

Opinnäytetyö (AMK)
Tietotekniikan koulutusohjelma
Mediatekniikka
2015

Daniel Asteljoki

KÄYTETTÄVYYDEN ARVIOINTI SOVELLUSKEHITYKSESSÄ

– Terveysthallintasovellus



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Daniel Asteljoki

KÄYTETTÄVYYDEN ARVIOINTI SOVELLUSKEHITYKSESSÄ

Käytettävyyden arviointi sovelluskehityksessä on tärkeä osa kehitysprosessia, koska mobiililaitteet ja niissä käytettävät sovellukset ovat yleistyneet eksponentiaalisesti ja käyttäjiä on monenlaisia, noviiseista harjaantuneisiin. Tässä opinnäytetyössä käsiteltiin käytettävyyttä ja sen arviointia tietokone- ja mobiilisovellusten käyttöliittymissä.

Työssä käsiteltiin käytettävyyden historiaa tietotekniikassa ja perehdyttiin käyttäjäkeskeiseen suunnitteluun. Työssä arvioitiin yleisten mobiilialustojen ominaisuuksia ja niiden muotokieliä. Tarkasteluun sisällytettiin Windows-, iOS- ja Android-alustat. Arviointiin käytettiin Jakob Nielsenin luomaa heuristista listaa, joka on suosittu menetelmä käytettävyyden arvioinnissa.

Arviointia sovellettiin kehityksessä olevaan terveydenhallintasovellukseen, joka julkaistaan yleisimmille mobiilialustoille ja HTML-pohjaisena sovelluksena. Opinnäytetyön yhteydessä suunniteltiin sovellukselle käyttöliittymä Android- ja HTML-alustoille ja muokattiin Windows-alustalle aiemmin suunniteltua käyttöliittymää vastaamaan paremmin käytettävyyden periaatteita.

Terveydenhallintasovellus suunniteltiin yhdessä InnoHealth-projektin asiantuntijoiden ja terveydenhoito-opiskelijoiden kanssa. Sovelluksesta julkaistaan pilottiversio kotimaan markkinoille ja mahdollisesti myöhemmin maailmanlaajuisesti.

ASIASANAT:

Sovelluskehitys, käytettävyys, käyttäjäkeskeinen suunnittelu, mobiilialusta, HTML

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology | Digital Media

2015 | 29

Principal Lecturer Mika Luimula, Ph.D

Daniel Asteljoki

USABILITY EVALUATION IN SOFTWARE DEVELOPMENT

The purpose of this thesis was to study usability in software development. Usability is an important part of the development process as the popularity of mobile devices and apps has risen exponentially during the last decade and users range from novices to experts.

The theoretical section of this thesis examines the history of usability as it pertains to software development and user-centered design. The features and design languages of the most popular mobile platforms, Windows, Android and iOS were examined. Their usability was evaluated using Jakob Nielsen's 10 usability heuristics, a popular method of usability evaluation in information technology.

Usability is further applied in the empirical section to the development of a mobile health management app. The app will be released for Windows and Android and as an HTML application. A user interface was designed for each platform and evaluated using Nielsen's list. An earlier version of the interface for Windows Phone was also modified for better usability.

The health management app was a joint effort with the InnoHealth project. The design process included healthcare professionals and students. The app will be soft launched in Finland and later worldwide.

KEYWORDS:

Software development, usability, user-centered design, mobile platform, HTML

SISÄLTÖ

1 JOHDANTO	1
2 KÄYTETTÄVYYS	2
2.1 Käytettävyyden määritelmä	2
2.2 Affordanssit	2
2.3 Käytettävyys tietotekniikassa	4
2.4 Nielsenin säännöt	4
2.5 Käyttäjakeskeinen suunnittelu	5
2.5.1 Käytettävyystestaus	6
2.5.2 Haasteet ja mahdollisuudet	6
3 MOBIILIALUSTA	8
3.1 Mobiilialustan määritelmä	8
3.2 Muotokieli	8
3.3 Yleisiä mobiilialustoja ja niiden muotokieliä	9
3.3.1 Android ja Material design	10
3.3.2 iOS ja skeuomorfismi	11
3.3.3 Windows ja Metro	12
4 CASE: TERVEYDENHALLINTASOVELLUS	15
4.1 Suunnittelu	15
4.1.1 InnoHealth	Error! Bookmark not defined.
4.2 Kohdekäyttäjät	16
4.3 Sovelluksen rakenne	16
4.4 Käyttöliittymä	17
4.4.1 Windows Phone	18
4.4.2 Android	22
4.4.3 HTML-sovellus	24
5 YHTEENVETO	27
LÄHTEET	28

KUVAT

Kuva 1: Erilaisia affordansseja	3
Kuva 2: Esimerkki Material Designin ulkomuodosta.	10
Kuva 3: Esimerkki skeuomorfisesta käyttöliittymästä	12
Kuva 4: Esimerkki Metro-käyttöliittymästä	14
Kuva 5: Koutsi-sovelluksen valikkorakenne	17
Kuva 6: Windows Phone –käyttöliittymä, tietopankki	19
Kuva 7: Android-käyttöliittymä, kalorilaskuri	22
Kuva 8: HTML-käyttöliittymä, tietopankki	25

TAULUKOT

Taulukko 1: Mobiilikäyttöjärjestelmien markkinaosuudet vuoden 2014 lopussa. [7].....	9
Taulukko 2: Windows Phone –version käytettävyyden arviointi.....	21
Taulukko 3: Android–version käytettävyyden arviointi	23
Taulukko 4: HTML –version käytettävyyden arviointi	26

1 JOHDANTO

Mobiililaitteiden yleistyminen jokapäiväiseen käyttöön suuren yleisön keskuudessa on aiheuttanut muutoksia sovellusten suunnitteluun. Kasvanut käyttäjäkunta vaatii entistä tarkemman suhtautumisen käytettävyyteen ja käytettävyydestäkööseen. Tarjolla on myös erilaisia alustoja, joilla käyttäjä haluaa käyttää sovellusta. Mobiilialustat sisältävät paljon erilaisuuksia ulkonäkönsä, suorituskykynsä ja ominaisuuksiensa puolesta. Haastavaksi osoittautuu usein, miten sovellukselle saadaan samanlainen käyttökokemus erilaisten ja –kokoisten laitteiden välillä.

Tämä opinnäytetyö on jaettu kolmeen osaan. Ensin käydään lyhyesti läpi käytettävyyden historia tietotekniikassa ja perehdytään käyttäjäkeskeiseen suunnitteluun. Lähteenä on käytetty Donald A. Normanin [1] ajatuksia käyttäjäkeskeisestä käytettävyyssuunnittelusta ja Jakob Nielsenin sääntöjä hyvän käyttöliittymän suunnitteluun [2].

Toisessa osassa tarkastellaan yleisiä mobiilialustoja ja miten niiden suunnittelussa on sovellettu käytettävyyttä. Tarkasteluun on otettu Googlen Android-, Microsoftin Windows- ja Applen iOS-alustat ja näiden omat muotokielet. Pohditaan myös, miten alustojen erot vaikuttavat sovellusten kehitykseen.

Lopuksi suunnitellaan ylläolevien periaatteiden mukaisesti käyttöliittymä kehitteillä olevaan terveydenhallintasovellukseen ja arvioidaan, miten se vastaa nykyajan vaatimuksia käytettävyytensä osalta. Sovellus on tarkoitettu parantamaan käyttäjän liikunta- ja syömistottumuksia ja on suunniteltu yhteistyössä terveydenhuollon opiskelijoiden ja Dato Systems Ay -yrityksen kanssa osana InnoHealth-hanketta. Arvioinnissa käytetään avuksi Nielsenin heuristista listaa käytettävyyden arviointiin.

2 KÄYTETTÄVYYS

2.1 Käytettävyyden määritelmä

Käytettävyydestä on olemassa useita määritelmiä. Usein kuitenkin puhutaan järjestelmästä, joka pitää sisällään käyttäjän, käytettävän asian tai esineen ja tilanteen, jossa käyttö tapahtuu. Näiden perusteella käytettävyys voidaan jakaa käyttäjäkeskeisestä, tuotekeskeisestä tai tilannekeskeisestä suunnittelusta.

ISO 9241-11 –standardin [3] mukaan käytettävyys on ”Se vaikuttavuus, tehokkuus ja tyytyväisyys, jolla tietyt määritellyt käyttäjät saavuttavat määritellyt tavoitteet tietyssä ympäristössä”. Vaikuttavuus tarkoittaa sitä, kuinka täydellisesti käyttäjä on saavuttanut tavoitteensa, tehokkuus sitä, kuinka hyvin tavoitteet on saavutettu suhteutettuna käytettyihin resursseihin, tyytyväisyys sitä, kuinka tyytyväinen käyttäjä on vuorovaikutuksen sujuvuuteen. Määritelmä on laaja ja sitä voidaan soveltaa kaikkiin tilanteisiin, joissa käyttäjä käyttää jotain välinettä jonkin tarkoituksensa saavuttamiseen.

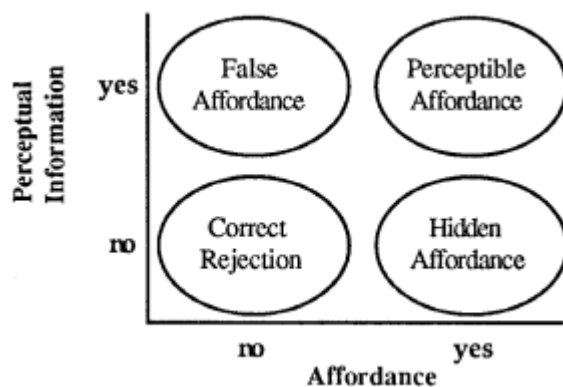
Jakob Nielsen [1] on lisännyt ISO-määritelmään opittavuus-, muistettavuus- ja virheiden vähyysskriteerit. Opittavuus tarkoittaa sitä, kuinka nopeasti käyttäjä oppii uuden toimintalogiikan, muistettavuus sitä, kuinka helppoa toimintalogiikka on palauttaa mieleen ja virheiden vähyydellä pyritään helpottamaan toimenpiteiden suorittamista.

2.2 Affordanssit

Affordanssit ovat käyttäjän havaittavissa olevia käyttömahdollisuuksia. Ne voivat olla joko suunniteltuja tai suunnittelemattomia tapoja käyttää järjestelmää ja ovat tärkeä osa käytettävyyden psykologiaa. Fyysisissä esineissä affordanssit voivat olla esimerkiksi ovenkahva, joka tarjoaa käyttäjälle affordanssin ”avaaminen”. Mainitussa tapauksessa affordanssi on näkyvässä ja ihmiselle havaittavissa.

On olemassa erilaisia affordansseja. Edellä oleva esimerkki ovenkahvasta on havaittava (perceptible) affordanssi. Se tarjoaa käyttäjälle tavan käyttää kohdetta ja johtaa haluttuun toimintaan. Havaittavan affordanssin lisäksi on olemassa piilotettuja (hidden) ja väärä (false) affordansseja (Kuva 1). [4]

Piilotettu affordanssi tarkoittaa sitä, että kohteelle on olemassa jokin käyttötapa, mutta se ei ole käyttäjälle ilmeinen. Jos ovesta ei ole kahvaa tai mitään osoitusta siitä, että sen voi avata, kyseessä on piilotettu affordanssi. Väärä affordanssi puolestaan on piilotetun affordanssin vastakohta. Väärä affordanssi on näkyvässä oleva käyttötapa, joka ei johda mihinkään oikeaan lopputulokseen. Esimerkki tästä voi olla painike, joka ei tee mitään. Piilotetut ja väärät affordanssit johtavat usein virhetilanteisiin.



Kuva 1. Erilaisia affordansseja [4]

Sovelluskehityksessä affordanssien pitää olla hyvin havaittavissa. Painikkeiden pitää olla sen näköisiä, että niitä painamalla tapahtuu jotain. Näkymästä pitää selvittää nopeasti, miten sitä voi katsella, onko kaikki tieto näkyvässä vai pitääkö käyttäjän vierittää sitä jne. Yleisimmät elementit virtuaalisissa käyttöliittymissä onkin suunniteltu siten, että ne muistuttavat vastaavia elementtejä oikeassa elämässä.

2.3 Käytettävyys tietotekniikassa

Käytettävyyden tutkiminen tietotekniikassa alkoi 80-luvulla, jolloin tietokoneiden hinnat laskivat sille tasolle, että niiden omistaminen yleistyi pelkästä ammattilaiskäytöstä myös kotikäyttöön. Näillä uusilla käyttäjillä oli joko hyvin vähän tai ei ollenkaan koulutusta tietokoneiden käytöstä. Ajan sovelluskehityskäytännöissä kuitenkin oletettiin, että asiakkailla ja käyttäjillä on jo tarvittavat taidot sovellusten käytössä. Tämä ristiriita herätti käyttäjien keskuudessa turhautumista ja tietokoneiden käyttö koettiin yleisesti vaikeakäyttöiseksi ja epäkäytännölliseksi. Tilanteen korjaamiseksi tuotteiden kehitysprosessissa nostettiin avainasemaan käytettävyyden tutkiminen ja sen parantaminen kaikissa prosesseissa, erityisesti jos käyttäjä ei ollut tekniikan ammattilainen. [5]

Kuten ISO-määritelmässäkin on määritelty, käytettävyydessä on usein tärkeää ottaa huomioon käyttäjän ja välineen lisäksi myös tilanne, jossa toiminta tapahtuu. Tietokoneohjelmissa tämä on kuitenkin harvinaista. Yleisempää on, että fyysinen ympäristö ei vaikuta ohjelman käyttöön. Tällöin sovellukset on hyvä suunnitella käyttäjäkeskeistä suunnittelufilosofiaa hyödyntäen. Mahdollisia poikkeuksia ovat sovellukset, joissa käytetään hyväksi ulkoisia antureita, kuten kiihtyvyyss- tai lämpötilamittaria.

2.4 Nielsenin säännöt

Jakob Nielsenin kehittämät säännöt hyvän käyttöliittymän kehittämiseen [2] on yleisesti käytetty periaate erityisesti ohjelmistokehityksessä. Seuraavat ovat Nielsenin 10 ohjetta:

1. *Järjestelmän tilan näkyvyys.* Järjestelmän tulee näyttää tietoa siitä, mitä sen sisällä tapahtuu, kohtuullisen ajan sisällä. Sovelluksissa tämä näkyy usein erilaisten ”Lataa”-viestien muodossa.
2. *Järjestelmän ja maailman tulee sopia yhteen toistensa kanssa.* Järjestelmän tulee puhua käyttäjän ymmärtämää kieltä, käyttää

samanlaisia käsitteitä ja yleisiä tapoja. Tämä näkyy esimerkiksi sovelluksen käyttämissä ikoneissa.

3. *Käyttäjän hallintakyky ja vapaus.* Käyttäjälle pitää antaa mahdollisuus suorittaa halutut komennot, kuten navigointi pois ei-halutusta tilasta.
4. *Yhdenmukaisuus ja yleiset tavat.* Järjestelmän tulee käyttää yhdenmukaista kieltä muiden samanlaisten järjestelmien kanssa. Samannimisten toimintojen täytyy toimia samalla tavalla.
5. *Virheiden estäminen.* Virheelliset tai virheelliset toiminnot pitää joko poistaa, korjata tai niistä pitää varoittaa käyttäjälle.
6. *Toimintojen näkyvyys.* Käyttäjän ei pitäisi joutua muistamaan tietoa saman dialogin sisällä, vaan kaiken mahdollisen tiedon tulee olla helposti näkyvillä.
7. *Joustavuus ja tehokkuus.* Järjestelmän käyttämisestä pitäisi olla mahdollista tulla nopeampaa sitä mukaa kuin käyttäjä kehittyy. Nämä nopeuttavat toiminnot ovat noviisikäyttäjälle tarpeettomia ja piilossa.
8. *Esteettinen ja minimalistinen suunnittelu.* Tarpeeton ja harvoin tarvittu tieto on hyvä piilottaa normaalissa näkymässä, koska se vähentää tärkeän tiedon näkyvyyttä.
9. *Virheilmoitusten selkeys.* Ilmoitusten tulee olla helppolukuista ja helposti ymmärrettävissä sekä ehdottaa ratkaisua ko. ongelmaan.
10. *Dokumentaatio.* Tarvittava avustus täytyy olla helposti saatavilla. [2]

2.5 Käyttäjäkeskeinen suunnittelu

Käyttäjäkeskeisellä suunnittelulla tarkoitetaan sitä, että käyttäjä on ratkaisevasti mukana suunnitteluprosessissa. Yleisesti käyttäjät ovat mukana vaatimusmäärittelyssä ja palautteen keräämisessä, mutta projektista riippuen voivat olla myös hyvin keskeisessä asemassa. Käyttäjäkeskeisen suunnittelun keskustassa on ajatus, että järjestelmän vaikeakäyttöisyys ei johdu käyttäjästä, vaan siitä että se on huonosti suunniteltu. Suunnittelun lähtökohtana ovat käyttäjän tarpeet.

Käyttäjäkeskeisestä suunnittelusta alettiin puhua 1980-luvulla, jolloin Donald Norman ja Stephen Draper julkaisivat aiheesta kirjan *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Norman myöhemmin laajensi teoriaa teoksessaan *The Psychology of Everyday Things*. Tätä pohjaa käyttäen mm. Ben Shneiderman ja Jakob Nielsen kehittivät omat sääntönsä käytettävyydestä. [6]

2.5.1 Käytettävyydestä

Järjestelmän testaus käyttäjillä on helppo ja kattava tapa kerätä palautetta projektin eri vaiheissa. Käyttäjättestaus on tärkeä osa käyttäjäkeskeistä suunnittelua, ja palautetta pyritäänkin keräämään niin usein kuin mahdollista. Tuotteen valmistuttua palautetta voidaan kerätä vaikka haastattelujen tai kohderyhmätutkimusten kautta.

Tuotekehityksen aikana tapahtuvalla testauksella on rajoituksia. Testaus tapahtuu yleensä hallitussa tilassa, kuten laboratoriossa, lyhyellä aikavälillä, jolloin on vaikeaa päätellä, miten tuote pärjää oikeissa tilanteissa tai päivittäisessä käytössä.

2.5.2 Haasteet ja mahdollisuudet

Jos käyttäjät ovat mukana tuotteen kehityksessä, heillä on parempi käsitys siitä, millainen lopputulos on, mikä johtaa tyytyväisempään käyttäjäkuntaan. Tämä myös vähentää suunnittelijan vastuuta lopputuloksesta ja lisää varmuutta siitä, että kehitys on menossa oikeaan suuntaan.

Käyttäjäkeskeisessä suunnittelussa vaatimusmäärittely on tärkeä osa prosessia. Asiakas tarjoaa yleensä ongelman, jonka tämä olettaa suunnittelijan ratkaisevan. Usein kyseessä ei kuitenkaan ole oikea ongelma, vaan oire jostain muusta ongelmasta. Tässä tapauksessa on hyvä työskennellä asiakkaan kanssa ja selvittää, mikä on oikea tapa lähestyä pulmaa.

Käyttäjien sisällyttäminen suunnitteluprosessiin voi aiheutua kalliiksi. Tiedon kerääminen käyttäjiltä, erityisesti oikeassa ympäristössä, on aikaavievää. Joskus myös tarvitaan ylimääräisiä henkilöitä, kuten psykologeja tai sosiologeja analysoimaan kerättyä tietoa ja kommunikoimaan asiakkaan ja kehittäjän välillä.

[6]

3 MOBIILIALUSTA

3.1 Mobiilialustan määritelmä

Mobiilialustalla tarkoitetaan käyttöjärjestelmää, joka toimii kannettavassa laitteessa, kuten älypuhelin tai taulutietokone. Vaikka kannettavat tietokoneet ovat määritelmältään mobiililaitteita, niissä käytetään usein samoja käyttöliittymiä kuin pöytätietokoneissa, jolloin niitä ei lasketa puhtaiksi mobiilialustoiksi. Poikkeuksena tähän sääntöön ovat viime vuosina kehitetyt käyttöjärjestelmät, jotka on suunniteltu työpöytä- sekä mobiilikäyttöön, kuten Microsoft Windows 8.

Nykyään valtaosa mobiilialustoista toimii sillä periaatteella, että ainoa syöttölaite on laitteen kosketusnäyttö. Vaikka on saatavilla muita syöttölaitteita, kuten ulkoisia näppäimistöjä tai ohjaimia, oletusarvona on, että kaikki toiminnot voidaan suorittaa kosketusohjauksella, mikä aiheuttaa haasteita käyttöliittymän ja sovellusten suunnittelussa.

Mobiililaitteet sisältävät myös usein toimintoja, joita ei työpöytälaitteissa tarvita, kuten puheluiden soittaminen tai kamera, jotka täytyy ottaa huomioon käyttöjärjestelmää suunnitellessa. Yleistä on myös antureiden, kuten kiihtyvyyss- tai GPS-antureiden käyttö.

3.2 Muotokieli

Yksi tärkeistä ominaisuuksista Nielsenin säännöissä on yleisten käytäntöjen seuraaminen. Yleiset käytännöt näkyvät mobiilialustoissa niiden muotokielessä. Muotokieli tarkoittaa tuotteen visuaalista ulkomuotoa. Useimmat alustat sisältävät hyvinkin tarkkoja määräyksiä ja ohjeita sovellusten välisen yhdenmukaisuuden saavuttamiseksi. Tämä näkyy väreissä, fonteissa, näyttöelementtien sijoittelussa, liikkeessä ja äänissä.

Sovelluskehityksessä muotokielellä on erityinen vaikutus tuotteen ulkonäön lisäksi sen käytettävyyteen. Sovellusta, joka noudattaa saman muotokielen

ohjeita kuin muut alustan sovellukset, on helpompaa ja nopeampaa käyttää kuin niitä, joissa poiketaan yleisistä sopimuksista. Tämä ei kuitenkaan tarkoita sitä, että ohjeita pitää noudattaa orjallisesti, vaan innovointi on edelleen tärkeä osa käyttöliittymäsuunnittelua. Muotokielen kehitys tulee pitää asteittaisena, eikä liian äkkipikaisena.

Tärkeitä muotokieliä mobiilisovelluskehityksessä ovat Material Design, skeuomorfismi ja Metro. Näitä käytetään Androidissa, iOS:ssä ja Windowsin mobiiliversioissa. Kielet eroavat huomattavasti toisistaan, mutta niiden toiminta-ajatukset ovat samankaltaisia.

3.3 Yleisiä mobiilialustoja ja niiden muotokieliä

Tämän hetken suosituimpia mobiilikäyttöjärjestelmien valmistajia ja kehittäjiä ovat Googlen Android, Applen iOS ja Microsoftin Windows. Muut alustat, kuten BlackBerry, Firefox OS ja Sailfish OS ovat osuudeltaan pieniä verrattuna yllämainittuihin ja niille suunnattu sovelluskehitys on vähemmän kannattavaa kuin suosituimmille järjestelmille. Tässä työssä käsitellään vain kolmea isointa alustaa, Androidia, iOS:ää ja Windowsia ja niiden omia muotokieliä.

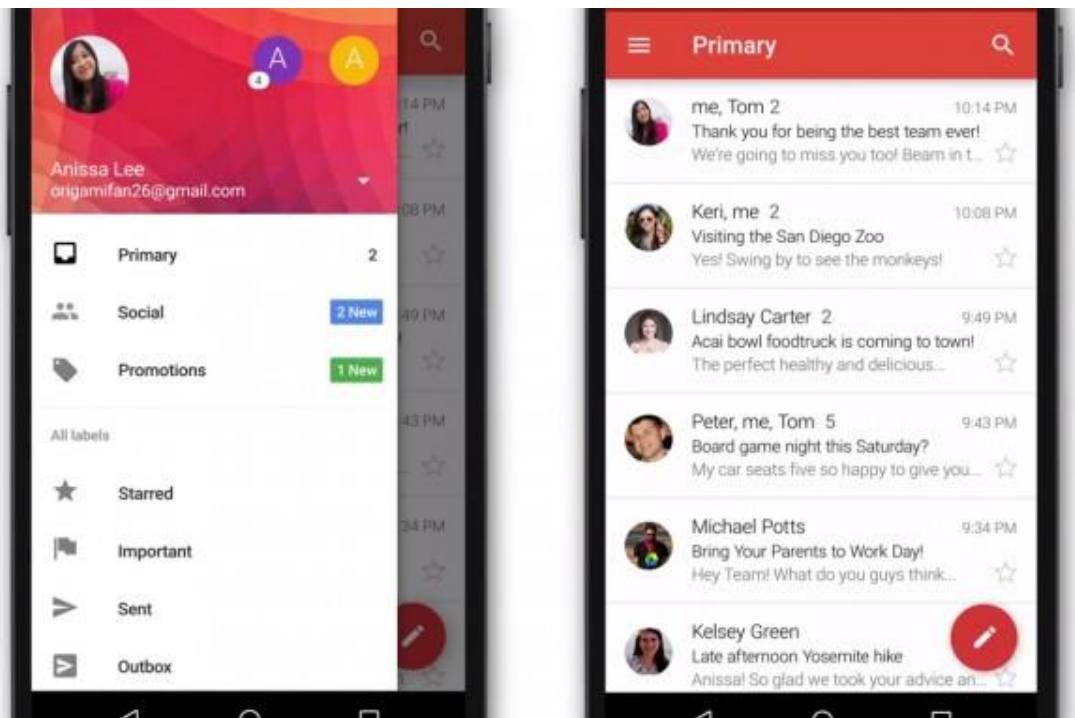
Taulukko 1: Mobiilikäyttöjärjestelmien markkinaosuudet vuoden 2014 lopussa. [7]

Käyttöjärjestelmä	Osuus markkinoista vuoden 2014 lopussa (%)
Android	81,5
iOS	14,8
Windows Phone	2,7
BlackBerry	0,4
Muut	0,6

3.3.1 Android ja Material design

Android johtaa markkinoita monipuoleisen laitevalikoimansa ja alustan avoimuuden avulla. Alusta on kehittynyt vuonna 2015 versioon 5.1, vaikka suurin osa laitteista käyttääkin vielä vanhempia 4.1- ja 4.4-versioita. [8] Androidin vahvuutena on alustan avoimuus ja edullisten laitteiden saatavuus teollistuneissa kehitysmaissa. Monet laitevalmistajat muokkaavat Androidista oman versionsa sopimaan omille laitteilleen. Androidia on kritisoitu alustan liiasta pirstaloitumisesta, mikä vaikeuttaa sovellusten kehittämistä alustalle. [9]

Material designin ensimmäisiä sovellutuksia oli Googlen Gmail-näkymän uudelleensuunnittelu vuonna 2011. Napit muuttuivat tasaisemmiksi, marginaalit kasvoivat ja näkymä muuttui kolmiulotteisemmaksi. Elementit eroteltiin toisistaan korkeuseroilla ja ne heittivät varjot alempien tasojen päälle. Värit muuttuivat kirkkaammiksi, selkeämmiksi ja yksinkertaisemmiksi (Kuva 2).



Kuva 2. Esimerkki Material Designin ulkomuodosta. [15]

Material Designin periaate on nykyään luonnollisen tuntuinen käyttöliittymä. Animaatioiden liike on johdonmukaista ja muistuttaa luonnossa tapahtuvaa

liikettä. Elementit on jaettu korteiksi, jotka liikkuvat toistensa päällä. Eri toiminnot muistuttavat itsestään hienovaraisilla animaatioilla oikeassa kontekstissa, kuten käyttäjän selattua sivu loppuun. Valo ja varjot käyttäytyvät kuin oikeassa tilassa. [10]

Material Designin animaatiot on suunniteltu siten, että ne kiinnittävät käyttäjän huomion sulavasti ja ne noudattavat luonnon lakeja. Liikkeellä on massaa ja kiihtyvyyttä ja se jatkuu samalla tavalla myös näköalueen ulkopuolelle, toisin kuin monessa muussa kielessä, jossa liike on mekaanista ja liike hidastuu epäluonnollisesti sen poistuessa alueelta.

3.3.2 iOS ja skeuomorfismi

Apple kehittää iOS:ää vain omille mobiililaitteilleen, kuten iPhoneille ja iPadille. iOS:ää ei ole saatavilla muiden valmistajien laitteille, mikä vähentää sen suosiota kasvavilla markkinoilla laitteiden korkean hinnan vuoksi. Alustan sulkeutuneisuus kuitenkin helpottaa sovellusten kehittämistä, koska kehityksessä ei tarvitse huomioida laitteiden eritasoisia suorituskykyjä tai ominaisuuksia. Eroavaisuuksia on olemassa mutta ei niin paljon kuin esim. Androidilla.

Skeuomorfismin ajatus on, että virtuaalisen käyttöliittymän elementit muistuttavat oikean elämän asioita, muotoja ja materiaaleja, mikä helpottaa niiden hahmottamista (Kuva 3). Tarkoitus on, että noviisikäyttäjänkin osaisi käyttää laitteita vaistonvaraisesti. Tämä näkyy esimerkiksi siinä, että käyttöjärjestelmissä käytetään virtuaalisia työpöytiä, roskakoreja ja dokumentteja. Skeuomorfismi helpottaa affordanssien havaitsemista. Skeuomorfisten käyttöliittymien sisältämät elementit vastaavat yleensä paremmin oikeaa elämää kuin muut käyttöliittymät, mikä nopeuttaa niiden havaitsemista.



Kuva 3. Esimerkki skeuomorfisesta käyttöliittymästä [16]

Viime aikoina Apple on vähentänyt skeuomorfisesta suunnittelua käyttöliittymissään. Valtaosan kilpailusta siirtyessä tasaiseen, minimalistiseen tyyliin skeuomorfismi visuaalisena tyylinä alkaa näyttää vanhanaikaiselta. Käyttäjien tottuessa laitteiden käyttöliittymiin skeuomorfinen tyyli menettää sen, mikä teki siitä helppokäyttöisen noviiseille ja tekee siitä sekavan kehittyneille käyttäjille. Mobiililaitteiden yleistymisen kautta käyttäjäkunta koostuu yhä enemmän kehittyneistä käyttäjistä, jotka hyötyvät enemmän suunnittelusta, joka keskittyy enemmän järjestelmän toimivuuteen ja jossa käyttöliittymäelementit on pidetty mahdollisimman yksinkertaisina.

3.3.3 Windows ja Metro

Microsoftin mobiilialustan juuret ovat vanhemmassa Windows Mobile – käyttöjärjestelmässä. Vuonna 2008 Microsoft uudisti alustan uuden nimen Windows Phone 7 alle. Windows Phone 8 julkaistiin vuonna 2012. Alusta sisältää allaan useita laitteita monilta valmistajilta, mutta se ei salli valmistajille samanlaista vapautta kuin Android. Microsoft on yhdistämässä kaikki käyttöjärjestelmänsä yhden brändin alle, jolloin se ei enää valmista erilaisia

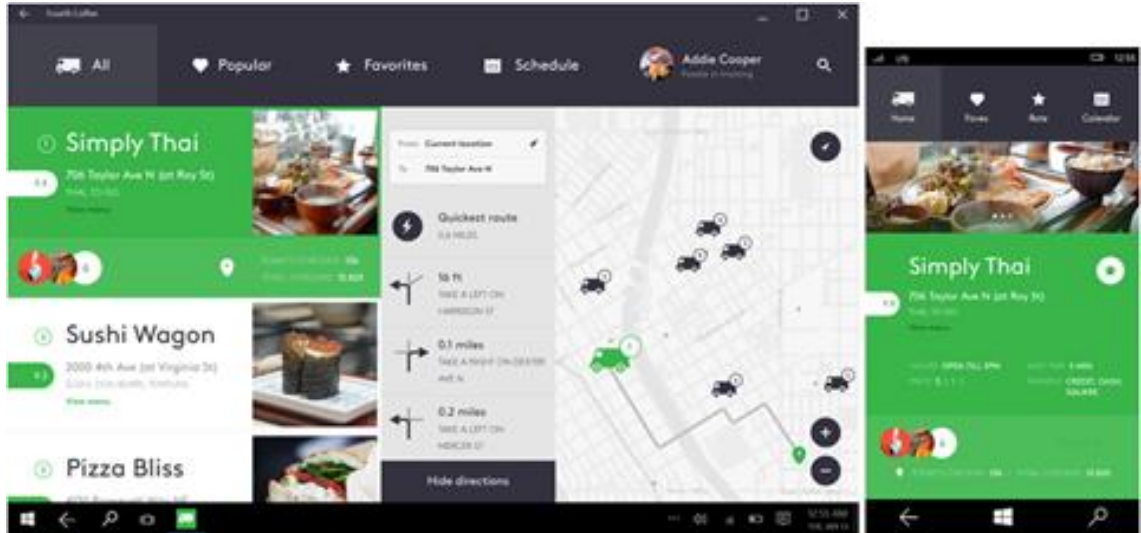
järjestelmiä eri laitteille, vaan kaikki käyttävät eri versioita samasta käyttöjärjestelmästä.

Microsoft julkaisi uuden muotokielensä Metron, Windows Phone 7:n julkistuksen yhteydessä. Metro on tasainen muotokieli, joka keskittyy vahvasti typografiaan suunnitteluelementtinä. Sitä käytetään Windows Phonen lisäksi Windowsin työpöytäversioissa versiosta 8 eteenpäin.

Metro-käyttöliittymät koostuvat usein värikkäistä elementeistä mustalla tai muuten hillityillä taustoilla. Teksti on tärkeässä osassa esimerkiksi otsikkona ja navigaatioon liittyen. Kuvien ja tekstuuriin käyttö on vähäistä, paitsi silloin kun sovellus sitä vaatii. (Kuva 4)

Metron kuvakkeet ja näyttöelementit ovat teräväkulmaisia, toisin kuin Googlen tai Applen kuvakkeet, jotka ovat usein pyöreitä tai pyöristettyjä. Terävät reunat ja kulmat yhdistettynä korkeaan kontrastiin värien välillä johtaa selkeään näkymään, josta on helppo hahmottaa tärkeät ohjauselementit.

Toisin kuin Googlen Material Design, joka tuo käyttöliittymiinsä eri tasoja ja varjoja, Metro toimii pääosin yhdellä tasolla. Metron käyttöliittymät ovat staattisia ja ne eivät sisällä Materialin suosimia hienovaraisia liikkeitä, jotka kiinnittävät huomiota aktiivisiin elementteihin. Animaatioita käytetään esimerkiksi silloin, kun käynnistetään sovellus tai vaihdetaan näkymää.



Kuva 4. Esimerkki Metro-käyttöliittymästä [17]

Koska Metro on suunniteltu mobiililaitteita varten, se ei käänny tehokkaasti suurille näytöille. Suuret ikonit ja tekstit vievät paljon tilaa näytöillä ja ennen Windows 10 -versiota Store -sovellukset eivät tukeneet useamman ikkunan näyttämistä yhtä aikaa. Windows 10:ssä on mahdollista käyttää sovelluksia ikkunoissa perinteisen Windowsin lailla.

4.2 Kohdekäyttäjät

Sovelluksen kohderyhmä on 18–50 vuotiaat miehet ja naiset, joilla on älypuhelin ja jotka ovat valmiita maksamaan terveyttä edistävästä sovelluksesta. Käyttäjällä ei tarvitse olla kokemusta muista sovelluksista eikä oleteta muuta teknistä osaamista kuin älypuhelimien käyttö. Sovellus toteutettiin aluksi vain suomen kielellä ja Windows Phone –alustalle.

Kohdekäyttäjät toivoivat yksilöllisiä ohjeita ja räätälöityjä ravitsemissuosituksia sekä muuta muokattavuutta. Heidän mielestään sovelluksen olisi hyvä sisältää vuorovaikutteisuutta muutenkin kuin vain kaloreiden ja urheilusuoritusten kirjaamisen lisäksi. Kohderyhmän haastattelujen aikana huomattiin, että motivaation pitäminen korkealla on tärkeää ja itseään vastaan kilpaileminen koettiin hyväksi kannustimeksi. Muita vastaan kilpaileminen esimerkiksi sosiaalisen median kautta jakoi mielipiteitä. [13]

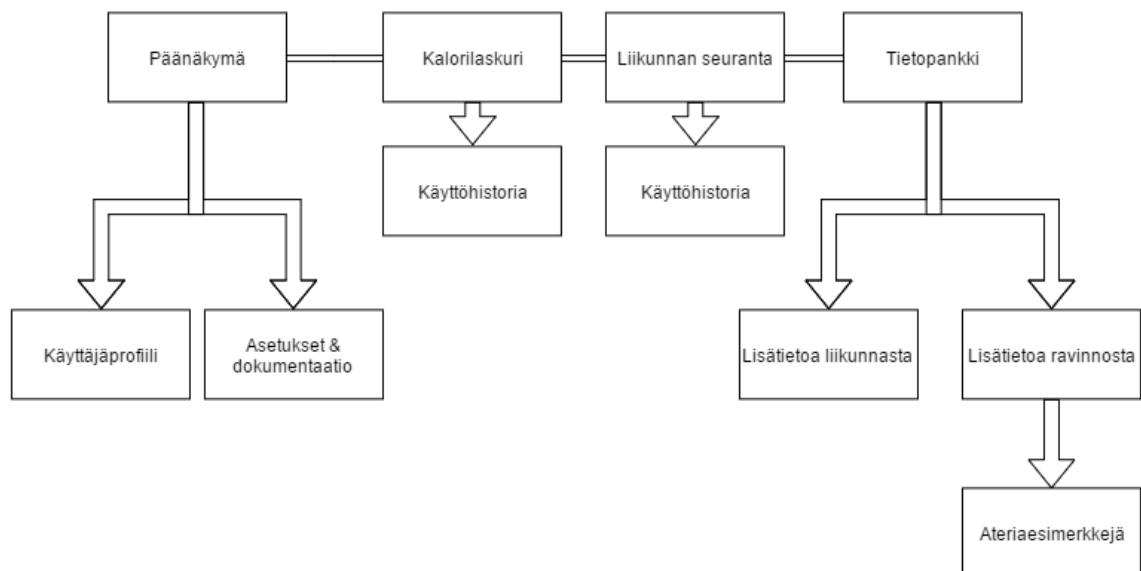
4.3 Sovelluksen rakenne

Sovellus sisältää kolme osiota: ravinnon seuraamiseen tarkoitettu kalorien laskenta, liikunnan seuranta, joka laskee kulutettuja kaloreita, ja tietopankki, joka sisältää tietoa terveellisemmän ravinnon ja elämän saavuttamiseksi, sekä esimerkkiaterioita helpottamaan paremman ruokavalion saavuttamista. Osiot on suunniteltu modulaarisiksi, mikä tässä tapauksessa tarkoittaa sitä, että yhtä niistä voi käyttää riippumatta muista. Tämä helpottaa käyttäjää, jos tämä ei haluakaan käyttää esimerkiksi kalorilaskuria tai haluaa käyttää siihen vaihtoehtoista sovellusta. Ohjelma muistuttaa käyttäjää halutuin väliajoin liikunnan puutteesta tai kalorien kerääntymisestä ja antaa ehdotuksia tilanteen parantamiseksi.

Sovelluksen betaversio kehitettiin Windows Phone –alustalle. Betaversio sisälsi kalorilaskurin ja osan tietopankista. Sovellus käännetään testausvaiheen päätyttyä myös Androidille ja HTML-pohjaiseksi sovellukseksi. Android- ja HTML-sovellukset käännetään myös englanniksi. Tämän työn aikana suunniteltiin käyttöliittymä Androidille ja HTML-sovellukselle.

4.4 Käyttöliittymä

Koutsi-sovelluksen valikkorakenne (Kuva 5) on jaettu samantasoisiin moduuleihin, jotta käyttäjän ei tarvitse navigoida monen näkymän kautta päästäkseen haluttuun päämäärään. Kaikki sovelluksen sisältämät alueet ovat näkyvissä jo päänäkymässä. Mobiilikäyttöliittymissä ensisijaisena näkymänä toimii aloitusruutu, jossa näkyy ajankohtaisia viestejä ja muut osiot saa näkyville liu'uttamalla sormea oikealle tai vasemmalle.



Kuva 5. Koutsi-sovelluksen valikkorakenne

Päänäkymässä näkyy ajankohtaisia viestejä käyttäjälle liikuntasuorituksista ja kalorien saannista. Viestit on räätälöity käyttäjälle tämän lähistorian sekä tavoitteiden mukaan, ja ne näkyvät mobiililaitteilla myös ilmoituksena sovelluksen ulkopuolella, jos käyttäjä niin haluaa. Näkymästä pääsee muokkaamaan käyttäjän profiilia ja asetuksia.

Kalorilaskuri on tarkoitettu käyttäjän avuksi tarkkailemaan kalorien saantia näyttämällä, mitä tämä on syönyt päivän aikana ja kuinka lähellä ollaan

määritettyä ylärajaa. Sovellus tarjoaa käyttäjän iän, painon ja sukupuolen perusteella ylärajaehdotuksen, jota voi vapaasti muokata. Sovellus sisältää tietokannan useimmista ruoka-aineista, ja sitä voi käyttää apuna tietoa syötettäessä.

Liikunnan seurantaan käyttäjä merkitsee liikuntasuorituksensa, jonka pohjalta ohjelma laskee, kuinka paljon kaloreita käyttäjä on päivän aikana kuluttanut. Kulutusta voi verrata siihen, kuinka paljon käyttäjä on saanut kaloreita.

Tietopankki sisältää apua ravitsemukseen ja liikuntaan, kuten painonhallintavinkkejä, liikuntasuunnitelman ja terveellisiä ateriaesimerkkejä.

4.4.1 Windows Phone

Windows Phonen käyttöliittymässä seurataan Metro-muotokielen ohjeita. Sovellus on toteutettu käyttäen Panorama-mallia. Sovelluksen ensimmäisellä sivulla on yleiskuva Koutsi-hahmosta, liikkumalla oikealle nähdään järjestyksessä tietopankki, kalorilaskuri ja liikunnan seuranta. Näiltä yleiskatsaussivuilta voidaan navigoida syvemmälle osion sisälle ja saada tarkempaa tietoa aiheesta.



Kuva 6. Windows Phone –käyttöliittymä, tietopankki

Koutsi-sovelluksen Windows Phone –versio noudattaa tarkasti Metron ulkonäkösuosituksia. Väreinä on käytetty mustaa ja valkoista, joita korostetaan sinisellä. Pääelementit ovat kooltaan isoja, jotta käyttäjä huomaa ne välittömästi. Ne on eroteltu selkeästi joko rajaten tai erottaen ne käyttäen tyhjää tilaa. Sovelluksen typografia noudattaa Metron yleisiä tapoja. Fonttiperheenä toimii Segoe, josta käytetään eri versioita eripaksuista ja -kokoista tekstiä varten.

Kalorilaskuri koostuu mittarista, joka näyttää käyttäjälle, kuinka paljon kaloreita tämä on päivän aikana syönyt ja kuinka paljon on vielä jäljellä ennen kuin suositeltu raja ylittyy. Tästä näkymästä pääsee muokkaamaan päivittäistä rajaa ja tarkastelemaan historiaa, jossa näkyy edellisten päivien lukemat. Liikuntaosio on ulkonäöltään samanlainen kuin kalorilaskuri. Mittari näyttää, kuinka monta kaloria käyttäjä on päivän aikana kuluttanut. Molemmissa tapauksissa käyttäjä

joutuu syöttämään itse haluamansa tiedon, mutta syöttövaiheessa sovellus osaa ehdottaa oikeita arvoja tietokannasta.

Tietopankki sisältää tietoa eri liikuntamuodoista ja ravitsemuksesta. Tarjolla on myös ehdotuksia erilaisiin terveellisiin aterioihin. Tieto löytyy artikkeleina joko selaamalla tai asiasanahaualla. Ateriaesimerkit on jaoteltu aamu- ja iltapalaan, lounaaseen, päivälliseen ja välipaloihin. Ateriaa klikkaamalla avautuu uusi näkymä, joka sisältää lisätietoa, kuten valmistusohjeet.

Jos sivulta toiselle siirtymisessä tai jossain muussa toiminnossa kestää kahta sekuntia kauemmin, tulee näkyviin ikoni, joka ilmoittaa käyttäjälle, että sovellus työskentelee taustalla. Virheilmoitukset annetaan pienessä ikkunassa, joka aukeaa tarvittaessa sovelluksen päälle. Ilmoituksessa pyritään kertomaan käyttäjälle, mikä aiheutti virhetilanteen. Sovelluksen asetukset ja lisätiedot on sijoitettu toimintopalkkiin, jonka saa näkyville hipaisemalla näytön yläreunaa. Koska sovellus kerää käyttäjän tietoja, täytyy täällä myös näkyä, mitä tietoja kerätään ja mihin niitä käytetään.

Microsoftin omat vaatimukset Windows Phone –sovelluksille asettavat omalta osaltaan vaatimuksia sovelluksen käytettävyydelle (Taulukko 2). Microsoftin suorittama varmennus sisältää ohjeita selkeyteen, navigointiin ja virheiden raportointiin. Käyttäjälle pitää näyttää viesti, jollei sovellus pysty näyttämään sisältöä muutaman sekunnin sisällä. Laitteen painikkeiden toiminta on tarkasti määritelty, kuten takaisin-painike, jonka täytyy viedä käyttäjä aina edelliselle sivulle, paitsi pääsivulla, jolloin se sulkee koko sovelluksen. [14]

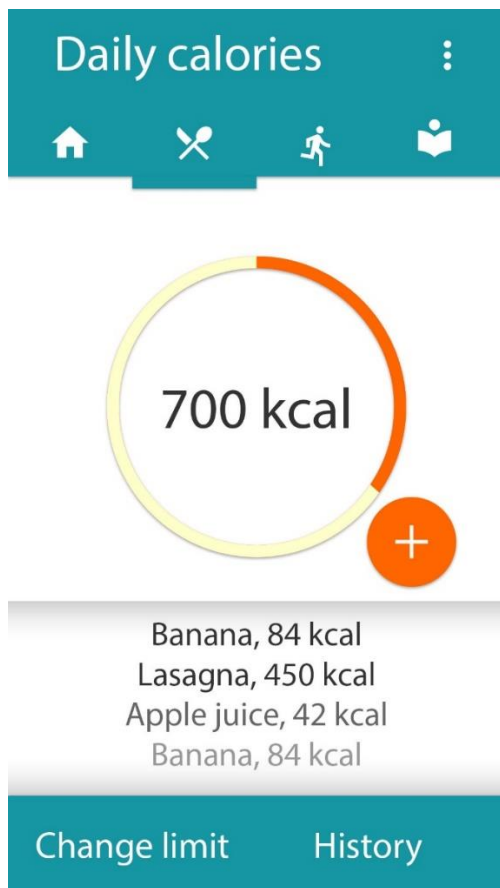
Taulukko 2: Windows Phone –version käytettävyyden arviointi

Järjestelmän tilan näkyvyys	Sovellus näyttää latausikkunan jos sivun lataamisessa menee tarpeettoman kauan. Huomioitu myös Microsoftin sertifiointissa.
Yhtenäisyys ohjelman ja maailman välillä	Sovellus käyttää käyttäjälle tuttua termistöä. Ikonit ovat verrattavissa fyysisen maailman esimerkkeihin.
Käyttäjän vapaus	Käyttäjä pääsee aina siirtymään edelliselle sivulle ja sulkemaan auki olevat ikkunat yhdellä painalluksella. Käyttäjä voi sulkea sovelluksen vain aloitussivulta.
Yhdenmukaisuus	Järjestelmän toiminnoista käytetään samoja nimiä kuin muissa ohjelmissa, kuten lataa ja tallenna.
Virheiden estäminen	Ohjelma antaa käyttäjälle ohjeita ohjelman käytöstä.
Toimintojen näkyvyys	Asiaankuuluvat toiminnot ovat koko ajan näkyvillä, paitsi lisätoimintopalkki, jonka saa näkyville sormeaa liu'uttamalla.
Joustavuus ja tehokkuus	Valikkorakenne on nopeasti navigoitavissa.
Minimalistinen suunnittelu	Näkyvissä ei ole turhaa tietoa. Huomioitu myös Metro-muotokielessä. Suuri osa elementeistä on tekstimuodossa.

Virheilmoitusten selkeys	Tunnetuista virheistä on tehty selkokieliset virheilmoitukset.
Dokumentaatio	Dokumentaatiota löytyy sovelluskaupasta ja asetusvalikosta.

4.4.2 Android

Android-käyttöliittymä seuraa rakenteeltaan Windows Phonen panoraamaa. Sivujen välillä liikutaan samaan tapaan sivusuunnassa, mutta niiden yli voi myös hypätä yläreunan valikon kautta. Material designin tapaan värit ovat kirkkaita ja selkeitä. Värimaailma on myös huomattavasti kirkkaampi kuin Windows Phone -versiossa. Tämä käyttöliittymä on tarkoitettu erityisesti puhelinten näytöille. Isokokoisemmilla Android -laitteilla käytetään käyttöliittymää, joka muistuttaa HTML-versiota.



Kuva 7. Android-käyttöliittymä, kalorilaskuri

Kalorilaskuri on toiminnaltaan samanlainen kuin Windows Phonen, mutta sen ulkonäkö on muokattu sopimaan paremmin Material designiin. Tässä versiossa se on ympyräkaavio, joka täyttyy päivän kiintiön mukaisesti. Liikuntapuolella käytetään pylväskaavioita, jotka näyttävät vierekkäin kulutetut ja syödyt kalorit. Sivujen elementit liikahtavat niiden tullessa näkyviin ja ne reagoivat myös kosketukseen. Ohjauselementeissä käytetään enemmän ikoneita kuin Windows Phone –versiossa.

Sovellus toimii virhetilanteissa samoin kuin Windows Phone –versio (Taulukko 3). Virheilmoitukset annetaan erilliseen ikkunaan ja niissä pyritään selvittämään, mikä johti virhetilanteeseen. Asetukset ja dokumentaatio on löydettävissä lisävalikosta, joka aukeaa painamalla ikonia oikeassa yläkulmassa.

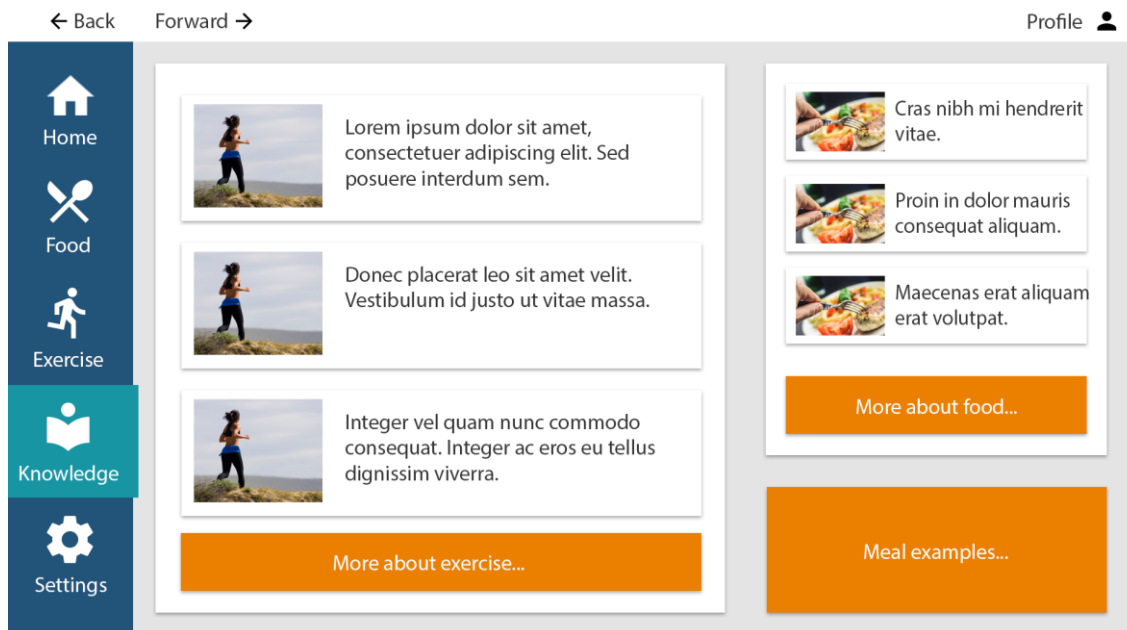
Taulukko 3: Android–version käytettävyyden arviointi

Järjestelmän tilan näkyvyys	Sovellus näyttää latausikkunan jos sivun lataamisessa menee tarpeettoman kauan.
Yhtenäisyys ohjelman ja maailman välillä	Sovellus käyttää käyttäjälle tuttua termistöä. Ikonit ovat verrattavissa fyysisen maailman esimerkkeihin.
Käyttäjän vapaus	Käyttäjä pääsee aina siirtymään edelliselle sivulle ja sulkemaan auki olevat ikkunat yhdellä painalluksella. Käyttäjä voi sulkea sovelluksen milloin vain.
Yhdenmukaisuus	Järjestelmän toiminnoista käytetään samoja nimiä kuin muissa ohjelmissa, kuten lataa ja tallenna. Ikoneina käytetään Androidin valikoimaa.
Virheiden estäminen	Ohjelma antaa käyttäjälle ohjeita ohjelman käytöstä.

Toimintojen näkyvyys	Asiaankuuluvat toiminnot ovat koko ajan näkyvillä. Lisävalikon saa auki painamalla ikonia sovelluksen yläkulmassa.
Joustavuus ja tehokkuus	Valikkorakenne on nopeasti navigoitavissa.
Minimalistinen suunnittelu	Näkyvissä ei ole turhaa tietoa. Käyttöliittymäelementeissä käytetty hienovaraista liikettä käyttäjän huomion kohdistamista varten.
Virheilmoitusten selkeys	Tunnetuista virheistä on tehty selkokieliset virheilmoitukset.
Dokumentaatio	Dokumentaatiota löytyy sovelluskaupasta ja asetusvalikosta.

4.4.3 HTML-sovellus

HTML-sovellus muistuttaa tyyliltään Android-versiota, ja siinä on otettu mallia Material Designista, mutta se ei noudata mitään tiettyä muotokieltä. Ulkoasussa on käytetty tasaisia värejä ja teräviä kulmia. Koska käyttöliittymän täytyy toimia erilaisissa Internet-selainta käyttävissä laitteissa, siitä tehdään erikokoisia ja asetelultaan erilaisia versioita eri näyttökokoihin. Lähtökohtana on, että HTML-sovellusta käytetään isommilla näytöillä, kuten tietokoneella ja tableteilla, mikä mahdollistaa suuremman määrän informaatiota ruudulla samanaikaisesti. Tila mahdollistaa esimerkiksi sen, että tietopankin pääsivulle voidaan nostaa muutama artikkeli, joita voi lukea ilman, että tarvitsee navigoida alasivuille (Kuva 8).



Kuva 8. HTML-käyttöliittymä, tietopankki

Kaikki pääsivut on jaettu kahteen sarakkeeseen. ”Home”-sivun tapauksessa vasemmalle on sijoitettu käyttäjän tiedot ja ajankohtaisia viestejä liikuntaan ja ravintoon liittyen. Oikealle puolelle nostetaan satunnaisia artikkeleita tietopankista. Kalorilaskurissa käytetään samanlaista ympyräkaaviota kuin Android-versiossa, mutta sen lisäksi oikeassa reunassa on myös kolmen viime päivän kaaviot. Kolmea päivää vanhempia tietoja varten on erillinen ”History”-sivu. ”Exercise”-sivulla esitetään kuluvan viikon aikana kulutetut kalorit pylväskaaviona.

Eniten HTML-version lisätilaa hyödyntää tietopankki, jonka artikkelit voivat sisältää monipuolisempaa muotoilua, kuten kuvia, kaavioita ja taulukoita. Selainpohjaisuus myös helpottaa ulkoisten lähteiden linkkaamista, koska nämä voidaan näyttää samassa ikkunassa siirtymättä pois sovelluksesta (Taulukko 4).

Taulukko 4: HTML –version käytettävyyden arviointi

Järjestelmän tilan näkyvyys	Sovellus näyttää latausikonin jos tiedon näyttämässä menee tarpeettoman kauan.
Yhtenäisyys ohjelman ja maailman välillä	Sovellus käyttää käyttäjälle tuttua termistöä. Ikonit ovat verrattavissa fyysisen maailman esimerkkeihin.
Käyttäjän vapaus	Sovelluksessa on navigointipalkki, joka sisältää ”eteenpäin”- ja ”takaisinpäin” –komennot. Käyttäjä voi sulkea sovelluksen milloin vain.
Yhdenmukaisuus	Järjestelmän toiminnoista käytetään samoja nimiä kuin muissa ohjelmissa, kuten lataa ja tallenna. Ikoneina käytetään Androidin valikoimaa.
Virheiden estäminen	Ohjelma antaa käyttäjälle ohjeita ohjelman käytöstä.
Toimintojen näkyvyys	Asiaankuuluvat toiminnot ovat koko ajan näkyvillä. Lisävalikon saa auki painamalla ikonia sovelluksen yläkulmassa.
Joustavuus ja tehokkuus	Valikkorakenne on nopeasti navigoitavissa.
Minimalistinen suunnittelu	Näkyvissä ei ole turhaa tietoa.
Virheilmoitusten selkeys	Tunnetuista virheistä on tehty selkokieliset virheilmoitukset.
Dokumentaatio	Dokumentaatio löytyy asetusvalikosta. Kattavampi ohjeistus kuin muissa versioissa.

5 YHTEENVETO

Tämän työn tarkoituksena oli suunnitella tehokas ja käyttäjäystävällinen mobiilikäyttöliittymä terveydenhallintasovellukselle käyttäen hyödyksi käytettävyyssiiteen periaatteita. Käyttöliittymä suunniteltiin Android- ja Web-alustoille ja arviointiin sisällytettiin aikaisemmin kehitetty Windows Phone 7 – testiversio. Arviointiin käytettiin Jakob Nielsenin heuristista listaa, joka on yleinen työkalu käytettävyyden arvioinnissa. Sovelluksen kehitys jatkuu käyttöliittymän suunnittelun jälkeen, ja siitä on tarkoitus kehittää kansainvälisesti myytävä tuote. Sovellus julkaistaan aluksi Androidille ja Internet-selaimella toimivaksi HTML-sovellukseksi.

Suunnittelussa otettiin huomioon aikaisemmassa testausvaiheessa ilmenneet ongelmat sekä käyttäjäkunnan toiveet ja ideat. Terveystieteiden opiskelijat järjestivät testauksen, ja aiheesta toteutettiin opinnäytetyö. Aiheesta toteutettiin myös kaksi muuta opinnäytetyötä, jotka liittyivät sovelluksen kehitykseen ja muiden vastaavien sovellusten kartoitukseen.

Arviointia varten tutustuttiin käytettävyyssiiteen perusteisiin tietotekniikassa. Huomiota kiinnitettiin erityisesti affordansseihin ja käyttäjäkeskeiseen suunnitteluun. Työssä pohdittiin, mitä hyötyä tai haittoja käyttäjien sisällyttämisellä voi olla tuotekehitysprosessiin.

Työssä käytiin läpi myös yleisimmät mobiilialustat ja muotokielet, joita niille tehdyissä sovelluksissa tulisi noudattaa. Pohdittiin, miten alustat eroavat toisistaan ja miten erot vaikuttavat sovelluksen kehitykseen.

LÄHTEET

- [1] Nielsen Norman Group, Usability 101: Introduction to usability. Viitattu 6.4.2015
<http://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [2] Nielsen Norman Group, 10 Usability Heuristics for User Interface Design. Viitattu 6.4.2015
<http://www.nngroup.com/articles/ten-usability-heuristics/>
- [3] International Organization for Standardization. Viitattu 7.12.2015
<http://www.iso.org/>
- [4] Gaver, W. 1991. Technology affordances. Cambridge: EuroPARC.
- [5] Smashing Magazine, The History Of Usability: From Simplicity To Complexity. Viitattu 21.4.2015
<http://www.smashingmagazine.com/2012/05/23/the-history-of-usability-from-simplicity-to-complexity/>
- [6] Abras, C.; Maloney-Krichmar, D. ; Preece, J. 2004. User-Centered Design
- [7] International Data Corporation, Press Release. Viitattu 21.4.2015
<http://www.idc.com/getdoc.jsp?containerId=prUS25450615>
- [8] Engadget, Lollipop isn't making a dent in Android usage numbers (yet). Viitattu 21.4.2015
<http://www.engadget.com/2014/12/01/android-version-share-december-2014/>
- [9] Webopedia.com, Android fragmentation. Viitattu 21.4.2015
http://www.webopedia.com/TERM/A/Android_fragmentation.html
- [10] Google, Material Design Guidelines, Viitattu 21.5.2015
<http://www.google.com/design/spec/material-design/introduction.html>
- [11] ITPro, Windows 10 vs. Windows 8.1: What's new? Viitattu 21.5.2015
<http://www.itpro.co.uk/desktop-software/23216/windows-10-vs-windows-81-what-s-new>
- [12] Turun ammattikorkeakoulu, InnoHealth. Viitattu 23.4.2015
http://www.turkuamk.fi/media/pdf/innohealth_17.pdf
- [13] Leminen, A. & Muli, M. 2013. Mobiiliterveyspeliin liittyvät tarpeet, toiveet ja ideat
- [14] Microsoft, App certification requirements, Viitattu 1.9.2015
<https://msdn.microsoft.com/en-us/library/windows/apps/hh694083.aspx>

[15] AndroidOS.in, Updated Gmail app with material design arrives this week in Play Store, Viitattu 1.9.2015

<http://www.androidos.in/2014/11/updated-gmail-app-material-design-arrives-week-play-store/>

[16] Morricks.me, Skeuomorphism and the eye of the beholder, Viitattu 1.9.2015

<http://morricks.me/archives/3727>

[17] Windowsadmins.com, Adding polish to improve the look and feel of your app, Viitattu 1.9.2015

<http://windowsadmins.com/adding-polish-to-improve-the-look-and-feel-of-your-app-10-by-10/>