



Rutiinipäiväkirja – mobiilityökalu äideille

Jose Kortelahti

Opinnäytetyö
Joulukuu 2013
Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka
Ohjelmistotekniikka

JOSE KORTELAKI:
Rutiinipäiväkirja – mobiilityökalu äideille

Opinnäytetyö 37 sivua
Joulukuu 2013

Tämän opinnäytetyön tarkoituksena oli opettaa tekijälleen syvemmin mobiilisovelluksen suunnittelua ja toteutusta. Työn yhtenä merkittävänä osana oli Internet-kyselyn laatiminen kohderyhmälle ja siten kerätä arvokasta tietoa sovellukseen halutuista ominaisuuksista sekä sovelluksen kohdealueista. Lisäksi kokemuksen saanti asiakaslähtöisestä sovelluskehityksestä koettiin tärkeänä – mutta ei opinnäytetyön kannalta ratkaisevana – asiana.

Työtä tehdessä tutustuttiin tarkemmin sekä halutun mobiilikäyttöjärjestelmän – Androidin – kehittämiseen ja tekniseen puoleen että kyseessä olevaa käyttöjärjestelmää käyttäviin laitteisiin. Kohdealueista käsitellään myös Android-ohjelmoinnissa tarvittavaa ohjelmistokehitystä ja muita työkaluja. Android-sovelluksien keskeistä jakelupaikkaa Google Play –kauppaa sekä yleisiä sovelluksien turvallisuus- ja yksityisyysasioita käsitellään tässä osassa.

Prototyypisovelluksen suunnittelukappaleissa käydään läpi tiivistetyn vaatimusmäärittelyn pohjalta valittujen ominaisuuksien suunnittelua sekä perusteltiin, miksi näihin ominaisuuksiin tai toteutustapoihin päädyttiin. Sovelluksen toteutuskappaleissa eriteltiin joitain toteutettuja ominaisuuksia. Pohdinta-osassa summataan opinnäytetyön aikana koettuja onnistumisia, virheitä, analysoidaan suunnittelun merkitystä sovelluskehityksessä sekä pohditaan sovelluksen jatkokehitysmahdollisuuksia.

Ensisijainen ohjelmointikieli Android-sovelluksia ohjelmoitaessa on Java. Monipuolisena alustana Androidissa voidaan käyttää myös XML-merkintäkieltä, jolla on helppo toteuttaa Android-sovelluksissa käytettävät käyttöliittymät tai usein tarvittavaa dataa, kuten valmiiden käyttöliittymäkomponenttien laajentamiseksi käytettäviä arvoja.

Työssä toteutettu rutiinipäiväkirjasovellus tehtiin tarkoituksellisesti prototyypinä viimeistellyn sovelluksen sijaan ensinnäkin ajan puutteen vuoksi, mutta prototyypin tarkoitus oli lähinnä osoittaa kyseiselle kohderyhmälle tarkoitettun sovelluksen potentiaali. Tärkeä tavoite tätä sovellusta toteuttaessa oli oppia uusia tekniikoita ja tapoja Android-sovelluksia ohjelmoitaessa. Valitsemalla sovellukseen tarkoituksella tiettyjä Android-ohjelmoinnin osa-alueita, joista oli aiemmin hyvin vähän tai ei ollenkaan kokemusta, luotiin hyvää pohjaa tietotaidolle tämän sovelluksen jatkokehitystä mutta myös muuta mobiiliohjelmointia varten.

Asiasanat: Android, Java, mobiiliohjelmointi, rutiinit, päiväkirja

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Computer Science
Software Engineering

JOSE KORTELAHTI:
Routine diary – mobile application for mothers

Bachelor's thesis 37 pages
December 2013

The purpose of this thesis for the author was to learn designing and development of mobile application in more depth. One significant aspect of this thesis was the making of an Internet survey for the targeted users and thus to collect valuable information about the desired features and the target platform of the application. Moreover, gaining experience from customer-oriented software development was important, but not crucial in terms of this thesis.

While working on the thesis the author familiarized himself with the evolution and the technical side of the target mobile operation system (Android) but with the devices using the operation system as well. In the section about target platform the framework and other software tools needed in this thesis are viewed. The central distribution channel of Android applications, Google Play store, and common safety and privacy questions are dealt in this section, too.

The sections about designing the prototype application are dedicated for planning of features described in the simplified requirement specification and for reasoning how the author ended up in such features and implementation methods. In the chapters of developing the application some developed features were specified. The final chapter is about analyzing achievements, mistakes and the importance of the design process in software development and possible further development of the application is pondered.

The primary programming language for Android software development is Java. As a versatile platform like Android is, XML markup language can also be used to build user interfaces. It can also be used to create custom data within the binary file, e.g. values to expand existing user interface components.

The routine diary application in this thesis was intentionally developed as a prototype instead of polished application due to lack of time. The purpose of the prototype was to demonstrate the potential of the application itself to the targeted users. An important goal for the author developing the application was to learn new techniques and methods on Android development. By intentionally choosing some parts in Android development that the author had little or no experience at all built a good foundation for the knowledge for further development of the application but also for mobile development in general.

Key words: Android, Java, mobile application development, routines, diary

SISÄLLYS

1	JOHDANTO	7
2	TIEDONKERUU JA ANALYSOINTI.....	8
	2.1 Sovelluksen kohderyhmä.....	8
	2.2 Internet-kysely kohderyhmälle	9
	2.3 Kyselyn tulokset.....	10
	2.4 Tiivistetty vaatimusmäärittely	13
	2.4.1 Käyttöjärjestelmä.....	13
	2.4.2 Rutiinit	13
	2.4.3 Käyttöliittymä ja käyttäjälle näkyvä rakenne.....	16
	2.4.4 Tietojen tallennus	16
3	KOHDEALUSTA JA KÄYTETTÄVÄ OHJELMISTOKEHYS	18
	3.1 Android-käyttöjärjestelmä.....	18
	3.2 Android-ohjelmistopino	18
	3.3 Android-pohjaiset laitteet	20
	3.4 Kehitysympäristö	23
	3.5 Google Play –sovelluskauppa.....	24
4	PROTOTYYPIN TEKNINEN SUUNNITTELU	26
	4.1 Tavoitteena helppokäyttöisyys	26
	4.2 Laite-erojen huomiointi.....	26
	4.3 Käyttöliittymä ja näkymät	27
	4.4 XML-tiedosto tallennusformaattina	30
5	PROTOTYYPIN TOIMINNALLISUUKSIEN TOTEUTUS.....	31
	5.1 Rutiinin käynnistäminen, lopettaminen ja muokkaaminen	31
	5.2 Käyttäjän tietojen salaaminen.....	31
	5.3 Lokalisointi.....	32
6	POHDINTA	33
	LÄHTEET.....	36

ERITYISSANASTO

Android	Googlen kehittämä mobiilikäyttöjärjestelmä, joka perustuu avoimeen lähdekoodiin.
Android SDK	Kehitystyökalukokoelma Android sovellusten ohjelmointia varten.
.apk-paketti	Android application package file –tiedostotyyppi, joka sisältää sovelluksen ajamiseen tarvittavat lähdekoodit ja ohjelmistot.
Dalvik	Android-sovellusten ajamista varten kehitetty virtuaalikone.
.dex-tiedosto	Dalvik Executable –tiedostotyyppi, pakkaustiedosto Javasta käännetylle tavukoodille.
.class-tiedosto	Tiedostotyyppi, joka sisältää Java-tavukoodin. Java-virtuaalikone suorittaa tavukoodiksi käännetyt sovellukset.
API	(engl. Application programming interface) Ohjelmointirajapinta, tarkoitus on tarjota sovellukselle mahdollisuus yleisimpien toimintojen käyttämiselle. Ohjelmat voivat jakaa tietoa ja ”keskustella” keskenään.
Google	Alkujaan hakukoneyhtiö, nykyään yksi merkittävimmistä ohjelmistoyrityksistä, jonka tarjontaan kuuluu sekä ilmaisia että maksullisia web-ohjelmistoja.
Eclipse IDE	Ilmainen kehitysympäristö, jota käytetään muun muassa Android-ohjelmien kehitykseen.
emulaattori	Tietokonesovellus, joka jäljittelee mahdollisimman tarkasti alkuperäistä laitetta, tässä tapauksessa Android-järjestelmää.
Java	Oliopohjainen, yksi yleisemmistä ohjelmointikielistä. Android-sovelluskehityksen pääkieli.
ohjelmistokehys	(engl. framework) Ohjelmistotuote, josta muodostuu varsinaisen sovelluksen runko.
XML	Merkintäkieli. Käytetään Android-sovellusten käyttöliittymän rakentamiseen ja muun helposti luettavan tiedon tallentamiseen.

älypuhelin	Kehittynyt matkapuhelin, joka sisältää yleisimpien matkapuhelintoimintojen lisäksi laitteiston Internetin käyttämiseen, navigointiin sekä kehittyneeseen tiedonsiirtoon. Älypuhelimia käytetään yleisimmin kosketusnäytön kautta.
tablet-tietokone	Älypuhelimien kaltainen, mutta kannettavan tietokoneen korvaajaksi suunniteltu älylaite.
prototyyppi	Valmiin tuotteen toimintaa tai ominaisuuksia esittelevä demonstraatio.
vaatimusmäärittely	Ohjelmistokehityksessä asiakkaan tarpeiden pohjalta laadittu dokumentti halutuista ominaisuuksista. Tämä dokumentti toimii referenssinä koko suunnitteluprosessin ajan, ja saattaa elää jonkin verran asiakkaan tarpeiden muuttuessa.
kernel	Käyttöjärjestelmäydin, joka sisältää käyttöjärjestelmän toiminnan kannalta tärkeimmät komennot, esimerkiksi muistinhallinnan tai laitteiston käytön osalta.

1 JOHDANTO

Tämän työn tarkoituksena on opettaa tekijälleen mobiilisovelluksen suunnittelua ja ohjelmointia kohderyhmältä kerättyä palautetta hyväksikäyttäen. Saatujen palautteiden pohjalta suunniteltiin ja ohjelmoitiin rutiinipäiväkirjan prototyypisovellus yhdelle markkinoilla olevalle mobiilikäyttöjärjestelmälle. Sovelluksen suunnittelu rakennettiin sovelluksen kohderyhmälle tehdyn Internet-kyselyn tulosten varaan, ja tulosten pohjalta valittiin myös sovelluksen ensisijaisesti tukema käyttöjärjestelmä, joka oli ennalta-arvattuna tällä hetkellä yleisin käyttöjärjestelmä Android.

Tässä työssä käsitellään palautteenkeräyksen vaiheita ja tulosten analysointia sekä käydään tarkemmin sovelluskehityksen työkaluja Androidia koskien. Android-käyttöjärjestelmääkin ja sen teknistä puolta raapaistaan pintaa syvemmillä. Läpi käydään myös prototyypisovelluksen suunnittelua ja toteutusta tiivistettyyn vaatimusmäärittelyyn pohjautuen.

Tämän työn luku 2 käsittelee kohderyhmälle tehdyn kyselyn palautteiden keräämistä ja analysointia sekä tulosten pohjalta toteutetun tiivistetyn vaatimusmäärittelyn. Luku 3 pureutuu syvemmin Android-käyttöjärjestelmän tekniseen puoleen, käydään läpi Android-sovelluskehityksessä tarvittavia ohelmistoja ja kehitysympäristöä yleisesti sekä puhutaan lyhyesti Google Play –kauppapaikasta. Luvussa 4 listataan tiivistetyn vaatimusmäärittelyn pohjalta suunniteltuja ominaisuuksia. Luku 5 käsittelee suunniteltujen ominaisuuksien toteuttamista. Luvussa 6 pohditaan prototyypisovelluksen onnistumista suhteessa suunniteltoon ja tavoitteiden täyttymistä.

2 TIEDONKERUU JA ANALYSOINTI

Tässä luvussa käsitellään prototyypisovelluksen alkutaivalta ideasta suunnitelmaksi ennen varsinaisen ohjelmoinnin aloittamista. Käsiteltävinä asioina on kohderyhmän merkitys ja kohderyhmän kyselytulosten analysointi ennen sovelluksen lopullisten ominaisuuksien päättämistä.

2.1 Sovelluksen kohderyhmä

Työn kohteena oleva sovellus on suunnattu ensisijaisesti esikoisen saaneille äideille. Joillakin äideillä saattaa olla tarvetta kirjata muistiin lapsensa toistuvia rutiineja sekä yksittäisiä suunnittelemattomia, spontaaneja tapahtumia (esim. lapsen sairastuminen) sekä oppimia asioita. Koska vanhemmuus on suuri muutos ihmisen elämässä aikaisempaan verrattuna, on täysin normaalia, että vanhempi kokee usein epävarmuutta omasta vanhemmuudestaan ja äitinä tai isänä toimimisesta. Arki pienen lapsen kanssa saattaa olla usein hyvin stressaavaa, jolloin muistamattomuus ja asioiden unohtelu ei ole poikkeuksellista. Parhaimmillaan sovelluksesta on apua arjen hahmottamisessa ja tapahtumien muistamisessa niiden kirjaamisen kautta, sekä se (sovellus) voi auttaa kohottamaan äidin itsevarmuutta rutiinien muodostumisen kautta. Sovellusta ei ole siis tarkoitus käyttää orjallisesti päivittäin vaan enemmänkin auttaa käyttäjää rutiinien muodostamisessa sekä muistamisessa jälkikäteen.

Ajoittain lastenhoitorutiinien seuranta voi olla tarpeen jo pelkästään neuvolakäyntien vuoksi. On hyvin mahdollista, että lapsi ei kasvakaan odotetulla tavalla, jolloin on syytä seurata esimerkiksi imetyksen tai syöttämisten tiheyttä ja kestoja, jotta varmistetaan lapsen riittävästä ravinnonsaannista. Tällöin muistiinpanot tehdään usein kynällä ja paperilla, mikä on useimmiten melko työlästä ja aikaavievää. Lisäksi papereilla on usein taipumusta hävitä vauva-arjen tohinoissa, jolloin muistiinpanojen eteen on nähty turhaa vaivaa ja täten neuvolassa joudutaan turvautumaan pelkän muistin varaan. Matkapuhelimesta, sen sijaan, on tullut ihmiselle varmasti yksi tärkeimmistä, ellei tärkein, arjen hyödykkeistä, eikä sitä hukata niin herkästi. Tänä päivänä matkapuhelin kulkee lähes jokaisen mukana miltei kellon ympäri ja paikasta riippumatta, ja enenevässä määrin älypuhelimet alkavat olla jo yleisempiä kuin peruspuhelimet. Hyvin monet ovat siirtyneet käyttämään puhelimen valmiita kalenteri- ja

muistiinpanosovelluksia, jolloin taskukalenterit ja muistiinpanovihkot ovat vastaavasti alkaneet vähitellen harvinaistumaan.

2.2 Internet-kysely kohderyhmälle

Idea sovellukseen tuli tämän työn tekijän vaimoltani – tuoreelta äidiltä – talvella vuosien 2012 – 2013 taitteessa. Vaikka sovellusta ei silloin alettu vielä suunnittelemaan, karkeat ominaisuudet olivat muodostuneet vaimon ideoiden pohjalta. Tämä ei yksinään riitä, kun suunnitellaan sovellusta suuremmalle käyttäjämäärälle kuin yhdelle, vaan tarvitaan jonkinasteista palautetta kohderyhmältä projektin kannattavuuden takaamiseksi. Sovellusta suunniteltaessa päätettiin täten käyttää hyväksi muiden äitien mielipiteitä ja ehdotuksia ominaisuuksien muokkaamiseksi mahdollisimman hyvin kaikkien tarpeita vastaaviksi, ja tämä päätettiin toteuttaa Internet-kyselyn kautta.

Kyselylomake muodostui yksinkertaisuuden vuoksi vain muutamasta kohdasta: kohderyhmän laajuuden selvittämisestä kysymyksestä, mitä rutiineja käyttäjä haluaisi olevan mahdollista merkitä sovelluksessa, millä alustalla käyttäjä mieluiten sovellusta käyttäisi sekä vapaasanakenttä muun palautteen ja mielipiteiden saamiseksi. Kyselyä tehdessä tarkoituksena oli pääasiassa kartoittaa sovelluksen tarpeellisuus muiden äitien keskuudessa. Työn tekijän kannalta toiseksi tärkein kohta oli selvittää, millä alustalla olisi toistaiseksi eniten käyttäjiä, eli mille alustalle sovellus ensisijaisesti suunniteltaisiin ja toteutettaisiin. Kaksi muuta kysymyskohtaa auttoivat valitsemaan tärkeimmät prototyypivaiheen ominaisuudet sekä hienosäätämään niitä käyttäjän tarpeisiin sopiviksi.

Kohderyhmälle tehtiin Google Drive –palvelussa helppokäyttöisellä ja ilmaisella Lomakkeet-työkalulla kyselylomake. Kyseisessä palvelussa lähetetyt vastaukset saa helppolukuisessa muodossa taulukkolaskentataulukkoon, sekä palvelu generoi tulosten pohjalta myös graafisen yhteenvedon sisältäen esimerkiksi diagrammeja ja tilastoja, lomakkeen luonteesta riippuen.

Jotta kysely olisi saanut mahdollisimman suuren vastaajakunnan, päätettiin se laittaa esille suosituimmille vauva-aiheisille Internet-sivuille sekä erilaisiin äitiryhmiin sosiaalista mediaa edustavalla Facebook-sivustolla. Pelkästään Suomen tasolla, mutta

myös kansainvälisesti tarkasteltuna erilaisia vauvasivustoja olisi ollut lukematon määrä, mutta ajan puutteen vuoksi tyydyttiin vain muutamaan.

Ensimmäisten vuorokausien aikaan vastauksia annettiin hyvää tahtia, yli puolet kaikista vastauksista saatiin tänä aikana. Vastausten tahti hiipui kuitenkin yllättävän nopeasti. Tämä selittynee kyseisten sivujen suurella käyttäjä- ja kommentointimäärällä, jolloin kyselyä varten laadittu keskustelu katosi nopeasti uusien keskusteluiden joukkoon. Vastausaikaa annettiin noin kuukausi, jona aikana vastauksia annettiin kaiken kaikkiaan tyydyttävä määrä, ja joiden pohjalta on hyvä lähteä sovellusta suunnittelemaan.

2.3 Kyselyn tulokset

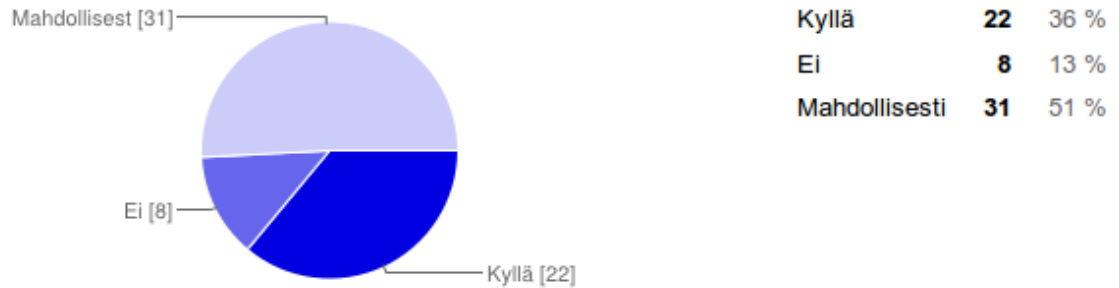
Vastausajan päättämisen jälkeen alkoi vastausten läpikäynti. Vastauksia käytiin etukäteen tutkimassa ennen vastausajan päättämistä, ja jo tällöin enemmistö vastaajista oli antanut positiivista palautetta sovelluksen puolesta. Eräät hieman kielteisemmät palautteenantajat olivat epäilevällä kannalla ja ihmettelivät, onko tällaiselle sovellukselle käyttöä. Kielteisten palautteiden kohdalla vaikutti siltä, että he olivat käsittäneet sovelluksen olevan siten sitova, että jatkossa pitäisi toimia täysin samoin kuin päivänä jolloin sovellusta käytti. Toiset ihmisistä ovat myös niin hetkessä eläviä, etteivät koe omakseen kirjata menneitä tapahtumia ylös/muistella menneitä. Suurin osa kuitenkin oli sitä mieltä, että sovellukselle löytyy käyttäjiä, ja eräs vastaaja kuvaili, että ”valmis sovellus olisi kätevä, tervetullut ja varmasti tätä päivää”.

Sovellusta puoltavilta vastaajilta tuli erittäin hyviä huomioita ja mielipiteitä ominaisuuksien suhteen. Koska tämä työ käsittelee vain prototyypisovellusta, ei kaikkia ominaisuuksia valitettavasti pystytty tekemään annetun ajan puitteissa, saatiika saattamaan sovelluksesta valmista tuotetta. Annettujen palautteiden perusteella valitaan muutama toteutettava toiminnallisuus ja keskitytään niiden tarkempaan suunnitteluun ja ohjelmointiin. Prototyypisovellukseen tulevia ominaisuuksien suunnittelua käsitellään tarkemmin seuraavassa kappaleessa.

Kyselyn vastasi kaiken kaikkiaan 61 äitiä, ja tulokset olivat erittäin selviä. Ensimmäisenä esitettävänä kysymyksenä oli, ajattelisivatko äidit sovelluksen olevan

heille hyödyllinen, jos äideillä olisi mahdollisuus käyttää sovellusta. Vastausten jakautuminen on esitetty alla olevasta kuvassa.

Jos tällainen sovellus olisi käytettävissäsi, luuletko siitä olevan Sinulle hyötyä?

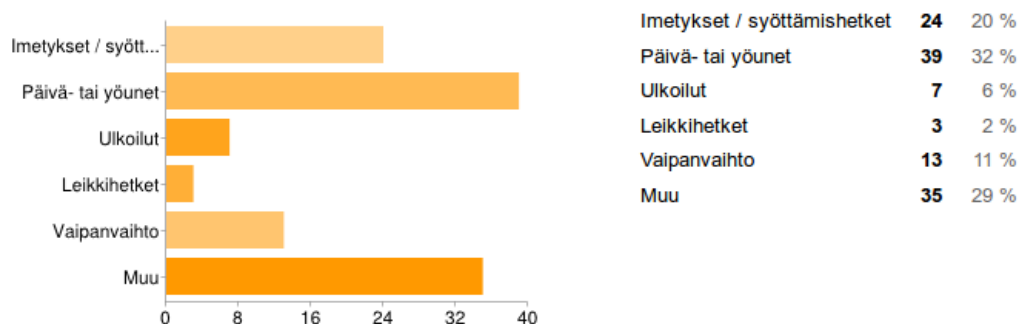


KUVA 1. Vastausten jakautuminen sovelluksen tarpeellisuudesta.

Kuten tuloksista näkyy, sovellusidean vastaanotto oli varsin positiivinen. Yhteensä 87 prosenttia vastaajista käyttäisi tai mahdollisesti käyttäisi sovellusta, mikä yllätti työn tekijän positiivisesti.

Kuten aiemmin on jo mainittu, tämän työn prototyypisovellukseen ei voida toteuttaa kaikkia suunniteltuja ja haluttuja ominaisuuksia, vaan on tyydyttävä parin kolmen toiminnallisuuden toteuttamiseen siten, että niiden toiminta olisi verrattavissa valmiin tuotteen toiminnallisuuksiin. Tässä tapauksessa toiminnallisuudella tarkoitetaan kirjattavaa rutiinia tai tapahtumaa. Prototyyppiin tulevat toiminnallisuudet valitaan puhtaasti kyselytulosten pohjalta. Alla on esitetty vastausten jakautuminen toivottujen, kirjattavien rutiinien välille.

Onko tai oliko Sinulla rutiineja, joita olisit halunnut seurata tai muistaa paremmin lapsesi vauva-aikana?



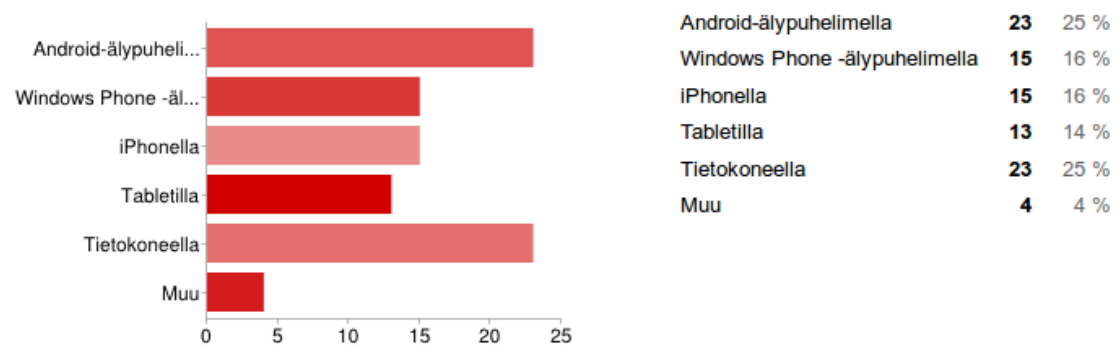
KUVA 2. Vastausten jakautuminen toivottujen rutiinien välillä.

Tämän kohdan vastauksista huomataan hyvin, mitä rutiineja äidit ovat eniten halukkaita merkitsemään sovelluksessa. Valmiiksi annetuista vaihtoehdoista selvästi enemmistön

vastauksista saivat syöttöjen ja lapsen unien seuraaminen. Kaikista eniten vastauksia annettiin kuitenkin Muu-kohtaan, jossa vastaajat antoivat ns. lisäominaisuuksia valmiisiin vaihtoehtoihin tai kokonaan uusia rutiineja seurattavaksi. Suunnitteluvaiheen alkaessa työn tekijä valitsee parhaaksi näkemänsä rutiinin yhdeksi toiminnallisuudeksi valmiiksi määriteltyjen lisäksi. Sovellukseen toteutettavat tapahtumat määrittellään tarkemmin kappaleessa Tiivistetty vaatimusmäärittely.

Tämän työn kannalta ehkä merkittävin kysymys koski sitä, millä alustalla äidit haluavat sovellusta käyttää. Tämänhetkisen älypuhelimien markkinaosuuksien huomioonottaen tulokset eivät olleet yllättäviä, päin vastoin lopputulosta osattiin odottaa. Alustakysymyksen tulokset alla.

Millä alustalla käyttäisit sovellusta mieluiten?



KUVA 3. Vastausten jakautuminen alustojen kesken.

Kuten tämän kysymyksen tuloksista nähdään, mobiililaitteissa käytetyimpänä ovat Android-puhelimet. Odotetusti iPhonea on äänestetty toiseksi eniten, mutta hieman yllättäen Windows Phone on saanut myös verrattain paljon vastauksia, ja on saanut yhtä paljon kannatusta kuin iPhone. Windows Phonen kannatus selittynee sillä, että kysely on tehty pelkästään suomalaisilla sivustoilla, ja viime aikoina Nokian Lumia-sarjan yleistymisen, hintojen alenemisen ja halvempien puhelinten julkistamisen myötä suomalaiset käyttäjät ovat löytäneet tiensä Windows Phonen luokse.

Yllättävimpänä seikkana alustoja koskien on tietokoneen suosio muiden laitteiden rinnalla. Selvästi suurempi osa äideistä käyttäisi rutiinipäiväkirjaa mieluiten tietokoneella kuin tablet-tietokoneella. Tähän huomioon ei saatu vastausta kyselyn kautta, ja sitä voidaan vain spekuloida, miksi niin suuri osa haluaisi käyttää sovellusta tietokoneella mobiililaitteen sijasta tai sen rinnalla.

2.4 Tiivistetty vaatimusmäärittely

Virallisissa, suurissa ohjelmistoprojekteissa käytettyjen vaatimusmäärittelyjen ollessa hyvin yksityiskohtaisia, formaaleja ja eläviä vaatimusten muuttuessa tai päivittyessä, tämän tiivistetyn ja yksinkertaistetun vaatimusmäärittelyn tarkoituksena on kattaa vain prototyypisovelluksen ominaisuuksien sekä sen vaatimusten määrittelyt. Tästä vaatimusmäärittelystä on siis jätetty pois kaikki, mitä ei tarvita prototyypin suunnitteluun.

2.4.1 Käyttöjärjestelmä

Uusissa, jälleenmyyjiltä saatavissa Android-laitteissa on pääosin joko Ice Cream Sandwich- tai Jelly Bean-koodinimellä varustettu käyttöjärjestelmäversio. Silti käytössä olevissa rekisteröidyissä Android-laitteissa elokuun 1. päivään 2013 mennessä 33 prosentissa laitteista oli käytössä Gingerbread-käyttöjärjestelmäversio (Android Developers: Dashboards). Yleisyytensä vuoksi tätä versiota pyritään myös tukemaan mahdollisimman pitkälle, joten laitteessa vaaditaan olevan vähintään Gingerbreadin 2.3-versio (Android API 10).

2.4.2 Rutiinit

Tässä kappaleessa on hieman laajennetusti ensin kerrottu jokaisen rutiinin ”taustaa”, jota seuraa tiivis analyysi määritellyistä rutiinista.

1. Lapsen ruokailut

Sana ”ruokailu” voi olla joko imettämistä, maidon antamista pullosta tai kiinteän ruuan syöttämistä, ja näillä on hieman erilaiset mitattavat ominaisuudet keskenään vaikka peruskäsite on sama.

Imetyksissä on suuria eroja äitien välillä, sillä joillain äideillä maidon tulo voi olla niukkaa, jolloin lapsi yleensä syö kauemmin ja tarvittaessa molemmista rinnoista, ja

vastaavasti joillain voi olla maidon tulo hyvinkin runsasta, jolloin lapsi tulee nopeammin kylläiseksi ja imetyksen kesto on melko lyhyt. Imettäessä on miltei mahdoton mitata määrällisesti, kuinka paljon lapsi on saanut maitoa, jolloin ainoa konkreettinen mittakeino imettäessä on aika ja lapsen syöntiaktiivisuus.

Imettäviä – varsinkin runsaasti maitoa tuottavia – äitejä kehoitetaan pumppaamaan maitoa imetysten välissä rintatulehduksien estämiseksi ja maidon tuotannon tasaamiseksi. Maitoa pullosta annettaessa on helppo seurata maidon kulumista millilitroina, jolloin aika ole niin tärkeässä asemassa.

Koska lapsen ravintona ensimmäisten kuukausien ajan on maito, ei kiinteiden ruokien merkitseminen ole ajankohtaista vastasyntyneen lapsen äidille. Tämä on kuitenkin hyödyllinen rutiini seurattavaksi, sillä kiinteitä ruokia aloittaessa määrät ovat pieniä, ja niitä kasvatetaan vähitellen. Allergioidenkin takia äitien on hyvä merkitä lapsen reaktioita tiettyihin ruoka-aineisiin.

Tämän analysoinnin pohjalta työn tekijä päätti jättää kiinteiden ruokien syöttämisen omaksi rutiinikseen, ja imetykset ja pullomaidon antamisen yhdeksi yhdistetyksi rutiiniksi, eli karkeasti jaettuna kiinteisiin ja nestemäisiin ruokiin. Tämä ratkaisu yksinkertaistaa rutiinien tarkempaa suunnittelua, kun kaikki syöttötavat eivät ole saman rutiinin alla.

2. Lapsen nukkumiset

Vauvaksi luettava lapsi nukkuu yleensä elämänsä ensimmäisen puolen vuoden aikana suurimman osan vuorokaudesta ja useita kertoja päivän aikana. Tämä ei tietenkään päde absoluuttisesti samalla tavalla kaikkiin vauvoihin, vaan jokaisella vauvalla ja yksilöllä on oma rytmensä.

Toisena kyselyn perusteella toivottuna rutiinina oli lapsen nukkumisten seuraaminen. Lapsen uni voi toisinaan olla hyvinkin katkonaista syystä tai toisesta ja toisinaan taas voi uni maittaa koko yön läpi, ja vanhemman voi olla hyödyllistä seurata unen kestoa ja muistiinpanojen avulla lisätä yksityiskohtaisemmin mahdolliset tapahtumat heräilyjen välissä. Sovelluksen avulla olisi siten helppo merkitä lapsen unien alkamis- ja loppumisajankohdat, jolloin unien yleinen kesto on yksinkertainen määrittää, sekä

käyttäjälle annetaan mahdollisuus kertoa yksityiskohtaisemmin erillisen tekstikentän avulla.

Lapsen unien seuranta pidetään yhtenä rutiinina vaikka unet voitaisiin jakaa omiksi rutiineiksi, päivä- ja yöuniksi. Tämä jako on kuitenkin yksinkertainen toteuttaa yhdenkin rutiinin sisällä, eikä niitä ole tarpeen jakaa. Tämä rutiini siis käsittää päivä- tai yöunien merkitsemisen, aloitus- ja lopetusajan ylösottamisen sekä mahdollisuuden kirjoittaa lisähuomioita unista.

3. Vaipan vaihdot

Yksi olennainen tärkeä osa vauvan tarpeiden täyttämässä on huolehtia säännöllisestä vaipanvaihdosta. Vauvan tarpeiden huolehtimisen lisäksi vaipanvaihtovälien seuraamisesta voi olla hyötyä, jos halutaan seurata, saako lapsi tarpeeksi ruokaa tai nesteitä, jos lapsi ulostaa normaalia tiheämmin tai ulosteen laadussa tapahtuu merkittävä muutos.

Kyselytulosten perusteella kolmanneksi äänestetyin rutiini oli siis vaipanvaihdot. Tämä rutiini on ohjelmistoteknisesti aika yksinkertainen rakenteeltaan, sillä rutiinissa tarvitsee seurata vain milloin vaippa on vaihdettu ja käyttäjä voi tekstikentässä kirjoittaa lisätietoja.

4. Käyttäjän itse luoma rutiini

Koska tämän työn osalta kaikkiin kohderyhmän tarpeisiin ja toiveisiin ei pystytä vastaamaan, päätettiin harjoituksen vuoksi toteuttaa ”tyhjä” rutiini, jolle käyttäjä voi itse määrittää otsikon sekä sen, onko rutiini pitkäkestoinen (yli 5 minuuttia) vai kertaluontoinen. Jokaisessa ”tyhjässä” rutiinissa on valmiina tekstikenttä muistiinpanoja varten. Käyttäjä voi halutessaan luoda niin monta rutiinia valmiiden lisäksi kuin keksii.

Työn tekijän kannalta huomioitava seikka tyhjän rutiinin käytön kanssa on se, että mikäli käyttäjä luo yhdenkin oman rutiinin, täytyy rutiinisymbolit sisältävän listan laajentua uusien rutiinien myötä. Mikäli tämä jää huomiotta, tullaan luoneeksi uusi rutiini vanhan ”päälle” siten, että vanha rutiini jää sovelluksen muistiin mutta käyttämättömäksi.

2.4.3 Käyttöliittymä ja käyttäjälle näkyvä rakenne

Sovelluksen käyttöliittymä rakentuu karkeasti kahteen näkymään, rutiini- ja menneet rutiinit -näkymään. Nämä näkymät sijaitsevat omilla välilehdillään, joiden välillä navigoidaan joko välilehti-ikonia painamalla tai mahdollisesti sivuttaissuuntaisella sormen liikkeellä.

Rutiininäkymässä on esitetty kunkin rutiinin käynnistävä painike omalla symbolillaan. Tässä näkymässä on neljä vakiopainiketta, joihin käyttäjä ei voi vaikuttaa, mutta näiden lisäksi on Lisää-painike (Plus-symboli), jonka avulla käyttäjä voi lisätä rutiinilistaan haluamansa määrän omiin tarpeisiinsa soveltuvia rutiineja. Rutiinilista päivittyy dynaamisesti rutiineja lisätessä. Rutiinipainikkeiden näkymisen oletuksena on, että käyttäjä on lisännyt sovellukseen vähintään yhden lapsen, jonka tietoihin tehtävät rutiinit tallennetaan.

Menneet rutiinit -näkymässä on esitetty edelliset rutiinit listana allekkain, uusin ensin. Aina rutiinin päätyttyä lista päivittyy edellisellä tehdyllä rutiinilla. Tämän näkymän kautta on myös mahdollista muokata aiempaa rutiinia tarkemmilla muistiinpanoilla.

Puhelimen Menu-painiketta painamalla avautuu pikavalikko. Tässä valikossa ovat toiminnot lapsen lisäämistä ja sovelluksen asetusten määrittämistä varten.

2.4.4 Tietojen tallennus

Lapsen tiedot sekä rutiinimerkinnät tullaan tallentamaan XML-tiedostotyyppiä käyttäen. Tähän päädyttiin tiedostotyyppin yksinkertaisuuden vuoksi, mutta myös siksi, että sovelluksen käyttöikä yhtä lasta kohden jää todennäköisesti lyhyeksi, arviolta yhden kahden vuoden mittaiseksi. Täten tietoa ei ehdi kertyä niin paljoa, että olisi perusteltua vaihtaa tiedontallennustapaa tietokantapohjaiseen ratkaisuun.

Tallennuskohteeksi valittiin Android-laitteen Internal Storage (tallennus sisäiseen muistiin) sen turvallisuuden vuoksi. Android-laitteissa sovelluksille on mahdollista määrittää tietojen tallennus yksityisinä tiedostoina sisäiseen muistiin, jolloin laitteen muut sovellukset tai käyttäjäkään ei pääse käsittelemään näitä tiedostoja. Kun

kyseessäoleva sovellus poistetaan, luodut tiedostot poistuvat sovelluksen mukana (Android Developers: Storage Options).

3 KOHDEALUSTA JA KÄYTETTÄVÄ OHJELMISTOKEHYS

3.1 Android-käyttöjärjestelmä

Android-käyttöjärjestelmän on alunperin kehittänyt Android Inc –startup-yritys (perustettu 2003). Google tuki yritystä ensin taloudellisesti mutta osti kuitenkin yrityksen tuotteineen vuonna 2005. (Business Week: Google Buys Android for Its Mobile Arsenal)

Nykyään Androidin kehityksestä vastaa Open Handset Alliance. Open Handset Alliance on vuonna 2007 perustettu konsortio, joka julkistettiin Androidin kanssa samaan aikaan. Se koostuu 84:stä maailman suurimmasta laitteisto- ja ohjelmistovalmistajista sekä teleoperaattoreista, esimerkkeinä laitevalmistajista mainittakoon Acer ja HTC. (Open Handset Alliance: Members, Open Handset Alliance: FAQ)

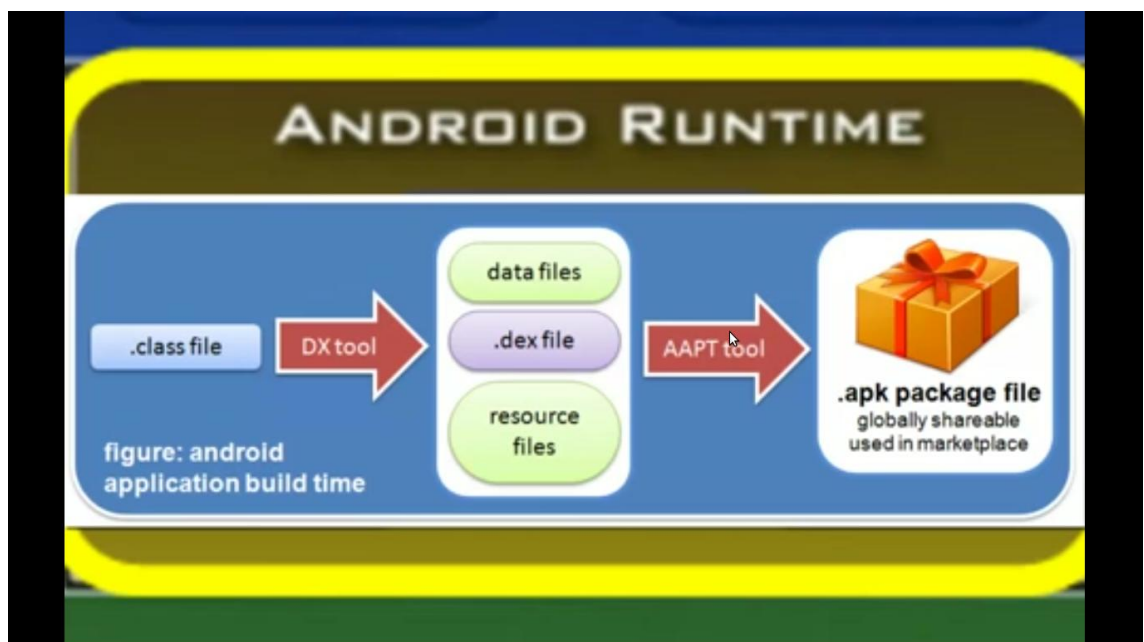
3.2 Android-ohjelmistopino

Android on ohjelmistopino, joku koostuu karkeasti käyttöjärjestelmästä, väliohjelmistoista ja käyttäjän perusohjelmista. Ohjelmistopinon alimpana kerroksena on Linux-käyttöjärjestelmäytimen (versio 2.60) pohjalta ohjelmoitu kernel, joka sisältää muistinhallintaohjelmia, turvallisuusasetuksia, virranhallintaohjelmistoja sekä useita laiteajureita. (Wikipedia: Android, HowStuffWorks: How the Google Phones Works)

Seuraavana kerroksena ohjelmistopinossa ovat Android-kirjastot. Kirjastot ovat kokoelma komentoja, joilla ohjataan laitteen erityyppistä tietoa. Esimerkiksi Media Framework –kirjasto sisältää tuen äänentoistolle sekä kuva- ja videoformaateille, ja SGL-grafiikkakirjasto (Skia Graphics Library) käytetään grafiikan välittämiseen ja ohjaamiseen Android-laitteen näytölle. Tässä samassa kerroksessa sijaitsevat myös Android Runtime layer (sisältää kokoelman Java-kirjastoja, joita sovelluskehittäjät käyttävät ohjelmoidessaan Android-sovelluksia), sekä Dalvik-virtuaalikone. Dalvik-virtuaalikonetta käytetään Android-sovellusten ajamiseen, toisin sanoen jokainen Android-sovellus on oma prosessinsa, jota ajetaan omassa virtuaalikoneessaan. Tällainen sovellusten ajaminen mahdollistaa sovellusten turvallisen ajamisen toisistaan

riippumatta, ja mikäli jokin sovellus kaatuu tai lakkaa toimimasta, tämän ei pitäisi vaikuttaa muiden sovellusten toimintaan. (HowStuffWorks: How the Google Phones Works, Wikipedia: Android software development)

Kuten normaalissa Java-virtuaalikoneessa, myös Dalvikissa suoritettava koodi on tavukoodia. Java-virtuaalikoneessa ajetaan .class-tiedostoista käännettyä tavukoodia. Dalvik-virtuaalikoneen tapauksessa ajettava ohjelmakoodi on .dex-tavukoodia (Dalvik Executable), joka on Javan tavukoodi käännettynä uudestaan .dex-tavukoodiksi, jota Dalvik ymmärtää. Dalvik kirjoitettiin alusta asti Android-käyttöjärjestelmää ja mobiililaitteita silmällä pitäen, mistä syystä .dex-tavukoodi on pitkälle optimoitu esimerkiksi mahdollisimman pienen muistinkäytön osalta. Alla on esitetty kuva .dex-tavukoodin kääntämisestä. (Electronics Weekly: What is the Dalvik virtual machine?, Android Central: Android A to Z: What is Dalvik?)

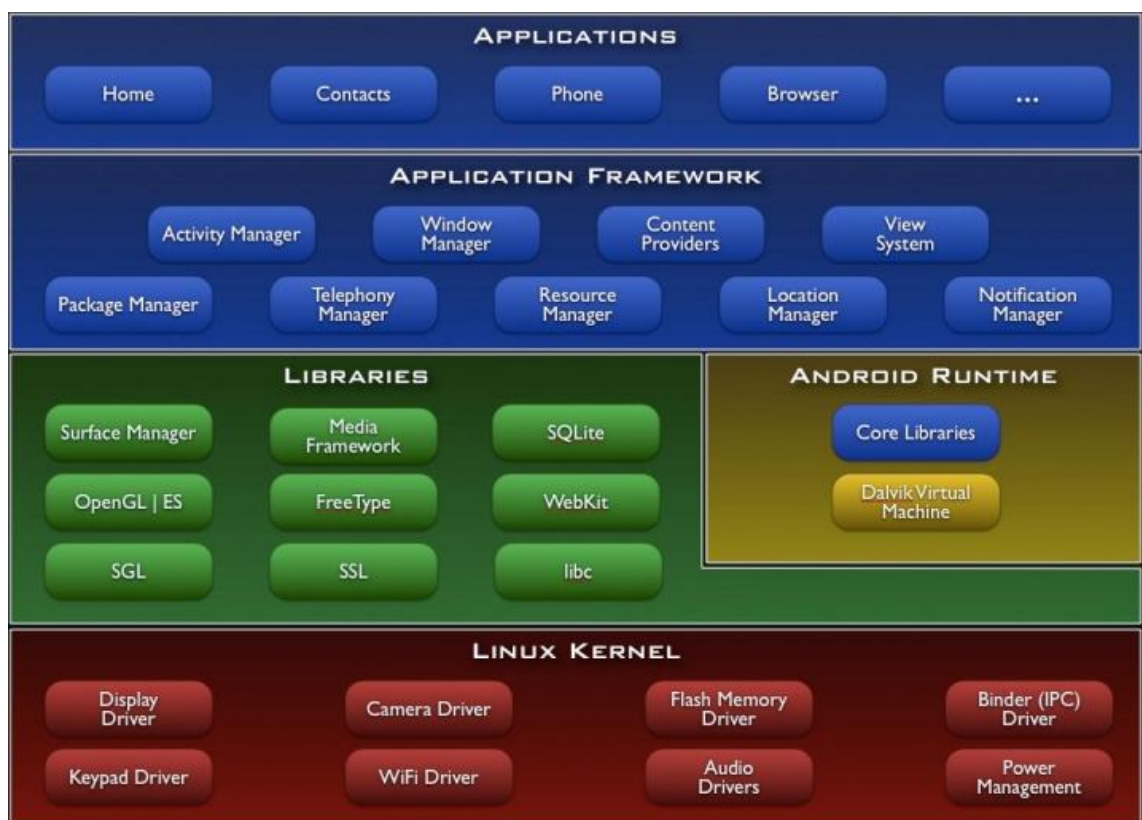


KUVA 4. Java-lähdekoodin kääntämisprosessi .dex-tiedostoksi (Catazine: Dalvik Virtual Machine)

Toiseksi ylimpänä kerroksena pinossa on sovelluskehyskerros. Kerros sisältää Android-laitteen perustoimintoja hallitsevat ohjelmat, kuten puheluiden soittamisen hoitavat ohjelmat tai resurssien varaamisen ohjelmat. Sovelluskehittäjillä on suora pääsy tähän kerrokseen, mikä mahdollistaa laitteen suorituskyvyn lähes täydellisen hyödyntämisen sovelluksia ohjelmoitaessa. Sovelluskehyskerrosta voidaan ajatella eräänlaisena

työkalulaatikkona, joka sisältää kaikki perustyökalut, joiden avulla sovelluskehittäjät voivat kehittää monimutkaisiakin ohjelmia.

Pinon ensimmäisenä, eli päällimmäisenä, kerroksena ovat laitteen varsinaiset sovellukset (sekä laitteen mukana tulevat valmistajan räätälöimät sovellukset että Google Play –sovelluskaupasta ladattavat). Tämän kerroksen kautta ohjataan laitteen toimintoja ja tapahtumia, se on käytännössä siis laitteen käyttöliittymä. Tavallinen keskiverto puhelimenkäyttäjä ei pääse käsiksi tätä kerrosta alempiin kerroksiin, vaan ne ovat sovelluskehittäjien ja laitevalmistajien saatavilla. Alla on esitetty Android-ohjelmistopino kokonaisuudessaan.



KUVA 5. Android-ohjelmistopino. (Embedded Linux Wiki: Android Architecture)

3.3 Android-pohjaiset laitteet

Maailman ensimmäinen Android-puhelin julkaistiin lokakuussa 2008, ja se tunnetaan joko HTC Dreamina tai T-Mobile G1 –nimellä markkina-alueesta riippuen. Laite käyttää Androidin 1.6-versiota. Tämän päivän Android-puhelimet sisältävät paljon enemmän laskentatehoa ja massamuistia, mutta Dream sisälsi kuitenkin kaiken

välttämättömän ollakseen täysiverinen älypuhelin. Laite tukee kaikkia siihen aikaan yleisimpiä tietoverkkoja (GPRS, EDGE, 3G, WLAN) ja se sisälsi tavallisimmat älypuhelimien käyttämät sensorit (kiihtyvyyssanturi, kompassi, GPS). Useissa ensimmäisissä Android-puhelimeissa oli esiinliu'utettava fyysinen QWERTY-näppäimistö, mutta nykyään tämän esiintyminen on harvinaista. (GSM Arena: HTC Dream, CNET: A brief history of Android phones)

Tarkkaa myyntimäärää ei ole tiedossa, mutta kaiken kaikkiaan laitetta myytiin ennakkoon 1,5 miljoonaa kappaletta ja pelkästään Yhdysvalloissa miljoonan myydyn laitteen raja ohitettiin noin puolessa vuodessa. (Android Central: T-Mobile G1 hits a million units sold, Softpedia: T-Mobile's G1 counts 1.5 million pre-sold units)

Ensimmäisen Android-puhelimen ilmestymisen jälkeen kului lähes vuosi, ennen kuin julkaistiin uusi laite, mutta toisen laitteen julkaisun jälkeen julkaisutahti lähti rajuun nousuun. Eniten laitteita julkaisivat HTC, Motorola sekä Samsung (suurin osa on laitteita, joita ei tullut Suomen markkinoille), ja tällä hetkellä Samsung on ylivoimaisesti suurin ja menestyksekkäin Android-puhelinten valmistaja. Laitteiden tekninen kehitys on luonnollisesti ollut nopeaa, kuten tietokoneiden kehitys on ollut. Ensimmäiset laitteet eivät jaksaneet pyörittää kovin raskasta grafiikkaa, mutta laitteiden ja valmistajien yleistyessä sekä varsinkin mobiilipelialan herätessä Android-laitteiden mahdollisuuksiin, puhelimien graafinen suorituskyky ja laskentateho on parantunut huomattavasti. (Android Headlines: Top 10 Best Android Manufacturers of 2012, CNET: A brief history of Android phones)

Kuten olettaa voidaan, sekä Android-laitteiden että itse käyttöjärjestelmän kehityksen kanssa on tultu huimia askeleita kuluneen viiden vuoden aikana. Edistystä on tapahtunut laitteiston puolella paljon, mutta käyttäjälle näkyvämpi ero huomataan käyttöjärjestelmän ulkoasussa. Alla on esitetty kuvat Androidin versioista 1.6 ja 4.0.



KUVA 6. Android 1.6 aloitussivu. (Things about stuff: Android home screen)



KUVA 7. Android 4.0 aloitussivu. (Wikipedia: Android version history)

Android-laitteet eivät kuitenkaan rajoitu enää ainoastaan puhelimiin. Toinen merkittävä Android-puhelimien rinnalle tullut laite on taulu- eli tablet-tietokone. Ensimmäisen nykyaikaisen kosketusnäytöllä varustetun taulutietokoneen, iPadin, julkaisi Apple vuonna 2010. Vuoden 2011 helmikuussa julkaistiin ensimmäinen Android-tablet (Motorola Xoom), jonka käyttöjärjestelmänä oli Android 3.1 (Honeycomb). Tässä Android-versiossa oli uusi ja nykyaikainen käyttöliittymä, ja huolimatta laitteen suhteellisen hyvästä suorituskyvystä, oli laite erittäin buginen ja hidas. Melko pian tämän jälkeen Google ja Android-laitteiden valmistajat aloittivat yhteistyönä

kehittämään Android 4.0 -versiota, jossa oli yhdistetty versioiden 2.3 ja 3.1 parhaat ominaisuudet, ja ensimmäinen Android Jelly Bean –pohjainen laite (Google Nexus 7) julkaistiin kesällä 2012. Tämän jälkeen myös muut valmistajat ovat tuoneet markkinoille varsin eri laatuista ja hintaisia Jelly Bean –tabletteja. Android-tabletit onnistuivat lyömään iPadin sen omilla markkinoilla varsinkin edullisella hinnalla mutta myös laitteiden monimuotoisella kirjolla. (ReadWrite: How the iPad stalled and Android took over the tablet market)

3.4 Kehitysympäristö

Pari kolme vuosikymmentä taaksepäin, kun ei ollut olemassa vielä kovin kehittyneitä sovelluskehitysympäristöjä, ohjelmointiprosessi tapahtui useimmiten joko puhtaasti komentoriviltä, tai tekstieditorin ja komentorivin avulla. Ohjelmointi oli tällä tavalla tehtynä hyvin aikaa vievää ja virhealtista. Tietokoneiden ja tietotaidon kehittyessä myös ohjelmistokehityksessä ja sen työkalujen kehittämisessä on menty valovuosia eteenpäin, ja tämän päivän ohjelmistokehitysympäristöt ovatkin hyvin monipuolisia ja - joissain tapauksissa - monimutkaisia.

Kun ohjelmistokehittäjä ohjelmoi jollekin valitsemalleen kohdealustalle sovellusta, käyttää tämä työhön pääsääntöisesti jonkinlaista ohjelmistokehityspakettia, software development kittiä (SDK). Näin on myös Androidin kohdalla; Android-sovelluksia ohjelmoidakseen kehittäjä tarvitsee Android-ohjelmointia varten tehdyn SDK:n. SDK on usean ohjelmistokehityksessä tarvittavan työkalun paketti. Kaksi todennäköisesti tärkeintä työkalua kyseisessä paketissa ovat ohjelmointirajapinta (Application programming interface, API) sekä integroitu kehitysympäristö (Integrated development environment, IDE). Sen hetkiseen tarkoitukseen suunnattu API sisältää ohjelmoijan tarvitsemat komennot sovelluksen luomiseen, oli kyseessä sitten käyttöliittymän luominen tai tietokannan ominaisuuksien toteutus. IDE on ohjelmoijan konkreettinen väline ohjelmiston tekemiseksi, ja se sisältää esimerkiksi seuraavat ominaisuudet: editori-ikkunan, johon ohjelmistokoodi kirjoitetaan, ohjelmistoprojektin puunäkymän sekä toiminnot debuggerin (virheenetsijän) ja kääntäjän käyttämiseksi.

Jotta Android-ohjelmointi voidaan aloittaa, tarvitaan ensiksi siihen käytettävät työkalut. Androidin sivuilta on ladattavissa ADT Bundle –niminen pakattu tiedosto, joka sisältää

Android SDK:n. Ohjelmoijan työtä helpottaakseen Android SDK sisältää Android-ohjelmointia varten esiasetetun Eclipse IDE:n. Google yhteistyökumppaneineen on nähnyt vaivaa rakentaakseen Eclipseen monipuoliset hallinta-, debuggaus- ja monitorointityökalut. SDK Manager ja Virtual Device Manager ovat näistä Android-ohjelmoinnin alussa tärkeimmät. SDK Managerin avulla hallitaan tarvittavia API-versioita, emulaattoreiden image-tiedostoja sekä muita Googlen sovelluskehitykseen tarjoamia apuvälineitä. Virtual Device Managerilla luodaan ja muokataan kuhunkin tarpeeseen soveltuvia Android-emulaattoreita. Emulaattoreita käsitellään lisää tulevassa kappaleessa.

Android-sovelluksia tarvitsee myös testata kehityksen yhteydessä, jotta nähdään, että toteutettu ominaisuus toimii suunnitellusti ja että nähtäisiin virheen tapahtuessa, mikä napin painallus tai tapahtuma aiheutti virheen. Sovelluksia voidaan ajaa kahdella tavalla: joko suoraan USB-yhteydessä olevalla Android-laitteella tai Android-emulaattorilla. Emulaattori on tietokoneohjelmisto, joka jäljittelee alkuperäisen laitteen toimintaa mahdollisimman hyvin. Emulaattorit vaativat yleisesti ottaen paljon muistia sekä prosessoritehoa pyöriäkseen siedettävästi.

3.5 Google Play –sovelluskauppa

Kuten muillakin älypuhelinvalmistajille, myös Androidilla on oma kauppapaikkansa, josta käyttäjät voivat ladata sovelluksia: Play Store (entinen Android Market).

Aiemmin Play Storen tarjonta rajoittui pelkästään erilaisiin sovelluksiin ja peleihin. Applen tuotteita käyttävillä on ollut mahdollisuus nauttia ladattavasta musiikista iTunes-ohjelman kautta jo vuosien ajan, kun taas Androidia käyttävillä on jouduttu tyytymään kolmannen osapuolen musiikkiohjelmistoihin, kuten Spotifyihin. Nykyään kuitenkin Android-käyttäjilläkin on mahdollisuus ladata ja ostaa laitteisiinsa kauppapaikan kautta myös e-kirjoja ja musiikkia.

Applen iPhoneen ja iPadin sovellustarjonnan ohella Android-laitteille on hämmentävän paljon sovelluksia tarjolla. Tämän kappaleen kirjoittamishetkellä Play Storessa oli hieman alle 880 000 sovellusta kun kilpailevalla puolella rikkoutui miljoonan

sovelluksen raja. (AppBrain: Number of available Android apps, About.com: How many apps are in the iPhone App Store)

Viime kuukausina on noussut Androidista puhuttaessa sen sovelluskaupan sisältämien haittaohjelmien määrä sekä itse Android-järjestelmän turvallisuus. Toisin kuin Apple ja Microsoft, Google mahdollistaa sovellusten lataamisen ja asentamisen muistakin lähteistä kuin virallisesta kauppapaikasta, mikä tekee haittaohjelmien tahattoman lataamisen todennäköisemmäksi. Apple ja Microsoft suorittavat kauppapaikkoihinsa ladattaville sovelluksille tiukat tarkastusrutiinit, ja vasta tarkistukset läpäistyään sovellukset päätyvät käyttäjien ladattaviksi. Googlellakin on tätä nykyä oma tarkistusohjelmansa, pilvipalvelu nimeltään Bouncer. Bouncer paljastettiin 2. helmikuuta 2012, mutta se on ollut toiminnassa ainakin vuoden 2011 aikana, ja Googlen mukaan tuona aikana se on onnistunut vähentämään haitallisten ohjelmien määrää 40 %:lla. Joidenkin lähteiden mukaan Google olisi kehittämässä Androidiin sisäänrakennettua haittaohjelmien tarkistajaa, mutta Google ei ole varmistanut tätä tietoa. Käyttäjillä on kuitenkin mahdollisuus ladata Play Storesta tunnetuimpien antivirusohjelmien kehittäjien tekemiä mobiilikäännöksiä haittaohjelmien estämiseksi. (Google Mobile Blog: Android and Security, About.com: Is Android safe?)

4 PROTOTYYPIN TEKNINEN SUUNNITTELU

Tässä kappaleessa/luvussa käsitellään Tiivistetty vaatimusmäärittely –kappaleessa esiintulleita ominaisuuksia ja niiden suunnittelua. On hyvä huomioida, että tässä kappaleessa käsitellyt suunnitelmat eivät välttämättä päädy prototyyppiin sellaisinaan, vaan ovat nimensä mukaisesti vain suunnitelmia, joten käytännön toteutus voi poiketa suunnitellusta.

4.1 Tavoitteena helppokäyttöisyys

Kuten Tiedonkeruu ja analysointi –luvussa painotettiin, sovelluksen on oltava yksinkertainen, helposti käytettävä ja nopeasti omaksuttava tätä kohderyhmää ajatellen. Äideillä ei ole aikaa etsiä sovelluksesta monen painalluksen kautta oikeaa toimintoa, vaan tarvittavan on oltava saatavilla mahdollisimman helposti. Tästä syystä sovelluksen käyttöliittymä on pyritty pitämään niin riisuttuna kuin mahdollista, jotta käyttäjälle näkyisi vain kaikista oleellisiin tieto sekä halutut toiminnot.

Sovelluksen käynnistyttyä avautuu päänäkymä sisältäen välilehtiwidgetin näkymien välillä navigointia varten, pudotusvalikko lapsen valintaa varten sekä lista sovelluksen sisältämistä valmiista ja käyttäjän määrittelemistä rutiineista. Laitteen Menu-nappia painettaessa ilmestyy pikavalikko, jossa on toiminnot lapsen lisäämistä sekä asetusten määrittämistä varten. Sovelluksen toisella välilehdellä on lista menneistä rutiineista, sisältäen tiedot rutiinin ajankohdasta ja kestosta. Jokaisen menneen rutiinin kohdalla on mahdollista lisätä rutiinia tarkentava muistiinpano.

Koska sovelluksessa käytetään tiedostoa tietojen tallennusta varten hyvinkin aktiivisesti, tämä voi syödä puhelimen resursseja, mikäli rutiineja lisätään paljon päivän aikana. Siksi tietojen tallennus on pyritty tekemään siten, ettei se hidastaisi sovelluksen käyttöä merkittävästi.

4.2 Laite-erojen huomiointi

Android-laitteiden – varsinkin näyttötarkkuuksien – suurissa eroissa sovelluskehityksen ongelmaksi voi nousta joustavan käyttöliittymän teko. Esimerkiksi pienelle näytölle

sovitetty käyttöliittymä voi näyttää hyvinkin rumalta suuremmalla näytöllä, tai olla epäkäytännöllisempi käyttöliittymäkomponenttien asettelultaan. Jos sovellusta ohjelmoitaisiin tablet-laitteelle (prototyypisovellus ei kata tablet-laitetta tässä vaiheessa), jouduttaisiin sillekin mahdollisesti toteuttamaan omanlaisensa käyttöliittymä, kun puhelimelle suunniteltu käyttöliittymä ei välttämättä sopisi tablet-laitteen näyttökoolle ollenkaan.

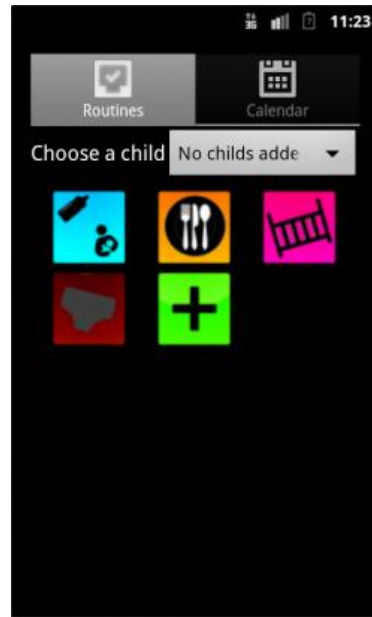
Mobiililaitteiden suorituskyvyssä on myös suuria eroja, eikä Android-puhelimet ole poikkeus. Ajettaessa keskiraskasta ohjelmaa low-end- ja high-end-laitteilla, laitteiden välinen suorituskykyero kyseessäolevalla sovelluksella voi olla merkittävä. Siksi sovellusta suunniteltaessa on pyrittävä huomioimaan ja välttämään mahdollisimman hyvin pullonkaulojen muodostumista. Kuten edellisessä kappaleessa mainittiin, tiedostonluku ja –kirjoitus voi olla hyvinkin raskas toimenpide, ja sen vaivattomaan toimintaan tulisi panostaa. Yleensä suositaan raskaiden tai pidempikestoisten metodien ajamista omissa säikeissään, ja tämän sovelluksen tapauksessa luultavasti päädytään tähän ratkaisuun.

4.3 Käyttöliittymä ja näkymät

Käyttöliittymän saumaton toiminta ja käyttöliittymäelementtien järkevä käyttö on kriittinen tekijä minkä tahansa mobiili- tai tietokonesovelluksen miellyttävän käyttökokemuksen aikaansaamisessa. Sovelluksesta tulee välittömästi epämiellyttävä käyttää, jos sovelluksen ruudulla on liikaa tavaraa tai näytettävä tieto ei hyödytä käyttäjää millään tavalla. Tässä työssä koetettiin panostaa erityisesti käyttöliittymän informatiivisuuteen ja yksinkertaisuuteen, eikä häiritäisi tätä työtä koskevan kohderyhmän käyttökokemusta millään ylimääräisellä tiedolla.

Tämän työn käyttöliittymää suunniteltaessa tehtiin lukuisia vedoksia paperille helpomman työskentelyn takia, joten niitä ei nähdä tässä dokumentissa. Kun paperiversiossa oleviin suunnitelmiin oltiin tyytyväisiä, tehtiin niiden pohjalta Androidin IDE:n avulla lopulliset käyttöliittymät. Seuraavaksi esitetään tätä sovellusta varten toteutetut käyttöliittymänäkymät.

1. Rutiininäkymä




KUVA 8. Rutiininäkymä.

2. Menneet rutiinit –näkyvä



KUVA 9. Menneet rutiinit –näkyvä.

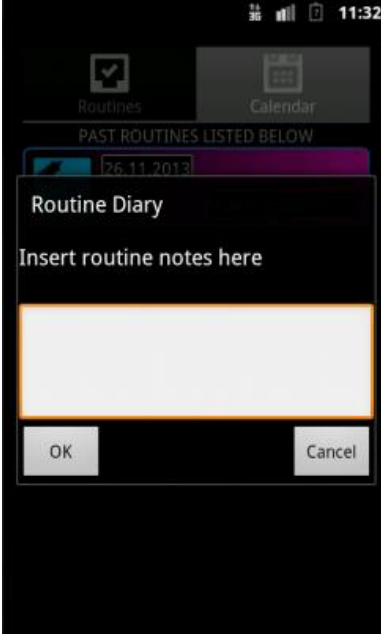
3. Lisää lapsi –näkyvä



The screenshot shows a mobile application interface for adding a child. The title is "Routine Diary" and the section is "Add child". There is a text input field for "First name". Below it are two date/time pickers: "Date of birth" set to "26.11.201" and "Time of birth" set to "11:31". At the bottom are "Save" and "Cancel" buttons.

KUVA 10. Lisää lapsi -näkyvä.

4. Lisää muistiinpano –näkyvä



The screenshot shows a mobile application interface for adding a note. The title is "Routine Diary" and the section is "Insert routine notes here". There is a large text input field. At the bottom are "OK" and "Cancel" buttons. The background shows a "Routines" screen with a "Calendar" icon and a date "26.11.2013" highlighted.

KUVA 11. Lisää muistiinpano -näkyvä.

4.4 XML-tiedosto tallennusformaattina

Kuten Tiivistetty vaatimusmäärittely –kappaleessa esitettiin, lapsien tiedot ja rutiinidata tullaan tallentamaan XML-tiedostomuotoa käyttäen. Vaihtoehtoisesti tietoa oltaisiin voitu tallentaa SQLiteä hyväksikäyttäen, mutta XML valittiin olemassa olevan kokemuksen ja helppolukuisuuden perusteella.

XML-kieli pohjautuu hyvin vahvasti HTML-kielen syntaksiin, ja on siksi helppolukuista, sekä tiedoston rakennetta on vaivatonta hahmotella etukäteen. XML-tiedostoa on myös nopea käsitellä kirjoitus- ja lukumetodeissa, kunhan tiedoston koko ei kasva liian suureksi. Tämä tarkoittaa kymmeniä megatavuja, ja prototyypisovellusta käytettäessä tuskin päästään tähän kokoluokkaan. Jos kohderyhmältä saatavien palautteiden pohjalta huomataan, että XML-tiedostomuoto ei sovellukaan tietojen tehokkaaseen tallentamiseen, tullaan tekemään tarvittavia muutoksia tehokkuuden parantamiseksi.

5 PROTOTYYPIN TOIMINNALLISUUKSIEN TOTEUTUS

Tässä kappaleessa käsitellään tarkemmin prototyypisovelluksen valmiiden toimintojen toteutusta, ja niiden mahdollista eroavaisuutta suunniteltuun nähden.

5.1 Rutiinin käynnistäminen, lopettaminen ja muokkaaminen

Rutiinin käynnistäminen tapahtuu painamalla Rutiinit-välilehdessä olevaa symbolia. Rutiinin luonteesta riippuen se on joko pitkäkestoinen tai kertarutiini. Pitkäkestoinen rutiini vaatii erikseen lopettamisen, jonka yhteydessä lopetusaika tallennetaan muistiin rutiinin kokonaiskeston laskemista varten, kun taas kertarutiini ei vaadi lopettamista, vaan se on erityisesti lyhyitä rutiineja, kuten vaipan vaihtoa tai lääkkeen antoa, varten. Kertarutiinissa tallennetaan vain aloitusaika, koska ei ole tarpeen laskea rutiinin kesto.

Sekä kerta- että pitkäkestoisen rutiinin tapauksessa, kun rutiini on käynnistetty ja lopetettu onnistuneesti, ilmestyy se välittömästi Menneet rutiinit –välilehteen. Kunkin menneen rutiinin kohdalla on nähtävissä rutiinin symboli, kesto, päivämäärä ja tyhjä kenttä muistiinpanoja varten. Menneet rutiinit –välilehti on nimensä mukaisesti menneiden rutiinien tarkastelua varten, mutta tässä välilehdessä on mahdollista lisätä suoritettuun rutiiniin sitä täydentävä muistiinpano. Jos käyttäjällä on rutiinin suorittamisen jälkeen tarve kirjoittaa muistiin jotakin tärkeää rutiinia koskien, onnistuu se helposti rutiinin päällä napauttamalla, jolloin aukeaa erillinen ikkuna tekstinsyöttöä varten. Lopetettuaan muokkaamisen OK-nappia painamalla muistiinpano tallentuu halutun rutiinin kohdalle.

5.2 Käyttäjän tietojen salaaminen

Aika ajoin mobiililaitteista keskusteltaessa aiheeksi nousee mobiilisovellusten turvallisuus ja luottamus käyttäjätietojen turvaamisesta. Huoli ei ole aiheeton, sillä haittaohjelmat ja virukset yleistyvät jatkuvasti myös mobiililaitteissa.

Tämän sovelluksen tapauksessa käyttäjän antamissa tiedoissa ei ole mitään rikollisia kiinnostavaa tietoa, kuten pankki- tai muita henkilökohtaisia arkoja tietoja, vaan enemmänkin tunnearvoa sisältäviä tietoja, joita käyttäjä ei välttämättä halua ulkopuolisten näkevän ja käsittelevän. Android tarjoaa sovelluskehittäjille mahdollisuuden tallentaa tiedostoja niin kutsuttuun Internal Storageen, johon on pääsy vain kyseessäolevalla sovelluksella.

Käyttäjätietojen salaamiseksi voitaisiin käyttää myös jotain salausalgoritmia, mutta sellaisen käyttämiselle ei nähty perusteita, ainakaan prototyypisovelluksen tapauksessa. Prototyypivaihetta ajateltaessa riittäväksi tietosuojaksi katsottiin Internal Storagen käyttö.

5.3 Lokalisointi

Lokalisointi voi olla joillekin hieman outo sana, mutta se on yleinen termi varsinkin ohjelmistokehityksen parissa. Lokalisointi tarkoittaa erityispiirteiden lisäämistä tuotteeseen, jota käytetään jossain tietyssä maassa. Lokalisointi on laajempi käsite edellämainituille erityispiirteille, joista esimerkkinä mainittakoon kääntäminen, symbolit ja paikallinen sisältö. Tätä työtä ajatellen lokalisoinnilla viitataan ainoastaan tuotteen kääntämiseen usealle eri kielelle.

Yleisesti ottaen mobiilisovelluksia käyttävät ihmiset haluavat käyttää sovelluksia joko äidinkielellään tai eniten osaamallaan toisella kielellä (suomalaisten tapauksessa käyttökieli on yleensä suomi, ruotsi tai englanti). Tämän sovelluksen tapauksessa käytössä on suomi ja englanti sen mukaan, mikä kieli käyttäjällä on puhelimessaan käytössä. Jos käyttäjä on asettanut puhelimensa kieleksi suomen, myös sovellus on suomen kielellä, mutta muussa tapauksessa sovelluksessa on käytössä englanti. Ajan puutteen ja taitamattomuuden vuoksi ruotsi ei ole tällä hetkellä tuettu kieli, vaikka se Suomen toinen virallinen kieli onkin. Tätä asiaa käsitellään lisää Pohdinta-kappaleessa.

6 POHDINTA

Opinnäytetyöni idean ”äitinä” toimi vaimoni, joka ehdotti mobiilisovelluksen toteuttamista lapsenhoitorutiinien helppoa seuraamista varten. Kyseisen sovelluksen toteuttaminen opinnäytetyönäni olikin projektina melkoinen hyppy tuntemattomaan. Projektin hyväksyminen kuitenkin mahdollisti perehtymisen suhteellisen valloittamattomalle alueelle, eli vanhemmuudesta ideansa ammentavien sovelluksien maailmaan.

Vaikka minulla oli jo aikaisempaa kokemusta Android-ohjelmoinnista, en ollut kuitenkaan aikaisemmin ohjelmoinut itsekseen yhtä laajaa sovellusta. Minulla ei ollut kokemusta myöskään asiakaslähtöisen projektin toteuttamisesta ennen tätä työtä, joten tämän mahdollisuuden saaminen oli hyvin kasvattava kokemus, IT-maailmassa kun suuri osa projekteista saadaan toimeksiantoina asiakkailta. Olkoonkin, että ”asiakaslähtöinen” oli tässä työssä passiivista vuorovaikutusta ja aloite tuli sovelluksen ohjelmoivalta puolelta. Toteutin yksinkertaisen mutta työtäni merkittävästi helpottavan Internet-kyselyn suurimmilla äitien keskustelupalstoilla ja Facebookin äitiryhmässä, ja runsaan palautemäärän pohjalta valikoin parhaimmat ominaisuudet, joiden pohjalta päätin toteuttaa sovelluksestani prototyypin. Palautteissa oli loistavia ideoita valmiin ja kaupallisen sovellukseen toteutettavaksi, mutta niitä olisi ollut yhden miehen mahdoton saada aikaiseksi niin lyhyessä ajassa.

Olen kaikissa aikaisemmissa projekteissani koittanut suunnitella sovelluksen vähintäänkin tyydyttävästi, mutta varsinkin tätä työtä tehdessä todella paljastui suunnittelun merkitys sovelluskehityksessä. Joitain sovelluksen toimintoja jouduin ohjelmoimaan uudestaan useitakin kertoja, kun joidenkin toimintojen välillä tuli ristiriitatilanteita yhden sovellusosa-alueen valmistuessa. Kaikissa toteuttamissani ominaisuuksissa toteutus ei täten aina vastannut täysin suunniteltua, mutta annoin sen itselleni anteeksi, sillä minulla ei ole kovinkaan vankkaa kokemusta varsinaisesta ohjelmistosuunnittelusta tai sen vaiheista. Lopputuloksena oli kuitenkin sovelluksen toimintaa demonstroiva prototyyppi.

Ajankäytön priorisoinnin tärkeys paljastui myös hyvin nopeasti työn alettua, varsinkin näin lyhyen työntekoajan huomioon ottaen. Hyvin usein huomasin olevani tilanteessa, jossa jouduin punnitsemaan toiminnon loppuun asti suunnittelemisen ja ohjelmoinnin

aloittamisen välillä. Toistuvasti ajauduinkin tilanteeseen, jossa sekä ohjelmoin että suunnittelin ohjelmoimaani toimintoa samanaikaisesti. Vaikka kukaan ei ollut antamassa minulle virallisia deadlineja toimintojen valmistumiseksi, antoi itse luotu paine saattaa toiminto valmiiksi mahdollisimman nopeasti kuvan siitä, millaista työskentely ohjelmistotalossa oikeiden aikarajojen varassa voisi olla.

Vaikeuksista huolimatta koin työtä tehdessä myös inspiroivia ja voimia-antavia onnistumisen hetkiä saadessani jonkin ominaisuuden toimimaan tai oivaltaessani uuden toteutustavan Android-ohjelmoinnista. Nämä onnistumisen tunteet olivat mielestäni merkittäviä, sillä ne todella antoivat lisäpuhtia työn jatkamiseen sellaisina hetkinä, kun yhden yksinkertaiselta vaikuttavan ominaisuuden toteuttaminen olikin erityisen haastavaa. Varsinkin perehtyminen työssä käyttämieni käyttöliittymäkomponenttien toimintaan ja toteutukseen suhteessa sovellukseeni oli palkitsevaa onnistuneen toteutuksen jälkeen, mutta ajoittain myös turhauttavaa tuntemattomaan aiheeseen opettelemiseen. Voin kuitenkin rehellisesti sanoa olevani tämän työn jälkeen etevämpi Android-sovellusten ohjelmoinnissa kuin aiemmin.

Tätä työtä kirjoittaessani löysin Androidin ja Windows Phonen sovelluskaupoista samalta kehittäjältä julkaistut sovellukset, jotka vastasivat täysin omaa ideaani ja visiotani sovelluksestani. Tämä oli hyvä opetus minulle siitä, että saadessaan omasta mielestään hyvän idean mahdollisesta kaupallisesta sovelluksesta, on kriittistä selvittää, onko vastaavia kilpailevia sovelluksia jo markkinoilla entuudestaan. Niin sanotusti kotiläksynsä tehdessään on vielä mahdollista suunnitella ja toteuttaa omasta sovelluksestaan kilpailijoitaan paremman, tai kehittää sovellukseensa jonkin ”koukun”, jolla tehdä siitä houkuttelevamman ladata ja käyttää.

Huomatessani omaa sovellustani vastaavan tuotteen sovelluskaupassa päätin, etten hylkää projektiani suoraan pöytälaatikkoon, vaan yritän toteuttaa visioni jopa laajemmin kuin olin alunperin suunnitellut. Jos onnistun suunnittelemaan hieman muokatun sovellusideani loppuun saakka hyvin, voisi siitä saada aikaiseksi potentiaalisen kilpailevan tai jopa uuden tuotteen (tai palvelun), joka toimisi kaikilla merkittävillä mobiili- ja työpöytäalustoilla.

Loppujen lopuksi saavutin suurimman osan opinnäytetyölleni asettamista tavoitteista vähintäänkin tyydyttävästi. En onnistunut saattamaan prototyyppiäni täysin valmiiksi,

sillä joitain ominaisuuksia, jotka olisivat demonstroineet valmiin sovelluksen toimintaa vielä paremmin, jäivät uupumaan. Lisäksi en ehtinyt turhia bugeja karsimaan pois. Tästä kaikesta huolimatta sain toteutettua prototyypisovelluksen, josta käy juuri sen tärkeimmän ominaisuuden toiminta selville, jota lähdettiin tavoittelemaankin. Tämän projektin toteuttaminen (joka jatkuu tämän opinnäytetyön jälkeenkin) on ollut erittäin hyvää kokemusta ohjelmistokehityksestä ja sen vaiheista yleisesti, sekä antanut hieman viitteitä myös siitä, millaista on tehdä sovelluksia jonkin ulkopuolisen tahon tarpeiden täyttämiseksi. Olen saanut tämän projektin myötä runsaasti itseluottamusta omiin kykyihini ohjelmoijana ja olen saanut kartutettua taitojani lisää Android-ohjelmoinnin saralla. Uskon tämän opinnäytetyön antaneen minulle ohjelmoijana yleisesti runsaasti lisää tietotaitoa ja valmiuksia tulevissa työtehtävissäni, vaikkei ne välttämättä koskisikaan juuri Android- tai muuta mobiiliohjelmointia.

LÄHTEET

About.com: How many apps are in the iPhone App Store. Luettu 31.10.2013.
<http://ipod.about.com/od/iphonesoftwareterms/qt/apps-in-app-store.htm>

About.com: Is Android safe? Luettu 31.10.2013.
<http://antivirus.about.com/od/wirelessthreats/a/Is-Google-Play-Safe.htm>

Android Central: Android A to Z: What is Dalvik? Luettu 27.9.2013.
<http://www.androidcentral.com/android-z-what-dalvik>

Android Central: T-Mobile G1 hits a million units sold. Luettu 9.10.2013.
<http://www.androidcentral.com/t-mobile-g1-hits-million-units-sold>

Android Developers: Dashboards. Luettu 29.8.2013.
<http://developer.android.com/about/dashboards/index.html>

Android Developers: Storage Options. Luettu 23.9.2013.
<http://developer.android.com/guide/topics/data/data-storage.html>

Android Headlines: Top 10 Best Android Manufacturers of 2012. Luettu 16.10.2013.
<http://www.androidheadlines.com/2012/12/featured-top-10-best-android-manufacturers-of-2012.html>

AppBrain: Number of available Android apps. Luettu 31.10.2013.
<http://www.appbrain.com/stats/number-of-android-apps>

Business Week: Google Buys Android for Its Mobile Arsenal. Luettu 24.9.2013.
<http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>

Catazine: Dalvik Virtual Machine. Luettu 28.9.2013.
<http://catazine.files.wordpress.com/2012/02/dvm.png>

CNET: A brief history of Android phones. Luettu 9.10.2013.
http://reviews.cnet.com/8301-19736_7-20016542-251/a-brief-history-of-android-phones/

Embedded Linux: Android Architecture. Luettu 28.9.2013.
<http://elinux.org/File:Android-system-architecture.jpg>

Electronic Weekly: What is the Dalvik virtual machine? Luettu 27.9.2013
<http://www.electronicweekly.com/eyes-on-android/what-is/what-is-the-dalvik-virtual-machine-2011-10/>

GSM Arena: HTC Dream. Luettu 9.10.2013.
http://www.gsmarena.com/htc_dream-2665.php

Google Mobile Blog: Android and Security. Luettu 31.10.2013.
<http://googlemobile.blogspot.fi/2012/02/android-and-security.html>

HowStuffWorks: How the Google Phones Works. Luettu 24.9.2013.
<http://electronics.howstuffworks.com/google-phone2.htm>

Open Handset Alliance: FAQ. Luettu 24.9.2013.
http://www.openhandsetalliance.com/oha_faq.html

Open Handset Alliance: Members. Luettu 24.9.2013.
http://www.openhandsetalliance.com/oha_members.html

ReadWrite: How the iPad stalled and Android took over the tablet market. Luettu 17.10.2013.
<http://readwrite.com/2013/08/06/how-the-ipad-stalled-and-android-took-over-the-tablet-market#awesm=~oksFD64hFBI5S5>

Softpedia: T-Mobile's G1 counts 1.5 million pre-sold units. Luettu 9.10.2013.
<http://news.softpedia.com/news/T-Mobile-039-s-G1-Counts-1-5-Million-Pre-Sold-Units-95496.shtml>

Thins about stuff: Android home screen. Luettu 16.10.2013.
<http://nhenze.net/uploads/donut-home-screen.png>

Wikipedia: Android. Luettu 23.9.2013.
<http://fi.wikipedia.org/wiki/Android>

Wikipedia: Android software development. Luettu 24.9.2013.
http://en.wikipedia.org/wiki/Android_software_development

Wikipedia: Android version history. Luetty 16.10.2013.
[http://en.wikipedia.org/wiki/File:Android_4.3_on_Nexus_7_\(2013\).jpg](http://en.wikipedia.org/wiki/File:Android_4.3_on_Nexus_7_(2013).jpg)