
KÄYTTÖLIITTYMÄN SUUNNITTELU ANDROID-SOVELLUKSELLE

Case oldtimerTimer



Ammattikorkeakoulun opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Visamäki, syksy 2014

Erkka Wessman



Visamäki
Tietojenkäsittelyn koulutusohjelma
Systeemityö

Tekijä	Erkka Wessman	Vuosi 2014
Työn nimi	Käyttöliittymän suunnittelu Android-sovellukselle Case oldtimerTimer	

TIIVISTELMÄ

Työn tavoitteena oli suunnitella käyttöliittymä oldtimerTimer-mobiilisovellukselle, joka on uusi vanhusten käyttöön tarkoitettu avustus-, muistutus- ja seurantasovellus. Sovellusta kehittää Hämeen ammattikorkeakoulu. Työn toimeksiantajana ja yhteistyökumppanina toimi Linnan Ateria Oy, joka tarjoaa vanhuksille mahdollisuuden ruokailla Hämeenlinnassa paikoissa, joihin Linnan Ateria toimittaa ruokaa.

Sovelluksesta oli tarkoitus alun perin tulla ruokalätietosovellus, jonka avulla ikäihmiset löytäisivät ruokailupaikkoja helpommin. Kehitysprojektin aikana sovellus kuitenkin laajentui ja siitä tuli osa suurempaa järjestelmää, johon kuuluu myös verkkosivusto. Järjestelmän tarkoituksena on helpottaa vanhusten arkielämää. Mobiilisovellus auttaa vanhuksen hakeutumista erilaisiin aktiviteetteihin sekä helpottaa erilaisten rutiinien järjestelyä.

Työ on toteutettu konstruktiivisella tutkimusmenetelmällä, mutta siihen kuuluu teoriapainotteista pohjatutkimusta. Tietolähteinä työssä käytettiin mahdollisimman tuoreita kirjoja ja internet-artikkeleita. Osa taustateoriasta on hypoteettista, eli omaa päättelyä erilaisten hyviksi ja toimiviksi todettujen käyttöliittymäratkaisujen pohjalta. Myös oldtimerTimerin käyttöliittymän testauksesta kerättyjä mielipiteitä käytetään tiedon lähteenä.

Sovelluksen varsinainen kehitys aloitettiin lähes samanaikaisesti kuin sen käyttöliittymän suunnittelu, mutta sovelluksen kehitys tulee jatkumaan vielä syksyllä tämän opinnäytetyön jälkeen. Työn tuloksena syntyi interaktiivinen käyttöliittymän prototyyppi ja käyttöliittymäspeksi, jotka asettavat sovelluksen jatkokehitykselle selkeät tavoitteet. Työ toimii myös suunnannäyttäjänä muille vanhusten käyttöön suunnatuille sovelluksille tulevaisuudessa.

Avainsanat käyttöliittymä, suunnittelu, grafiikka, mobiilisovellus, Android

Sivut 38 s. + liitteet 12 s.

Visamäki
Degree Programme in Business Information Technology
System Engineering

Author	Erkka Wessman	Year 2014
Subject of Bachelor's thesis	Android application user interface design Case oldtimerTimer	

ABSTRACT

The objective of this thesis was to design a user interface for the oldtimer-Timer mobile application, which is a new application for the elderly meant to aid, remind and monitor its users. The application is being developed by HAMK University of Applied Sciences. The thesis is commissioned and collaborated by Linnan Ateria Oy. They are offering a chance for elders to dine at places to which they are supplying food in Hämeenlinna.

The purpose of the application at first was to provide lunchroom information, so that elders would more easily find places to dine at. However, during the development project, the application expanded and became a part of a larger system that includes a website. The purpose of the system is to ease the everyday life of elders. The mobile application helps the person to seek various activities and simplifies the arrangement of various routines.

A constructive research method was used in this thesis, but it includes theory-oriented research. Books and internet articles used for information were as recent as possible. Part of the background theory is hypothetical based on various successful user interface solutions. In addition, the feedback gathered from the oldtimerTimer user interface testing is used as a source of information.

The development of the application was started almost simultaneously with the user interface design, but the development will continue in the fall after this thesis. The result of this thesis was an interactive user interface prototype and a user interface specification. These results set clear objectives for the further development of the application. The thesis also acts as future direction for other applications designed for the elderly.

Keywords user interface, design, graphics, mobile application, Android

Pages 38 p. + appendices 12 p.

KÄSITELUETTELO

Bittikarttagrafiikka

Yleinen tapa esittää kuvia digitaalisessa muodossa. Kutsutaan yleisesti myös pikseligrafiikaksi. Se koostuu pikseleistä eli kuvapisteistä, jotka muodostavat suoran ruudun. Jokaiselle pikselille on määritetty tietty väri.

GUI (Graphical User Interface)

Graafinen käyttöliittymä, tietokonesovelluksissa käytettävä näkymä tai työpöytäympäristö, jonka avulla käyttäjä pystyy käyttämään sovelluksen toimintoja. Tyypillisiä käyttöliittymän elementtejä ovat ikkunat, painikkeet, kuvat, valikot ja tekstit. Usein graafisesta käyttöliittymästä käytetään pelkästään lyhennettä UI.

Haptinen teknologia

Tarkoittaa tuntoaistiin perustuvaa teknologiaa. Sitä hyödynnetään yleisimmin syöttölaitteissa, kuten ohjaimissa ja näppäimistöissä. Tärinä on haptista eli tuntoaistiin perustuvaa palautetta.

Layer

Termi, joka kuvaa näkymän tai kuvan kerrosta. Tarkoittaa Photoshopissa kuvan yksittäistä muokattavaa osaa, ja käyttöliittymissä näkymän päälle avautuvaa toista näkymää.

Layout

Visuaalista asettelua tarkoittava englanninkielinen sana. Tarkoittaa käyttöliittymäsuunnittelussa graafisten elementtien asettelua yksittäisessä näkymässä.

Natiivi

Termi, jota käytetään tietotekniikassa kuvaamaan sovellusta tai tietoformaattia, jota tietty käyttöjärjestelmä tukee. Natiivi sovellus ei tarvitse lisäkomponentteja, jotta se toimisi järjestelmässä.

Pikselitiheys

Määre, jota kuvataan yleisimmin mittayksiköllä pixels per inch (PPI). Mittayksikkö tarkoittaa kuvaa pikselien määrää neliötuuman alueella. Sitä käytetään muun muassa näyttölaitteiden tarkkuuden ilmaisuun.

Resoluutio

Termi, jolla kuvataan bittikarttakuvan erotuskykyä eli yksityiskohtien määrää. Resoluutio voi myös merkitä kuvantoistolaitteen erottelukykyä eli tarkkuutta. Resoluutio ilmaistaan yleensä kuvan pikselien määränä vaaka- ja pystysuunnassa, esimerkiksi 1920x1080.

UX (User Experience)

Käyttäjäkokemus, termi jota on alettu soveltaa tietoteknisten tuotteiden suunnitteluun. Tuotteen käyttäjäkokemukseen kuuluu käyttäjän kokonaisvaltainen elämys, käyttäytyminen ja asenne tuotetta kohtaan. Käyttöliittymä on merkittävä osa käyttäjäkokemusta.

Vektorigrafiikka



Digitaalinen grafiikka, joka perustuu koordinaatistoon ja matemaattisiin funktioihin. Koordinaatistoon sidotaan objekteja, joilla voidaan muodostaa kuva. Vektorigrafiikan riippumattomuus pikseleistä tekee siitä helposti skaalattavaa.

Widget

Termi, joka tarkoittaa tietokonejärjestelmässä toimivaa pienen sovellusta, joka toimii osana suurempaa käyttöliittymää tai näkymää. Androidin widgetit ovat kotinäkömään lisättäviä pieniä sovelluksia tai sovelluksen osia.

XML (Extensible Markup Language)

Merkintäkieli, jossa tiedon merkitys voidaan kuvata tiedon sekaan. Sitä käytetään tallentamiseen ja järjestelmien välisen tiedonvälityksen formaattina.

SISÄLLYS

1	JOHDANTO.....	1
2	TAVOITTEET	3
2.1	Käyttöliittymäsuunnittelu.....	3
2.2	Käytännön tulokset.....	3
3	KÄYTETTÄVYYS	5
3.1	Käytettävyyden suunnittelu.....	5
3.2	Esteettömyys vanhuksille.....	6
3.3	Androidin käytettävyys	7
3.4	Vallitsevat trendit	9
3.4.1	Minimalistisuus	10
3.4.2	Monitasoisuus ja sumennus.....	10
3.4.3	Kortit ja jatkuvat sivut.....	11
4	MENETELMÄT JA TYÖKALUT	13
4.1	Photoshop.....	13
4.2	Illustrator	13
4.3	Sketch.....	14
4.4	InVision.....	15
4.5	Fluid UI.....	16
4.6	Paperiluonnokset	17
4.7	CamScanner	18
5	SOVELLUKSEN SUUNNITTELU.....	20
5.1	Lähtötilanne.....	20
5.2	Prototyyppien luonti.....	21
5.3	Prototyyppi iPhonelle.....	22
5.4	Näkymien rakenne.....	23
5.4.1	Aloituskäyttö.....	24
5.4.2	Asetukset	24
5.4.3	Apua	25
5.4.4	Ilmoitukset.....	26
5.4.5	Uuden painikkeen luonti.....	27
5.4.6	Oma muistilista.....	28
5.5	Visuaalinen tyyli ja värimaailma.....	29
5.6	Prototyyppien testaus	30
5.7	Käyttöliittymän toteutus.....	31
5.8	Skaalaus.....	32
5.9	Projektiryhmän osana toimiminen	33
6	JATKOKEHITYS	35
7	YHTEENVETO	37
	LÄHTEET	38

1 JOHDANTO

Tämän työn aiheen valinta oli monen vaiheen tulos. Syksyllä 2013 työharjoittelupaikkaa etsiessä graafiset käyttöliittymät alkoivat kiinnostaa tämän opinnäytetyön tekijää. Silloin mielessä oli web-käyttöliittymät, mutta älypuhelimien kasvattaessa suosiotaan mobiilisovellukset ovat entistäkin ajankohtaisempia. Sitten työharjoittelussa Espanjassa kansainvälinen työskentely alkoi tuntua kiehtovalta tulevaisuuskuvalta. Käyttöliittymäsuunnittelun pohjalta olisi hyvinkin mahdollista lähteä jatkossa ulkomaille töihin.

Työn yhteistyökumppani Linnan Ateria Oy tarjoaa vanhuksille mahdollisuuden ruokailla Hämeenlinnassa paikoissa, joihin Linnan Ateria toimittaa ruokaa. Tällaisia paikkoja ovat esimerkiksi koulujen ja kokopäivälaitoksien ruokalot. Linnan Ateria toivoi, että vanhuksille saataisiin sovellus, jonka avulla he löytäisivät nämä mahdolliset ruokailupaikat helpommin. Sovellusta aloitettiin tekemään Linnan Aterian toiveiden pohjalta, mutta Hämeen ammattikorkeakoulu omistaa sovelluksen.

Hämeen ammattikorkeakoulun kesäprojektina oli oldtimerTimer-sovelluksen kehittämisen aloitus, johon kuuluu sekä web-sivusto että mobiilisovellus. oldtimerTimerin pääasiallinen kohderyhmä oli ikäihmiset, joten helppokäyttöisyyden toteuttaminen oli siinä avainasemassa. Mobiilisovellus oli tarkoitus kehittää ensiksi Androidille. Sovelluksen käyttöliittymän suunnittelussa oli paikka vapaana, joten tilaisuuteen täytyi tarttua. Aihe oli sopiva minulle, ja koska itselläni ei juurikaan ollut aikaisempaa kokemusta käyttöliittymien suunnittelusta, helppokäyttöisyydestä aloittaminen oli luontevaa.

Työn keskeisenä ideana on tuottaa oldtimerTimer-mobiilisovellukselle käyttöliittymä, joka toimii hyvin kehitysajankohdan uusimmilla Android-puhelimilla. Sen pitää olla mahdollisimman helppokäyttöinen ikäihmisille, mutta kuitenkin noudattaa mahdollisimman paljon Androidin johdonmukaista linjaa. Työssä toteutetaan valmis käyttöliittymäprototyyppi, joka helpottaa sovelluksen kehitystä kesän jälkeen. Sovelluksen tekniseen toteutukseen eli ohjelmointiin työ ottaa kantaa vain teoreettisella tasolla. Työssä tutkitaan myös käyttöliittymäsuunnittelussa käytettäviä erilaisia suosittuja toimintamenetelmiä ja työkaluja, vaikka kaikkia tutkittuja menetelmiä ei työssä hyödynnetä.

Opinnäytetyön keskeisimmät tutkimuskysymykset:

- Mitkä toimintatavat ja työkalut ammattimaisen käyttöliittymäsuunnittelijan pitää hallita?
- Millä tavoin käyttöliittymäsuunnittelussa tehdään yhteistyötä sovelluksen muiden toteuttajien kanssa?
- Miten sovelluksen käyttöliittymästä tehdään yhtäaikaaisesti sekä moderni että vanhuksille sopiva?

-
- Miten sovelluksen jatkokehitys täytyy huomioida käyttöliittymän suunnittelussa?

2 TAVOITTEET

Työn perimmäinen tavoite on tehdä oldtimerTimer-sovelluksen käyttöliittymästä ajanmukainen, mutta myös vanhusten käyttöön sopiva. Tällaisen käyttöliittymän toteutus vaatii pohjatiedon keräämistä erityisesti vanhuksille sopivista käyttöliittymäratkaisuksista sekä nykyaikaisesta GUI-suunnitteluprosessista. Koska aikaisempaa kokemusta prosessista ei juurikaan ole, pitää selvitys aloittaa perusasioista. Tässä luvussa pohjustetaan työn tietoperusta ja tulokset.

2.1 Käyttöliittymäsuunnittelu

Työ tarjoaa kattavan katsauksen käyttöliittymäsuunnittelusta ja keskittyy mobiilisovelluksiin, tarkennettuna Android-käyttöjärjestelmään. Siinä käydään läpi teoriaa käytettävyydestä ja esteettömyydestä. Android käyttöliittymäsuunnittelijan näkökulmasta tulee työn aikana tutuksi. Työssä keskitytään moderniin käyttöliittymäsuunnitteluun, johon kuuluu alan vallitsevat trendit, käytetyimmät menetelmät ja työkalut.

Työn tietoperustana käytetään mahdollisimman tuoreita kirjoja ja internet-artikkeleita, pääasiassa 2010-luvulta. Itse menettelytavat eivät ole ratkaisevasti muuttuneet viimeisen kahden vuosikymmenen aikana, mutta tekniikka on ottanut suuria harppauksia, ja älypuhelimien suosio lähti kasvuun vasta vuonna 2007 ensimmäisen iPhone:n myötä. Nykyisin kosketuskäyttöliittymiä alkaa olla kaikkialla, ja niitä on mahdotonta välttää mobiilipuolella. Tämän kehityksen takia hyödyllinen tieto useimmista vanhemmista kirjoista jää vähäiseksi. Suurin osa tiedosta onkin kerätty internetin tuoreista artikkeleista ja tutkimuksista.

Osa taustateoriasta on hypoteettista ja omaa päättelyä. Käyttöliittymäsuunnittelu on kuitenkin siinä määrin luovaa työtä, että suoraa kirjoitettua vastausta kaikkeen ei ole saatavilla. Internetissä on lähes loputtomasti listauksia ja esimerkkejä toimivista käyttöliittymäratkaisuksista. Näistä listauksista on mahdollista päätellä vallitsevia trendejä ja yleisesti pidettyjä ratkaisuja. Työstäessä oldtimerTimerin käyttöliittymää myös testauksesta saatavia mielipiteitä hyödynnetään käyttöliittymän kehittämiseen. Kaiken tämän tietoperustan pohjalta valitaan parhaita menetelmiä oldtimerTimerin kehittämiseen eli työn käytännön osuuteen.

2.2 Käytännön tulokset

Käytännön osuudessa tehdään oldtimerTimerin käyttöliittymää Android-puhelimille osana Hämeen ammattikorkeakoulun kesäprojektia. Ensinnä käydään läpi sovelluksen kehitystä edeltävä hahmotteluvaihe. Sen jälkeen, projektin alettua, tehdään paperille luonnoksia käyttöliittymän eri ruuduista ja testataan näitä. Tämän jälkeen tehdään lopullinen visuaalinen prototyyppi käyttöliittymälle FluidUI-sovelluksella, johon on koottu parhaat puolet kaikista aikaisemmista hahmotelmista. Lopuksi selvitetään, miten suunniteltu käyttöliittymä voidaan toteuttaa

ohjelmointiympäristössä, ja miten käyttöliittymä skaalataan erikokoisille näytöille.

Käyttöliittymän lopullinen toteutus ja skaalaus eivät kuitenkaan kuulu tähän opinnäytetyöhön, se on sovelluksen ohjelmoijien tehtävä. Sovelluksen kehitys tulee jatkumaan tämän opinnäytetyön jälkeen, ja saattaa laajentua paljonkin. Opinnäytetyön viimeinen osuus käsittelee sovelluksen jatkokehitysmahdollisuuksia ja sitä, miten jatkokehitys on otettu huomioon käyttöliittymässä.

3 KÄYTETTÄVYYS

Käytettävyysteoria ei rajoitu pelkästään tietokoneohjelmistoihin ja muihin teknisiin sovelluksiin. Se liittyy kaikkiin tuotteisiin, jotka on tarkoitettu käytettäväksi. Käyttäjää on monenlaisia ja heidän vaatimuksensa tuotteelta saattavat olla erilaisia, eivätkä kaikki käytettävyysteoriat eivät ole todistettuja, mutta käytettävyyseratkaisut harvemmin ovat makuasioita (Sinkkonen 2002, 10). Kulttuurilliset seikat voivat määritellä käyttäjien olettamuksia ja mieltymyksiä, mutta geneettisesti käyttäjät ovat hyvin samankaltaisia ja pystyvät suorittamaan samanlaisia toimintoja. Tässä luvussa käsitellään käytettävyyden teoriaa, helppokäyttöisyyttä painottaen. Aihetta on rajattu pääasiassa mobiilisovelluksiin ja kosketusohjaukseen.

3.1 Käytettävyyden suunnittelu

Sovellusten käytettävyyden suunnittelu jaetaan yleensä kahteen eri työnkuvaan: UX-suunnittelija ja UI-suunnittelija. UX on käyttäjäkokemuksen suunnittelua, UI taas käyttöliittymän suunnittelua. Käyttöliittymä on työkalu, jonka avulla käyttäjä kommunikoi sovelluksen kanssa. Käyttäjäkokemus on kokonaisuus, koko sovelluksen tarjoama elämys, jonka näkyvin osa on käyttöliittymä. UX on monialaista työtä, jossa vaaditaan psykologista ymmärrystä, kokonaiskuvan hallintaa ja kriittistä ajattelua. Monesta kokeneesta UI-suunnittelijasta tulee lopulta UX-suunnittelija, koska sitä pidetään haastavampana työnä. Käyttäjäkokemusta ei kuitenkaan ole ilman käyttöliittymää. Käyttöliittymän visuaalinen puoli on nykypäivänä tärkeämpi kuin koskaan, varsinkin nuorille suunnatuissa tuotteissa. Harva sovellus menestyy ilman kaunista ja toimivaa käyttöliittymää. (Miller 2012.)

Monissa yrityksissä UI- ja UX-suunnittelijan työroolit ovat yhdistetty. Tässä työssä keskitytään enemmän UI-suunnitteluun, mutta käyttäjäkokemuksellisia asioita ei voida jättää täysin huomioimatta. Käyttöliittymäsuunnittelussa kaksi työvaihetta, luonnostelu ja luonnosten testaus tai esittely vie suurimman osan suunnittelijan ajasta. Sovelluksen kehityksen aikana on muitakin työvaiheita, joissa UI-suunnittelijan kannattaa olla mukana.

Simõesin (2012) mukaan kehityksen alussa suunnittelijaa tarvitaan, kun päätetään mitä sovelluksella halutaan saavuttaa ja mitä käyttäjät haluavat siltä. Suosittu toimintatapa on fiktiivisten käyttäjien luominen, jolloin selkeytyy, minkälaisissa tilanteissa sovellusta pitäisi pystyä käyttämään. Tällä tavalla näkee sovelluksen vaatimukset käyttäjän näkökulmasta. Sovelluksen tavoitteiden selkeydyttyä analysoidaan jo olemassa olevia samantyyppisiä sovelluksia. Joskus sovellusta vastaavaan tarkoitukseen ei ole olemassa, jolloin pitää unohtaa sovellusten sisältö ja tutkia niiden rakenteita.

Rautalankamallien ja prototyyppien luonnostelun voi tehdä nopeastikin, mikäli sovelluksen tavoitteet ja ominaisuudet ovat jo täysin päätetty. Jos sovelluksen päämääriä muokataan vielä kehityksen alettua, se hidastaa UI-suunnittelijan työtä, puhumattakaan ohjelmoijien työstä. Luonnostelua

varten on olemassa kymmeniä vartenotettavia työkaluja, perinteisestä kynästä ja paperista alkaen.

Valmiit käyttöliittymäprototyypit vaativat testausta tai asiakasesittelyjä, riippuen sovelluksen vaatimuksista. Jos käyttöliittymää tehdään asiakkaalle, joka tietää minkälainen sovelluksesta pitää tulla, luonnoksia voidaan esitellä suoraan asiakkaalle. Asiakas huomaa prototyyppien epäkohdat ja ohjaa suunnittelijaa yrittämään toisenlaisia ratkaisuja, jonka jälkeen suunnittelija tekee paranneltuja prototyyppejä. Tämän jälkeen luonnokset taas esitellään asiakkaalle, ja sama kaava voi toistua useammankin kerran.

Prototyyppien testaus toimii yleensä niin, että luonnokset annetaan testikäyttäjän eteen, ja käyttäjää pyydetään suorittamaan erilaisia toimintoja. Näin selviää, mitä puutteita tai vaikeasti ymmärrettäviä toimintoja käyttöliittymässä on. Testaus on järkevää erityisesti silloin, kun ei vielä täysin tiedetä mitä käyttäjät haluavat sovellukselta.

Lopuksi, ennen sovelluksen julkaisua, UI- tai UX-suunnittelija huolehtii että käyttäjäkokemus on sulava. Sitten lisätään vielä kokemusta parantavia yksityiskohtia. Sovellus on harvemmin heti julkaisussa täydellinen, joten joskus paras strategia on epäonnistuminen alussa ja sen jälkeinen nopea iterointi ja jatkuvat parannukset. (Simões 2012.)

3.2 Esteettömyys vanhuksille

Vanhusten ongelmat tietoteknisten sovellusten kanssa eivät useimmiten johdu suoranaisesti iästä. Ikä kertoo ihmisestä loppujen lopuksi vähän. Terveiden ikäihmisten testatut suoritusajat vaihtelevat ikäryhmän sisällä merkittävästi enemmän kuin nuoremmilla ikäryhmillä. (Sinkkonen 2002, 45–46.) Ikäväestö on vain jakautuneempaa, koska nuoremmilla jonkinasteinen tietotekninen taito on välttämättömyys jo pelkästään työssä ja sosiaalisessa kanssakäymisessä.

Kulttuurin kehitys on suurin syy vanhusten tietoteknisiin rajoitteisiin. Tekniikan kehittyminen on ollut vauhdikasta, ja aikuisiällä töiden ulkopuolella uusien asioiden oppiminen vaatii omistautumista ja kiinnostusta. Ikäihmisten sukupolvi tällä hetkellä ei ole täysin omaksunut internetiä tai varsinkaan älypuhelimia. Tulevien vanhusten sukupolvi, eli nykyiset 40–60-vuotiaat ovat jo tottuneet teknologiaan työssään. Internetin tai älypuhelimien käyttö tuskin tuottaa heille sen suurempia ongelmia vanhuuspäivillään, olettaen että nämä teknologiat eivät ota käytettävyyden suhteen suuria harppauksia.

Vanhusten suorituskyky on toki heikompaa kuin nuoremmilla. Näkö- ja kuuloaistit heikentyvät monilla niin paljon, ettei niitä voida korjata. Myös lihasten motoriikka heikkenee, jolloin etenkin sormien käytön tarkkuus heikentyy ja monimutkaisista eleistä tulee vaikeita. Ajatuksen kulku voi myös hidastua ja erillisiä asioita on vaikeaa käsitellä yhtäaikaaisesti. Heikentynyt suorituskyky voidaan ottaa huomioon sovelluksen suunnittelussa, ja käytön helpotukseen on olemassa joitakin tekniikoita.

Haptinen teknologia järkevästi käytettynä voi olla käännteentekevä ominaisuus, joka tekee käyttöliittymistä helpommin ymmärrettäviä. Tärinä on haptista palautetta, ja se auttaa käyttäjää ymmärtämään, että painallus on suoritettu. Nykyhetken haptista teknologiaa ovat kosketusnäytöt ja muut kosketuspinnat. Tulevaisuuden haptiset näytöt voivat tunnistaa painallusvoiman, ja pystyvät hienoisilla tärinäefekteillä tuomaan fyysistä tuntumaa sormieleisiin ja painalluksiin. Tällaisia pintoja on jo olemassa prototyyppiasteella. Kosketuspintoihin tottumattomille erilaiset hienostuneet tärinäpalautteet voivat olla suuri apu. (Lewis 2013.)

Heikentyneet aistit voidaan ottaa huomioon älypuhelinsovellusta suunniteltaessa seuraavin tavoin (Díaz & Moreno 2013):

- Käytetään suurempia elementtejä ja valinnan selkeää korostusta.
- Grafiikkaa käytetään vain tarvittaessa, ei koristeena, ja animaatiot pidetään minimissä.
- Ikonit pidetään yksinkertaisina ja tarkoituksenmukaisina.
- Värejä käytetään harkitusti ja kontrasti pidetään suurena, sinisen ja vihreän eri sävyjä kannattaa välttää.
- Vältetään ponnahdusikkunoita ja muita päällekkäisiä ikkunoita.
- Käytetään helppoa ja selkeää kieltä, pidetään tärkeä tieto näkyvissä keskellä ja vältetään turhaa tietoa.
- Elementtien asettelu, navigaatio ja termistö pidetään yksinkertaisena ja yhdenmukaisena.
- Tiedon lukemiseen annetaan tarpeeksi aikaa.
- Äänenvoimakkuus pidetään tarpeeksi suurena.
- Tärinää hyödynnetään kaikissa valinnoissa.

3.3 Androidin käytettävyys

Android on käyttöjärjestelmänä vielä nuori, ja on nyt vasta viime vuosina vakiinnuttanut asemansa mobiilialustojen terävimmässä kärjessä. Androidin käyttöliittymä ei ole kokenut mullistavia muutoksia olemassaolonsa aikana viidessä vuodessa. Vanhat tavaramerkit, kuten widgetit, alas vedettävä ilmoituspalkki ja vahva integraatio Googlen palveluihin ovat edelleen mukana. Sittemmin Androidin käyttöliittymään on istutettu puheentunnistus, lisätty pieniä ominaisuuksia ja visuaalinen ilme on uudistunut, mutta perusta on pysynyt samana. (Verge 2011.)

Android-puhelinten osuus maailman älypuhelimien myynnistä oli huikea 84,7 % vuoden 2014 toisella neljänneksellä. Applen iOS:n osuus oli 11,7 %, ja Microsoftin Windows Phonen 2,5 %. (IDC 2014.) Vielä vuonna 2009 iOS hallitsi älypuhelinmarkkinoita, mutta Android on kasvattanut suosiotaan tasaisesti. Yhdysvalloissa iPhone on edelleen todella suosittu ja sillä on oma vankka kannattajaryhmänsä muuallakin, mutta muualla maailmassa Android on täysin ylivoimainen laitemäärässä mitattuna. Jos mobiilisovellukselle halutaan valita vain yksi kehitysalusta Suomessa, Android on ainoa järkevä valinta.

Kuten kilpailijansa, Android on pääosin sormikosketuksella ja puheella ohjattava käyttöjärjestelmä. Kynäkosketus toimii myös ja on tarkempaa kuin sormikosketus, mutta yleiskäytössä sormi on tarpeeksi tarkka. Lisäksi useimmissa Android-laitteissa on gyroskooppi, jonka avulla laite tietää oman asentonsa.

Koska kosketukseen pohjautuvassa käyttöliittymässä ei ole osoitinta, painallukset eivät ole yhtä tarkkoja kuin osoittimella. Tuplapainalluksia ei käytetä, ja Windowsista tuttu hiiren oikea klikkaus on myös poissa. Näppäimistö lähes kaikissa Android-puhelimissa on nykyisin virtuaalinen, ja niiden näytöt tukevat useita samanaikaisia kosketuksia. Kosketusnäytöllä erilaiset eleet, kuten pyyhkäisy ja nipistys toimivat hyvin helpolla, luonnollisella tavalla.

Androidissa on ollut versiosta 4.0 eteenpäin virtuaaliset painikkeet alareunan navigaatiopalkissa toiminnoille takaisin, koti ja viimeisimmät sovellukset (Back, Home ja Recents). Virtuaalisia painikkeita käytetään, jos laitteessa ei ole fyysisiä painikkeita. Navigaatiopalkkiin ilmestyy myös painike menua varten, jos käytettävä sovellus on kehitetty Android 2.3:lle tai sitä vanhemmille. Uusissa sovelluksissa menu on yleensä sovelluksen yläreunan toimintopalkissa. (Android Developers 2014.)

Kuvassa 1 on esitetty Android 4.4-version kotinäky (Home). Sekä yläreunan tilarivi että alareunan navigaatiopalkki voidaan asettaa läpinäkyväksi tai piilottaa kokonaan sopivissa tilanteissa. Ilmoituksille on varattu paikka tilarivin vasemmassa reunassa. Tilarivin voi vetää alas ja nähdä lisätietoja katsomattomista ilmoituksista.



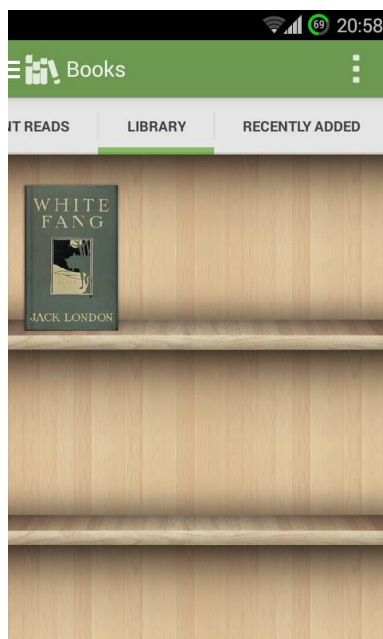
Kuva 1. Androidin kotinäky (Home) käyttöliittymä

3.4 Vallitsevat trendit

Käyttöliittymäsunnittelun trendit ovat muuttuneet käyttöjärjestelmien mukana aina ensimmäisistä graafisista käyttöliittymistä alkaen. Sovelluksen kehittäjän on helppoa ottaa mallia käyttöjärjestelmästä, jolle sovellusta kehitetään. Suunnittelijat suosivat käyttöjärjestelmän mukaisia käyttöliittymiä, koska sillä tavalla käyttäjäkokemus saadaan tuntumaan tutulta. Helppokäyttöisyys on useimmissa sovelluksissa tärkeää. Joskus UI-suunnittelijat kehittelevät uusia käyttötapoja, jotka voivat olla toimivia yksittäisissä sovelluksissa, mutta tuntuvat irrallisilta. Tällaisilla ratkaisuilla haetaan elämyksellistä käyttökokemusta.

Skeuomorfismi, eli käyttöliittymäelementtien muotoilu fyysisten esineiden näköisiksi, oli Applen käyttöliittymissä etenkin mobiilipuolella suosiossa vielä kaksi vuotta sitten. Tämänkin trendin nosti pinnalle Apple itse vuonna 2007 iPhonensa myötä. Suuntauksen tarkoituksena oli helppokäyttöisyys kokemattomille älypuhelin käyttäjille. Skeuomorfismi on toimiva tapa tehdä käyttöliittymästä helpompi, mutta se kuitenkin jakaa vahvasti mielipiteitä. Toiset ovat sitä mieltä, että se tuo turhaa kiiltoa ja yksityiskohtia käyttöliittymään, joka taas tekee kokonaisuudesta sekavan. Google ja Microsoft alkoivat suuntautua käyttöliittymissään yhä enemmän minimalismiin, jonka suosion kasvettua myös Apple jätti skeuomorfismin taakseen ja alkoi keskittyä puhtaampiin käyttöliittymiin iOS 7:n myötä. (Downer 2012.)

Kuva 2 on ruutukaappaus Aldiko Book Readerin Android-versiosta, jossa on skeuomorfinen näkymä kirjahyllystä. Tämänkin sovelluksen käyttöliittymä on muutoin hyvin puhdas ja minimalistinen, mutta tämä fyysistä kirjahyllyä muistuttava näkymä on sovelluksessa edelleen. Androidilta on vaikea löytää nykyisin täysin skeuomorfista sovellusta, iOS:lla niitä edelleen on paljon.



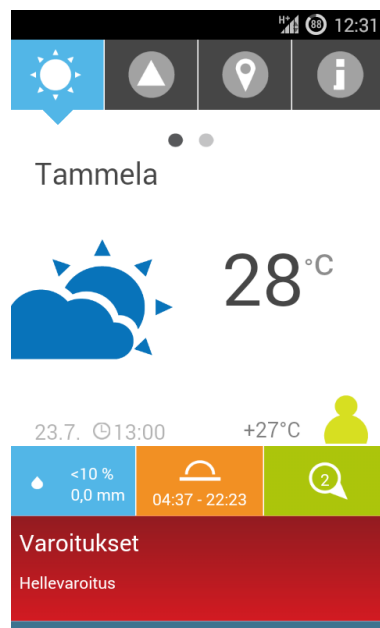
Kuva 2. Aldiko Book Readerin Library-näkymä

3.4.1 Minimalistisuus

Käyttöliittymien minimalismi alkoi yleistyä samaan aikaan kuin älypuhelimet yleistyivät, mutta todelliseen suosioon se nousi vasta Googlen Android 4.0:n, Applen iOS 7:n ja Microsoftin Windows 8:n myötä. Käyttöliittymäkomponenttien syvyysvaikutelmasta ja liukuväreistä luovuttiin. Siirryttiin yksiulotteisiin, monesti yksi- tai kaksivärisiin ikoneihin. (Salimi 2014.)

Windowsin uusi ilme on tästä trendistä pisimmälle viety esimerkki. Painikkeet ovat samankokoisia laatikoita, joissa on yksivärinen valkoinen ikoni. Tämänlainen käyttöliittymä on kuitenkin pääasiassa kosketusnäyttöjä varten, eikä koko käyttöjärjestelmää ole yksinkertaistettu näin radikaalisti. Muutkin modernit käyttöjärjestelmät ovat tasaisia, ylimääräisistä tilanjakoviivoista ja -laatikoista on luovuttu ja värejä käytetään todella harkitusti. Samaa fonttia käytetään aina kun mahdollista ja tekstin määrä yritetään pitää minimissä.

Minimalistisuus tuo käyttöliittymiin kontrastia, joka auttaa käyttäjää huomaamaan käyttöliittymän keskeiset osat. Se on tehnyt graafikoiden työstä helppoa. Se myös antaa tekstille enemmän tilaa, ja mahdollistaa suurempien fonttien käyttämisen. Kuvassa 3 esitetty Ilmatieteen laitoksen sääsovellus on hyvä esimerkki minimalistisesta käyttöliittymästä. Värejä siinä käytetään paljon, mutta ne ovat perusteltuja, ja niillä voi kätevästi rajata käyttöliittymän elementtejä.



Kuva 3. Ilmatieteen laitoksen sääsovellus

3.4.2 Monitasoisuus ja sumennus

Minimalismin yleistymisen ei tarkoita sitä, että käyttöliittymistä tulee täysin kaksiulotteisia. Käyttöliittymissä on alettu käyttää useampia tasoja, jotka teknisesti muistuttavat ponnahdusikkunoita. Ne tuovat

kolmiulotteista vaikutusta näkymään ja usein helpottavat ymmärtämään eri elementtien ja interaktioiden vuorovaikutuksia. Oikein käytettyinä ne myös nopeuttavat sovelluksen käyttöä, kun sovelluksen ei tarvitse siirtyä täysin uuteen näkymään valintatilanteissa. (Golden Gekko 2014.)

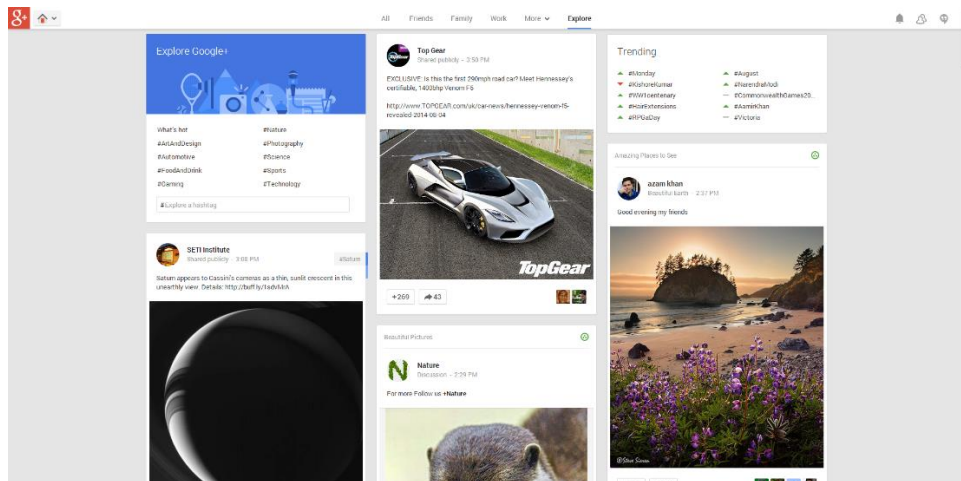
Monet mobiilisovellukset käyttävät sivuvalikkoa, jonka kautta navigoidaan sovelluksen eri osiin. Google on vähitellen siirtynyt käyttämään vedettävää navigaatiovalikkoa lähes kaikissa Android-sovelluksissaan. Tämä valikko avautuu ruudun vasemmasta yläkulmasta, painamishetkellä auki olevan näkymän päälle, mutta ei peitä sitä kokonaan. Applen iOS 7:ssa sivuvalikko avautuessaan siirtää alhaalla olevaa näkymää oikealle, ja jättää osan siitä näkyviin. Periaate on kuitenkin molemmissa tyyleissä sama.

Sumennusta käytetään usein monitasoisissa käyttöliittymissä havainnollistamaan tasojen välistä hierarkiaa. Sumennuksen avulla sovelluksista voi tehdä huomaamattomalla tavalla monitasoisia, esimerkiksi avattaessa tarkempaa informaatiota uuteen näkymään, voi vanha näkymä olla taustalla sumennettuna. Vaikka näkymää vaihdettaessa kaikki interaktiivisuus siirtyisikin uuteen näkymään, käyttöliittymä on visuaalisesti monitasoinen.

3.4.3 Kortit ja jatkuvat sivut

Kortti on kaikille tuttu käsite. Se on palanen, jolle on koottu tietynlainen kokonaisuus informaatiota, fyysisenä esimerkkinä käyntikortti tai postikortti. Digitaalisessa muodossa kortit toimivat samalla tavalla. Niitä voi järjestellä, kääntää, levittää laajempaa tarkastelua varten tai pinota tilan säästämiseksi. Seuraavissa kappaleissa puhutaan korteista digitaalisen sisällön erottelijana.

Korttien käyttö digitaalisena sisällön erotteluun on melko uusi suunnittelumalli, jota on alettu käyttää internetsivustoilla vasta viime vuosina. Näkyvimmin kortteja käytetään sosiaalisissa medioissa. Pinterest on yksi trendin edelläkävijöistä, ja sen etusivun käyttöliittymä perustuu täysin kortteihin. Pinterestin etusivu on täynnä pieniä kortteja, joissa on kaikissa oma sisältö, jonka saa laajemmin näkyviin painamalla korttia. Myös Facebookin käyttöliittymä perustuu nykyisin kortteihin, jokainen päivitys Facebookissa näkyy omana korttinaan. Kuvassa 4 esitetty Google+’n näkymä koostuu korteista, ja Google käyttää korttinäkymää joissain muissakin sovelluksissaan. (Adams 2013.)



Kuva 4. Google+:n tyypillinen näkymä

Korttinäkymä on kehitetty erityisesti mobiililaitteiden hyödyksi. Sisällön pilkkominen pieniin, itsenäisiin osiin on toimivaa erikokoisilla näytöillä, ja sitä on helppo skaalata. Älypuhelimien näytöllä kortit voidaan pinota yhteen, alas tai ylös selattavaan sarakkeeseen. Tabletilla voidaan käyttää kahta, kolmea tai neljää saraketta riippuen laitteen asennosta ja koosta, tai vaihtoehtoisesti näyttää suurempia kortteja joissa on enemmän informaatiota. Kortit ovat visuaalisesti selkeä ja yhdenmukainen tapa jakaa sisältöä, mutta varsinkin suuremmilla näytöillä informaation määrä saattaa helposti kasvaa liian suureksi. Suunnittelijan pitää huomioida, että joskus informaatiota on liikaa.

Jatkuvat sivut ovat kasvattaneet suosiotaan korttien vanavedessä. Vielä kymmenen vuotta sitten liian pitkän internetsivun tekeminen oli paha virhe, sillä sen lataamiseen meni liian kauan. Nykyään loputtomiin jatkuvat sivut ovat yleinen näky. Trendin suosion kasvamisen suurimmat syyt ovat verkkojen nopeuksien kasvu ja internetin ohjelmointikielten kehitys.

On vain ajan kysymys, että Google poistaa hakusivunsa alareunasta perinteisen pidennetyn Goooooooooogle-logon sivunumeroineen, ja tekee hakutuloksista loputtomasti rullattavan. Facebook on jo pitkään toiminut niin, että se lataa sivullisen uusia sisältökortteja näkyviin aina, kun käyttäjä pääsee edellisen sivun loppuun.

Jatkuvat sivut sisältävät toistaiseksi ongelmia, joista sisällöntuottajalle pahin on niiden näkymättömyys hakukoneissa. Tämän voi kuitenkin ratkaista pitämällä sivut edelleen teknisesti numeroituina ja erillisinä, vaikka sivun vaihto onnistuisi pelkästään rullaa käyttämällä. (Schofield 2014.) Myös sivujen selaaminen taaksepäin voi olla työlästä ilman sivunumeroita, joten sivunumeroilla navigointia ei kannata jättää täysin pois käyttöliittymästä.

Mobiilipuolella sivujen sulava jatkuvuus on sovelluksissa täysin normaali käytäntö kosketusohjauksen takia. Sivujen vaihtaminen napeista pienellä näytöllä ei ole mukavaa. Loputtomasti vieritettävät sivut yleistyvät

selainsovelluksissakin varmasti lisää jatkossa, mikäli ongelmat hakukoneiden kanssa selvitetään.

4 MENETELMÄT JA TYÖKALUT

Tässä luvussa käsitellään työssä käytettävät menettelytavat, menetelmät ja työkalut. Luvussa on myös tutkittu vaihtoehtoisia menetelmiä, joita ei työssä varsinaisesti hyödynnetty, mutta ovat alalla suosittuja. Suosituimmat työkalut ovat tietokoneohjelmistoja joko Macille tai Windowsille, joilla voidaan suunnitella ja jakaa käyttöliittymien luonnoksia.

Tableteille on myös olemassa joitakin työkaluja, mutta tarjonta on heikkoa. Yhtäkään hyvin tehtyä, harmaasta massasta erottuvaa käyttöliittymien luonnostelutyökalua ei Androidille löytynyt. Potentiaalia tableteissa varmasti olisi, sillä iPadille löytyy ainakin muutama kohtalainen työkalu, joista mainitsemisen arvoisia ovat Mocking Pad ja Blueprint.

4.1 Photoshop

Adoben Photoshop on kuvankäsittelyohjelma bittikarttagrafiikalle, jota moni osaa käyttää vähintään alkeellisella tasolla. Siksi se on monen aloittelevan käyttöliittymäsuunnittelijan valinta ensimmäiseksi digitaaliseksi työkaluksi. Photoshop on monipuolinen, sillä onnistuu monenlainen piirtäminen, mutta kokemattomalle käyttäjälle se ei ole erityisen nopea tapa tehdä UI-luonnoksia.

Vaikka Photoshop ei olisikaan käyttöliittymäsuunnittelijan tehokkain työkalu, se on varmasti monipuolisin, ja suunnittelijan on pakko hallita sen käyttäminen. Nopea työskentely Photoshopilla vaatii pikanäppäimien osaamista ja työkalujen käytön täydellistä hallintaa. Erittäin kokenut Photoshopin käyttäjä pystyy tekemään tarkkojakin prototyyppejä nopeasti, kun käyttää sovellukseen sopivia grafiikkapohjia. (McKinney 2013.)

Photoshopia käytettiin oldtimerTimer-projektissa käyttöliittymän ikonien tekemiseen ja sillä tehtiin myös sovelluksen aloitusruudusta prototyyppi iPhonelle. Prototyyppi tehtiin projektin loppuvaiheessa, kun käyttöliittymä alkoi olla lopullisessa muodossaan. Aloitusruudun pohjalle käytettiin ilmaista grafiikkapohjaa, jossa oli oikean kokoinen näyttö ja käyttöjärjestelmän tilarivi valmiina. Ruudun luonnostelussa kului yli tunti, vaikka ikonit olivat tehty valmiiksi ennen sitä.

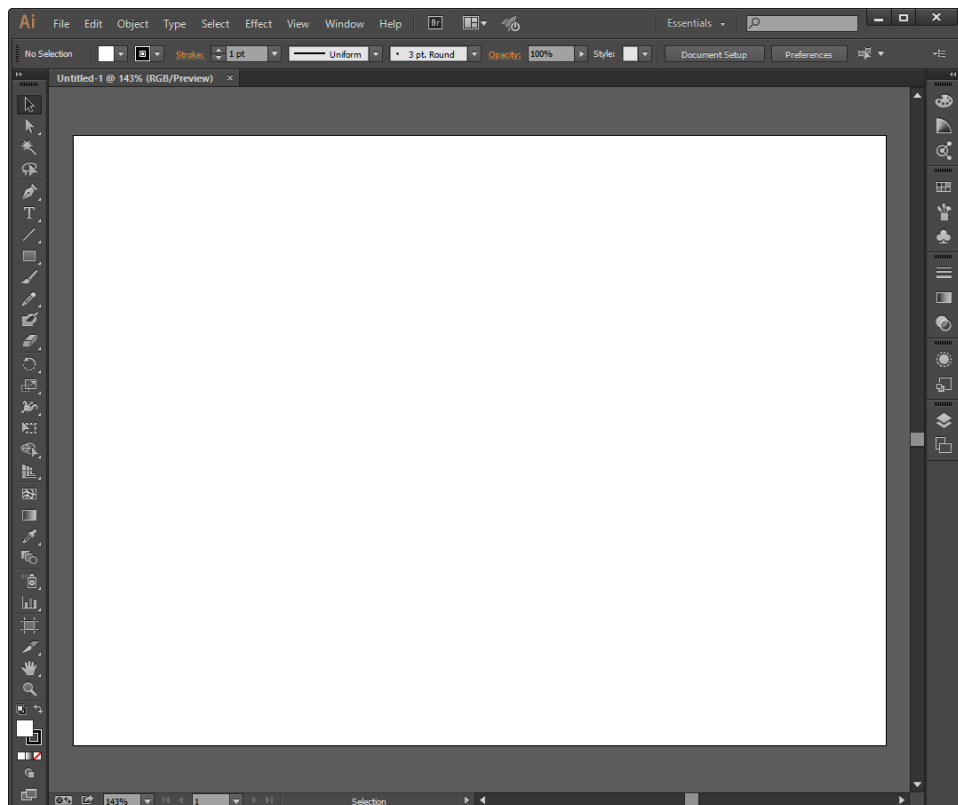
4.2 Illustrator

Illustrator on Adoben grafiikkaeditori vektorigrafiikalle. Sen käyttöliittymä on hyvin paljon Photoshopin kaltainen. Vektorigrafiikka kuitenkin tekee sen käytöstä erilaista. Jokainen piirretty muoto on oma objektinsa, ja niitä voi muokata ilman layerien käyttöä. Vektorigrafiikka

mahdollistaa kuvien skaalaamisen ilman laadun heikentymistä. Illustratorilla luodut grafiikat ovat helposti käytettävissä uudestaan missä tahansa koossa, mikä tekee luonnoksista johdonmukaisia. Illustratoriin voi myös tuoda bittikarttagrafiikkaa, jonka sovellus osaa muuntaa vektorigrafiikaksi.

Vektorigrafiikasta on vaikeaa tehdä yksityiskohtaista, joten Illustrator on parhaimmillaan logojen ja rautalankaluonnosten tekemisessä. Yksityiskohtaisissa käyttöliittymäluonnoksissa Photoshop on monesti nopeampi vaihtoehto, mutta myös Illustrator taipuu osaavissa käsissä moneen tilanteeseen. (Cianfrone 2012.)

Itselleni Illustrator oli kirjoittajalle täysin tuntematon työkalu, josta otettiin tutkimuksessa selvää. Sen käyttöä ei kannattanut alkaa opetella tämän työn aikana edes testimielessä. Opetusvideoita katsoessa kuitenkin huomasi, että monet sen työkalut toimivat samaan tapaan kuin Photoshopissa, joten Illustratorin opettelu on Photoshop-osaajille varmasti verrattain helppoa. Kuvassa 5 on esitetty ruutukaappaus Illustratorin käyttöliittymästä.



Kuva 5. Adobe Illustratorin käyttöliittymä

4.3 Sketch

Sketch on vain Macille saatavilla oleva editori vektorigrafiikalle, joskin sillä voi muokata myös bittikarttagrafiikkaa. Alkuperäisen Sketchin kehitys aloitettiin vuonna 2008 ja uusin versio siitä on Sketch 3. Se on kevyempi ja edullisempi kuin Illustrator, ja sen käytön opettelu on helpompaa. Sketch on tarkoitettu erityisesti GUI-suunnitteluun. Siitä on

jätetty pois kaikki valokuvaamiseen liittyvät muokkausominaisuudet, mikä tekee siitä huomattavasti suoraviivaisemman. (Heaton 2013.)

Sketchin toiminnot ovat hyvin samankaltaisia kuin Illustratorissa eikä se sisällä mullistavia uusia ominaisuuksia, mutta kaikki tarvittavat työkalut ovat nopeasti käytettävissä. Käyttöliittymäkomponenttien asettelusta, muokkauksesta ja uudelleen käytöstä on tehty nopeaa ja helppoa.

Sketch on vähitellen kerännyt suosiota ammattimaisessa käyttöliittymien suunnittelussa, mutta vanhat tutut Photoshop ja Illustrator edelleen hallitsevat alaa. Yritysmaailmassa Adobeen sovellusten osaaminen on yleinen vaatimus, mutta pienemmissä piireissä Sketch saattaa jo nyt olla tehokkaampi ja suositumpi työkalu. Mac on erittäin suosittu alan ammattilaisten pääasiallisena työalustana, ja Sketchin tekijätiimi ei edes harkitse sovelluksen tuomista Windowsille tai Linuxille (Bohemian Coding 2014).

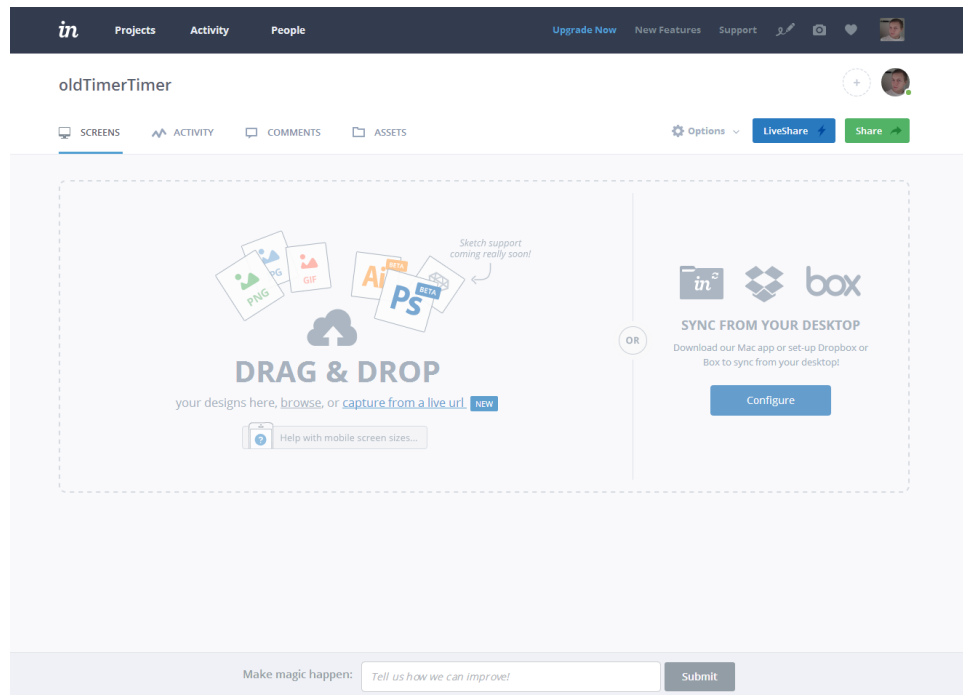
Työn kirjoittaja ei Mac-konetta omista, joten myöskään Sketchiä ei käytetty oldtimerTimerin kehityksessä. Sitä ei päästy edes kokeilemaan, mutta koska Sketch on todella varteenotettava ja suosittu työkalu, sen käytöstä otettiin selvää tätä tutkimusta varten.

4.4 InVision

InVision on selainpohjainen projektinhallintatyökalu, joka on tarkoitettu helpottamaan UI- ja UX-suunnittelijoiden työtä. InVision ei sovellu kokonaisvaltaiseen projektinhallintaan, vaan keskittyy käyttöliittymäprototyyppien organisointiin, jakamiseen ja testaamiseen. InVisionilla ei ole mahdollista tehdä luonnoksia, vaan se vaatii erillisen piirto-ohjelman, kuten Photoshop, Illustrator tai Sketch. InVision on kuitenkin yksi suosituimmista käyttöliittymäsuunnittelun työkaluista varsinkin suuremmissa yrityksissä.

InVisioniin voi ladata luonnoksia kuvamuodossa, Photoshopin psd-tiedostomuodossa tai Illustratorin ai-muodossa. InVisionissa voi tehdä luonnoksistaan interaktiivisia. Kuviin voi lisätä hotspotteja, eli interaktiivisia alueita, joita voi linkittää muihin ruutuihin. Ruutujen eri osia voi vapaasti kommentoida suunnitteluryhmän kesken. InVisionissa voi pitää konferenssipuheluita ja yhteisiä luonnostelukokouksia joissa voi piirtää reaaliajassa ruudulle. Työkaluja yhteistyöhön on enemmänkin, ja palvelua kehitetään jatkuvasti.

Työn sovelluksen, oldtimerTimerin kehityksessä InVisionia ei käytetty, sillä käyttöliittymää suunnittelemassa ei ollut muita. InVision työkaluna oli myös täysin tuntematon projektin alussa, eikä muuta projektiryhmää voitu vaatia käyttämään sitä kesken projektin. Järjestäytyneessä projektissa InVision voi kuitenkin olla suuri apu UX-suunnitteluun liittyvässä kommunikoinnissa. Potentiaalia siinä on, jos projektiryhmä käyttää sitä aktiivisesti. Kuvassa 6 on esitetty ruutukaappaus InVisionin käyttöliittymästä. Kuvassa näkyy tyhjä projekti, joka tehtiin esimerkiksi.



Kuva 6. Projektinäkö InVisionissa

4.5 Fluid UI

Fluid UI on selainpohjainen työkalu mobiilikäyttöliittymien rautalankamallinnukseen ja prototyyppien tekemiseen. Sillä on mahdollista suunnitella myös työpöytäsovelluksia, mutta painopiste on selkeästi mobiilisovelluksissa. Prototyyppiä voi myös jakaa ja testata, sillä niistä on mahdollista tehdä interaktiivisia linkitysten, eleiden ja vaihtoanimaatioiden avulla. Testausta varten on olemassa myös Androidille ja iOS:lle ladattava sovellus Fluid Player, jolla valmiiden prototyyppien testaus onnistuu laitteen näytöltä.

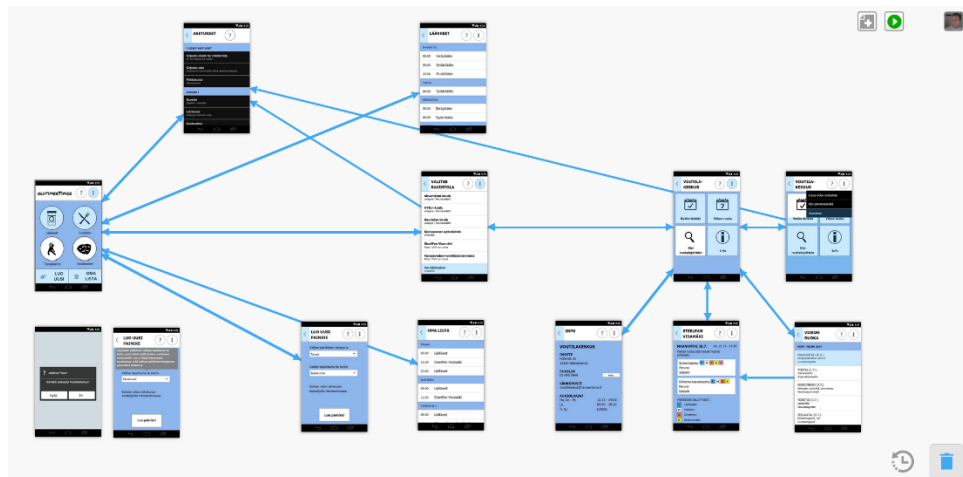
Fluid UI sisältää rakennuspalikoita eli tyypillisimpiä käyttöliittymäelementtejä Androidista, iOS:sta ja Windows Phonesta, joita raahataan valikosta suunnitteluruudulle. Tableteille löytyy myös omat elementit. Elementtejä voi myös itse piirtää ja ladata Fluid UI:hin, jolloin prototyyppistä saadaan enemmän lopullista sovellusta vastaava. Käyttöliittymäsuunnittelu Fluid UI:lla on erittäin nopeaa valmiiden elementtien ja elementtien raahaamiseen perustuvan systeemin ansiosta.

Fluid UI on melko uusi, julkaistu heinäkuussa vuonna 2012, ja siitä on nopeasti tullut suosittu työkalu suurtenkin yritysten käyttöliittymäsuunnittelussa. Sen valtteja ovat käyttämisen helppous, nopeus ja interaktiiviset prototypit, mutta tarkempien suunnitelmien ja ruutujen sisäisten muutosten havainnollistaminen sillä on vaikeampaa. Kommentointi on epätarkempaa kuin InVisionissa, ja yhteisominaisuudet ovat muutenkin paljon suppeammat.

Fluid UI valittiin pääasialliseksi digitaaliseksi työkaluksi luonnosteluun, kun Digital Service Development -pintojaksolla huomattiin, kuinka

nopeaa ja helppoa sen käyttö on. Fluid UI:ta käytettiin oldtimerTimerin mobiilisovelluksen käyttöliittymän prototyyppien tekemiseen enimmäkseen projektin loppuvaiheessa elokuussa. Siitä otettiin kokeiluun maksullinen Lite-tili, jolla oli mahdollista ylittää kymmenen ruudun rajoitus, joka ilmaisella tilillä oli. Opinnäytetyön viimeiset testaukset tehtiin valmiilla, interaktiivisella prototyypillä Fluid Playerin avulla.

Kuvassa 7 esitetään ruutukaappaus Fluid UI:n käyttöliittymästä projektinäkömässä. Kuvassa näkyy oldtimerTimer-projekti työn lopussa. Siniset viivat kuvassa ovat linkkejä ruutujen välillä. Tässä näkömässä yksittäistä ruutua klikkaamalla siirrytään ruudun muokkaustilaan.

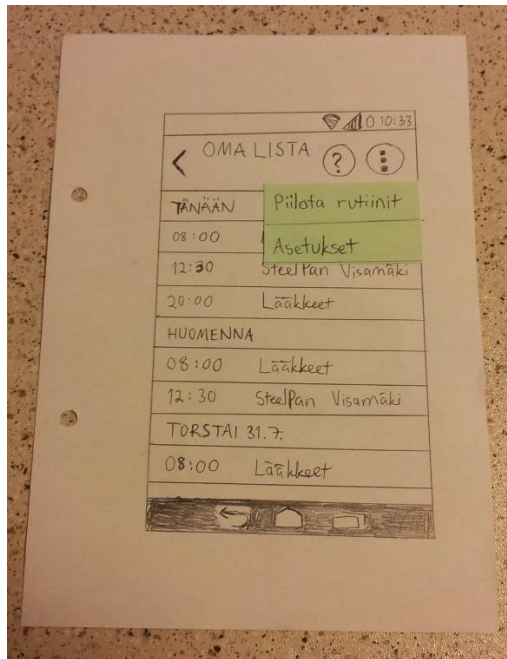


Kuva 7. Projektinäkömä FluidUI:ssa

4.6 Paperiluonnokset

Käyttöliittymien luonnostelu paperille on yksinkertaista ja halpaa. Se on myös monille luonnollisin tapa piirtää. Digitaalisten sovellusten valmiiden, siirrettävien elementtien myötä paperille piirtäminen ei nykyisin ole tarkoin tai edes nopein tapa luonnostella. Interaktioita on mahdollista toteuttaa käyttämällä esimerkiksi erivärisiä muistilappuja ja mielikuvitusta. Ponnahdusikkunoiden toteuttaminen onnistuu paperilla jopa paremmin kuin monilla luonnosteluohjelmilla. (Medero 2007.)

Kuva 8 on valokuva oldtimerTimeria varten tehdystä paperiluonnoksesta. Luonnoksessa esitetään sovelluksen näkömä omasta muistilistasta, ja vihreällä muistilapun palalla on esitetty kolmea pistettä esittävästä painikkeesta aukeava valikko. Saman esittäminen digitaalisilla ohjelmilla vaatii näkömän kopioimisen, tai valikkoelementin piilotustoiminnon.



Kuva 8. oldtimerTimerin ruudusta tehty paperiluonnos

Paperiluonnosten testaus on erilaista kuin digitaalisten luonnosten. Niiden testaaminen ei peruskäyttäjälle vaikuta niin ammattimaiselta, mikä tekee testauskokemuksesta vapautuneemman. Toisaalta, nykyisin pelkkien paperiluonnosten esitleminen asiakkaalle saattaa vaikuttaa amatöörimaiselta. Asiakasesittelyihin tarkemmat, digitaalisesti tehdyt luonnokset ovat yleensä parempi vaihtoehto. (Medero 2007.)

Työssä tehtiin paperille luonnoksia kaikista sovelluksen mahdollisista ruuduista, koska se vaikutti aloittelevalla suunnittelijalla sopivimmalta tavalta. Sitä ennen oli FluidUI:lla jo tehty prototyyppejä, mutta FluidUI:n ilmaisversiossa oli kymmenen yhtäaikaisen ruudun rajoitus. Paperille oli helppoa luonnostella muutoksia ja kauempaakin haettuja ratkaisuja ilman rajoitteita.

Testatessa paperiluonnoksia huomattiin, että niitä on hankalampaa testata järjestelmällisesti. Interaktiivisuuden toteuttaminen on kömpelöä, ja ilman värejä käyttöliittymän hahmottaminen on vaikeampaa. Tärkeimmät asiat, eli puutteet ja tarvittavat muutokset pystytään kyllä huomaamaan. Myös luonnosten jakaminen etänä vaatii ylimääräisen työvaiheen: skannaamisen. Skannaukseen voitiin kuitenkin käyttää CamScanneria, joka oli kätevä apu.

4.7 CamScanner

CamScanner on kamerapohjainen skannaussovellus, joka toimii Androidilla, iPhoneella, iPadilla ja Windows Phonella. Websivustokin on olemassa, josta pääsee CamScannerin Cloud Synciin eli pilvitalennuksiin ja sitä kautta voidaan ladata skannattuja dokumentteja tietokoneelle. Sovelluksesta on olemassa ilmainen ja maksullinen versio, joiden näkyvin ero skannauksessa on ilmaisen version pieni vesileima skannattujen

dokumenttien alareunassa. Ilmaisella versiolla yksityinen käyttäjä pärjää hyvin, siinä on kaikki tärkeät ominaisuudet. Sovelluksella skannaus onnistuu nopeasti, ja sillä voi korvata perinteisen skannerilaitteen, kunhan paperit eivät ole pahasti rypyssä. Valaistuksen pitää myös olla kohtuullinen, ja skannausalustan kohtuullisen tasainen.

Skannaus CamScannerilla toimii siten, että ensin dokumentista otetaan kuva kännykän tai tabletin kameralla, jonka jälkeen sovellus poistaa kuvasta ylimääräiset varjot ja värit. Kuva myös rajataan automaattisesti, mutta rajausta voi muuttaa manuaalisesti. Jopa väriskannaus onnistuu, mutta jonkin verran varjoja saattaa dokumenttiin jäädä. Skannauksen laatu on varsin hyvää, se suurin piirtein vastaa erillistä skannerilaitetta.

CamScanneria käytettiin projektissa paperiluonnosten skannaamiseen, jonka jälkeen skannatut luonnokset jaettiin muulle projektin kehitystiimille. Sovelluksessa riittää käyttöpotentiaalia laajempaankin käyttöön kehitystiimin yhteistyössä, sillä dokumenttien jakaminen CamScanner-tilien välillä on tehty helpoksi. Skannaaminen ei ole sovelluksen ainoa tarkoitus, sillä voi myös järjestellä ja jakaa skannattuja dokumentteja.

5 SOVELLUKSEN SUUNNITTELU

Tässä luvussa käsitellään oldtimerTimer-mobiilisovelluksen käyttöliittymäsuunnittelua projektin alusta eli kesäkuun 2014 alusta elokuun 2014 loppuun. Elokuun loppuun mennessä sovelluksesta oli tarkoitus olla käyttökelpoinen prototyyppi valmiina. Sovelluksen kehitys ei siis ehtinyt vielä pitkälle, joten vain perusominaisuudet ja toiminnot sekä niiden alustava käyttöliittymä saatiin valmiiksi. Sovelluksen kehitys aloitettiin tabletilla, mutta tämän työn käyttöliittymäsuunnitelma on tehty kännykkäversiota ajatellen.

Tässä työssä suunniteltu käyttöliittymä toiminee jatkokehityksen suunnannäyttäjänä varsinkin siinä vaiheessa, kun sovellusta käännetään pienemmille näytöille. Se on suunniteltu kestävään jatkokehitystä, joten uusien ominaisuuksien lisäämisen ei pitäisi olla ongelmallista.

5.1 Lähtötilanne

Projektin sovelluksen, oldtimerTimerin suunnittelu aloitettiin 23.4.2014 alkaneella opintojaksolla Digital Service Development. Opintojakson toisella lähikerralla sovelluksen suunnittelun eri tehtäväalueet olivat jo jaettu ryhmän jäsenille. Mobiilisovelluksen käyttöliittymäsuunnittelun vastuu oli jaettu Ilpo Kiurulle. Hänen työparikseen kuitenkin pääsi, koska Kiurun oli tarkoitus olla mukana suunnittelussa vain Digital Service Developmentin ajan. Tässä vaiheessa projekti keskittyi vain ruokalatoimintaan, ja suunnittelussa edettiin sen mukaisesti.

Opintojakson aikana pidettiin tapaaminen Linnan Ateria Oy:n Mari Tähtisen kanssa, tehtiin muutama käyttäjähaastattelu, ja kehitettiin mobiilisovelluksesta ensimmäinen käyttöliittymäprototyyppi Fluid UI:lla. Prototyyppiin ehdittiin tekemään muutoksiakin jo opintojakson aikana, lähinnä muiden ryhmän jäsenten ehdotuksien pohjalta. Opintojakso päättyi 3.6.2014, jolloin työn tulokset esiteltiin.

Projekti oli alusta lähtien hieman sekaisin, sillä Digital Service Development keskittyi vain sovelluksen ruokailuosioon. Sovelluksen ja websivuston varsinainen kehitystiimi joka oli Hämeen ammattikorkeakoululla kesätöissä, tähtäsi kuitenkin suurempaan kokonaisuuteen. Digital Service Development keskittyi kännykkäsovellukseen, töissä oleva kehitystiimi tablettisovellukseen. Kesäprojektin alkaessa opintojaksosta jäljelle jääneet henkilöt muodostivat yhdessä uuden projektiryhmän, ja tavoitteita yhtenäistettiin. OldtimerTimerista oli tulossa laajempi kokonaisuus, ei pelkästään ruokalätietosovellus.

Projektiryhmä kokoontui Hämeen ammattikorkeakoululla joka arkipäivä keskiviikkoja lukuun ottamatta. Projektihallinnan viitekehyksenä toimi Scrum, joka mahdollisti iteratiivisen ja joustavan kehitystavan. Projektissa käytettiin kahden viikon sprinttejä eli kehitysjaksoja, joita ennen sprint-palaverissa sovittiin mitä kukin ryhmän jäsen tekee sprintin aikana.

Mobiilisovelluksella ei ollut varsinaista kokonaisvaltaista käyttöliittymäsuunnitelmaa, ainoastaan Digital Service Developmentin aikana suunniteltu ruokailuosio oli visualisoitu. Sovellukseen tuli varsinkin alkuvaiheessa ennen heinäkuuta suuria rakennemuutoksia jotka rajoittivat GUI-suunnittelua. Sovelluksesta muotoutui tapahtumien ja rutiinien kokonaisuus, jonka nitoi yhteen aloitusruutu muokattavine painikkeineen sekä oma muokattava muistilista.

Tapahtuma on aina yksittäinen, se voi olla esimerkiksi ruokailu, musiikkitapahtuma tai lavatanssit. Rutiini on päivittäinen tai viikoittainen niin, että joka viikko on samanlainen. Rutiinia voi kuitenkin muokata jos se muuttuu. Rutiinien ja tapahtumien tiedot haetaan käyttöliittymään XML-tiedostoista. Rajoittavaa ja haastavaa tässä oli se, että rutiinit ja tapahtumat olivat monimuotoisia, mutta niiden tietoa sisältävät näkymät piti suunnitella yhdenmuotoisiksi. Viikon ruokalistan piti pystyä muuntumaan myös viikon esiintyjälistaksi, käyttöliittymän pysyessä muuttumattomana.

Tämän opinnäytetyön kirjoittaja oli mukana projektissa lähinnä opinnäytetyönsä takia, kuten muutama muukin projektiryhmän jäsen. Keskittyminen oli mobiilisovelluksen kännykkäversion suunnittelussa, mutta tablettiversion kehitykseen annettiin myös prototyyppejä ja ideoita. Vastuu sovelluksen käyttökokemuksesta kokonaisuutena oli myös suurelta osin kirjoittajalla.

5.2 Prototyypin luonti

Projektin alkuvaiheessa tehtiin selväksi, mitkä näkymät sovellukseen tarvitaan. Jo tässä vaiheessa piti ajatella jatkokehitystä eteenpäin, koska sovelluksen varsinainen kehitys oli jäämässä tauolle heinäkuun jälkeen. Tärkeimmät näkymät olivat aloitusruutu, uuden painikkeen luontiruutu, tapahtumapaikan valintaruutu, rutiinien viikkonäkymä, oma muistilista sekä tapahtumapaikan oma valikko, info, päivän tapahtuma ja viikon tapahtumalista. Näiden lisäksi sovellukseen tarvittiin asetukset, apukuplat, ilmoituksia ja muistutusikkunoita, mutta niiden tiedettiin rakentuvan pääosin Androidin omista rakennuspalikoista.

Suunnitteluvaihetta sekoitti se, että sovelluksen ohjelmoijat, Mikko Suni ja Mikko Jussilainen kehittivät sovellusta ensisijaisesti tabletilaitteille, ja oman suunnittelutyöni piti keskittyä kännykkäversioon. Oli kuitenkin opinnäytetyön kannalta liian myöhäistä alkaa keskittymään tablettiversion käyttöliittymän suunnitteluun, joten kännykkäversiossa pitäydettiin. Tablettiversiona jotkin näkymät ovat täysin erilaisia kännykkäversioon verrattuna, mutta käyttöliittymästä tehtiin myös joitakin luonnoksia tabletteja ajatellen.

Pitkin kesä- ja heinäkuuta tehtiin paperiluonnoksia. Niiden tekeminen alkoi tuntua pian puuduttavalta, kun samaa kuvaa piti piirtää uudestaan useita kertoja. Osa luonnoksista päätyi pian roskakoriin kun huomattiin, ettei niissä olleita ideoita voitu tai haluttu käyttää. Heinäkuussa alettiin keräämään luonnoksia kansioon, johon niitä ei ehtinyt enää paljon kertyä

ennen projektin loppuvaiheita. Jatkossa jos tulee tehtyä paperiluonnoksia, kansio pitää ottaa heti prosessiin mukaan, jottei luonnoksia mene hukkaan.

Työn loppupuolella elokuussa Fluid UI otettiin uudestaan käyttöön. Maksullisella Lite-tilillä oli mahdollista saada paljonkin aikaiseksi. Käyttöliittymästä tehtiin viimeiset prototyypit tämän opinnäytetyön osalta. Fluid UI:lla luonnostelu oli sujuvaa, koska ruutuja ei koskaan tarvinnut tehdä uudestaan alusta asti. Sulavuus oli kloonaustyökalun sekä valmiiden elementtien ansiota. Kaikkia elementtejä ei saatu muokattua juuri sellaiseksi kuin olisi haluttu, mutta pääpiirteittäin Fluid UI:lla tehty prototyyppi vastaa suunnittelijan näkemystä.

Lopuksi valmiista käyttöliittymäprototyypistä ladattiin kaikki ruutukuvat, ja tehtiin käyttöliittymäspeksi. Käyttöliittymäspeksi on suunnittelun lopputulos, joka havainnollistaa sovelluksen toimintalogiikkaa. Se voi olla räätälöity asiakkaan tai kehittäjän tarpeisiin. Sillä annetaan mahdollisimman selvä kuva siitä, minkälainen sovellus tulee olemaan valmistuttuaan. (Interacta 2014.)

Käyttöliittymäspeksi tehtiin Word-dokumentiksi sovelluksen jatkokehittäjiä varten. Siinä esitellään sovelluksen jokainen ruutu prototyypistä otetulla kuvalla ja selityksellä. Dokumentin tarkoitus on olla visuaalinen muistutus kehittäjille, mihin suuntaan sovellusta kehitetään. Speksi ei ole täydellinen lopputulos tai pakollinen kehityssuunta, ja siinä käydään myös läpi joitakin vaihtoehtoja, joita kehittäjiä pitää itse pohtia. Käyttöliittymäspeksi oldtimerTimer-sovellukseen on tämän opinnäytetyön liitteessä 1.

5.3 Prototyyppi iPhonelle

Sovelluksen pääasiallinen alusta on Android, mutta iPhonelle tehtiin yksi prototyyppi projektin lopussa. Samalla testattiin Photoshopin soveltuvuutta GUI-suunnitteluun. Käyttöjärjestelmän teknisiä vaatimuksia tai käyttöliittymäsuunnittelun ohjeistuksia ei juurikaan selvitetty, joten käyttöliittymä on hyvin samankaltainen Android-version kanssa. Takaisinnäppäintä ei iPhonessa ole, mutta oldtimerTimerin käyttöliittymässä on painike sitä varten Android-versiossakin. Aloituskäytössä sitä ei tarvita.

Prototyyppi oldtimerTimerin aloituskäytöstä tehtiin Photoshopilla, ja grafiikkapohjana käytettiin ilmaista iOS 7:n sovelluspohjaa. Sovelluspohjassa on iPhone 5S:n näytön mitat ja käyttöjärjestelmän tilarivi valmiina. Kun tätä prototyyppiä tehtiin, käyttöliittymän viimeiset luonnokset olivat jo suurimmaksi osaksi valmiina FluidUI:ssa, ja mallia otettiin sieltä.

Käyttöliittymien piirtämisessä Photoshopissa käytetään paljon suorakaide- ja ellipsityökaluja, tekstityökalua, ja ikonien piirtämisessä Pen toolia. Pen toolin käyttö vaatii harjoittelua jotta sillä piirtäminen on sujuvaa, mutta sillä saa aikaiseksi tarkkoja ja helposti muunneltavia kuvioita. Layereita kertyy prototyyppijä työstäessä paljon, joten niiden organisointi on tärkeää. Aloituskäytön piirtämisessä käytettiin myös häivytystä, jolla

saatiin kolmas painikerivi häviämään osittain alareunan painikkeiden alle. Häivytyksestä käyttäjä tietää että näkymää vierittämällä painikkeita saa näkyviin lisää.

Lopuksi prototyypistä tehtiin esittelyversio. Photoshopiin avattiin Pixeden-sivustolta ladattu ilmainen 3D-mallinnos iPhone 5S:sta. Tähän malliin liitettiin valmis GUI-luonnos, ja hieno esittelykuva oli valmis. Kuvassa 9 esittelykuva, josta saa käsityksen, miltä valmis sovellus näyttää puhelimen näytöllä.



Kuva 9. Photoshopilla tehty luonnos iPhone-version aloitusnäkyvästä

5.4 Näkymien rakenne

Sovellusten eri osa-alueilla on paljon erilaisia hyviksi todettuja rakennemalleja. Google, Apple ja Microsoft kuitenkin asettavat käyttöliittymilleen suosituksia ja rajoituksiakin ominaisuuksien toteuttamiseen. Tässä luvussa tutkitaan Googlen Androidille suosittomia rakenteita, joita käyttökokemuksen suunnittelijan toivotaan noudattavan. Googlen ohjeistusten lisäksi tutkitaan suosittujen ja hyviksi havaittujen sovellusten ratkaisuja erilaisissa käyttönäkymissä. Näissä ratkaisuisissa nähdään myös käytännössä, miten tämän hetken GUI-trendit vaikuttavat suunnitteluun.

Näiden tutkimusten pohjalta suunniteltiin oldtimerTimerin käyttöliittymä, mutta sovelluksen kohderyhmä pidettiin mielessä. Luvussa käydään läpi oldtimerTimerin näkymissä käytettyjä ratkaisuja. Kaikki ruutuja ei esitellä tässä, vaan ne löytyvät oppimateriaalin liitteestä 1.

5.4.1 Aloitusnäkyvä

Sovelluksen aloitusnäkyvän tulee olla palkitseva ja toimiva kokemus sekä uusille käyttäjille että sovellusta usein käyttäville (Android Developers 2014). Navigointi sovelluksen eri osiin pitää olla sujuvaa aloitusnäkyvästä alkaen. Se on useimmissa tapauksissa käyttökokemuksen kannalta sovelluksen tärkein sivu.

Projektin sovelluksen aloitusnäkyvästä ei tehty montaa erilaista versiota. Alusta lähtien oli selvää, että aloitussivulle tulee suuria painikkeita, joita voi lisätä tai poistaa tarpeen mukaan. Aloitusruutuun piti saada tapahtumapainikkeiden lisäksi näkyviin kaksi tärkeää toimintoa, eli uuden painikkeen lisääminen ja omaan muistilistaan pääseminen. Nämä toiminnot olisi voitu sijoittaa ylös toimintopalkkiin, mutta sovelluksen kohderyhmää ajatellen näkyvämmät painikkeet ruudun alareunassa toimivat paremmin. Uuden painikkeen lisäys olisi voinut olla myös tapahtumapainikkeiden joukossa, ja tällaiseen ratkaisuun päädyttiinkin tablettiversioiden kehityksessä. Kuvassa 9 on esitetty kännykkäversion aloitusnäkyvä.

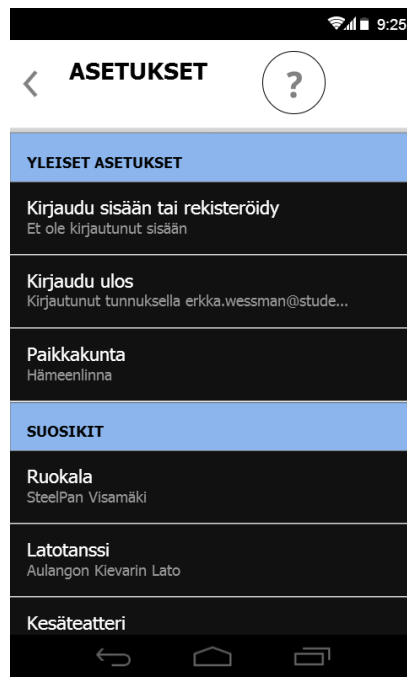
5.4.2 Asetukset

Asetuksia tarvitaan lähes jokaisessa sovelluksessa, koska käyttäjien mieltymykset ovat erilaisia. Asetukset mahdollistavat sovelluksen käyttäytymisen halutulla tavalla niin, että käyttäjältä ei tarvitse kysyä hänen mieltymystään aina, kun tietynlaisia tilanteita nousee esiin. Lisäksi asetusten kautta voidaan tallentaa käyttäjän tietoja järjestelmään, jotta niitä ei tarvitse aina itse syöttää sovellukselle.

Android Developersissa (2014) annetaan asetuksille käyttäjäkokemusta parantavia ohjeita. Asetuksien määrään on hyvä kiinnittää huomiota, ja mikäli niitä on listassa enemmän kuin seitsemän, ryhmittely on suotavaa. Asetusten liiallinen määrä on monen käyttäjän mielestä musertavaa. Kaikesta ei pitäisi tehdä asetusta, vaan pohtia voisiko asetuksesta tehdä esimerkiksi pikavalinnan käyttöliittymään. Staattista tietoa tai toimintoja ei pitäisi myöskään laittaa asetuksien taakse.

Asetuksen tyyppiä kannattaa myös harkita. Valintaruutua käytetään, jos asetus on yksinkertaisesti käytössä tai ei käytössä, kuten ”Automaattinen aikavyöhyke”. Jos asetus vaatii kuitenkin tarkempaa selitystä kuin listamuodossa on mahdollista selittää, asetuksesta voidaan tehdä uuteen ruutuun johtava kytkin. Ajan ja päivän valintaan voidaan käyttää Androidin omaa valintaruutua. Liukusäädintä käytetään niissä tapauksissa kun asetuksen valinta on jatkuvasti muuttuva, jonka ei tarvitse olla eksakti. (Android Developers 2014.)

Sovelluksen käyttöliittymän osa-alueena asetusvalikko on sellainen, että käyttöjärjestelmän suosituksissa kannattaa pitäytyä mahdollisimman paljon. Brändin värimaailmaa voi asetuksissakin pitää yllä, mutta kaikki muu kikkailu on suositeltavaa jättää pois. Käyttäjä haluaa asetusten säädöstä mahdollisimman nopean ja helpon tapahtuman. Tuttu valikkorakenne helpottaa asetusten käyttöä. oldtimerTimer-sovellukseen asetuksia ei tarvita alussa kovinkaan montaa, joten asetusnäkyvästä tulee pakostikin yksinkertainen. Jatkokehityksessä asetusnäkyvän harkittu suunnittelu on tarpeellista, ja tärkeä osa sovelluksen helppokäyttöisyyttä. Kuvassa 10 on viimeisin luonnos sovelluksen asetusvalikosta.



Kuva 10. Luonnos oldtimerTimerin asetusvalikosta

5.4.3 Apua

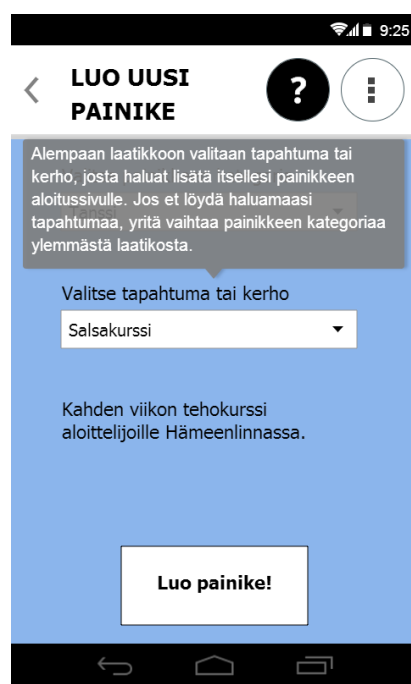
Monet sovellukset tarvitsevat tuekseen apuruutuja tai selitteitä, joita ei voida pitää käyttöliittymässä pysyvästi. Apua-toiminnon tai ylimääräisten selitteiden pitää olla käyttöliittymän lisäosa, ei pysyvä osa. Tällöin sovelluksen kokeneet käyttäjät eivät joudu katselemaan turhaa tietoa ruudulla.

Android Developersissa (2014) on ohjeistusta aputoimintojen sijoittamiseen. Apua ei suositella näyttämään ilman pyyntöä, ellei se liity tärkeään toimintoon jota käyttäjä ei muuten osaisi käyttää. Avun pitäisi löytyä johdonmukaisesta paikasta, Google suosittelee ylhäällä olevaa toimintopalkkia.

Itse avun antamisen voi toteuttaa monenlaisilla tavoilla. Toteutuksellisesti helppo tapa on avata websivu, josta löytyy ohjeistusta yleisimpiin ongelmiin, tai ruutukohtaisesti. Tämä ei kuitenkaan ole käyttäjän kannalta helpointa. Käyttäjäkokemuksen kannalta toimivampi ratkaisu on

käyttöliittymän päälle ilmestynvä ohjeistus, joka neuvoo ruutukohtaisesti käyttäjää. Erillisen apua-valikon avaaminen ei ole suositeltavaa, sillä ohjeiden antamisen pitäisi olla mahdollisimman mutkatonta. (Android Developers 2014.)

Koska oldtimerTimerin kohderyhmä on pääasiassa vanhukset, yksinkertaisinkin käyttöliittymä voi sisältää hankalasti ymmärrettäviä kohtia. Käyttöohjeistusta tulisi olla saatavilla varsinkin ensimmäisiä käyttökertoja ajatellen. Toimintovalikosta avattava näkymän läpikäynti on tässä tapauksessa käyttäjälle selkein tapa oppia sovelluksen käyttö. Kuvassa 11 on esitetty viimeisin luonnos oldtimerTimerin aputoiminnosta. Kuvassa toimintopalkin kysymysmerkki-ikonia on painettu, ja ruudulle on ilmestynyt apukuplia. Niistä näkyy vain yksi kerrallaan, ja ne ovat lievästi läpinäkyviä.



Kuva 11. Luonnos oldtimerTimerin apuohjeistuksesta

5.4.4 Ilmoitukset

Ilmoitukset ja muistutukset Androidilla ilmoittavat itsestään äänellä ja värinällä järjestelmän asetuksista riippuen, sekä joillain laitteilla merkkivalolla. Lisäksi ilmoitus näkyy ylimmällä tilarivillä, jonka alas vetämällä ilmoituksista näkee tarkempia tietoja.

Android Developersissa (2014) kerrotaan millaisia Androidin ilmoituksista pitäisi tehdä. Ohjesuunnat ovat tarkat, sillä ilmoitukset ovat osa käyttöjärjestelmää ja niistä halutaan mahdollisimman yhtenäisiä. Androidilla ilmoitukset koostuvat sovelluksen ikonista, ilmoituksen otsikosta, viestistä ja aikaleimasta. Jos ilmoitus koskee esimerkiksi viestiä toiselta henkilöltä, ikonin tilalla voi näkyä henkilön kuva ja tällöin ilmoituksessa on viestisovelluksen ikoni aikaleiman alla. Nämä ovat ilmoituksen minimivaatimukset, mutta ilmoitukseen voidaan tehdä

laajennettu näkymä. Ilmoituksen laajennetun näkymän saa avattua sormieleellä, ja siinä voi olla kuvan esikatselu, enemmän tekstiä tai pikatoimintoja.

Ilmoitusten prioriteetin määrittystä suositellaan, jotta järjestelmä osaisi näyttää tärkeimmät ilmoitukset listan alussa. Jos samasta sovelluksesta tulee useampia ilmoituksia ennen kuin käyttäjä niitä ehtii huomioida, nämä ilmoitukset tulee pinota uusimman ilmoituksen alle. Vaihtoehtoisesti uudet ilmoitukset voi lisätä jo olemassa olevaan: jos olemassa on ilmoitus uudesta sähköpostista, jonka otsikkona on sähköpostin otsikko, ja käyttäjä saa toisen uuden sähköpostin, ilmoituksen otsikkona voidaan näyttää ”2 uutta sähköpostia”.

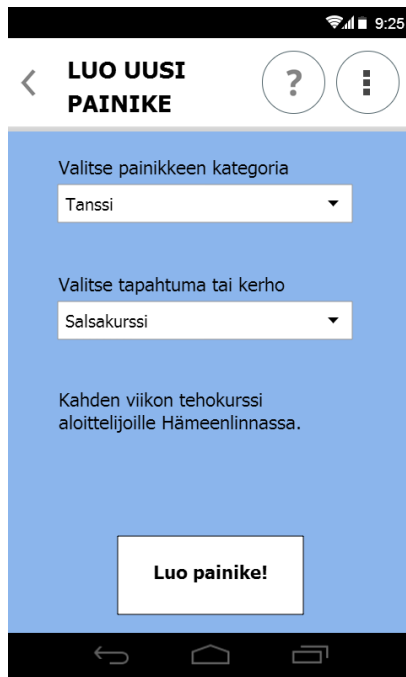
Ilmoitukset ja muistutukset ovat tärkeä osa `oldtimerTimer`-sovellusta. Sovellukseen kehitettiin muistutus ennen tapahtumaa, sekä palautekysely tapahtuman jälkeen. Tärkeää oli, että nämä eivät olisi käyttäjälle liian painostavia, vaan olisivat hänelle avuksi. Molemmat näistä toimivat Androidin ilmoituksille tyypillisesti, eikä niissä ole pikatoimintoja. (Android Developers 2014.)

5.4.5 Uuden painikkeen luonti

Työn sovelluksen aloitusnäkymä on muokattava joukko painikkeita. Näiden painikkeiden lisääminen osoittautui käyttökokemuksen kannalta hankalaksi. Uuden painikkeen luonti voi helposti olla sekava tilanne, jossa käyttäjä ei täysin tiedä, mitä on tekemässä. Tästä näkymästä tehtiin monta erilaista versiota, joiden käyttöliittymä oli melko samantapainen, mutta toiminnot vaihtelivat.

Sovelluksen ohjelmoijilla oli myös oma visionsa painikkeen luomisesta. Siinä valitaan ensin, luodaanko tapahtuma vai rutiini. Sen jälkeen esiin ponnahtaa tapahtuman tai rutiinin kategorian valinta. Jos ollaan luomassa tapahtumapainiketta, kategorian valinnan jälkeen näytetään lista valittavista tapahtumapaikoista. Tämän toiminnon tekeminen jäi prototyyppisovelluksessa vielä kesken, joten sekavuus on ymmärrettävää. Toiminto on tällä hetkellä tehty ohjelmoijille helpolla tavalla, mutta uusi käyttäjä ei osaisi hahmottaa tilannetta.

Omassa suunnitelmassani yritettiin tehdä painikkeen luonti loppukäyttäjälle helpommaksi. Ensimmäinen valinta on jätetty pois, jolloin käyttäjän ei tarvitse miettiä, löytyykö haluttu painike tapahtumista vai rutiineista. Ensin valitaan painikkeen kategoria, esimerkiksi tanssi tai musiikki. Valitusta kategoriasta valitaan haluttu tapahtuma tai kerho, jolloin ruutuun ilmestyy hieman tietoa valitusta kohteesta tekstimuodossa. Tämän jälkeen painikkeen voi luoda. Tapahtumapaikkoja ei käydä läpi tässä vaiheessa. Jos tapahtumalla on useampia mahdollisia tapahtumapaikkoja, kuten ravintoloita, niitä voidaan tutkia luodun painikkeen takaa. Kuvassa 12 on viimeisin luonnos painikkeen luontinäkymästä `oldtimerTimer`issa.



Kuva 12. Luonnos uuden painikkeen luontinäkömstä oldtimerTimerissa

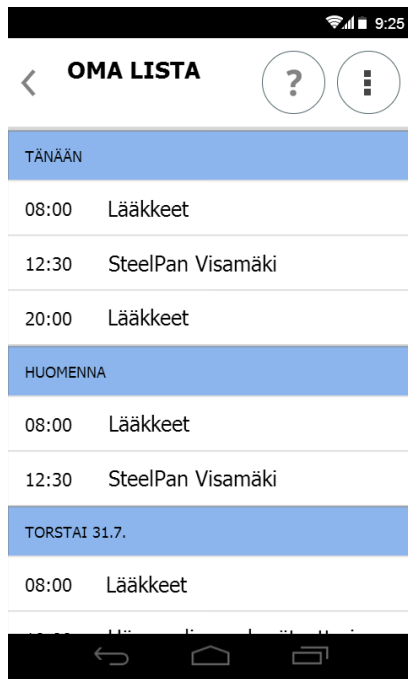
5.4.6 Oma muistilista

Oma muistilista -näkömää tunnettiin projektin edetessä muun muassa nimityksillä ”kalenteri” ja ”tulevat tapahtumat”. Muistilista kerää kaikki tapahtumat, joihin on ilmoittauduttu, ja tekee niistä helposti luettavan listan. Listassa näkyy oletuksena myös tulevat rutiinit, mutta ne on mahdollista piilottaa. Tämä näkömää oli muutosten keskipisteessä, sillä tapahtumien ja rutiinien viikkolistat oli tarkoitus suunnitella samantyylliseksi kuin oma muistilista.

Listasta oli tarkoitus aluksi tehdä kalenterin näköinen, mutta todettiin, että puhelimen näytöllä kalenterin käyttö on liian monimutkaista. Tabletilla kalenteri voisi toimia paremmin. Sovelluksessa ei ole tarkoitus ilmoittautua tapahtumiin, jotka ovat ajankohtaisia vasta kuukausien jälkeen. Sovellus tarjoaa käyttäjälle pikemminkin viikkosuunnittelua, ja siinä tarkoituksessa kalenteri voisi olla turha.

Listamuoto sopii näkömäälle paremmin varsinkin puhelimen näytöllä. Listassa väliotsikkoina on sinisiä rivejä, joissa lukee viikonpäivä ja päivämäärä. Jos päivämäärän alla ei ole tapahtumia, se automaattisesti piilotetaan listasta. Tapahtuman kohdalla näkyy ajankohta ja tapahtuman nimi. Lista on todella yksinkertainen ja käyttää tilansa hyvin.

Muistilista on tarkoitettu vain muistamiseen, ainakin tässä vaiheessa sovelluksen kehitystä. Rutiineja ei ole tarkoitus muokata sitä kautta, mutta tapahtumien ilmoittautumisia voi poistaa. Rutiinien muokkaus on luontevampaa yksittäiselle rutiinille tarkoitettussa viikkonäkömäässä. Kuvassa 13 on esitetty luonnos oldtimerTimerin muistilistan näkömäästä.



Kuva 13. Luonnos oldtimerTimerin muistilistanäkymästä

5.5 Visuaalinen tyyli ja värimaailma

oldtimerTimer-sovelluksen visuaalinen tyyli alkoi hahmottua vähitellen tutkimuksen aikana. Minimalismi on tyyli-suuntaana tällä hetkellä suosittu, ja sovellus sopi hyvin tähän yksinkertaiseen muottiin. Koska kohderyhmänä oli iäkkäämpi väestö, yksinkertaiset ruutunäkymät ja hallittu värien käyttö olivat käyttöliittymässä tärkeitä.

Käyttöliittymän elementeissä on terävät kulmat ja suuret tekstit. Pyöristettyjä kulmia vältettiin, pois lukien muutamat täysin pyöreät elementit. Teräväkulmaisuus oli vain tyyli-suunta, sen valinnalla ei ollut käyttökokemuksen kannalta juurikaan merkitystä. Teksti pyrittiin mahdollistamaan ruudulle mahdollisimman suurena, kuitenkin välttämättä liiallista tilan täyttämistä, jotta puhdas ja selkeä yleisilme säilytettiin. Tekstin fontiksi valittiin Tahoma, joka on selkeä ja suosittu fontti, ja se soveltuu mobiiliympäristöön.

Sovelluksen painikkeissa päädyttiin välttämään nykyistä tyyli-suuntaa, jossa painikkeet erotellaan värein tai ikonien sijoittelulla ilman rajaviivoja. Painikkeiden rajaviivat haluttiin säilyttää kohderyhmän takia, jotta painikkeiden tunnistaminen helpottuisi. Osa painikkeista on myös eroteltu värein, ja suurimmassa osassa on yksinkertainen ikoni.

Suunnitelmassa olevat ikonit ovat yksinkertaisia, eivätkä ne välttämättä ole lopullisia. Yksinkertaisuus ikoneissakin on nykypäivää, mutta kokemattomille älypuhelimien käyttäjille suunnattu sovellus voisi hyötyä tarkemmista, yksityiskohtaisista ikoneista. Käyttöliittymäsuunnittelija ei kuitenkaan välttämättä ole loistava piirtäjä, ja yksityiskohtaisten ikonien tekoon tarvittaisiin graafikon taitoja.

Käyttöliittymäprototyypin päävärit ovat toistaiseksi musta, valkoinen ja vaaleansininen (heksadesimaaliarvoltaan #8BB5EC). Rajausvärinä on myös käytetty harmaata. Värit voivat helpostikin muuttua jatkossa, mutta useampaa kuin kolmea pääväriä ei kannata käyttää. Sinisen ja vihreän eri sävyjä tulee välttää vanhuksille tarkoitettussa järjestelmässä varsinkin, jos niillä on tarkoitus rajata elementtejä.

Visuaalista tyyliä on melko helppoa muokata sovelluksen jatkokehityksessä, joten mitään lopullisia valintoja ei tehty. Nykyisellään käyttöliittymän suunnitelma näyttää selkeältä ja suhteellisen modernilta, joskin hieman viimeistelemättömältä. Vanhusten tarpeet on otettu siinä huomioon mahdollisimman hyvin. Jos sovelluksesta olisi haluttu tehdä vielä yksinkertaisempi, olisi ominaisuuksia pitänyt karsia. Vaihtoehtoisesti navigaatiotasoja olisi voinut olla enemmän, mutta se olisi hidastanut sovelluksen käyttöä.

5.6 Prototyyppien testaus

Käyttöliittymää testattiin pääasiassa prototyyppitasolla, sillä mobiilisovelluksen kehitys jäi kesän aikana melko varhaiseen vaiheeseen. Testausta tehtiin jossain määrin jo Digital Service Development -opintojakson aikana, kun käyttöliittymän luonnoksia esiteltiin projektiryhmälle. Muutosehdotuksia saatiin paljon, ja sovelluksen toimintaan tuli suuriakin muutoksia siinä vaiheessa.

Järjestelmällistä prototyyppitestausta aloitettiin heinäkuun puolessa välissä, kun sovelluksen kaikista ruuduista oli saatu valmiiksi paperiluonnoksia. Tässä vaiheessa sovelluksen ominaisuudet oli pääosin lyöty lukkoon kehityksessä, mutta jatkokehityksen mahdollisuuksia harkittiin. Käyttöliittymää alettiin suunnittelemaan enemmän tulevaisuutta ajatellen. Paperiluonnokset käytiin läpi projektiryhmän jäsenten kanssa, yksi henkilö kerrallaan.

Aivan työn lopulla tehtiin vielä kaksi testausta valmiin interaktiivisen prototyypin avulla. Varsinaisia virheitä ei enää löytynyt, mutta muutosehdotuksia tuli muutama. Muutosehdotuksia voidaan pitää mielipiteinä, joita pitää harkita sovelluksen jatkokehityksessä. Muutokset vaativat usein kompromisseja, joita ei aina kannata tehdä yhden tai kahden mielipiteen takia. Hyvä käyttöliittymäsuunnittelija kuitenkin harkitsee kaikkia mahdollisia ehdotuksia.

Testausten tuloksia varten tehtiin muistiinpanodokumentteja, joihin kirjattiin ilmenneitä puutteita ja muutosehdotuksia. Dokumenttien rakenne oli yksinkertainen: sovelluksen kaikki näkymät listattiin allekkain, ja jokaisen näkymän alle merkittiin ranskalaisilla viivoilla ilmenneet puutteet ja korjausehdotukset.

5.7 Käyttöliittymän toteutus

Kesän projektin aikana tehty oldtimerTimerin prototyypiversio Eclipse-ohjelmointiympäristöllä jossa on ADT (Android Developer Tools) lisäosa, jota Google suosittelee Android-sovellusten kehittämiseen. Kehitteillä on uusi ympäristö nimeltään Android Studio, joka on betavaiheessa. Android Studio on aloittelevalle Android-kehittäjälle tutustumisen arvoinen, sillä ADT:n aktiivinen tuki aiotaan lopettaa. (Android Developers 2014.)

Aluksi suunnitelmana oli, että kun sovelluksen ohjelmoijien projektityö päättyy heinäkuun lopulla, sovelluksen käyttöliittymää olisi hiottu omasta toimestani opinnäytetyötä varten. Prototyypin käyttöliittymä jäi niin alkeelliseksi, että oma aikani elokuussa ei olisi riittänyt käyttöliittymän toteuttamiseen. Niinpä tässä luvussa kerrotaan käyttöliittymän toteuttamisesta yleisemmällä tasolla, mutta oldtimerTimerin ratkaisut huomioon ottaen.

Kaikki natiivit Android-sovellukset ohjelmoidaan Java-kielellä. Käyttöliittymäkomponenttien asettelu määritellään XML-tiedostoissa. Nämä tiedostot sijaitsevat /res/layout -hakemistoissa, ja layout-kansioita voi olla useampiakin, jotta käyttöliittymä voidaan skaalata erikokoisille näytöille. Skaalauksesta kerrotaan lisää seuraavassa luvussa. Eclipse sisältää myös käyttöliittymäeditorin, jolla XML-tiedoston määrittämiä voi hahmotella graafisesti. Lopulliset asetelut kannattaa kuitenkin aina viimeistellä manuaalisesti XML-tiedostossa. (Harju 2013, 53–62.)

Android tukee useita erilaisia asettelumalleja, joista yleisimmin käytettyjä ovat LinearLayout ja RelativeLayout. LinearLayout mahdollistaa käyttöliittymäelementtien asettelun peräkkäin pysty- ja vaakasuunnassa. RelativeLayout mahdollistaa elementtien asettelun toisiinsa ja asettelun reunoihin suhteutettuna, mikä helpottaa sovelluksen skaalausta. Asettelumallin määrittelevää tagia käytetään XML-tiedoston juuritagina. Asettelumalleja voidaan käyttää myös sisäkkäin, mikä onkin yleensä tarpeen halutun asettelun aikaansaamiseksi. (Harju 2013, 53-58.)

Alla olevassa XML-esimerkissä on määritelty RelativeLayout-asettelumallilla yksinkertainen näkymä. Näkymä on yksinkertaistettu versio oldtimerTimerin ”Uuden painikkeen luonti”-ruudusta. Padding- ja width-arvoissa käytetty mittayksikkö dp on Androidin käyttämä virtuaalinen pikseliyksikkö.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp" >
    <Spinner
        android:id="@+id/category"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <Spinner
        android:id="@+id/target"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/category" />
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/target"
    android:layout_centerHorizontal="true"
    android:text="@string/create" />
</RelativeLayout>

```

5.8 Skaalaus

Android Developers (2014) tarjoaa kattavan tietosivun käyttöliittymän skaalaamisesta. Skaalaus tässä yhteydessä tarkoittaa käyttöliittymän sovittamista erikokoisille näytöille niin, että elementit eivät näytä venytetyiltä. Varsinkin sovellukset joita kehitetään sekä puhelimille että tableteille tarvitsevat vähintäänkin skaalausta, joskus jopa täysin erillisen käyttöliittymän.

Android-laitteiden näyttökoossa, pikselitiheydessä ja resoluutiossa on paljon vaihtelua. Sovelluksia skaalatessa näyttökoolla ja pikselitiheydellä on suurin merkitys. Google on jaotellut molemmat neljään yleistettyyn ryhmään, joiden avulla skaalauksesta tulee tarpeeksi monipuolinen. Lisäksi skaalauksessa otetaan usein huomioon näytön asento, joka voi olla pystyssä tai vaakatasossa.

Androidin kehittäjien helpotukseksi on otettu käyttöön virtuaalinen pikseliyksikkö, tiheydestä riippumaton pikseli, englanniksi Density-independent pixel (dp). Sen avulla voidaan määritellä käyttöliittymän mittasuhteet pikselitiheydestä riippumattomalla tavalla. Järjestelmä osaa skaalata dp-yksiköt käytössä olevan laitteen pikselitiheyden mukaisesti. Bittikarttagrafiikan laatu saattaa kuitenkin heikentyä ylöspäin skaalatessa, joten grafiikasta kannattaa olla olemassa useita versioita eri resoluutioilla.

Android tukee vaihtoehtoisten resurssien käyttöä, esimerkiksi grafiikkaa eri pikselitiheyksillä, ja osaa käyttää oikeita resursseja laitteen mukaan. Nämä resurssit pitää vain sijoittaa oikeaan paikkaan, jotta järjestelmä osaa ne hakea. Taulukossa 1 on havainnollistettu esimerkkiresurssien oikeat kansiosijainnit sovelluksen rakennekansiossa.

Taulukko 1. Vaihtoehtoisten resurssien kansiosijainnit

Kansio	Resurssi
res/layout/my_layout.xml	Layout normaalille (vakio) ruutukoolle
res/layout-small/my_layout.xml	Layout pienelle ruutukoolle
res/layout-large/my_layout.xml	Layout suurelle ruutukoolle
res/layout-xlarge/my_layout.xml	Layout extrasuurelle ruutukoolle
res/layout-xlarge-land/my_layout.xml	Layout extrasuurelle ruutukoolle vaakasuunnassa
res/drawable-mdpi/my_icon.png	Pikseligrafiikka pienelle tiheydelle
res/drawable-hdpi/my_icon.png	Pikseligrafiikka keskiverto tiheydelle
res/drawable-xhdpi/my_icon.png	Pikseligrafiikka korkealle tiheydelle

Käytännön skaalausta oldtimerTimer-sovelluksessa ei toistaiseksi tehty, johtuen sovelluksen keskeneräisyydestä. Sovelluksen prototyyppi on tehty erittäin suurelle ruutukoolle, eli 10-tuumaiselle tabletille jossa on korkea pikselitiheys, mutta sitä ei ole optimoitu, ja osa elementeistä ei skaalaudu oikein. Skaalaus on monimutkainen työvaihe, jossa täytyy ottaa huomioon grafiikan ja piirrettävien elementtien lisäksi myös tekstin koko, ja se vaatii kärsivällistä testausta.

5.9 Projektiryhmän osana toimiminen

Projektiryhmään oldtimerTimerin tapauksessa kuului projektimanageri, kaksi websivuston kehittäjää, kaksi mobiilisovelluksen ohjelmoijaa, tietokantavastaava, tietoturvavastaava, palvelinvastaava, websivuston käyttöliittymän suunnittelija sekä mobiilisovelluksen käyttöliittymän suunnittelija eli minä itse.

Eniten yhteistyötä tein mobiilisovelluksen ohjelmoijien ja projektimanagerin kanssa. Vuorovaikutus muihin projektiryhmän jäseniin rajoittui lähes täysin kokouksiin, ja edes websivuston käyttöliittymän suunnittelijan kanssa en juurikaan tehnyt yhteistyötä. Websivuston ja mobiilisovelluksen käyttöliittymä pidettiin suurimmaksi osaksi erillisinä kesän projektin aikana, sillä kumpaakaan ei tehty valmiiksi. Websivuston kohderyhmä on myös erilainen kuin mobiilisovelluksen, joten sen ei tarvitse olla aivan yhtä paljon yksinkertaistettu. Kun järjestelmä alkaa olla valmiimpi, sivuston ja sovelluksen visuaalista tyyliä ja värimaailmaa kannattaa yhtenäistää.

Projektiryhmä kokoontui koululla Visamäellä joka arkipäivä, paitsi keskiviikkoisin. Keskiviikko oli varattu opinnäytetöiden kirjoittamiseen, mutta kehitystiimin töissä olevat henkilöt olivat koululla töissä myös keskiviikkoisin. Scrum-kokouksia pidettiin kahden viikon välein

maanantaisin sprinttien välissä. Itse kävin projektitapaamisissa vain kerran tai kaksi kertaa viikossa, koska en asunut Hämeenlinnassa, ja yhteistyö ei vaatinut jokapäiväisiä käyntejä. Koululla käydessäni annoin konsultointia sovelluksen kehitykseen, sekä testautin käyttöliittymän luonnoksia. Myös projektinmanagerin kanssa keskustelimme sovelluksen käyttökokemuksesta ja kävin läpi tekemääni työtä.

Tiedon kulku oli projektiryhmässä toimivaa, ja kokouksiin osallistuminen etänä toimi Microsoftin Lyncin avulla kohtalaisen hyvin. Projektinhallintaan käytettiin selainpohjaista teamwork.com-sovellusta, joka piti ryhmän jäsenet ajan tasalla projektin kulusta. Ryhmässä oli pääosin hyvä työmotivaatio, ja ongelmia yritettiin ratkoa mahdollisimman paljon yhdessä.

6 JATKOKEHITYS

Järjestelmän kehitystyö jäi kesken, mutta oldtimerTimer-sovelluksen sekä websivuston kehityksen on tarkoitus jatkaa syksyllä 2014 tietojenkäsittelyn koulutusohjelman ICT-projektina. Kesällä tehty pohjatyö on uutta projektia ajatellen arvokasta. Kesän projektissa saatiin järjestelmä osittain toiminnalliseksi, ja useiden muutosvaiheiden jälkeen järjestelmästä saatiin jokseenkin johdonmukainen ja sen kehityssuunta on selkeämpi.

Mobiilisovelluksen kohderyhmä saattaa jatkossa laajentua. Tätä spekuloidtiin jo kesäprojektin aikana. On myös mahdollista, että kohderyhmää ei tulevaisuudessa ole lainkaan. Sovelluksen konsepti on sellainen, että se voisi olla hyödyllinen muillekin kuin vanhuksille. Kohderyhmän laajennus on kuitenkin hankala asia käyttöliittymäsuunnittelun kannalta, koska kokeneemmat älypuhelimien käyttäjät saattavat pitää nykyistä käyttöliittymää liian yksinkertaisena ja hidaskäyttöisenä. Tällöin voi joutua tekemään kompromisseja, jotka taas saattavat aiheuttaa vaikeuksia vanhuksille.

Sovelluksen käyttöliittymä jäi vielä lähes täysin toteuttamatta, ja se ei toimi aivan samalla tavalla kuin tämän työn suunnitelmassa. Tabletilla osa ruuduista pitää vielä suunnitella uudestaan niin, että suuremman ruudun tarjoama tila käytetään mahdollisimman hyvin. Puhelimelle suunniteltuja näkymiä voidaan käyttää tablettikoossakin, mutta tabletille suunnitellut näkymät kutistuvat liian pieniksi puhelimen näytöllä. Siksi kännykkänäkymien toteutus pitää laittaa etusijalle jatkokehityksessä.

Ominaisuuksien lisääminen on normaali käytäntö monissa sovelluksissa, eikä oldtimerTimer varmastikaan ole poikkeus. Kilpailutilanne on tässä tapauksessa erityinen, sillä suoria kilpailijoita järjestelmällä ei ole. Uudistumisen paineita ei ole, joten lisäominaisuudet eivät ole heti tarpeellisia, mutta joitakin käytettävyyttä parantavia ominaisuuksia kannattaa harkita. Tapahtumapaikkojen etsiminen GPS-paikannuksella oli yksi puhutuimmista ominaisuuksista, jota ei vielä sovellukseen toteutettu. Myös hakuominaisuus tapahtumien etsimiseen tullee ajankohtaiseksi ennemmin tai myöhemmin.

Sovelluksen prototyyppi ja käyttöliittymän prototyyppi toimivat eri tavoin, mikä johtuu osittain sovelluksen varhaisesta kehitysvaiheesta ja osittain huonosta kommunikaatiosta sekä päätöksenteosta. Käyttöliittymän prototyyppi tehtiin käytettävyyttä ajatellen, mutta Linnan Aterian, Lasse Seppäsen ja Tommi Lahden toiveiden perusteella. Nämä toiveet olivat ristiriitaisia, mikä aiheutti projektiryhmän jakautumista. Sovelluksen kehittäjät toimivat lähinnä Tommi Lahden ohjeistuksen mukaisesti, ja osittain käyttöliittymäsuunnitelman pohjalta. Tekniset ongelmat ohjelmoinnissa kuitenkin pakottivat heidät tekemään omiakin ratkaisuja.

Jos ohjelmoijilla olisi ollut enemmän aikaa, olisi sovelluksesta tullut luultavasti enemmän suunnitelmaa vastaava. Jotta oldtimerTimer-sovelluksesta tulisi valmis ja toimiva kokonaisuus, jatkokehityksessä on

oltava selkeämpi kehityssuunta, selkeä visio. Tällä tavalla ohjelmoijien työstä tehdään paljon helpompaa.

Sovelluksen nimestä ei projektiryhmän kesken keskusteltu yhtään. Selvää on, että oldtimerTimer on vain kehitysnimi, ja että lopullinen sovellus tarvitsee helpommin ymmärrettävän tai muulla tavalla mieleenpainuvamman nimen. Mikäli kohderyhmä pysyy vanhuksissa, nimestä pitää tehdä lyhempi tai suomenkielinen. Jos kohderyhmä laajenee ja sovellukselle halutaan laajempaa huomiota, viittaukset vanhuksiin pitää jättää nimestä pois. Nykyinen nimi on joka tapauksessa harhaanjohtava, ja vaatii päivitystä.

7 YHTEENVETO

Opinnäytetyön alkuperäisenä tavoitteena oli suunnitella ja jossain määrin myös toteuttaa käyttöliittymä oldtimerTimer-mobiilisovellukselle. Suunnittelu onnistui hyvin, ja iteratiivinen suunnittelutapa toimi projektissa kutakuinkin odotetulla tavalla. Erilaisia versioita näkymistä suunniteltiin vähemmän kuin oli tavoitteena, mutta käyttöliittymä kehittyi koko projektin ajan. Käyttöliittymän toteutus jäi vielä kesken, koska sovelluksen kehitys oli odotettua hitaampaa. Osa sovelluksen toiminnoista jäi toistaiseksi toteuttamatta, ja käyttöliittymän toteutus Eclipsellä osoittautui hankalaksi. Suunnitelma kuitenkin asettaa selkeämmät tavoitteet jatkokehitykseen, joten toteutus sujuu luultavasti paremmin jatkossa.

Työn määritteleviä tutkimuskysymyksiä oli alussa kolme, mutta yksi niistä muuttui projektin aikana. Muuttunut kysymys liittyi käyttöliittymän skaalaukseen, mutta sen tilalle vaihdettiin menetelmien ja työkalujen tutkimukseen liittyvä kysymys. Kävi ilmi, että Android sisältää hyvän kehyksen skaalauksen toteuttamiseen, mikä tekee skaalauksesta suoraviivaista, joten skaalaukseen ei haluttu keskittyä. Loppuvaiheessa lisättiin vielä neljäs kysymys, ”Miten sovelluksen käyttöliittymästä tehdään yhtäaikaaisesti sekä moderni että vanhuksille sopiva?”. Sovelluksen käyttöliittymän ajanmukaisuus ja sopivuus vanhuksille olivat suunnitteluprosessin tärkeimmät tavoitteet alusta asti.

Ensimmäinen alkuperäisistä tutkimuskysymyksistä koski käyttöliittymäsuunnittelijan ja muun projektiryhmän välistä yhteistyötä. Yhteistyötä tarvittiin ehkä yllättävänkin vähän, mutta se on eittämättä osa käyttöliittymäsuunnittelijan työtä. Läheisintä yhteistyötä suunnittelussa tehtiin projektimanagerin ja sovelluksen ohjelmoijien kanssa, mutta muu ryhmä jäi selkeästi työskentelyssä ulkopuolelle. Projektiryhmän toimintaan liittyvää asiaa on erityisesti luvussa 5.9, mutta yhteistyötä käsitellään muissakin työn osissa. Toinen alkuperäinen tutkimuskysymys koski sovelluksen jatkokehitystä ja sen vaikutuksia käyttöliittymäsuunnittelussa. Jatkokehitysmahdollisuudet huomioitiin suunnittelussa hyvin, varsinkin loppuvaiheessa.

Työn alussa ammattimainen käyttöliittymäsuunnittelu oli kirjoittajalle tuntematon aihealue. Työn oppimistavoitteena oli tutustua aiheeseen mahdollisimman laajasti mobiilisovellusten näkökulmasta. Koska aihealue rajoittui pääasiassa mobiilisovelluksiin, oppia tuli pintaraapaisua syvemmältä. Kerättyä oppia on myös helppoa soveltaa jatkossa webikäyttöliittymiin. Photoshopin käyttötaito vahvistui myös huomattavasti, vaikka sitä ei käytetty työn pääasiallisena työkaluna. Android kehitysalustana tuli hyvin tutuksi, ja uusia teknisen puolen asioita opittiin käyttöliittymän toteutuksen ja skaalauksen osalta. Projektitoiminnassa uuden sovelluksen kehitysprosessi ja sen viitekehyksenä käytetty Scrum toivat uudenlaista kokemusta.

Opinnäytetyö vahvisti tekijän omaa mielenkiintoa käyttöliittymäsuunnitteluun. Alalla työskentely kiinnostaa nyt entistäkin

enemmän, ja omiakin projekteja olisi tarkoitus aloittaa lähitulevaisuudessa. Käytännön kokemusta alalta tarvitaan lisää, ja sen keräämisen täytyy olla jatkossa ensimmäinen prioriteetti.

LÄHTEET

Adams, P. 2013. Why cards are the future of the web. Intercom Blog
<http://insideintercom.io/why-cards-are-the-future-of-the-web/>
Viitattu 29.7.2014

Android Developers 2014. Design
<http://developer.android.com/design>
Viitattu 21.7.2014

Android Developers 2014. Develop
<http://developer.android.com/develop>
Viitattu 20.8.2014

Bohemian Coding 2014. Sketch for Windows and Linux
<http://bohemiancoding.com/sketch/support/faq/02-general/5-windows.html>
Viitattu 5.8.2014

Cianfrone, J. 2012. Which is better – Photoshop or Illustrator?
<http://www.creativebloq.com/adobe/which-better-photoshop-or-illustrator-12121500>
Viitattu 4.8.2014

Díaz, J.M. Moreno, L. 2013. Accessibility to mobile interfaces for older people. Elsevier B.V.
<http://www.sciencedirect.com/science/article/pii/S1877050914000106>

Downer, C. 2012. Skeuomorphism and the User Interface
<http://realmacsoftware.com/blog/skeuomorphism-and-the-user-interface>
Viitattu 28.7.2014

Golden Gekko 2014. 2014 UX/UI trends for mobile solutions
<http://www.slideshare.net/goldengekko/mobile-apps-design-trends-2014>
Viitattu 3.8.2014

Harju, J. 2013. Android-ohjelmoinnin perusteet. Books on Demand GmbH

Heaton, J. 2013. Made for UI Design: A Review of Sketch
<http://www.bignerdranch.com/blog/made-ui-design-review-sketch/>
Viitattu 5.8.2014

IDC 2014. Smartphone OS Market Share, Q2 2014
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
Viitattu 16.8.2014

Interacta 2014. GUIDe - käyttöliittymäspekssi

<http://www.interacta.fi/speksi.html>

Viitattu 19.8.2014

Lewis, J. 2013. What's The Difference Between Haptics Now And Then?

<http://electronicdesign.com/components/what-s-difference-between-haptics-now-and-then>

Viitattu 13.7.2014

McKinney, D. 2013. How To Set Up Photoshop For UI Design

<http://davidmckinney.com/blog/2013/12/31/designing-iphone-apps-how-to-setup-photoshop>

Viitattu 21.7.2014

Medero, S. 2007. Paper Prototyping

<http://alistapart.com/article/paperprototyping>

Viitattu 16.7.2014

Miller, D. 2012. UI vs UX: what's the difference? Web Designer Depot

<http://www.webdesignerdepot.com/2012/06/ui-vs-ux-whats-the-difference/>

Viitattu 23.7.2014

Salimi, D. 2014. 2014 Mobile UX Design Trends: Embracing Minimal

<http://blog.gfk.com/2014/05/2014-mobile-ux-design-trends-embracing-minimal/>

Viitattu 10.7.2014

Schofield, J. 2014. Google tries to save the web from the curse of 'infinite scrolling'. ZDNet

<http://www.zdnet.com/google-tries-to-save-the-web-from-the-curse-of-infinite-scrolling-7000026900/>

Viitattu 29.7.2014

Simões, T. 2012. 7 Steps To A Killer Mobile UI. InformationWeek

Sinkkonen, I. Kuoppala, H. Parkkinen, J. Vastamäki, R. 2002. Käytettävyyden psykologia, 2. painos. Edita Publishing Oy

Verge Staff 2011. Android: A visual history. The Verge

<http://www.theverge.com/2011/12/7/2585779/android-history>

Viitattu 25.7.2014

OLDTIMERTIMER-MOBIILISOVELLUKSEN KÄYTTÖLIITTYMÄSPEKSI

Aloitus

Sovelluskokonaisuuden aloitusruutu, jossa on painikkeita eri rutiineihin ja tapahtumiin.



Ylhäällä vasemmalla on logo. Kuvassa esimerkkilogo.

Logon vieressä, Action Barissa oikealla *Apua* -painike, josta saa avustavia puhekuplia ruutuun.

Apua -painikkeen oikealla puolella Action Overflow -painike, josta avautuu ponnahduslista sivun toiminnoista. Myös asetukset löytyvät täältä.

Keskellä painikkeet joista pääsee tarkastelemaan luotuja rutiineita ja tapahtumia. Puhelimella painikejoukkoja selataan ylös-alas, mikäli painikkeita on enemmän kuin neljä.

Pyöreiden tapahtuma / rutiinipainikkeiden alla teksti.

Tapahtumapainikkeiden alapuolella ruudun alareunassa on painikkeet *Uuden painikkeen luontiin* ja *Omaan muistilistaan*.

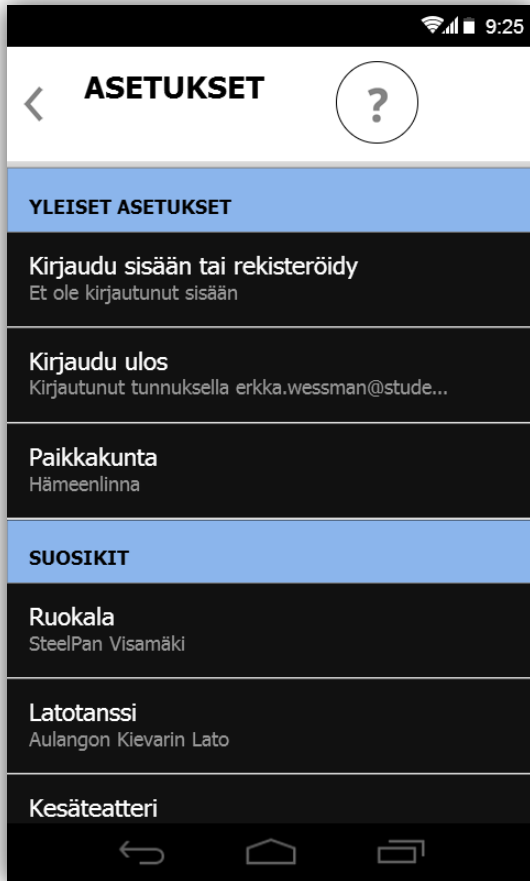
Ikonit ovat alustavia esimerkkejä, ja lisää ikoneita pitää tehdä erilaisia tapahtumia/rutiineja varten.

Heti aloitusnäkyvästä alkaen sovellus yrittää väreillä esittää, että valkoiset alueet ovat interaktiivisia ja siniset eivät.

Aloitusrudussa käytetyt värit ovat #8BB5EC, #FFFFFF ja #000000. Harmaa väri tulee Fluid UI:n valmiista elementeistä, se ei kuulu varsinaiseen suunnitelmaan.

Asetukset

Sovelluksen asetukset, joita ei tässä kehitysvaiheessa ole montaa. Asetukset -sivu mukailee Googlen ohjeistuksia, mutta sivulle on lisätty hieman väriä selkeytykseksi.



Action Barissa vasemmalla on Androidin up-painike, josta pääsee tason taaksepäin sovelluksen hierarkiassa. Vieressä teksti josta tietää millä sivulla ollaan.

Yläreunasta puuttuu Action Overflow -painike, koska asetuksissa sitä ei tarvita.

Asetuksissa on väliotsikot, joissa ei ole interaktiivisuutta. Väliotsikoille on annettu sovellukselle ominainen vaaleansininen väri.

Interaktiiviset asetuspainikkeet ovat mustataustaisia. Tässä on tehty tietoinen eroavaisuus sovelluksen muihin osiin.

Kuvassa näkyy esimerkin vuoksi sekä sisäänkirjautuminen että uloskirjautuminen, todellisuudessa asetuksissa näkyy vain toinen niistä.

Asetuspainikkeet ovat kaksirivisiä. Toisella rivillä näkyy asetuksen tämänhetkinen valinta/tila.

Asetuksista pääsee myös vaihtamaan

tapahtumien suosikkipaikkoja.

Lista asetuksista:

- Kirjaudu sisään tai rekisteröidy / Kirjaudu ulos
- Paikkakunnan valinta
- Suosikkien valinta, jokaisesta oma painike

Kirjautumista ja rekisteröitymistä varten avautuu yksinkertainen lomakeikkuna. Ulos kirjautuessa sovellus kysyy, "Haluatko kirjautua ulos tunnukseelta ...".

Uuden painikkeen luonti

Alasvetovalikoita, josta uuden painikkeen kategoria ja kohde. Tämä luo aloitusruutuun uuden painikkeen. Ruudusta on tehty mahdollisimman yksinkertainen ja helposti ymmärrettävä.

Action Bar samanlainen kuin *Asetuksissa*, mutta Action Overflow on taas mukana.

Näkymässä on kaksi alasvetovalikkoa, joista alempi on osittain häivytetty/sumennettu aluksi.

Ylemmästä valikosta valitaan painikkeen kategoria, joka voi olla esimerkiksi tanssi tai musiikki.

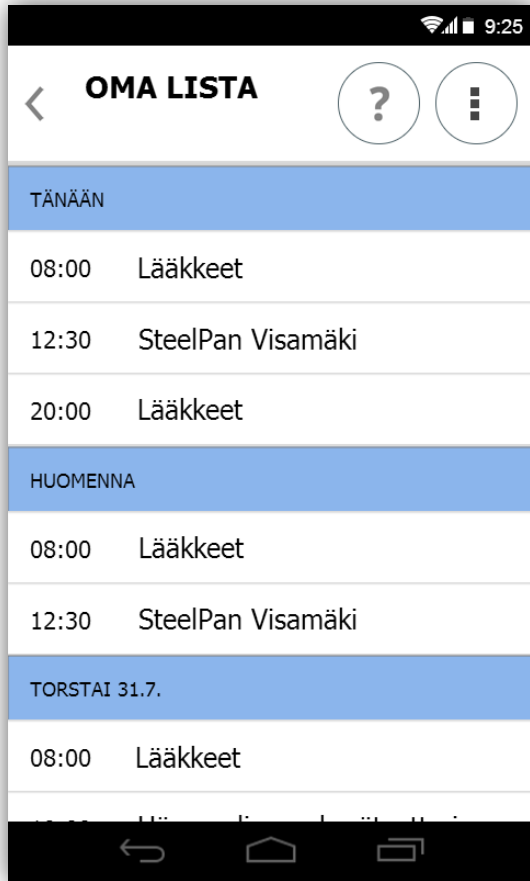
Kun kategoria on valittu, alempi valikko tulee kokonaan näkyviin. Siitä valitaan tarkemmin haluttu kohde. Itse tapahtuman paikkaa ei tässä vaiheessa valita.

Valikkojen alle ilmestyy valintojen jälkeen lyhyt kuvaus kohteesta.

Ruudun alareunassa oleva luontipainike on myös häivytetty, kunnes valinnat on tehty. Se on tarkoituksella pienehkö, jotta vahinkopainallukset jäisivät minimiin. Sitä painamalla siirrytään takaisin aloitussivulle, johon ilmestyy uusi painike, sekä väliaikainen teksti "Uusi painike luotu".

Oma muistilista

Lista tulevista tapahtumista ja rutiineista, jota voi rullata alaspäin niin kauan kun tapahtumia riittää.



Action Bar pysyy tutunlaisena.

Listaa vain ne päivät, joihin on ilmoittauduttu, sekä käynnissä olevat rutiinit.

Jos tapahtumia ei ole merkitty, rutiineja ladataan näkymään kaksi viikkoa eteenpäin.

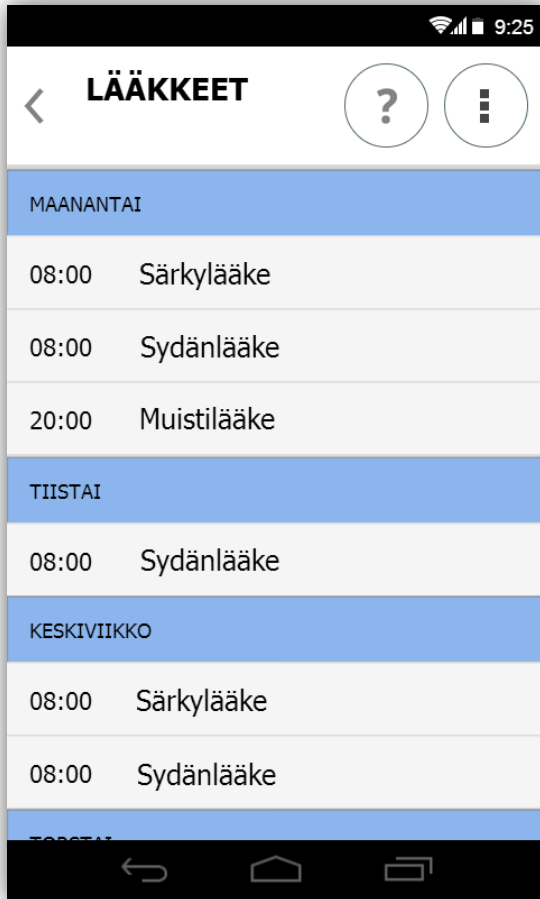
Action Overflow:ssa on painike josta voidaan piilottaa rutiinit listasta.

Sinisillä riveillä erotellaan päivät. Myös päivämäärä näytetään. Ne eivät ole interaktiivisia.

Listassa näkyvä kellonaika on tapahtuman alkamisaika, ei välttämättä se aika jolloin käyttäjä haluaa mennä (aikaa ei määritellä tapahtumaan ilmoittautuessa).

Rutiininäkymä

Viikoittainen rutiini näkyy tässä. Näkymässä on myös mahdollisuus muokata tietoja. Esimerkkikuvassa näkyy moniosainen lääkerutiini.



Action Bar pysyy samanlaisena, vain teksti muuttuu. Action Overflow:sta löytyy valinta muokkausta varten.

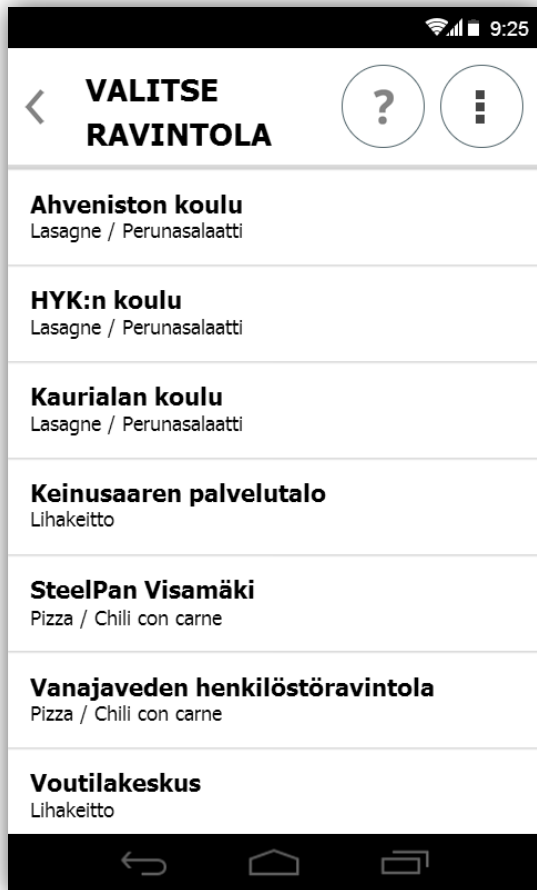
Muokkauksessa riviä voi painaa, jolloin esiin ponnahtaa lomake, josta voi vaihtaa lääkkeen ottoajan tai otsikon.

Siniset rivit erottelevat päivät. Päivämääriä ei ole, koska ideana on, että jokainen viikko on samanlainen.

Lääkerivit ovat harmahtavia, koska ne eivät ole interaktiivisia ilman muokkaustilaa.

Tapahtuman paikan valinta

Tässä ruudussa näkyy lista valittavista tapahtumapaikoista, oli kyseessä sitten ruokat tai esimerkiksi bingot.



Action Barissa aina "Valitse ravintola / bingo / elokuvateatteri", tapahtuman tyyppin mukaan.

Keskellä näkyy lista paikoista. Tässä voi olla kaikki paikkakunnan vaihtoehdot, tai vaihtoehtoisesti GPS:llä haetut lähimmät paikat.

Painikkeet listassa ovat kaksirivisiä. Alemmassa rivissä näkyy kyseessä olevan päivän tapahtuma.

Listan riviä painamalla pääsee *Tapahtumapaikan valikkoon*.

Tämä ruutu esitetään käyttäjälle joka kerralla, ellei käyttäjä aseta suosikkipaikkaa Action Overflowsta tai *Asetuksista*.

Tapahtumapaikan valikko

Tapahtumapaikan valikko avautuu kun tapahtumapaikka on valittu. Se sisältää painikkeet jotka johtavat ruutuihin *Tapahtuma tänään*, *Viikon tapahtumat* ja *Tapahtumapaikan info*.



Action Bar pysyy tutunlaisena, mutta siinä on tapahtumapaikan nimi.

Action Overflow -painikkeen takana voi olla päivän valinta, jos haluaa löytää tietyn päivämäärän tapahtuman. Valinnan jälkeen siirrytään ruutuun *Tapahtuma tänään*, joka näyttää kyseisen päivän tiedot.

Action Overflow:ssa myös mahdollisuus koko tapahtumalistan lataamiseen. Se voi olla pdf- tai kuvamuodossa.

Keskellä on optimitilanteessa neljä painiketta. Kolme näistä on tärkeitä, tapahtumalajien etsimisen voi siirtää Action Overflow:hon, jos tilalle tulee tärkeämpi toiminto.

Tapahtuma tänään

Ruudussa näkyy painikkeena päivän ruoka/esiintyjä. Jos samana päivänä on useampi ruokavaihtoehto tai esiintyjä, painikkeita on enemmän. Tapahtumia voi myös selata sivuttaissuunnassa, jolloin tulee näkyviin eri päivä.



Action Barin tekstinä on edelleen tapahtumapaikan nimi.

Ylimpänä tieto-osassa näkyy valittu päivä ja tapahtumien ajankohta/aukioloaika.

Päivän alle on lisätty ohjeteksti, koska valintamahdollisuus saattaa muuten mennä käyttäjältä ohi.

Keskellä näkyy valintapainikkeet. Niitä voi olla yksi tai useampia. Ruokien kohdalle on lisätty merkkejä, joilla voi antaa lisätietoa. Useimmissa tilanteissa merkkejä ei tarvita.

Alimpana merkkien selitykset, mutta nämä voisi myös siirtää Action Overflow:hon, mikäli niitä ei aina tarvita.

Merkkien selitysten tilalla voi pitää myös esittelytekstiä.

Tapahtumien toteuttaminen samalla muotilla voi olla ongelmallista. Joillakin tapahtumilla on jokin tietty aikamääre, mutta ruokaloiden tapauksessa tällaista tilannetta ei ole.

Tämän viikon tapahtumat

Ruudussa näkyy viikkonäkymä tapahtumista.



Action Bar on tutunlainen.

Sinisellä taustalla näkyy, mikä kuukausi on kyseessä. Tässä tapauksessa viikko sattuu olemaan kahden kuun välissä. Viikkonumero olisi hyvä näyttää myös, vaikka se tästä puuttuu.

Alla on lista viikonpäivistä päivämäärineen. Lista alkaa aina maanantaista, vaikka se olisi jo mennyttä aikaa.

Painike listassa sisältää niin monta riviä kun on tarpeen. Jos tapahtumia on päivässä monta, ne kaikki näytetään tässä.

Lauantai ja sunnuntai näytetään myös, vaikka mitään tapahtumaa ei olisi. Niissä voi lukea esimerkiksi "Suljettu".

Tapahtumapaikan info

Inforuudusta näkee tapahtumapaikan osoitteen, yhteystietoja ja aukioloajat.



Action Bar pysyy tuttuna.

Ylimpänä on paikan nimi, sen jälkeen paikan osoite.

Puhelinnumero, johon voi suoraan soittaa viereisestä painikkeesta. Painikkeesta on tehty pieni, jotta vahinkoja ei sattuisi. Käyttäjältä kysytään myös varmistus ennen soittoa.

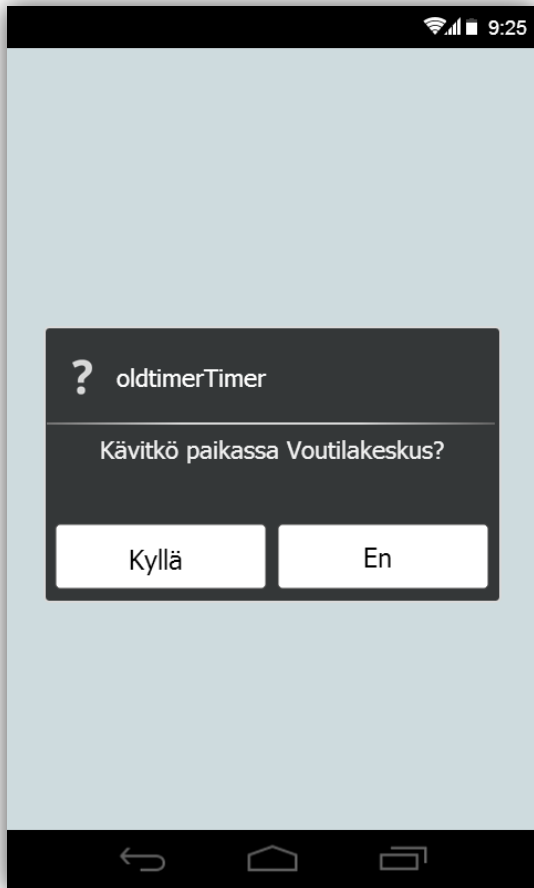
Sähköpostiosoite, johon ei voi lähettää mitään suoraan sovelluksesta.

Aukioloajat ovat pakattu mahdollisimman pieneen tilaan. Ei jokaista päivää erikseen, ellei ole pakko.

Tausta on sininen, jotta käyttäjä ymmärtää, ettei se ole interaktiivinen.

Ilmoitus

Tässä kuvassa on esitetty esimerkki sovelluksen lähettämästä ilmoituksesta. Se voi ilmestyä suoraan ruudulle tai Androidin ilmoituksiin, josta sen voi avata. Kuvassa oleva ilmoitus on Kysely, jolla varmistetaan onko käyttäjä käynyt ilmoittamassaan tapahtumassa.



Kyselyssä kysytään, onko käyttäjä käynyt siellä missä oli tarkoitus. Siihen voi vastata myöntävästi tai kieltävästi.

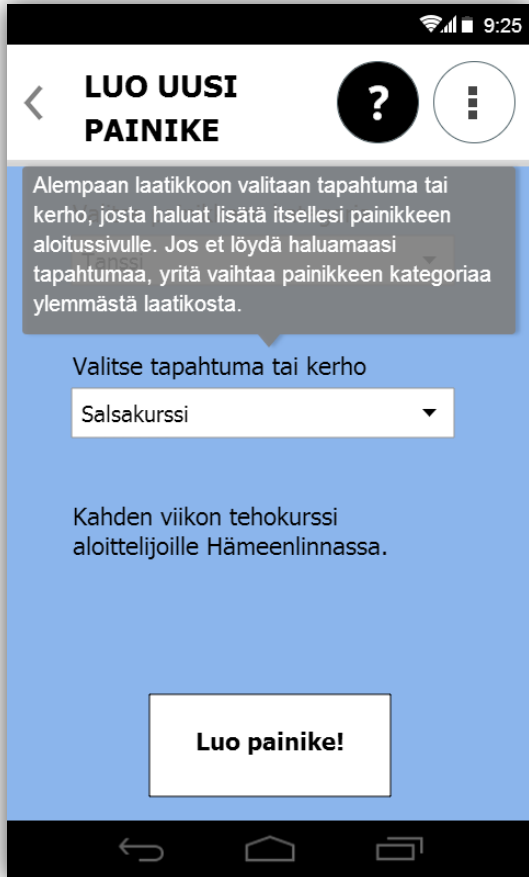
Seuraavaksi voidaan kysyä, vastauksesta riippuen, miksei käynyt tai haluaisiko käyttäjä antaa tapahtumasta palautetta.

Palautetta antaessa oldtimerTimer-sovellus avautuu, ja ohjaa suoraan palautelomakkeeseen.

Ilmoituksia on sovelluksessa monenlaisia. Voidaan tehdä muistutusilmoituksia ennen tapahtumaa, tapahtuman alkaessa (esim. lääkkeiden ottaminen) tai kuten kuvan tapauksessa eli jälkikyselyä varten.

Apua

Sovelluksen jokaisessa näkymässä on mahdollisuus saada apua. Apua annetaan puhekuplilla, jotka ovat lievästi läpinäkyviä. Esimerkkikuvassa näkyy yksi apukuplista *Uuden painikkeen luonti* -ruudussa, kun Apua -painiketta on painettu.



Action Bar pysyy paikoillaan. Apua -painike täyttyy mustalla ja ikoni muuttuu valkoiseksi.

Apukupla voisi olla myös musta. Pääasia on, että se erottuu, mutta näyttää hetkelliseltä.

Apukuplan näkyessä, kun ruutua painetaan, seuraava apukupla tulee näkyviin ja edellinen katoaa.

Apukuplan ilmestyessä voidaan näyttää yksinkertainen ponnahdusanimaatio. Kun apukuplia ei ole enempää, Apua -toiminto sulkeutuu ja sovellusta voi käyttää normaalisti.