrolleja pöytäkonetta varten suunniteltuun käyttöliittymään. Lisäksi graafisten objektien määrä tulisi pitää vähäisempänä kosketuskäyttöliittymissä, jotta objektien koko voisi olla suurempi. (Nielsen & Budiu 2013, 41.) Toisaalta myös kahden käyttöliittymän toteuttamista samalle sisällölle tulee välttää. Kahden käyttöliittymän toteuttaminen on oikeutettua vain, jos ne korostavat eri näkökulmia sisällöstä. (Nielsen & Budiu 2013, 54.)

Käyttöliittymäelementtien määrää tulee pitää silmällä. Tilaa tuhlautuu varsinkin kun sovelluksia on auki useammalla tasolla sisäkkäin ja jokaisella tasolla on omat käyttöliittymäelementtinsä. Tämän takia Nielsen suosittelee piilottamaan osan käyttöliittymäelementeistä ja näyttämään ne vain tarvittaessa. Toisaalta tässä piilee omat vaaransa, koska käyttöliittymäsuunnittelussa ei tulisi luottaa lyhytkestoiseen muistiin.

Piiloutuvat käyttöliittymäelementit toimivat, jos käyttöliittymäelementit paljastava operaatio on yksinkertainen ja luotettava. Käyttöliittymän on oltava myös johdonmukainen, jotta käyttöliittymäelementtien olemassaolo tallentuu käyttäjän pitkäaikaiseen muistiin toiston avulla. Käyttäjää on myös autettava sopivien vihjeiden ja progressiivisen paljastamisen avulla. Vihjeitä tulisi näyttää yksi kerrallaan ja oikeassa asiayhteydessä. (Nielsen & Budiu 2013, 57–59.) Progressiivisella paljastamisella tarkoitetaan sovelluksen tai internetsivun ominaisuuksien jakamista useaan näkymään. Tärkeät ja usein käytetyt toiminnot tulee sijoittaa päänäkymään ja vähemmän käytetyt ja monimutkaiset toiminnot toissijaisiin näkymiin. (Wikipedia 2014).

Lähes jokaisessa nykypäivänä valmistetussa älypuhelimessa on kosketusnäyttö ensisijaisena ohjauslaitteena. Kosketusnäytöllä käyttöliittymän ja fyysisten hallintalaitteiden välillä on suora yhteys. Käyttäjä käyttää kosketuskäyttöliittymää eleiden avulla. Käyttöjärjestelmä tai sovellus tulkitsee käyttäjän sormellaan kosketusnäytölle piirtämät eleet komennoiksi, jotka suorittavat jonkin toiminnon. Eleitä on olemassa useita erilaisia, jotka vaihtelevat yhden sormen eleistä usean sormen avulla toteutettaviin eleisiin. (Lehtimäki 2012, 38.) Taulukossa 1 ovat Android-käyttöjärjestelmän vakiona tukemat kosketuseleet.

Taulukko 1. Androidin kosketuseleet (Lehtimäki 2012, 38–39.)

Ele	Kuvaus
Napautus	Käyttäjä koskettaa ruutua
Kaksoisnapautus	Käyttäjä napauttaa kaksi kertaa nopeasti samaa kohtaa
Pyyhkäisy	Käyttäjä vetää sormea näyttöä pitkin nopeasti yhteen samaan suuntaan
Raahaus	Käyttäjä vetää sormea näyttöä pitkin (Ei yhtä nopea kuin pyyhkäisy, eikä tarvitse vetää samaan suuntaan)
Nipistys, aukeneva	Käyttäjä vetää kahta sormea näytöllä pois päin toisistaan
Nipistys, sulkeva	Käyttäjä vetää kahta sormea näytöllä toisiaan kohti
Napauta ja paina pohjassa	Käyttäjä napauttaa näyttöä nostamatta sormeaan pohjassa

Nielsenin mukaan käyttöliittymäelementit voi korvata eleillä. Esimerkiksi sähköisten kirjojen lukusovelluksessa voidaan eliminoida nappeja käyttämällä pyyhkäisyelementtiä. (Nielsen & Budiu 2013, 59.) Eleet ovat kuitenkin Nielsenin mukaan vaikeasti muistettavia. Jos sovelluksessa on useita eri eleitä, voi syntyä häiriöitä ja eleet voivat myös unohtua käyttäjältä. Käyttäjällä voi myös olla vaikeuksia muistaa mikä ele tekee mitään. (Nielsen & Budiu 2013, 61.)

Nielsenin mukaan parhaiten toimivat eleet ovat kytköksissä oikeaan maailmaan. Esimerkkeinä Nielsen nostaa esiin sivujen käynnön pyyhkäisyllä ja nipistys-eleen. Nipistys-eleellä näyttöä ikään kuin nipistetään, jolloin kuva pienenee ja venytetään, jolloin kuva suurenee. (Nielsen & Budiu 2013, 62.)

Suunnittelijoiden tulisi ottaa huomioon eri käyttäjien tarpeet. Toisaalta käyttäjät haluavat paljon ominaisuuksia ja valinnanvaraa, mutta toisaalta taas yksinkertaisuutta. Progressiivisen paljastamisen avulla voidaan saavuttaa nämä molemmat tavoitteet. Aluksi tulisi näyttää vain tärkeimmät ominaisuudet ja valinnat mutta tarjota käyttäjälle pääsy vähemmän tärkeisiin ominaisuuksiin. Harvinaisempien valintojen piilottaminen auttaa aloittelevaa käyttäjää kiinnittämään huomiota heille tärkeisiin ominaisuuksiin ja välttymään virheiltiltä. Myös edistyneet käyttäjät säästävät aikaa, kun valintoja ei ole esitetty pitkänä listana. On kuitenkin tärkeää tehdä oikea jako tärkeiden ja harvinaisten ominaisuuksien välillä, jotta progressiivinen paljastaminen toimisi. (Nielsen & Budiu 2013, 60.)

Nielsenin tutkimuksen mukaan käyttäjät vihaavat kirjoittamista mobiililaitteilla. Fyysisiä näppäimistöjä käyttäessään käyttäjä oppii painikkeiden sijainnit, minkä johdosta käyttäjän ei tarvitse katsoa näppäimistöä kirjoittaessaan. Kosketusnäyttöjen virtuaalinäppäimistöt taas vaativat käyttäjältä enemmän huomiota, mikä tekee kirjoittamisesta

haastavampaa. Kosketusnäytöllä käyttäjän on myös vaikea tietää osuiko hän painikkeeseen. Tämä ongelma on helppo korjata haptisella palautteella. (Nielsen & Budiu 2013, 81.) Haptinen teknologia käyttää hyväksi tuntoaistia. Mobiililaitteilla haptista teknologiaa käytetään, kun käyttäjä koskettaa näyttöä ja laite värähtää vastaukseksi käyttäjän tekemään syötteeseen. (Wikipedia 2014).

Nielsen kehottaa välttämään pakollista rekisteröitymistä mobiisisovelluksen ensimmäisenä käyttökokemuksena. Sovelluksen käytön alkuvaiheessa käyttäjät eivät ole vielä tarpeeksi sitoutuneita nähdäkseen tarpeeksi vaivaa rekisteröidäkseen itselleen käyttäjätiliä. Rekisteröityminen voi tarjota käyttäjälle parempaa käytettävyyttä, mutta ensin sovelluksen on todistettava käyttäjälle arvonsa, sillä muuten se saattaa karkottaa käyttäjän. (Nielsen & Budiu 2013, 81.)

Mobiililaitteilla vierityspalkkia käytetään ilmaisemaan vieritettävät akselit ja suhteellinen sijainti vieritettävällä alueella. Vierityspalkkeja ei tulisi voida liikuttaa suoraan, vaan sallia sisältöalueen vieritys eleen tai erillisten suuntanäppäimien avulla. (Hoover & Berkman 2012, 12.)

Ilmoituspalkin tulisi olla näkyvissä kaikissa näkymissä. Se voidaan piilottaa tai sen näkyvyyttä voidaan vähentää esimerkiksi kokonäytön videoistoa tai pelaamista varten. (Hoover & Berkman 2012, 18.) Ilmoituspalkki sijaitsee näkymän yläalareunassa ja kertoo laitteiston ominaisuuksien tilan, kuten akun varauksen. (Hoover & Berkman 2012, 10.)

Ilmoituspalkin piilottaminen lisätilan saamiseksi tulisi harkita tarkkaan. Esimerkiksi videon toistamista, useimpia pelejä ja diaesityksiä varten ilmoituspalkin piilottaminen on sopivaa. Selaimet, e-lukulaitteet ja PDF-lukuohjelmat voivat myös hyötyä ilmoituspalkin piilottamisesta vierityksen ja lukemisen aikana. (Hoover & Berkman 2012, 18.)

Ammattislangia ja sisäisten prosessien paljastamista tulisi välttää. Myös liian ankaria virheilmoituksia ja muita asioita, jotka saattavat hämmentää tai ärsyttää tyyppillistä käyttäjää tulee välttää. (Hoover & Berkman 2012, 29.) Nielsen taas kehottaa seuraamaan käytettävyyden peruskäskyä: ”Puhu käyttäjän omaa kieltä” (Nielsen & Budiu 2013, 111).

Useat sivustot, sovellukset ja mobiilipalvelut ovat mainosrahoitteisia. Jos tuotteeseen pitää asettaa mainoksia, niitä ei tule piilotella, mutta ei myöskään tehdä niistä niin silmiinpistäviä, että ne vahingoittavat käyttäjäkokemusta. Mobiilialustoilla mainonta tulee olla helposti erotettavissa sovelluksen sisällöstä. Mainoksien tulee olla samassa paikassa jokaisessa näkymässä ja helposti käytettävissä. (Hoover & Berkman 2012, 52.)

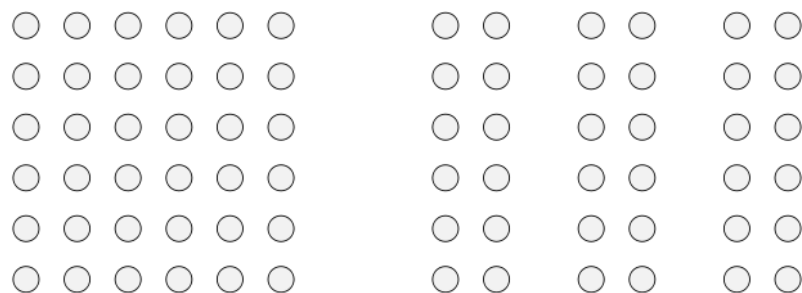
Kaksi yleistä virhettä mainonnan toteuttamisessa ovat liian ilmiselvä mainonta ja mainokset jotka eivät erotu sovelluksen sisällöstä. Mainoksien tulee olla selkeästi erotettavissa sisällöstä ja sovelluksen toiminnallisuuden näkökulmasta huomaamattomia. Sovelluksessa esiintyvien mainoksien

tulee olla samalla tavoin sijoitettu ja samaan tyyliin toteutettuja. On myös tärkeää, että mainokset on helposti luettavissa ja helposti käyttäjän käytettävissä. (McWherter 2009, 107.)

Kun käyttäjät käyttävät sovellusta, he haluavat saada jotain aikaan. Käyttäjät arvioivat sovelluksen sen perusteella kuinka hyvin se auttaa heitä saavuttamaan tavoitteensa. Käyttäjä on tyytyväinen sovellukseen, jos kehittäjä tunnistaa käyttäjien tavoitteet, joita sovellus tukee, ja suunnittelee käyttöliittymän tukemaan näiden tavoitteiden saavuttamisessa. (Lehtimäki 2012, 10.)

McWherter suosittelee ottamaan huomioon hahmopsykologian periaatteet käyttöliittymäsuunnittelussa. Hahmopsykologian periaatteiden mukaan käyttäjä käsittää jokaisen ärsyksen sen yksinkertaisimmassa muodossa. Käyttöliittymäsuunnittelun kannalta keskeisiä periaatteita ovat läheisyys, sulkeutuvuus, jatkuvuus, kohteen jäsentäminen kuvioksi ja taustaksi sekä samankaltaisuus.

Hahmopsykologian läheisyyden lain mukaan käyttäjillä on tapana ryhmitellä objekteja yhteen. Tämän vuoksi lähelle toisiaan asetellut objektit havainnoidaan ryhmänä. Tätä hyväksi käyttäen käyttöliittymää suunniteltaessa ikonit, jotka suorittavat samankaltaisia toimintoja, on hyvä järjestää lähelle toisiaan. Kuvassa 1 on esimerkki hahmopsykologian läheisyyden laista.



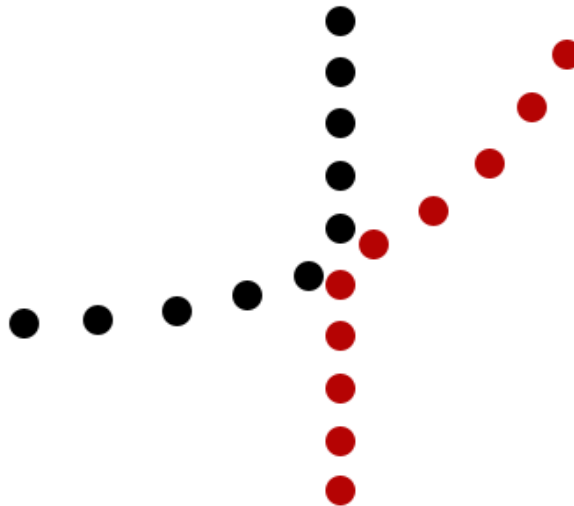
Kuva 1. Hahmopsykologian läheisyyden laki (Wikipedia: Gestalt psychology[viitattu 15.9.2014]. Saatavissa: http://en.wikipedia.org/wiki/Gestalt_psychology)

Sulkeutuvuuden lain mukaan, jos muotoa on tarpeeksi näkyvissä, ihmismieli täydentää puuttuvat osat. Sulkeutuvuuden lain näkökulmasta ikoneja suunniteltaessa tulisi tehdä vahva ensisijainen siluetti ja jättää käyttäjää kuormittavat yksityiskohdat toteuttamatta. (McWherter 2009, 91.) Kuvassa 2 on esimerkki hahmopsykologian sulkeutuvuuden laista.



Kuva 2 Hahmopsykologian sulkeutuvuuden laki (Wikipedia: Gestalt psychology[viitattu 15.9.2014]. Saatavissa: http://en.wikipedia.org/wiki/Gestalt_psychology)

Hyvän jatkon lain mukaan objektit, jotka risteävät keskenään jatkuvat risteyskohdissa niin, että niiden suunta muuttuu mahdollisimman vähän. Tätä voidaan käyttää mobiilikäyttöliittymäsuunnittelussa hyväksi käyttäjien opastuksessa. Linkki voi esimerkiksi nuolen avulla osoittaa seuraavaan linkkiin. Kuvassa 3 on esimerkki hyvän jatkon laista.



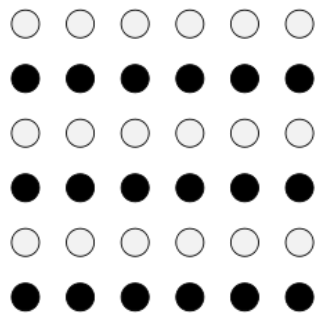
Kuva 3. Hahmopsykologian hyvän jatkon laki (Steven Bradley: Gestalt Principles: How Are Your Designs Perceived?[viitattu 15.9.2014]. Saatavissa: <http://www.vanseodesign.com/web-design/gestalt-principles-of-perception/>)

Kuvion jäsentäminen kuvioksi ja taustaksi voi yksinkertaisimmillaan tarkoittaa sivulla olevaa kirjainta, jonka ympärillä on valkoista aluetta. Monimutkaiset designit voivat leikkiä tämän käsitteen kanssa, mutta mobiilialustoilla käyttöliittymissä tulisi olla selvä erottelu kuvion ja taustan välillä. Kuvassa 4 on esimerkki kuvion jäsentämisestä kuvioksi ja taustaksi.



Kuva 4. Hahmopsykologian kuvion jäsentäminen kuvioksi ja taustaksi (Steven Bradley: Gestalt Principles: How Are Your Designs Perceived?[viitattu 15.9.2014]. Saatavissa: <http://www.vanseodesign.com/web-design/gestalt-principles-of-perception/>)

Samankaltaisuuden lain mukaan samankaltaiset kohteet ryhmitellään yhteen. Asettelun tulisi edistää objektien oikeaa ryhmittelyä, eikä sekoittaa käyttäjää yhdistämällä toisistaan erillisiä objekteja yhteen. (McWherter 2009, 92.) Kuvassa 5 on esimerkki hahmopsykologian samankaltaisuuden laista.



Kuva 5. Hahmopsykologian samankaltaisuuden laki (Wikipedia: Gestalt psychology [viitattu 15.9.2014]. Saatavissa: http://en.wikipedia.org/wiki/Gestalt_psychology)

2.2 Rajoitteet ja vahvuudet

R.I. Singh ja kollegat Albertan yliopistosta tulivat tutkimuksessaan siihen lopputulokseen, että monimutkaista tietoa on vaikeampi ymmärtää, kun se luetaan pienen aukon kautta. Singhin tutkimuksessa 50 osallistujaa suorittivat aukkotehtäviä lukien eri sivustojen rekisteriselosteita (privacy policy) joko työpöydän näytöltä tai iPhoneen ruudun kokoiselta näytöltä. Työpöydän näytöltä luetunymmärtämisprosentti oli 39.18 prosenttia, kun mobiililaitteen näytöltä se oli 18.93 prosenttia. (Nielsen & Budiu 2013, 102.)

Mobiililaitteita käyttäneiden huonompi menestys tutkimuksessa johtui pienemmästä näytöstä, koska se oli ainoa eroavaisuus tutkimusryhmien

välillä. Nielsenin mukaan pienempi ruutu heikentää käyttäjän kykyä ymmärtää lukemaansa kahdesta syystä. Käyttäjä joutuu turvautumaan omaan muistiinsa yrittäessään ymmärtää asioita, jotka eivät mahdu kokonaan ruudulle ja käyttäjän pitää vierittää ruutua päästäkseen käsiksi eri kohtiin tekstiä.

Vierittäminen luo myös omat ongelmansa. Kun käyttäjä vierittää ruutua, kuluu aikaa ja käyttäjän muisti heikkenee. Vierittäminen myös kiinnittää käyttäjän huomiota oikean kohdan etsimiseen tekstistä, mikä vaikeuttaa käyttäjän mahdollisuutta ymmärtää. Myös edellisen sijainnin löytäminen sivulla vaikeutuu verrattuna suureen näyttöön, jolla koko teksti mahtuu näkyviin. (Nielsen & Budiu 2013, 104–105.) Koska pieni ruutu heikentää ymmärrystä, sitä ei tulisi täyttää tarpeettomalla tiedolla (Nielsen & Budiu 2013, 106).

Koska mobiililaitteita käytetään paikoissa, joissa sovellus joutuu taistelemaan käyttäjän huomiosta, tulee käyttöliittymän olla käytettävä pelkän vilkaisun perusteella. Useimmissa tilanteissa käyttöliittymä ei saa vaatia käyttäjän täyttä huomiota. Käyttöliittymän tulee tämän takia olla intuitiivinen ja kaikkien kontrollien tulee olla selvästi näkyvillä. (Lehtimäki 2012, 32.)

3 KÄYTTÖLIITTYMÄN SUUNNITTELEMISEN EROAVAISUUDET MOBIILILAITTEIDEN JA TYÖPÖYTÄLAITTEIDEN VÄLILLÄ

Mobiilialustat eroavat työpöytäympäristöstä usealla eri tavalla. Mobiililaitteita esimerkiksi käytetään erilaisissa ympäristöissä kuin pöytäkoneita. Myös käyttötavat eroavat toisistaan. Mobiililaitteita käytetään pääsääntöisesti informaation kuluttamiseen, kun taas tietokoneita informaation tuottamiseen. (Lehtimäki 2012, 31.)

Mobiilialustoille suunnitteleminen on vaikeaa, koska käyttökokemusten ongelmat tulevat paremmin ilmi. Kognitiivinen kuormitus kasvaa, kun huomio kiinnittyy navigoimiseen, minkä lisäksi samalla pitää muistaa mitä ruudulla oli ja löytää uudelleen alkuperäinen asiayhteys.

McWherter suosittelee käyttämään minimalismia hyväksi käyttöliittymää suunniteltaessa ja rajoittamaan yhdellä ruudulla olevien ominaisuuksien määrää. Näkymässä ei tulisi olla mitään turhaa vaan kaikella pitäisi olla tarkoitus. Selkeän informaatiohierarkian avulla voidaan auttaa käyttäjää selättämään kognitiiviset häiriötekijät.

Tärkeää sisältöä tulisi korostaa visuaalisesti. Suuremmat ja intensiivisemmän väriset objektit tai elementit kiinnittävät helpommin käyttäjän huomion. Käyttäjät selaavat nopeammin läpi vaaleammat värikontrastit, vaaleammat sävyt, pienemmät objektit ja paljon tekstiä sisältävät kappaleet. Johdonmukainen hierarkia auttaa saavuttamaan hyvän käytettävyyden. Mobiilikäyttöliittymiä suunniteltaessa hierarkiaa voidaan luoda sijainnin, koon, muotojen, värien ja kontrastin avulla. (McWherter 2009, 90.)

Tietokoneet käyttävät syötelaitteena hiirtä, kun taas mobiililaitteita käytettäessä syötelaitteena toimii usein käyttäjän sormi. Kun suunnitellaan käyttöliittymää tietylle alustalle, on tärkeää ottaa huomioon syötelaitteen vahvuudet ja heikkoudet. (Nielsen & Budiu 2013, 24.) Hiiri- ja kosketus-syötteiden eri vahvuudet on yksi pääsyy miksi työpöytä- ja mobiiliympäristöön tulee toteuttaa alustoille varta vasten suunnitellut käyttöliittymät. Muita syitä ovat esimerkiksi työpöytäympäristössä saatavilla oleva fyysinen näppäimistö ja suuri ero näyttöjen kokojen välillä. (Nielsen & Budiu 2013, 24.)

Koska mobiililaitteet ottavat virtansa akuista, sovellukset tulee optimoida säästämään virtaa. Käyttäjät poistavat liikaa akkua kuluttavat sovellukset hyvin nopeasti. (Lehtimäki 2012, 33.) Kun aletaan suunnitella sovellusta, tulee miettiä mitä tehtäviä sen tulee suorittaa. Taulukossa 2 listataan yleisimmät Android-käyttöjärjestelmän toiminnot ja niiden virrankulutus. (Lehtimäki 2012, 34.)

Taulukko 2. Yleisimpien toimintojen virrankulutus Android-käyttöjärjestelmässä (Lehtimäki 2012, 34.)

Toiminto	Virrankulutus
Verkon käyttäminen WiFi:llä	Keskisuuri
Verkon käyttäminen hyvällä kuuluvuusalueella	Keskisuuri
Verkon käyttäminen huonolla kuuluvuusalueella	Korkea
Bluetooth	Keskisuuri
GPS	Keskisuuri
Paikantaminen puhelinverkon avulla	Matala
NFC (near-field communication)	Matala
Ruudun sulkemisen estäminen	Korkea
Mikrofonin käyttäminen	Matala
Kameran käyttäminen	Keskisuuri
Puhelimen herättäminen lepotilasta	Korkea

Mobiilisovellusten käyttäjätestaaminen on Lehtimäen mukaan haastavaa, koska mobiilisovelluksia käytetään usein liikkeessä. Jos sovellusta testataan hiljaisessa toimistossa, saattavat kohdatut käytettävyysongelmat erota suuresti niistä, joita löydettäisiin oikeassa ympäristössä, jossa sovellusta käytetään. Tämä tulee pitää mielessä, kun käyttäjätestauksen tuloksia tarkastellaan. (Lehtimäki 2012, 29.)

3.1 Käyttöliittymän suunnitteleminen pienille näytöille

Nielsen toteaa, että ruutuala on äärimmäisen arvokasta, joten sitä ei tule tuhjata mutta ei myöskään säästää. Suunnittelijan tulisi siis löytää

tasapaino tilankäytön ja objektien määrän suhteen. Näyttö on mobiililaitteilla niin pieni, että Nielsenin mukaan tilaa ei tulisi haaskata. (Nielsen & Budiu 2013, 52.)

Mobiililaitteilla käyttöliittymä on tärkeää suunnitella niin, että käyttäjä pääsee helposti käsiksi sisältöön. Tätä vaikeuttaa kuitenkin se, että mobiililaitteiden näytöt ja resoluutiot ovat pienempiä kuin pöytätietokoneiden. Pienet näytöt vaativat suuremmat painikkeet, mikä taas rajoittaa ruudulle mahtuvan sisällön määrää. Jopa suuremmat näytöt omaavat mobiililaitteet vaativat suuremmat painikkeet, kuin vastaavan kokoinen tietokone, jota käytetään hiirellä. Myös käyttäjän oma sormi ja käsi voivat peittää osan näytöstä käytön aikana. (Hoover & Berkman 2012, 199.)

Nielsenin mukaan onnistuneen mobiilisovelluksen tai -sivuston suunnittelussa tulee ottaa huomioon ruudun pieni koko. Muussa tapauksessa käyttökokemusta haittaavat liian pienet painikkeet, joita on vaikea painaa. Myös ominaisuuksia tulee karsia ja jättää jäljelle vain mobiilikäytön näkökulmasta merkitykselliset ominaisuudet. (Nielsen & Budiu 2013, 52.)

Mobiililaitteiden pienien näyttöjen takia sisältö ja toiminnot pitää priorisoida käyttäjän tavoitteiden ja tarpeiden mukaan. Pääsy sisältöön tulee olla havaittavissa pelkällä vilkaisulla, koska mobiililaitteen käyttäjän huomio herpaantuu helposti ympäristöön. Hooverin ja Berkmanin mukaan kohteen koolla ja etäisyydellä käyttäjästä sekä ajalla, jonka käyttäjä käyttää sen valitsemiseen, on suora korrelaatio. (Hoover & Berkman 2012, 199.) Korkea informaatiotiheys voi tehdä kaavioiden, taulukoiden ja karttojen käyttämisestä vaikeaa pienen näytön omaavilla laitteilla (Hoover & Berkman 2012, 256).

3.2 Käyttöliittymän suunnitteleminen kosketusnäytölle

Lomakkeiden täyttäminen mobiililaitteilla on haastavaa. Tämän vuoksi Nielsen ehdottaa, että sovelluksen tulisi auttaa käyttäjää mahdollisimman paljon lomakkeen täyttämässä. Käyttäjän tulisi saada käyttää lyhenteitä, järkeviä oletusarvoja ja käyttöhistorian perusteella valittuja ehdotuksia. Syötteenä tulisi myös antaa käyttää laitteen kameraa, GPS-signaalia tai ääntä. Sovelluksen tulisi myös osata itse laskea tietoa, kuten postinumeron avulla paikkakunta tai luottokortin numeron avulla kortin tyyppi. On myös suositeltavaa, että lomake tukee kopioi ja liitä -toimintoa. Sovelluksen pitäisi täyttää valmiiksi jo tiedetyt tiedot, kuten sähköpostiosoite, jos käyttäjä on sisäänkirjautuneena. Kaikkien edellisten lisäksi lomakkeen tulisi olla mahdollisimman lyhyt ja sen tulisi pyytää vain tärkeitä tietoja.

Kirjoittamisen ohella myös kohteiden painaminen voi olla hankalaa, jos ne ovat pienikokoisia. Nielsenin mukaan paras koko kohteelle on 1 cm kertaa 1 cm. (Nielsen & Budiu 2013, 76–78.)

Hoover ja Berkman suosittelivat sallimaan vierityksen vain yhdellä akselilla aina sen ollessa mahdollista. Kun zoomataan sisältöä, joka alun

perin mahtui kokonaisuudessaan näytölle, voidaan tarjota toisen akselin suuntainen vierityspalkki. Harvoissa tapauksissa molemmat vieritysakselit voivat olla yhtä tärkeitä. Esimerkiksi kuvien zoomauksessa informaatio on, joka suunnassa yhtä tärkeää.

Pystysuora vieritys on käyttäjille helpoin käytettävä ja ymmärrettävä. Tämä johtuu siitä, että kirjoitetussa kielessä vaakasuoraa tilaa käytetään tehottomasti, jolloin lisätieto pitää esittää pystysuoralla akselilla. Tämän takia pystysuora vieritys on tällä hetkellä käyttäjille luontevaa.

Vaakasuora vierityspalkki voi toisinaan Hooberin ja Berkmanin mukaan olla hyödyllinen tietynlaiselle informaatiolle, mutta hyödyllisimmillään se on, kun pitää tarjota toissijainen vieritysalue. Pystysuoraan vieritettävällä sivulla on mahdollista tarjota alueita, joilla on sallittu vaakasuora vieritys. Tällä voidaan tarjota enemmän selkeyttä ja poistaa ristiriitoja, jotka liittyvät kohdistetusta vierityksestä.

Hooper ja Berkman myös ohjeistavat pitämään vierityspalkin aina näkyvässä sen ollessa mahdollista. Tilaongelmat tulisi heidän mukaansa ratkaista muilla tavoin kuin poistamalla vierityspalkit näkyvistä. Vieritys tulee sallia vain objektilta, joka on valittuna. Kosketuksella ja kynällä käytettävillä laitteilla ei tulisi mahdollistaa vierityspalkin suoraa käyttöä. Tämä vain lisäisi virheellisten syötteiden määrää mobiililaitteiden pienen koon vuoksi. (Hooper & Berkman 2012, 12–14.)

Käyttäjän tekstinsyöttöä tulisi helpottaa toteuttamalla tekstinsyöttöä avustava ominaisuus. Vaikka käyttöjärjestelmä tarjoaisi vastaavan ominaisuuden, voi silti olla kannattavaa rakentaa oman sovelluksen ehdot täyttävä automaattinen täydennys.

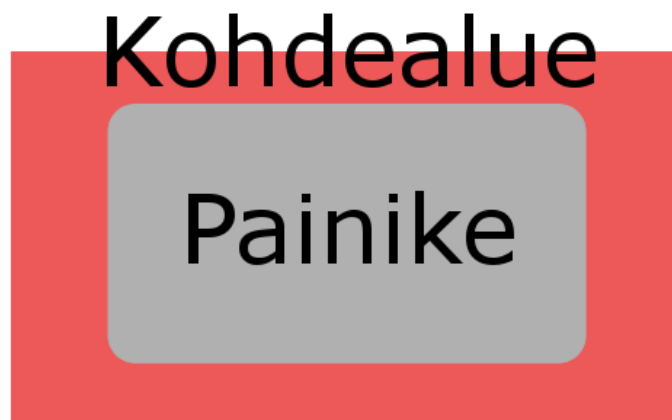
Automaattisesta täydennyksestä ja ennakoivasta tekstinsyötöstä on tullut vakiintuneita teknologioita jokapäiväisessä tietojenkäsittelyssä ja etenkin mobiilialustoilla, koska ne nopeuttavat alustalle ominaista hidasta tekstinsyöttöä. (Hooper & Berkman 2012, 308–309.) Käyttäjän tekstinsyöttöä voi helpottaa ja nopeuttaa ennakoivalla tekstinsyötöllä. Myös alasetoalikoita voi käyttää tekstikentän tilalla esimerkiksi maan valitsemisessa listasta. (Hooper & Berkman 2012, 368.)

Kokonäytön tekstikenttien käyttöä tulisi välttää. Monissa käyttöjärjestelmissä tekstikentän valitseminen avaa kokonäytön paneelin oletuksena. Kokonäytön tekstikentät peittävät muun tekstin ja vihjeet alleen ja samalla käyttäjä kadottaa asiayhteyden. Lomakkeiden kentät ja tekstikentät tuottavat käyttäjille eniten turhautumista ja virheitä vapaamuotoisen syötteen takia. (Hooper & Berkman 2012, 372.)

Kosketusnäytöllä navigoiminen on virhealtista. Tämän takia sovelluksessa tulisi olla kumo- ja takaisin-toiminnot. Jokaisen prosessin tulee suojella käyttäjän syötettä. Käyttäjän näkökulmasta on äärimmäisen turhauttavaa täyttää uudelleen samat tiedot virhetilanteen jälkeen. Käyttäjien syöttämä data tulisikin tallentaa mahdollisimman usein, jotta esimerkiksi käyttäjän vahingossa sulkema lomake voidaan uudelleen täyttää aiemmin syötetyllä

tiedolla. (McWherter 2009, 94.) Sovelluksen tulee myös muistaa tilansa keskeytysten varalta. Esimerkiksi saapuvan puhelun jälkeen käyttäjän tulee pystyä jatkamaan siitä mihin hän jäi ennen puhelua. (Lehtimäki 2012, 32.)

Kohteita suunniteltaessa on suositeltavaa pohtia havainnoidun kohdealueen ja varsinaisen kohdealueen suhdetta. Kun kohteen lähellä ei ole muita eleillä käytettäviä kohteita, on varsinaisesta kohdealueesta hyvä tehdä havainnoitua kohdealuetta suurempi. Jos esimerkiksi painikkeen läheisyydessä ei ole muita kohteita, voidaan varsinaisesta kohdealueesta tehdä isompi ja ottaa täten huomioon erikokoiset sormet ja vähentää virheitä. Kohteiden ympärille pitäisikin suunnitella tyhjää tilaa mahdollisimman paljon, jotta ei syntyisi tilanteita joissa käyttäjä vahingossa koskettaa läheistä kohdetta. (McWherter 2009, 93–94.) Kuvassa 6 on esimerkki painikkeen ja kohdealueen suhteesta.



Kuva 6. Painike ja kohdealue

4 OMA PROJEKTI

Projektin tarkoituksena oli toteuttaa yksinkertainen älypuhelinsovellus ja suunnitella sovelluksen käyttöliittymä. Projektin tärkein tavoite oli luoda edellä mainittujen periaatteiden avulla mahdollisimman helppokäyttöinen mobiilisovellus.

4.1 Mahdolliset ympäristöt

Android

Android-käyttöjärjestelmää kehittää Google. Suurin osa Androidista on ilmaista tai avointa lähdekoodia. Vuoden 2011 marraskuussa Android saavutti 52.5 prosentin osuuden älypuhelinmarkkinoista. (Wikipedia 2014.)

Android on suunniteltu ensisijaisesti kosketusnäyttöllisille mobiililaitteille, kuten älypuhelimille ja tableteille. Yksi Androidin heikkouksista kilpailijoihin nähden on Android-laitteiden laaja kirjo. Koska Android-laitteita on paljon eri laitteistokokoonpanoilla ja eri ohjelmistoversioilla on

erittäin vaikeaa kehittää sovellus, joka toimii jokaisessa Android-laitteessa. (Wikipedia 2014).

iOS

Applen kehittämä iOS-käyttöjärjestelmä on suljettua lähdekoodia. Applen laitteista iOS-käyttöjärjestelmää käyttävät iPhone, iPod Touch, iPad ja toisen sukupolven Apple TV. Tällä hetkellä kaikki iOS-laitteet ovat Applen kehittämiä ja Foxconin tai jonkun muun Applen yhteistyökumppanin valmistamia. (Wikipedia 2014.) Vuoden 2012 neljännellä neljänneksellä iOS-laitteilla oli 21 prosentin osuus älypuhelin markkinoista (Wikipedia 2014).

Windows Phone

Applen iOS-käyttöjärjestelmän tavoin myös Microsoftin Windows Phone-käyttöjärjestelmä on suljettua lähdekoodia. Tärkeimmät Windows Phone-valmistajat ovat Nokia, HTC, Samsung ja Huawei. (Wikipedia 2014.) Vuoden 2013 lopulla Windows Phone-käyttöjärjestelmän osuus älypuhelin markkinoista oli 3 prosenttia (Wikipedia 2014).

Päädyn toteuttamaan sovellukseni Android-alustalle. Android oli alustoista itselleni kaikista tutuin, koska olen käyttänyt Android-laitteita eniten edellä mainituista vaihtoehtoista. Tämä helpottaa käyttöliittymän suunnittelua, koska olen Android-laitteita käyttäessäni jo tottunut moniin Androidille ominaisiin käyttöliittymäratkaisuihin.

Toinen tärkeä syy valintaani oli ohjelmointikieli, jolla sovellus toteutetaan. Android-sovelluksia ohjelmoidaan Javalla (Wikipedian 2014). Java on ohjelmointikielenä itselleni kaikkein tutuin, mikä on sovelluksen kehittämisen näkökulmasta prosessia helpottava tekijä.

4.2 Android-sovellus

Halusin keskittyä opinnäytetyön käytännön osuudessa toimivan käyttöliittymän suunnitteluun ja toteutukseen. Liian monimutkainen sovellus olisi tuhannut aikaa opinnäytetyön kannalta epäolennaisiin asioihin, minkä takia halusin pitää sovelluksen mahdollisimman yksinkertaisena.

Edellä mainitun perusteella sovellukseksi valikoitui BMI-laskuri. Tarkoitukseni oli toteuttaa sovellus natiivisovelluksena. Pyrin toteuttamaan laskurin käyttöliittymän ja toiminnallisuuden tässä opinnäytetyössä eriteltyjen periaatteiden mukaisesti.

4.2.1 Kehitysympäristö

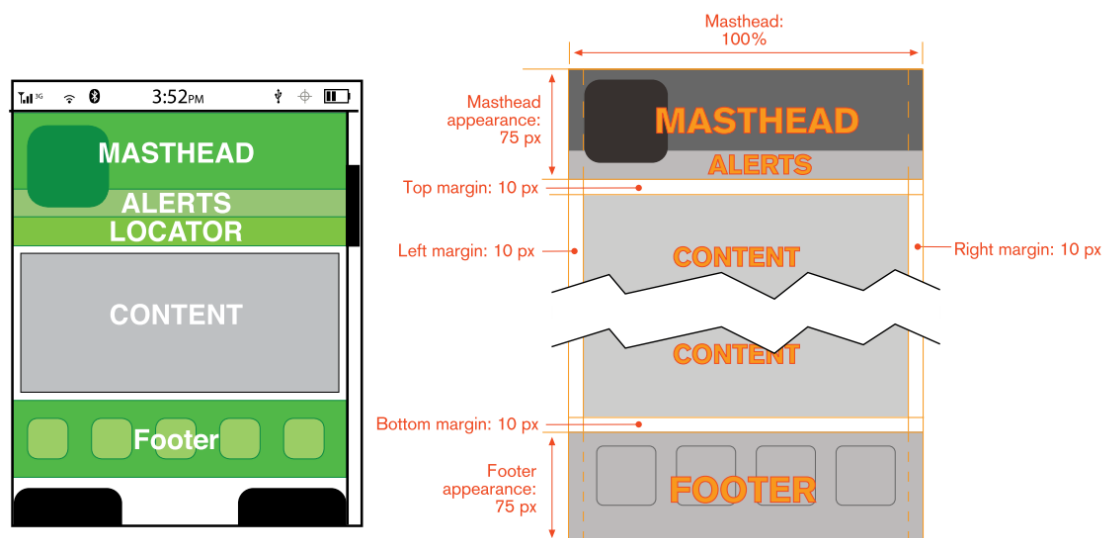
Kehitysympäristöä valitessani mielessäni oli kaksi vaihtoehtoa: NetBeans ja Eclipse. Edellä mainituista NetBeans oli itselleni entuudestaan tuttu,

koska olin käyttänyt sitä niin koulussa kuin työharjoittelussakin, kun taas Eclipse oli vain nimenä tuttu.

Kehitysympäristöksi valitsin Eclipsen, koska halusin laajentaa tietämystäni Javan eri kehitysympäristöistä. Android-sovellusten kehittäminen vaatii Java-kehitysympäristön lisäksi Android SDK:n (Android Developers 2014). Eclipsestä on tarjolla Android Developers -sivustolla Android-kehitystä aloittaville valmis paketti, joka sisältää Android SDK:n ja on asennuksen jälkeen valmis käytettäväksi. Koska tämä on ensimmäinen Android-projektini, tämä oli myös yksi syy valita Eclipse.

4.2.2 Suunnittelu

Hooperin ja Berkmanin ohjeiden mukaan päätin aloittaa käyttöliittymän suunnittelemisen luomalla sovelluksen näkymille yhteisen pohjan. Tämän pohjan päälle on helppo suunnitella sovelluksen eri näkymät ja saada aikaan yhtenäinen sovellus. Kuvassa 8 Hooperin ja Berkmanin esimerkki sovelluksen käyttöliittymäsuunnittelun pohjaksi. (Hooper & Berkman 2012, 2.)



Kuva 7. Esimerkki pohjasta käyttöliittymälle (Hooper & Berkman 2012, 2).

Hooperin ja Berkmanin esimerkin pohjalta loin omalle sovellukselleni käyttöliittymähierarkian. Tämän hierarkian pohjalta suunnittelin sovelluksen muut näkymät. Valmiit suunnitelmat testasin lähipiirilläni.

4.2.3 Käyttöliittymä

Sovellus jättää käyttöjärjestelmän tarjoaman ilmoituspalkin näkyviin, koska sen piilottaminen on suositeltavaa vain peleissä tai videota toistettaessa (Hooper & Berkman 2012, 18). Ilmoituspalkin alapuolella on tila otsikolle ja sovelluksen logolle.

Halusin sisällyttää sovellukseen myös useamman välilehden. Välilehtien navigointipalkin sijoitin otsikkorivin alapuolelle. Navigointipalkki on sijoitettu ruudun yläosaan, koska tämä on yleinen käytäntö Android-käyttöjärjestelmässä ja Android-sovelluksissa (McWherter 2009, 110). Navigointipalkin alapuolella on varsinainen sisältöalue, johon tulee näkymän sisältö. Sisällön määrästä riippuen oikealle puolelle näyttöä tulee näkyviin vierityspalkki, joka näyttää käyttäjälle tämän sijainnin sisältöalueella. Kuvassa 8 on valmis suunnitelma käyttöliittymän pohjaksi.



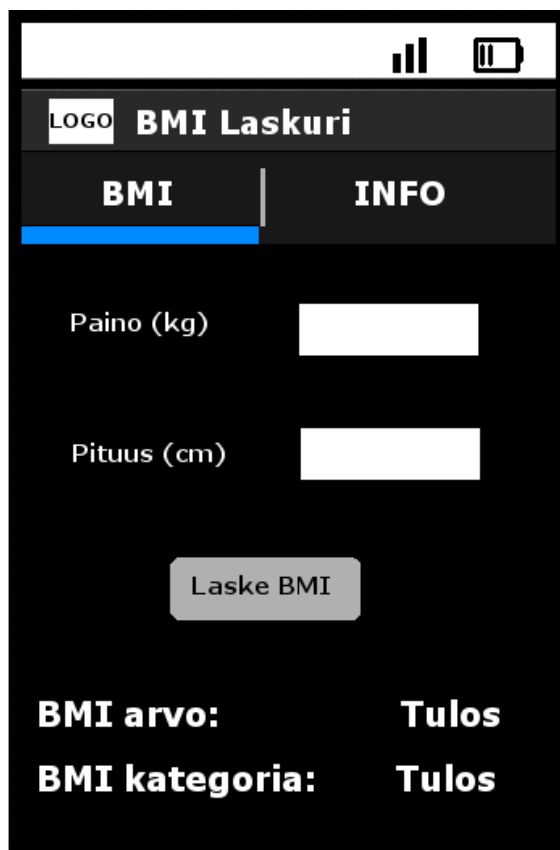
Kuva 8. Käyttöliittymän pohja

Saatuani käyttöliittymän pohjan valmiiksi suunnittelin näkymät. Sovelluksessa tulee olemaan kaikkiaan kaksi näkymää, jotka ovat kumpikin omilla välilehdillään. Toinen näkymä sisältää varsinaisen laskurin ja toinen on info-sivun.

Otsikko- ja välilehti-rivit ovat molemmilla näkymillä samat. Otsikkoriville tulee sovelluksen nimi ja logo. Välilehti-rivillä on kaksi painiketta, BMI ja info. Valittu näkymä on korostettu pienellä värillisellä alueella välilehti-painikkeen alaosassa.

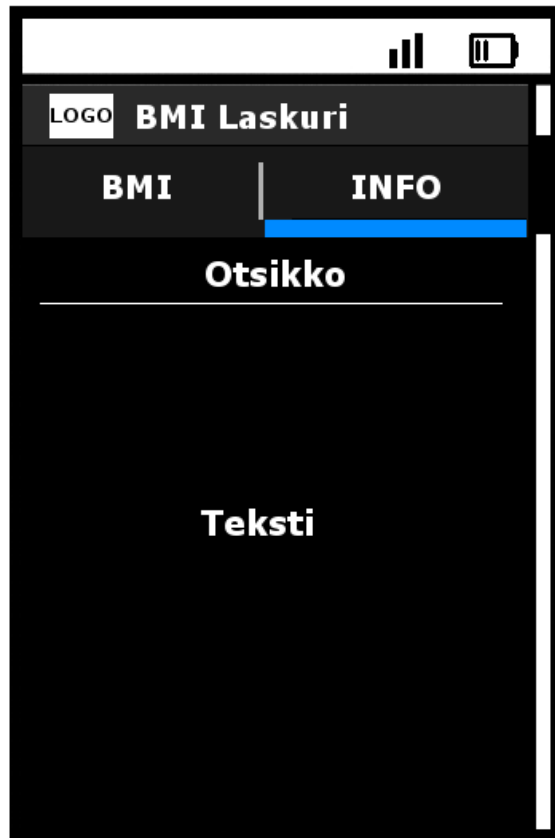
Laskurinäkymän sisältöalueella on tekstikentät, joihin syötetään paino ja pituus. Tekstikentät ovat päällekkäin ja molempien vasemmalla puolella on teksti, joka kertoo käyttäjälle mitä kyseiseen tekstikenttään syötetään. Näiden alapuolella, horisontaalisesti keskitettynä, on ”Laske BMI”-painike. Painiketta klikattaessa sovellus laskee BMI-arvon tekstikenttiin syötettyjen arvojen perusteella.

Lopuksi sisältöalueen alapuolella on kaksi tuloskenttää päällekkäin. Tuloskenttien vieressä vasemmalla puolella on tekstit, jotka kertovat käyttäjälle mitä tietoja kentät pitävät sisällään. Kuvassa 9 on valmis laskurinäkymä.



Kuva 9. BMI-laskurinäkymä

Info-välilehden sisältöalueella on otsikkorivin ja välilehtien lisäksi paikat otsikolle ja tekstille. Sisältöalueen tekstiin viittaava otsikko sijaitsee välilehtien alapuolella horisontaalisesti keskitettynä. Otsikon alapuolelle sijoitetaan viiva, joka luo selkeän erottelun otsikon ja tekstin välille. Viivan alapuolelle sijoitetaan teksti. Tekstin pituudesta riippuen näkymään tulee myös vierityspalkki. Kuvassa 10 on valmis infonäkymä.



Kuva 10. Infonäkymä

4.2.4 Toiminnallisuus

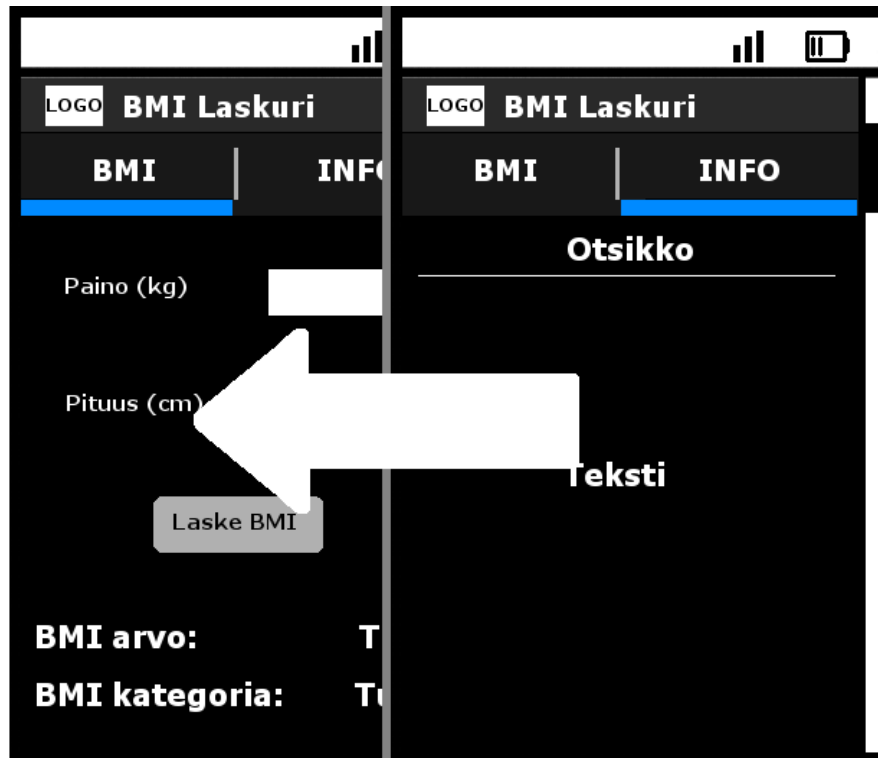
Sovellus avautuu laskurinäkymään. Laskurinäkymässä Paino- ja pituus-tekstikenttien painaminen tuo esiin virtuaalisen näppäimistön, jonka avulla kenttiin on mahdollista syöttää paino ja pituus. Kun paino ja pituus on syötetty, ”Laske BMI”-painiketta painamalla sovellus laskee BMI arvo-kenttään painoindeksiin ja BMI kategoriakenttään painoindeksiä vastaavaan kuvauksen.

Painokenttään voi syöttää maksimissaan viisi merkkiä ja painon pystyy syöttämään yhden desimaalin tarkkuudella. Painokentän suurin mahdollinen arvo on 999.9kg ja pienin 1kg. Pituuskenttään pystyy syöttämään vain kolme merkkiä, mikä mahdollistaa pituuden syöttämisen sentin tarkkuudella. Syötteen maksimipituuden rajoittaminen estää liian suurien arvojen syöttämisen kenttiin. Laskutoimitus suoritetaan, joko ”Laske BMI”-nappia painamalla tai virtuaalinäppäimistön enter-painiketta painamalla, kun paino ja pituus on syötetty. Virtuaalinäppäimistö sulkeutuu virtuaalinäppäimistön enter-painiketta tai ”Laske BMI”-painiketta painamalla.

Laskutoimitus toteutetaan vain, jos molemmat kentät on täytetty. Jos käyttäjä yrittää suorittaa laskutoimituksen toisen tai molempien kenttien ollessa tyhjänä, sovellus avaa varoitusikkunan. Varoitusikkuna pyytää käyttäjää syöttämään painon ja pituuden ennen laskutoimituksen suorittamista. Painokentän arvon ollessa suurempi kuin 999.9 tai pienempi

kuin yksi avautuu myös varoitusikkuna, joka pyytää käyttäjää syöttämään painon yhden desimaalin tarkkuudella 999.9 ja 1 kilon väliltä. Molemmista varoitus-ikkunoista poistutaan painamalla ikkunassa olevaa OK-painiketta.

Välilehdeltä toiselle siirtyminen tapahtuu navigointipalkissa olevien nappien painamisella. Toinen tapa vaihtaa välilehteä on liu'uttaa sormea vaakasuunnassa näytöllä. Logiikan kannalta välilehtien vaihtaminen liu'uttamalla toimii samalla tavalla kuin kirjan selaaminen. Kun sivua vedetään vasemmalle, päästään seuraavalle sivulle ja oikealle vedettäessä palataan edelliselle sivulle, kuten kuva 11 osoittaa.



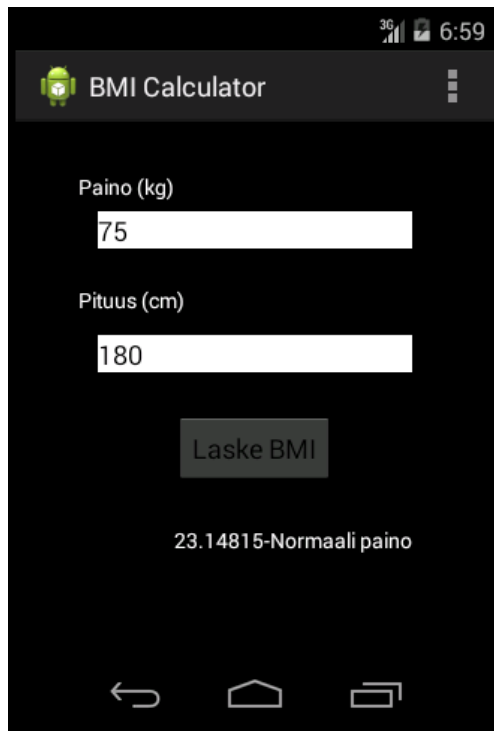
Kuva 11. Välilehden vaihtaminen kosketuksella

Info-välilehdellä olevaa tekstialuetta voi vierittää kosketuksella. Ruudun oikealla reunalla oleva vierityspalkki näyttää käyttäjän sijainnin sivulla.

Sovelluksesta poistutaan painamalla puhelimen takaisin-painiketta. Käyttäjän painaessa takaisin-painiketta sovellus avaa varoitusikkunan ja ilmoittaa käyttäjälle, että tämä on poistumassa sovelluksesta. Ikkuna kysyy käyttäjältä haluaako tämä varmasti poistua sovelluksesta. Käyttäjä vastaa kysymykseen painamalla ikkunan kyllä- tai ei-painiketta. Kyllä-painikkeen painaminen sulkee sovelluksen, kun taas ei-painikkeen painaminen sulkee varoitusikkunan. Sovelluksesta poistumisen varmentamisella pyrin estämään vahingossa tapahtuvat sovelluksen sulkemiset.

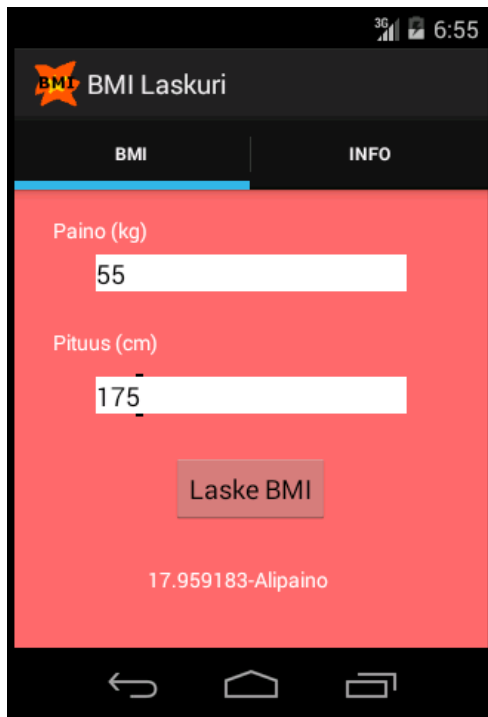
4.2.5 Toteutus

Aloitin sovelluksen toteuttamisen laskurinäkymästä. Ensimmäisenä asettelin tarpeelliset käyttöliittymäelementit paikoilleen käyttäen Eclipsen editorinäkymää. Seuraavaksi toteutin sovellukseen laskuri-toiminnallisuuden. Loin metodin jota kutsutaan ”Laske BMI”-painikkeen painamisen yhteydessä. Metodi hakee paino- ja pituus-kenttien sisällöt ja asettaa ne painoindeksin laskukaavaan. Lopuksi metodi palauttaa arvon joka tulostetaan tuloskenttään. Kuvassa 12 on sovellus ensimmäisessä muodossaan.



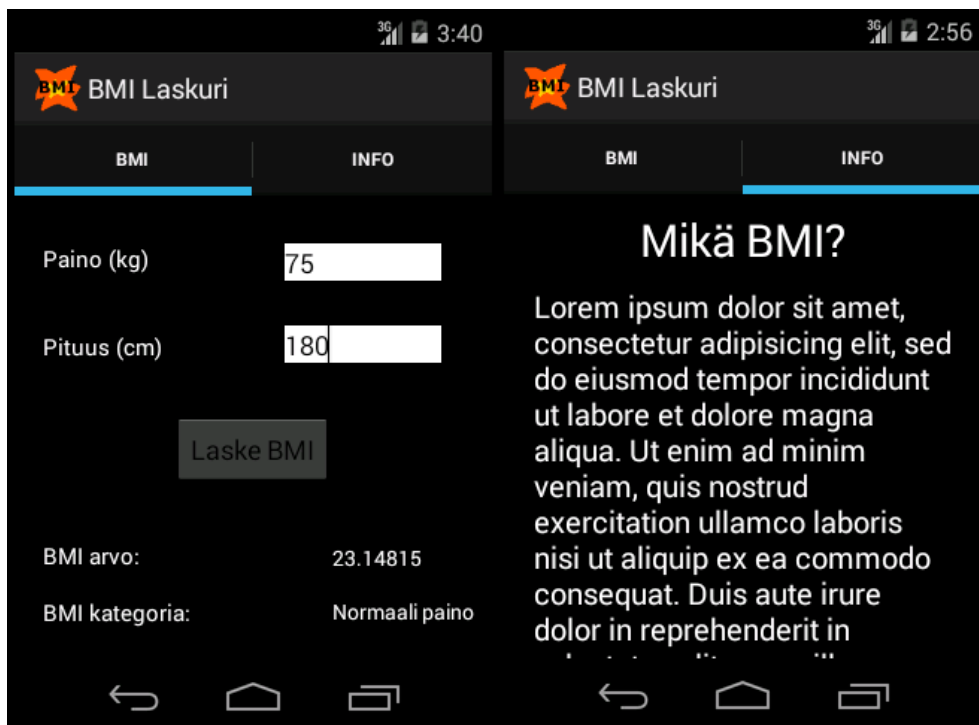
Kuva 12. Laskurinäkymän toteutus

Seuraavaksi lisäsin navigaatiopalkin välilehtien vaihtamista varten. Asetin välilehtien taustaväriin molemmille näkymille erivärisiksi, helpottamaan välilehtien toiminnallisuuden testauksessa. Vaihdoin myös sovelluksen logon kokeillakseni kuinka ominaisuus toimii. Kuvassa 13 on sovellus välilehtien lisäämisen jälkeen.



Kuva 13. Välilehtien lisääminen

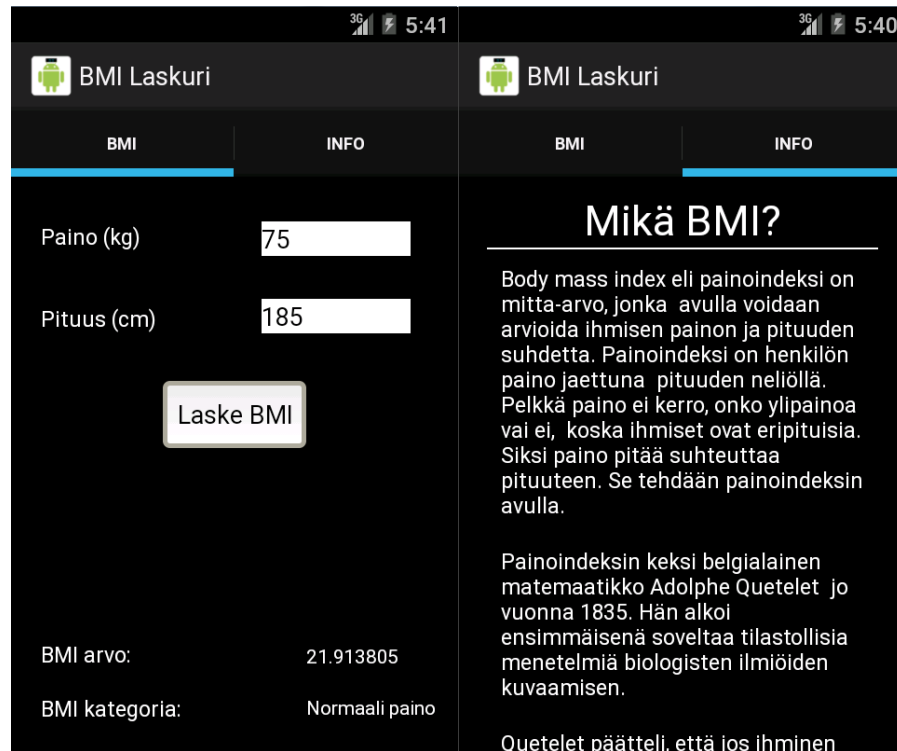
Seuraavana vaiheena asettelin laskurinäkymän elementit oikeille paikoilleen. Jaoin myös tuloksentän kahteen osaan käyttöliittymäsuunnitelmien mukaan. Lisäsin myös infonäkymään otsikon ja tekstialueen infotekstille. Tämän kaiken lisäksi tein tekstialueesta vieritettävän ja lisäsin tekstialueeseen pitkän pätkän lorem ipsum-tekstiä testatakseni vierityksen toiminnallisuutta. Kuvassa 14 ovat laskuri- ja infonäkymä käyttöliittymäelementtien asettelun jälkeen.



Kuva 14. Laskurinäkymän elementtien asettelu ja infonäkymän elementtien sijoittelu

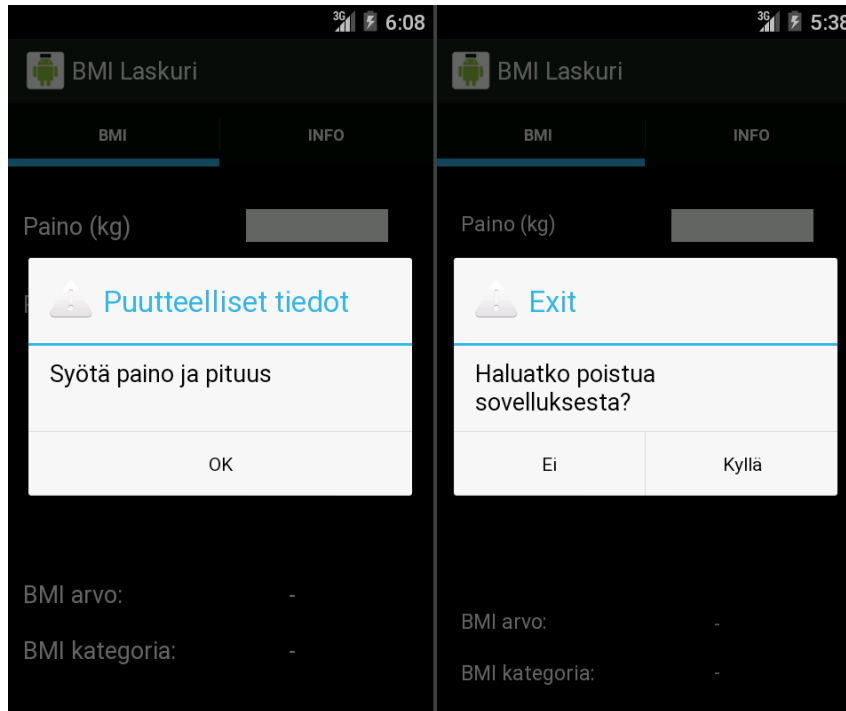
Kehitysympäristön tarjoama oletuspainike ei erottunut mielestäni tarpeeksi hyvin mustasta taustasta. Tämän takia loin oman painikkeen, joka paremmin mukaili alkuperäistä käyttöliittymäsuunnitelmaa. Vaihdoin myös molemmissa näkymissä näkyvän logon sovelluksen lopulliseen logoon. Ohjelmoin myös toiminnallisuuden, joka suorittaa painoindeksin laskutoimituksen, kun painetaan virtuaalinäppäimistön enter-painiketta.

Infonäkymään lisäsin otsikon alle viivan, joka erottaa otsikon tekstialueesta. Lisäsin myös tekstialueeseen infotekstin painoindeksiin liittyen. Kuvassa 15 ovat valmiit laskuri- ja infonäkymät.



Kuva 15. Valmiit laskuri- ja infonäkymät

Näkymien valmistuttua sovellukseen piti lisätä vielä käytettävyyden kannalta olennaisia toimintoja. Lisäsin laskurinäkymään ominaisuuden, joka pyytää käyttäjää syöttämään painon ja pituuden, jos toinen tai molemmat arvot puuttuvat. Lisäsin myös varoitusikkunan, joka ilmoittaa käyttäjälle sovelluksesta poistumisesta ja varmentaa haluaako käyttäjä varmasti poistua sovelluksesta. Kuvassa 16 ovat molemmat varoitusikkunat.



Kuva 16. Sovelluksesta poistumisen ja puutteellisen syötteen varoitusikkunat

Kun kokeilin lähes valmista sovellusta omalla Android-puhelimellani, huomasin sovelluksen avautuvan valkoiseen ikkunaan. Tämä valkoinen ikkuna oli näytöllä yhden tai kaksi sekuntia ennen kuin sovellus käynnistyi. Päätin varmuuden vuoksi lisätä sovellukseen vielä kevyen aloitusruudun, joka avautuu heti sovelluksen käynnistyessä ja on ruudulla sen aikaa, että itse sovellus on valmis. Kuvassa 17 on sovellukseen lisätty aloitusruutu.



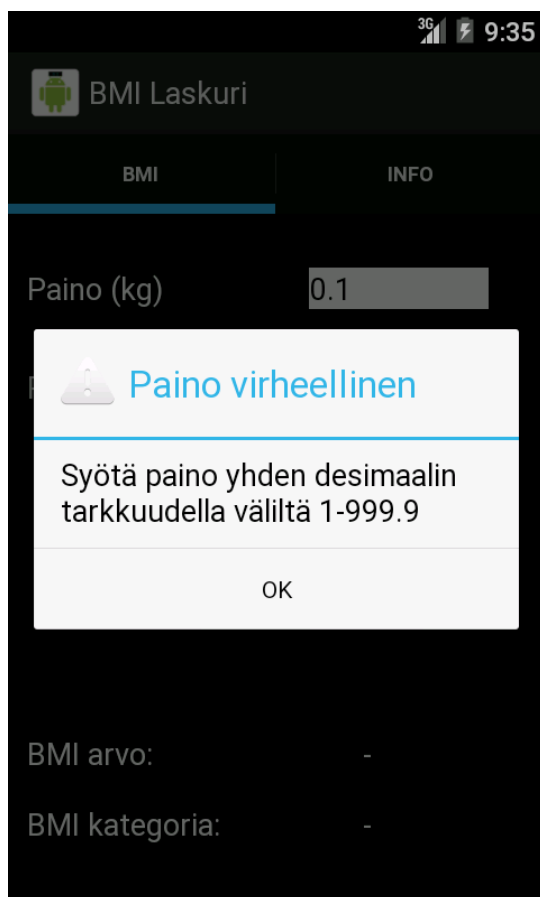
Kuva 17. Sovelluksen aloitusruutu

Tässä vaiheessa teetin lähes valmiilla sovelluksella käyttäjätestin lähipiirilläni. Testauksessa nousi esille liian pieni kirjasimen koko. Suurensin kirjasimen kokoa niin, että teksti on puhelimen näytöltä helppolukuista. Pyöristin myös BMI-arvon kahden desimaalin tarkkuudella. Kuvassa 18 on sovellus kirjasinkoon vaihdon jälkeen.



Kuva 18. Sovelluksen suuremmat fontit

Viimeisenä lisäsin vielä toiminnallisuuden, joka tarkistaa painokenttään syötetyn arvon. Jos arvo on suurempi kuin 999.9 kiloa tai pienempi kuin 1 kilo, sovellus avaa varoitusikkunan. Varoitusikkuna pyytää käyttäjää syöttämään painokenttään arvon, joka on sallittujen rajojen sisäpuolella ja sisältää ja on enintään yhden desimaalin tarkkuudella. Kuvassa 20 on varoitusikkuna, joka varoittaa käyttäjän syöttämän painon olevan sallittujen rajojen ulkopuolella.



Kuva 19. Varoitusikkuna, joka varoittaa painon olevan virheellinen

5 YHTEENVETO

5.1 Lopputulos

Mielestäni tässä opinnäytetyössä käydään kattavasti läpi käyttöliittymäsuunnittelun perusteet moderneille älypuhelimille. Tämän opinnäytetyön avulla lukijan tulisi saada hyvä käsitys miten käyttöliittymäsuunnittelu mobiilialustoille eroaa perinteisiin työpöytäympäristöihin nähden.

Projektin lopputuloksena syntyi Android-sovellus, joka toimii edellä mainitulla tavalla. Sovellus on pyritty toteuttamaan opinnäytetyön teoriaosuudessa mainittujen ohjeiden ja käytäntöjen mukaan. Päädyin toteuttamaan sovelluksen oman Android-puhelimeni resoluutioon, joka on 800x480. Sovellus on testattu toimivaksi Samsung Galaxy SII puhelimessa, jonka Android-käyttöjärjestelmä versio on 4.1.2.

5.2 Ongelmat

Aiheesta kertovien kirjojen löytäminen osoittautui hieman hankalaksi. Tämä johtuu hyvin todennäköisesti älypuhelimien ja mobiilisovelluksien lyhyestä historiasta.

Käyttöliittymän suunnittelussa suurimmat ongelmat syntyivät tilan rajoitetusta määrästä. Suunnitteluvaiheessa käyttöliittymää piti useampaan otteeseen muokata, koska siitä olisi tullut liian täyteen tupattu. Näkymistä nimenomaan laskurinäkymä tuotti enemmän ongelmia. Jouduin suunnittelemaan näkymän useampaan kertaan. Halusin laskurinäkymän olevan sen verran kompakti, että sitä ei tarvitse vierittää. Tämän takia jouduin suunnitteluvaiheessa hieman siirtelemään näkymän elementtejä ja poistamaan muutamia tarpeettomia elementtejä.

Kehitysvaiheessa ongelmia esiintyi sovelluksen toiminnallisuuden toteuttamisessa. Syynä tähän oli heikko Android-kehitysosaamiseni. Sain kuitenkin kaikki haluamani ominaisuudet toteutettua vaikeuksista huolimatta.

Vaikein yksittäinen ongelma kehityksessä syntyi kun päivitin Javan uuteen versioon. Tämän seurauksena Android SDK ei enää toiminut ollenkaan. Ongelman ratkaiseminen vaati Javan ja kehitystyökalujen uudelleen asentamisen. Myös Android-emulaattori piti määrittellä uudelleen ja tästä syystä kehityksen aikana ottamani ruutukaappaukset eroavat hieman toisistaan.

5.3 Havainnot

Päätin jo ennen opinnäytetyön käytännön osuuden aloittamista toteuttavani vain älypuhelinversion sovelluksesta. Halusin toteuttaa selkeästi rajatun sovelluksen ja tämän takia oli mielestäni loogista jättää tablettiversio toteuttamatta.

Havaitsin kuitenkin kehityksen aikana, että minun oli tehtävä vieläkin selkeämpi rajausta ja toteutettava sovellus vain yhdelle näyttöresoluutiolle ja Android-käyttöjärjestelmäversiolle. Android-laitteita on niin monia erilaisia, että sovelluksen optimointi eri resoluutioille ja käyttöjärjestelmäversioille olisi ollut liian suuri työ sisällytettäväksi opinnäytetyöhöni. Havaitsin myös, että olemassa ollut Java-osaamiseni loi hyvän pohjan Android-sovelluksen toteuttamiselle.

5.4 Pohdinta

Aloitin opinnäytetyön tekemisen tutkimalla aiheeseen liittyvää teoriaa. Tutustuin käyttöliittymäsuunnittelua mobiililaitteilla käsitteleviin kirjoihin ja valitsin mielestäni parhaat teokset opinnäytetyöni teoriaosuuden aineistoksi. Seuraavaksi kokosin kirjoista omasta mielestäni käyttöliittymäsuunnittelun kannalta tärkeimmän tiedon ja koostin siitä

opinnäytetyöni teoriaosuuden. Suurin osa opinnäytetyöhön käytetystä ajasta kului aineiston keräämiseen ja koostamiseen teoriaosuutta varten.

Kun teoriaosuus oli lähes valmis, siirryin työstämään käytännöosuutta ja siihen liittyvää älypuhelinsovellusta. Toteutin sovelluksen niiden käytäntöjen pohjalta, jotka olivat tulleet minulle tutuiksi teoriaosuuden kirjoittamisen aikana. Sovelluksen ja käytännöosuuden toteuttaminen tapahtui todella nopeasti.

Jos tekisin työn nyt uudelleen, haluaisin tehdä teoriaosuudesta laajemman. Lisäksi tutustuisin aiheeseen useamman lähteen avulla. Pyrkisin myös kirjoittamaan teoriaosuuden lyhyemmässä ajassa. Käytännöosuutta varten toteuttaisin monimutkaisemman sovelluksen, jonka suunnittelisin tarkemmin opinnäytetyön tekemisen aikana hankkimani tiedon avulla.

LÄHTEET

Hooper, S. & Berkman, E. 2012. Designing Mobile Interfaces. Sebastopol, CA, USA: O'Reilly Media, Inc.

Lehtimäki, J. 2012. Smashing Magazine Book Series: Smashing Android UI. Somerset, NJ, USA: John Wiley & Sons.

McWherter, J. 2009. Professional Mobile Application Development. Somerset, NJ, USA: John Wiley & Sons.

Nielsen, J. & Budiu, R. 2013. Mobile Usability. Berkeley, CA, USA: New Riders.

Wikipedian artikkeli Mobile operating system. 2014. Viitattu 25.8.2014. http://en.wikipedia.org/wiki/Mobile_operating_system

Wikipedian artikkeli Android (operating system). 2014. Viitattu 25.8.2014. [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

Wikipedian artikkeli iOS (Apple). 2014. Viitattu 25.8.2014. [http://en.wikipedia.org/wiki/iOS_\(Apple\)](http://en.wikipedia.org/wiki/iOS_(Apple))

Wikipedian artikkeli Windows Phone. 2014. Viitattu 25.8.2014. http://en.wikipedia.org/wiki/Windows_Phone

Wikipedian artikkeli Android software development. 2014. Viitattu 26.8.2014. http://en.wikipedia.org/wiki/Android_software_development

Android Developers www-sivut Get The Android SDK. 2014. Viitattu 27.8.2014. <http://developer.android.com/sdk/index.html>

Wikipedian artikkeli Progressive Disclosure. 2014. Viitattu 15.9.2014. http://en.wikipedia.org/wiki/Progressive_disclosure

Wikipedian artikkeli Haptic technology. 2014. Viitattu 15.9.2014. http://en.wikipedia.org/wiki/Haptic_technology