

Opinnäytetyö (AMK)

Tietojenkäsittely

Tietojärjestelmät

2015

Juho Heiskanen

# ANDROID-TABLETIN KÄYTTÖLIITTYMÄ VANHUKSILLE



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Tietojärjestelmät

Marraskuu 2015 | 33 sivua

Ohjaaja: Jarkko Paavola

Juho Heiskanen

# ANDROID-TABLETIN KÄYTTÖLIITTYMÄ VANHUKSILLE

Työn tavoitteena on Android-käyttöjärjestelmän ehdoilla kehitetty sovellus, joka on ainoa käyttäjälle näkyvä käyttöliittymä, ja näin ollen määrää käytettävissä olevat toiminnot.

Koska laitteet tulevat käyttöön kotihoidossa oleville vanhuksille, on käyttöliittymän ulkoasun ja käytettävyyden oltava niin selkeä ja intuitiivinen, että laitetta pystyy käyttämään ilman opastusta.

Sovelluksen kehitys tapahtui annetun vaatimusmäärittelyn perusteella ketterän sovelluskehityksen periaatteilla yhteistyössä Medbit Oy:n kanssa. Vaatimusmäärittely antoi sovellukselle vain suurpiirteisen rungon. Yksityiskohtiin liittyvät suunnittelupäätökset jäivät opinnäytetyön tekijän vastuulle.

Lopputuloksena syntyneessä käyttöliittymässä käytettävyyden helppous toteutui, mutta Android-käyttöjärjestelmän ominaisuuksista johtuen laitteen toimintojen rajoittaminen ei onnistunut täydellisesti.

Käyttöliittymä on ollut Rauman kaupungin kotihoidossa pilottiprojektissa ja sitä ollaan viemässä tuotantokäyttöön vuoden 2016 alkupuoliskolla. Sovelluksen rakenne on modulaarinen ja helposti muokattavissa, joten sitä voidaan tarpeen vaatiessa muunnella käytettäväksi muissakin käyttökohteissa.

## ASIASANAT:

Android, käyttöliittymä, käytettävyys, Java, ohjelmointi, sovelluskehitys, tabletti, taulutietokone, mobiili

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology | Data systems

November 2015 | 33 pages

Instructor: Jarkko Paavola

Juho Heiskanen

## ANDROID TABLET USER INTERFACE FOR ELDERLY PEOPLE

The purpose of this thesis was to develop an application, designed to comply with the restrictions of the Android operating system. The application would be the only interface that the user can use in certain devices, thus dictating the selection of available functions.

These devices will be given to elderly people currently in home care, which means that the user interface must be so simple and intuitive so that it can be used without instructions.

The application was developed in collaboration with Medbit Oy, according to the given requirements specifications and by the principles of agile software development. The requirements specification described only a rough structure for the application. The author of this thesis was responsible for any detailed design decisions.

The resulting user interface was indeed easy to use, but due to limitations in the Android operating system restrictions to available functions could not be implemented perfectly.

This interface has been tested in a home care pilot project by the city of Rauma. It will be ready to be released during the first half of 2016. The application has a modular structure and is easily modified. This enables it to be developed for different kinds of uses.

### KEYWORDS:

Android, user interface, usability, Java, programming, software development, tablet computer, mobile

# SISÄLTÖ

<b>SANASTO JA LYHENTEET</b>	<b>5</b>
<b>1 JOHDANTO</b>	<b>7</b>
<b>2 PROJEKTIN TAUSTATIEDOT</b>	<b>8</b>
2.1 Asetetut vaatimukset	8
2.2 Android-käyttöjärjestelmä yleisesti	8
2.4 Dalvik-virtuaalikone	9
2.5 Android-sovelluksen lähdekoodin rakenne	10
2.6 Android-sovelluksen elinkaari	10
<b>3 KÄYTTÖLIITTYMÄN TOTEUTUS</b>	<b>12</b>
3.1 Ikonit ja painikkeet	12
3.2 Näytön orientaatio	13
3.3 Tilapalkki	14
3.3.1 Tilapalkin piilottaminen	14
3.3.2 Verkkoyhteyden tila	16
3.3.3 Akun tila	17
3.4 Yhteyshenkilölista	18
3.4.1 Ruudun jakaminen osiin	18
3.4.2 Yhteyshenkilön lisääminen	20
3.4.3 Tiedostoon tallentaminen	23
3.5 Kotisovellus	24
3.6 Toisen sovelluksen käyttäminen videopuheluihin	25
3.7 Oletusnäkyvästä poistuminen	27
3.8 Ammattilaisnäkyvä	27
<b>4 LOPPUTULOKSEN ARVIOINTI</b>	<b>29</b>
<b>LÄHTEET</b>	<b>31</b>
<b>KUVAT</b>	<b>33</b>

## SANASTO JA LYHENTEET

.apk	<i>Android application package</i> . Muoto tiedostolle, johon Android-sovellukset paketoidaan asennusta varten.
.dex	<i>Dalvik executable</i> . Dalvik-virtuaalikoneessa suoritettavien ohjelmien tiedostomuoto.
Four Button	Työn aiheena olevan sovelluksen työnimi. <i>Four button application</i> . ”Nelinappinen sovellus”. Myöhemmin nappeja tuli kaksi lisää.
kotipainike	Painike, jolla voi palata oletusnäkyseen Android-käyttöjärjestelmässä. Laitteen valmistajasta riippuen joko graafinen tai fyysinen.
kotisovellus	Sovellus, joka näyttää laitteen oletusnäkyseen. Avautuu laitteen käynnistyessä tai kotipainiketta painamalla.
LTE	<i>Long term evolution</i> . 3G-yhteystekniikan seuraaja. Tunnetaan myös nimellä 4G. Tarkkuuden vuoksi tässä raportissa käytetään yksinomaan LTE-nimitystä, koska 4G-nimityksellä on markkinoitu harhaanjohtavasti yhteyksiä jotka eivät ole olleet LTE-tekniikkaa.
paluupainike	Painike, jolla voi palata edelliseen näkyseen Android-käyttöjärjestelmässä.

pääaktiiviteetti	Android-sovelluksen aliohjelma, joka näkyy ensimmäisenä sovellusta käynnistettäessä.
roottaaminen, roottaus	Android-laitteen saattaminen tilaan, jossa laitteen valmistajan asettamat estot poistetaan ja käyttäjä saa järjestelmänvalvojan oikeudet ( <i>eng. rooting</i> ).
sarjallistaminen	Java-ohjelmointikielessä olion muuntaminen tavuina esitettyyn muotoon. Edellytys olion kirjoittamiseksi tiedostoon. ( <i>Eng. Serialization</i> )
SSID	<i>Service set identifier</i> . Langattoman lähiverkon verkkotunnus.
tilapalkki	Android-käyttöjärjestelmässä ruudun yläreunassa näkyvä kapea alue, joka on varattu ilmoitusten ja laitteen tilaan liittyvien symbolien näyttämiseen.
URI	<i>Uniform Resource Identifier</i> . Merkkijono, jolla kerrotaan tietyn tiedon paikka.
valikkopainike	Painike, jolla voi avata sovelluksen valikon tai järjestelmävalikon Android-käyttöjärjestelmässä.
WYSIWYG	<i>What you see is what you get</i> . Suunnittelumetodi, jossa suunnittelunäkymässä oleva tuote vastaa täydellisesti valmista tuotetta.

# 1 JOHDANTO

Osallistuin vuoden 2014 alussa Medbit Oy:n projektiin, jonka päämääränä on kehittää vanhusten kotihoidon tueksi tarkoitettu tabletti. Käyttöliittymässä olisi neljä tai kuusi painiketta; laitteen muiden toimintojen käyttö olisi estetty. Käytettävissä oleviin toimintoihin kuuluivat muun muassa sähköposti, kansalaisen asiointitili ja potilastietojen katselu, mutta projektin alkuvaiheessa käytössä olisi vain videopuhelutoiminto, jolla vanhukset voivat pitää yhteyttä hoitajaan.

Oma osuuteni projektissa oli toteuttaa käyttöliittymä, joka antaa käyttäjälle tarvittavat tiedot laitteen tilasta ja mahdollistaa tiettyjen toimintojen käyttämisen. Viittaan sovellukseen sille antamallani työnimellä Four Button.

Rajallisista toiminnoista johtuen käyttöliittymästä piti tehdä kotisovellus, joka korvaa laitteen valmistajan tekemän alkuperäisen kotisovelluksen. Tällä tavalla voidaan rajata laitteen käyttömahdollisuudet haluttuihin toimintoihin. Samalla laitteen käyttäminen tulee helpommaksi; ongelmatilanteissa käyttäjä voi aina palata oletusnäkympään painamalla laitteen kotipainiketta.

Laitteeksi projektiin valikoituivat Samsung Galaxy -tuotesarjan edullisimmat tabletit: ensin Tab 3 ja myöhemmin Tab 4. Alun perin kyseiset laitteet olivat erästä toista projektia varten ja toimivat kehitysalustana, mutta ne osoittautuivat sopiviksi myös tähän projektiin. Laitteiden laatu on hintaan suhteutettuna hyvä. Lisäksi fyysinen kotipainike tekee laitteen käytön selkeämmäksi.

Keskityn työssäni siihen, miten helppokäyttöinen, laitetta hallitseva sovellus toteutetaan Android-käyttöjärjestelmän ehdoilla. Laitteen roottaaminen tai käyttöjärjestelmän muokkaaminen eivät olleet vaihtoehtoja.

Tärkeänä rajauksena mainittakoon, että opinnäytetyö käsittelee Androidin versioita 4.1 – 4.3.

## 2 PROJEKTIN TAUSTATIEDOT

### 2.1 Asetetut vaatimukset

Projektin alussa pääasiallisia vaatimuksia oli kaksi. Ensimmäkin käyttöliittymän tulisi hallita laitetta siten, että käyttäjällä on käytettävissä vain ne toiminnot mitkä käyttöliittymä tarjoaa, eikä käyttöliittymästä voi poistua. Käyttöliittymän tulisi olla laitteen oletusnäkyvä kaikissa tilanteissa, myös laitteen sammutuksen tai käyttöliittymän virhetilanteen jälkeen. Näin ollen sen tulisi korvata laitteen alkuperäinen käyttöliittymä kokonaan.

Toiseksi käyttöliittymän tulisi sisältää vain neljä tai kuusi isoa painiketta. Muita käyttöliittymän elementtejä ei poissuljettu, mutta pidin yleisilmeen yksinkertaisena ja pelkistettynä käytön helppouden maksimoimiseksi (Carroll & Reitman 1987, 25). Sain vapaat kädet käyttöliittymän graafisen suunnittelun suhteen.

Projektin luonteen vuoksi sovelluksen kehittäminen tapahtui ketterän sovelluskehityksen periaatteilla (Shore & Warden 2008, 3-5). Tein sovellukseen parannuksia ja uusia ominaisuuksia tietyn väliajoin. Yleensä aikaväli oli noin kaksi tai kolme viikkoa. Useiden demoversioiden jälkeen muodostui versio, jota kuvaillaan tässä raportissa.

### 2.2 Android-käyttöjärjestelmä yleisesti

Android on Linux-ytimeen perustuva avoimen lähdekoodin käyttöjärjestelmä, joka on suunniteltu kosketusnäytöisille mobiililaitteille. Jotkin laitevalmistajat, kuten Samsung, asentavat laitteisiinsa oman Android-versionsa, joka voi poiketa alkuperäisestä Androidista ulkonäöltään ja sisältää lisättyjä ominaisuuksia. (Hildenbrand 2015).



Vuonna 2008 julkaistiin historian ensimmäinen Android-käyttöjärjestelmällä varustettu älypuhelin T-Mobile G1 käyttöjärjestelmän versionumerolla 1.0 (Amadeo 2014). Sen jälkeen Androidin suosio on kasvanut tasaista vauhtia. Vuoden 2015 toisella vuosineljänneksellä Androidin markkinaosuus kaikista mobiililaitteiden käyttöjärjestelmistä oli 83 % (International Data Corporation 2015).

## 2.4 Dalvik-virtuaalikone

Dalvik-virtuaalikone on suunniteltu akkukäyttöisiin laitteisiin, joissa on hidas suoritin, verraten pieni määrä keskusmuistia ja virtuaalimuistiton käyttöjärjestelmä (Bornstein 2008).

Java-kielellä kirjoitetut sovellukset käännetään Java-virtuaalikoneen ymmärtämäksi tavukoodiksi, joka muunnetaan edelleen Dalvik-virtuaalikoneen tavukoodiksi ja tallennetaan dex-tiedostoon (Bornstein 2008). Kun dex-tiedosto on paketoitu apk-tiedostoon muiden tarvittavien resurssien kanssa, on sovellus valmiina asennettavaksi Android-laitteeseen (Morriell 2008).

Kaikki sovellukset Android-käyttöjärjestelmässä suoritetaan omana prosessinaan, ja jokainen prosessi saa oman Dalvik-virtuaalikoneensa. Virhetilanteessa ainoastaan sovelluksen oma virtuaalikone voi kaatua. Tämä parantaa käyttöjärjestelmän vakautta. (Kanaujia 2011.)

Virtuaalikone luo sovelluksen ja laitteen välille rajapinnan, joka antaa sovelluksen suorittaa vain sille luvallisia toimintoja. Ohjelmakoodi sisältää tiedoston, jossa sovellus pyytää luvan suorittaa erilaisia toimintoja. Näitä toimintoja ei voi päättää vapaasti, vaan ne on valittava ennalta määrätyistä vaihtoehdoista. Sovellusta asentavan käyttäjän on hyväksyttävä sovelluksen pyytämät käyttöoikeudet ennen kuin sovelluksen asentamista voidaan jatkaa. (Deep 2013.)

Sovelluksen suunnittelun kannalta virtuaalikone edellyttää tietynlaisia toimintatapoja. Esimerkiksi langattoman verkkosovittimen tilaa ei voida havaita

laitetasolla suoraan sovittimesta, vaan sovelluksen täytyy kuunnella käyttöjärjestelmän lähettämiä tiedotteita sovittimen tilan muutoksista. (Google Incorporated 2015d.)

## 2.5 Android-sovelluksen lähdekoodin rakenne

Lähdekoodi voidaan jakaa kolmeen pääosaan: ohjelmakoodiin, sisällön määrittelyyn ja graafiseen asetteluun (APC 2011).

Sisällön määrittely tapahtuu xml-tiedostoilla. Tärkein niistä on AndroidManifest.xml-tiedosto, jossa määritellään muun muassa sovelluksen versio, sovelluksen vaatimat käyttöoikeudet laitteessa ja sovelluksen sisältämät aliohjelmat. Muita määrittelyjä ovat esimerkiksi usein käytetyt merkkijonot, kuten sovelluksen nimi tai dialogi-ikkunoissa käytetyt vaihtoehdot. (APC 2011.)

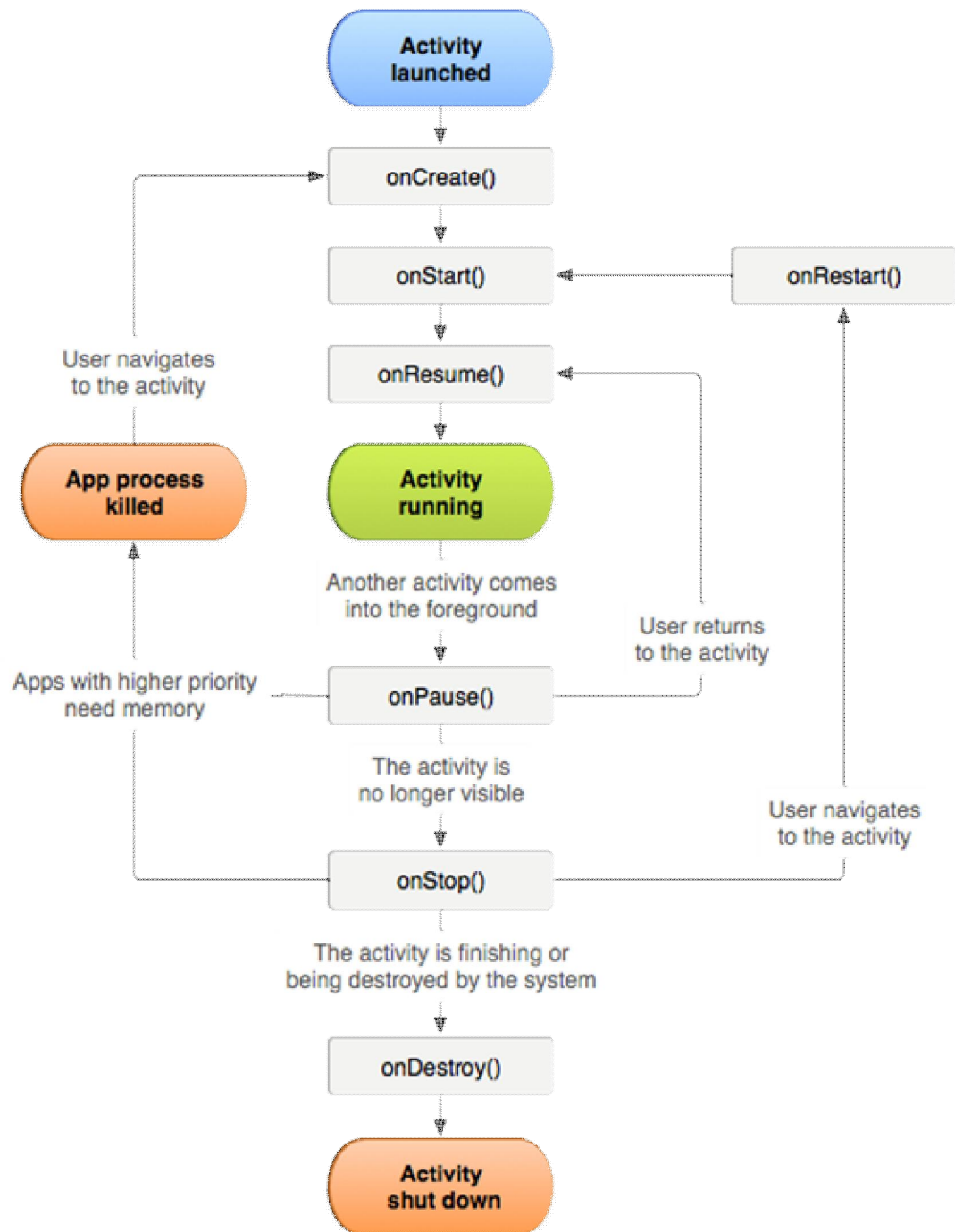
Myös graafinen asettelu tehdään xml-tiedostoissa. Kukin Android-sovellus vaatii tiedoston, jossa määritellään sovelluksen pääaktiviteetin näkymä. Sovelluksen muille aliohjelmille voidaan asettelu määritellä erikseen. (APC 2011.)

Suosituimmat kehitysympäristöt mahdollistavat asettelun suunnittelun graafisella käyttöliittymällä niin sanotulla WYSIWYG-periaatteella.

## 2.6 Android-sovelluksen elinkaari

Sovelluksen suoritus kulkee ennalta määrättyjen vaiheiden läpi (kuva 1). Ohjelmakoodissa voidaan määritellä toiminnot kullekin vaiheelle. Jos määrittelyä ei tehdä, käyttää käyttöjärjestelmä automaattisesti oletusarvoisia asetuksia, jotka periytyvät Activity-luokasta. (Google Inc. 2015j.)

Eri vaiheet on otettava huomioon esimerkiksi tilapalkin piilottamista varten. Jos käyttäjä on poistunut ohjelmasta tilapäisesti ja tulee myöhemmin takaisin, ei onCreate-metodissa olevaa koodia suoriteta ollenkaan. Tilapalkin piilottavaa metodia tulee siksi kutsua onResume-metodissa.



Kuva 1. Android-sovelluksen elinkaari (Yang 2012).

## 3 KÄYTTÖLIITTYMÄN TOTEUTUS

### 3.1 Ikonit ja painikkeet

Lopullinen ulkoasu sisältää kuusi painiketta. Toistaiseksi käytössä on vain videopuhelutoiminto. Muut painikkeet ovat varattu tulevaisuudessa lisättäviä ominaisuuksia varten ja ne voivat vielä muuttua (kuva 2).



Kuva 2. Käyttöliittymän oletusnäkyvä.

Mahdollisten tekijänoikeusongelmien vuoksi opinnäytetyön kirjoittaja on piirtänyt ikonit itse. Koska käyttöliittymä on tarkoitettu vain 8- ja 10-tuumaisille näytöille, on normaalisti skaalautuvien ikonien katsottu olevan riittävät.

Jos näyttöjen koossa olisi ollut suurempaa vaihtelua, olisi ikoneista voinut tehdä osittain skaalautuvat. Toisin sanoen ikoneista voidaan määritellä alue, jonka koko pysyy muuttumattomana. Tällä tavalla turvataan kuvien yksityiskohtien pysyminen terävänä. (Google Inc. 2015e.)

Kosketusnäytöllä olevien painikkeiden tulisi olla kooltaan niin suuria, että niiden sisälle mahtuu vähintään 7 mm<sup>2</sup>:n kokoinen alue, joka vastaa karkeasti sormen ja näytön välistä kosketuspintaa. Painikkeiden välissä tulee sen lisäksi olla vähintään 2 mm tilaa. (Gypps 2014, 48.) Käyttäjryhmä huomioon ottaen painikkeet on tehty niin suuriksi kuin kohtuuden rajoissa oli mahdollista. Painikkeiden väliin jäi yli 2 mm tyhjää tilaa virhepainallusten ehkäisemiseksi.

Ikonit ja painikkeet on yritetty saada näyttämään mahdollisimman hyvältä, koska esteettisesti miellyttävät käyttöliittymät koetaan myös helpommiksi käyttää, olivatpa ne oikeasti sellaisia tai eivät (Lidwell ym. 2010, 20-21).

### 3.2 Näytön orientaatio

Sovellusten käyttöliittymät Android-laitteissa voivat kääntyä automaattisesti laitteen asennon mukaan. Jos laite käännetään vaakatasosta pystyyn, myös käyttöliittymä kääntyy pystyyn, jolloin se on käyttäjän näkökulmasta oikein päin.

Automaattinen mukautuminen näytön orientaation muutokseen voi kuitenkin aiheuttaa ongelmia, koska se synnyttää sovelluksessa tilan muutoksen. Muita tilan muutoksen aiheuttajia voivat olla esimerkiksi näppäimistön saatavuus tai kieliasetukset. Tilan muutoksen tapahtuessa käyttöjärjestelmä lopettaa sovelluksen parhaillaan käynnissä olevan aktiviteetin suorittamalla sen `onDestroy()`-metodin, ja käynnistää aktiviteetin uudelleen suorittamalla sen `onCreate()`-metodin. (Google Inc. 2015g.)

Jos tilamuutoksen tapahtuessa sovelluksessa on ollut käynnissä esimerkiksi asynkronista laskutoimitusta suorittava olio, ei laskennan päätyttyä olion antama tulos enää päde laitteen uuteen tilaan (Lockwood 2013).

Suosittelava tapa hallita laitteen asennon vaihtumista on käyttää Activity-luokalta periytyvää `onSaveInstanceState()`-metodia, joka varastoi aktiviteetin tilan ennen sen lopettamista. Aktiviteetin käynnistyessä uudelleen aiempi tila voidaan palauttaa `onRestoreInstanceState()`-metodilla. (Google Inc. 2015g.)

Koska oli tiedossa, että Four Button –sovellusta tullaan käyttämään vaakatasoisissa pöytätelineissä olevissa tableteissa, on sovellukseen lisätty asetus, joka pakottaa sen käyttöliittymän pysymään vaakatasossa. Tämä tapahtuu lisäämällä aktiviteetin ominaisuuksiin AndroidManifest.xml-tiedostossa merkintä:

```
android:screenOrientation="landscape"
```

Vaihtoehtoisesti saman toiminnon voi tehdä ohjelmallisesti lisäämällä onCreate()-metodiin rivin:

```
this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
```

### 3.3 Tilapalkki

Käyttäjälle keskeiset tiedot laitteen tilasta näkyvät normaalisti tilapalkissa. Käyttöjärjestelmän oma tilapalkki piti kuitenkin peittää kahdesta syystä. Ensinnäkin tilapalkin hipaisu sormella tuli estää, jotta käyttäjä ei pääse avaamaan ilmoituspaneelia vahingossa tai tahallisesti. Ilmoituspaneelistä käyttäjä pääsisi käsiksi muun muassa laitteen asetuksiin. Toiseksi laitteen käyttäjälle tarpeettomat kuvakkeet ja ilmoitukset tuli piilottaa.

Tilapalkin piilottamisen seurauksena on se, että myös tarpeelliset tiedot, kuten kellonaika, katoavat näkyvistä. Käyttöjärjestelmä ei salli tilapalkin peittämistä tai muokkaamista; sen voi ainoastaan piilottaa (Google Inc. 2015b). Näin ollen piti luoda tilapalkin korvike, jossa näytetään päivämäärä ja kellonaika sekä verkkoyhteyden ja akun tila.

#### 3.3.1 Tilapalkin piilottaminen

Alkuperäisen tilapalkin piilottaminen tapahtuu sovelluksen käynnistyessä, heti kun graafinen asettelu on ladattu. Käyttöjärjestelmältä pyydetään koodissa

kokoruudun tila. Tämä ei kuitenkaan estä tilapalkkia tulemasta uudelleen näkyviin tietyissä tilanteissa, esimerkiksi dialogi-ikkunan auetessa. (Google Inc. 2015b.)

Tilapalkki täytyy piilottaa uudelleen aina kun näkymään on tullut jokin tilapalkin palauttava muutos. Tätä varten Four Buttonin koodissa on oma metodinsa, jossa suoritetaan kaikki tarvittavat näkymän asetukset, kuten tilapalkin piilottaminen (kuva 3). Kyseistä metodia kutsutaan muun muassa sovelluksen elinkaareen kuuluvassa onResume()-metodissa, joka suoritetaan aina kun sovelluksen tavalliseen näkymään palataan, myös dialogi-ikkunan tapauksessa.

```
private void hideBars()
{
    int uiOptions;
    View decorView = getWindow().getDecorView();

    uiOptions = View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION;
    decorView.setSystemUiVisibility(uiOptions);

    uiOptions = View.SYSTEM_UI_FLAG_LAYOUT_STABLE;
    decorView.setSystemUiVisibility(uiOptions);

    uiOptions = View.SYSTEM_UI_FLAG_HIDE_NAVIGATION;
    decorView.setSystemUiVisibility(uiOptions);

    uiOptions = View.SYSTEM_UI_FLAG_FULLSCREEN;
    decorView.setSystemUiVisibility(uiOptions);

    ActionBar actionBar = getActionBar();
    actionBar.hide();
}
```

Kuva 3. Koodiesimerkki: tilapalkin piilottaminen.

Vaikka tilapalkki onkin piilotettu, voi käyttäjä silti pyyhkäistä sormella ruudun yläreunaa ja tuoda esiin ilmoituspaneelin. Onneksi tämä vaatii tapahtuakseen hallitun ja määrätietoisen pyyhkäisyn. Tahaton ja nopea hipaisu ei vielä riitä.

Ilmoituspaneeliongelmaan ei löytynyt eleganttia ratkaisua. Jos ilmoituspaneelin saaminen esiin täytyy ehdottomasti estää, voidaan kuhunkin näkymään lisätä metodi, joka pakottaa kaikki järjestelmän valikot tai ikkunat sulkeutumaan heti kun ne ovat auenneet (kuva 4).

```
public void onWindowFocusChanged(boolean hasWindowFocus)
{
    super.onWindowFocusChanged(hasWindowFocus());
    if(!hasWindowFocus())
    {
        Intent closeDialog = new Intent(Intent.ACTION_CLOSE_SYSTEM_DIALOGS);
        sendBroadcast(closeDialog);
    }
}
```

Kuva 4. Koodiesimerkki: valikoiden piilottaminen.

Käyttäjän näkökulmasta ratkaisu ei ole paras mahdollinen, koska se aiheuttaa välkkymistä ja nykimistä käyttöliittymässä.

### 3.3.2 Verkkoyhteyden tila

Verkkoyhteys on toteutettu Medbit Oy:n tarjoamalla LTE-mobiiliyhteydellä. Langatonta lähiverkkoa ei käytetä.

Verkkoyhteyden tila osoitetaan selkokiekisellä tekstillä. Jos yhteys on poikki, näytetään punaisilla tikkukirjaimilla teksti "EI VERKKOYHTEYTTÄ". Jos verkkoyhteys on toiminnassa, näytetään harmaana teksti "Mobiiliyhteys toiminnassa (X)", jossa X on käytetyn yhteystekniikan lyhenne.

Vaikka langatonta lähiverkkoa ei ole tarkoitus käyttää, on sovelluksella valmius osoittaa langattomaan tukiasemaan muodostettu yhteys. Tämä näytetään harmaana tekstinä "Yhdistettynä langattomaan lähiverkkoon (X)", jossa X on tukiaseman SSID. Tästä on nähtävillä esimerkki kuvassa 2.

Harmaa teksti on suunniteltu varastamaan mahdollisimman vähän huomiota tärkeämmiltä asioilta. Tieto on kuitenkin luettavissa, jos sitä tarvitsee. Vaikka kohderyhmään kuuluvalla käyttäjälle ei olekaan tarpeen tietää, mitä yhteystekniikkaa käytetään milläkin hetkellä, on tieto oltava kuitenkin saatavilla esimerkiksi mahdollisia etätukitilanteita varten.



Verkkoyhteyden tilaa tarkistetaan reaaliajassa ohjelmakoodissa olevalla käyttöjärjestelmän ilmoituksia kuuntelevalla oliolla. Tässä tapauksessa ilmoitukset koskevat verkkoyhteyden tilan muutoksia. Käsittelyn jälkeen ilmoitusten antama tieto näytetään selkokielisessä muodossa käyttöliittymän yläreunassa.

### 3.3.3 Akun tila

Tieto akun varaustilasta ja lataustapahtumista saadaan käyttöjärjestelmältä samalla tavalla kuin kohdassa 3.3.2 mainittu verkkoyhteyden tila.

Akun varaustilaa kuvastaa paristoa esittävä ikoni, jonka pääasiallinen informaatio tulee ikonin väristä (Hooper & Berkman 2011, 19). Väriä käyttämällä mahdollisesti huononäköisen käyttäjän ei tarvitse tulkita pientä ikonia, jossa latauksen tila esitetään kokoaan muuttavalla palkilla. Tämä ratkaisu on tietysti mielessä suunnitteluvirhe, sillä se ei ota huomioon värisokeita käyttäjiä, vaikka puna-vihersokeus onkin suhteellisen harvinainen. Sitä esiintyy 8 %:lla miehistä ja 0,5 %:lla naisista (Saarema 2015).

Vihreä ikonin väri tarkoittaa, että akun varaustila on 50-100 %. Keltainen tarkoittaa 15-49 %:n varausta. Alle 15 %:n varaustila muuttaa ikonin punaiseksi ja sykkiväksi. Varaustila näkyy myös ikonin keskellä prosenttilukuna.

Ikonin toteutus perustuu erivärisiin kuviin, joita vaihdetaan tarpeen mukaan. Prosenttiluku näytetään tekstikentässä mustalla tekstillä kuvan päällä. Punaisen ikonin sykkivä animaatio on toteutettu Androidin omalla animointiluokalla `AlphaAnimation` (Kuva 5).

```
ani = new AlphaAnimation(1.0f, 0.1f);
ani.setDuration(500);
ani.setStartOffset(20);
ani.setRepeatMode(Animation.REVERSE);
ani.setRepeatCount(Animation.INFINITE);
ani.setFillAfter(true);
```

Kuva 5. Koodiesimerkki: `AlphaAnimation`-luokka.

Tässä tapauksessa animointi liu'uttaa kuvan alpha-kanavan arvoa tietyin aikaväleihin välillä 1,0 ja 0,1.

Alhainen varaustila (~10%) aiheuttaa käyttöjärjestelmän oman ilmoituksen ponnahdusikkunassa, jossa annetaan hälytysääni ja pyydetään käyttäjää kytkemään laite lataukseen.

Kun laite kytketään lataukseen, näytetään varaustilaikonin vieressä teksti "Akkua ladataan". Varauksen saavutettua 100 % muuttuu teksti muotoon "Akku täynnä", ja varaustilaikoni muuttuu siniseksi osoittamaan latauksen päättymistä.

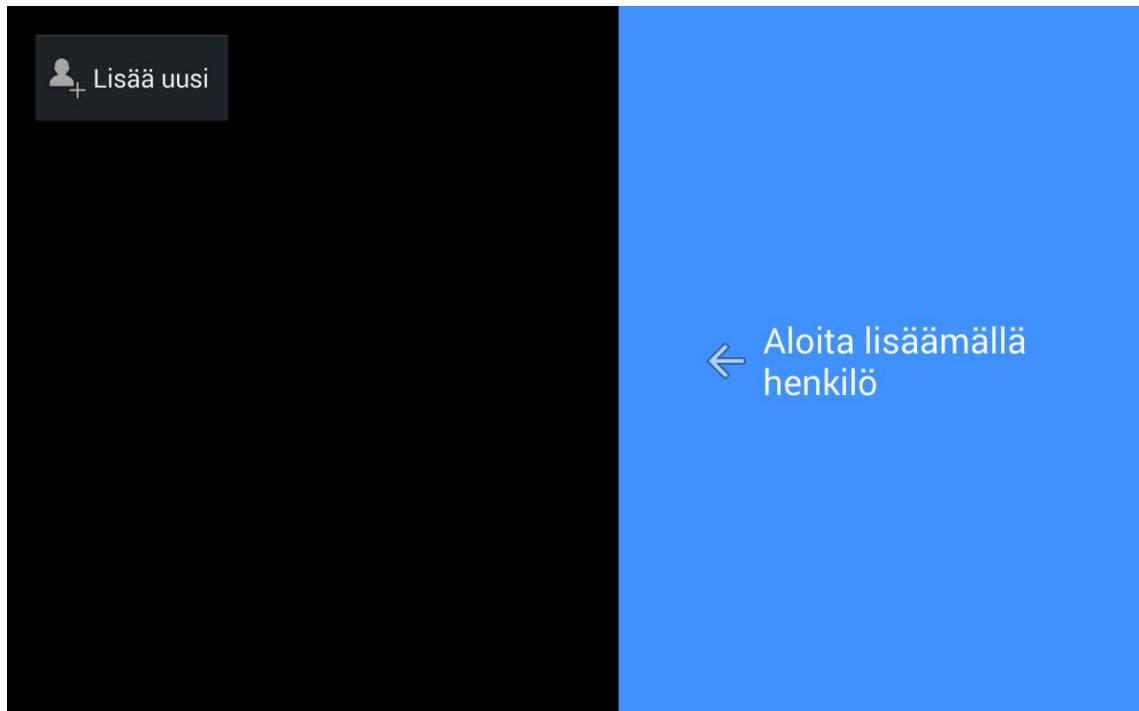
### 3.4 Yhteyshenkilölista

Videopuhelutoimintoa varten laitteen tulee pystyä näyttämään käyttäjälle lista yhteyshenkilöistä. Yhteyshenkilölista avautuu, kun sovelluksen päänäköymästä painetaan Videopuhelut-painiketta.

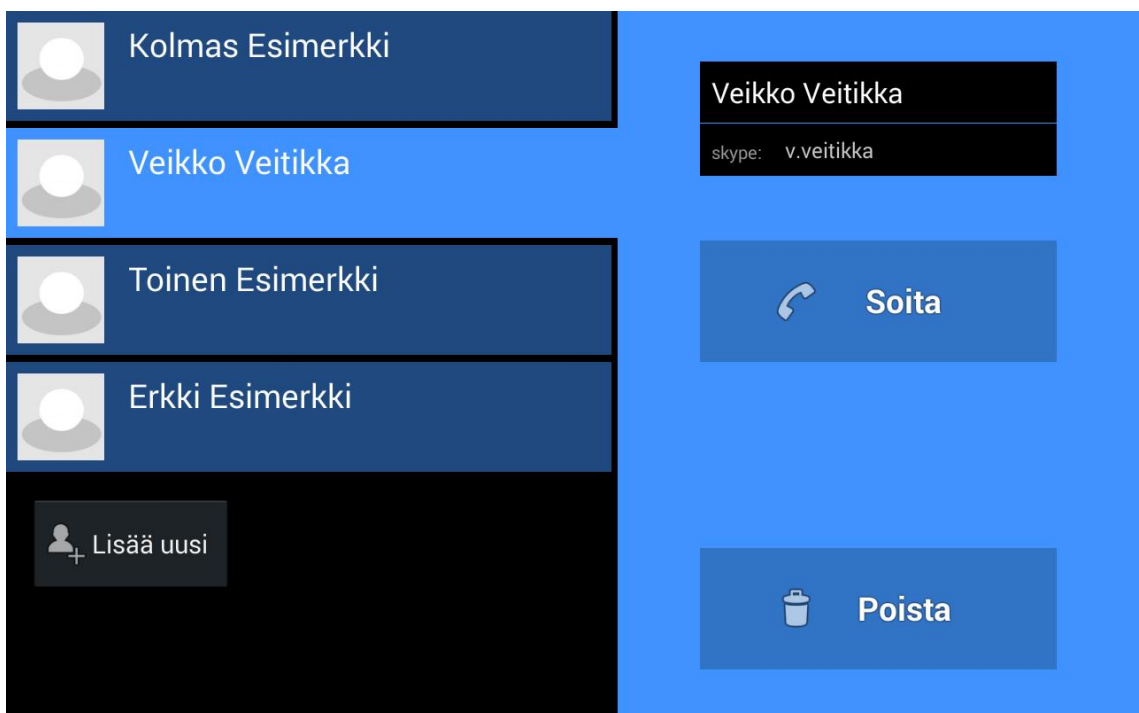
Oletusarvoisesti lista on tyhjä, ja käyttäjää opastetaan selkokielisellä tekstillä lisäämään käyttäjiä listalle (kuva 6). "Lisää uusi" -painike aukaisee Androidin oman osoitekirjan. Videopuheluihin käytetään demonstraatiotarkoituksessa Skype-sovellusta. Käytetyn Skype-tilin yhteyshenkilöt kopioituvat automaattisesti Androidin osoitekirjaan. Tästä käyttäjä voi lisätä henkilön Four Buttonin yhteyshenkilölistalle.

#### 3.4.1 Ruudun jakaminen osiin

Sovelluksen näkymä on mahdollista jakaa kahteen tai useampaan osaan. Englanninkielinen termi näille osille on *fragment*. Näkymä, jossa näkyy lista henkilöistä, on jaettu keskeltä kahteen osaan, vasempaan ja oikeaan puoliskoon. Oikealla puolella näytetään painikkeita, kun yhteyshenkilö on valittu listalta (kuvat 6 ja 7).



Kuva 6. Tyhjä yhteyshenkilölista.

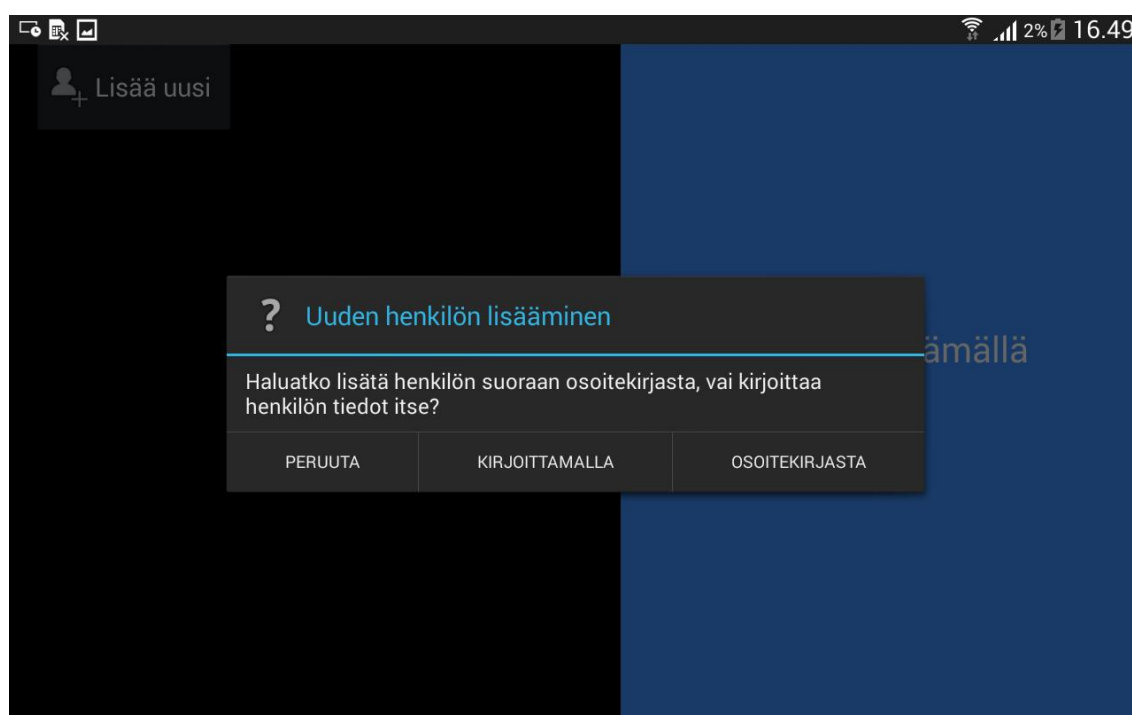


Kuva 7. Yhteyshenkilölista, jossa henkilö on valittu.

Puoliksi jaetun näkymän pohjana on asettelu, joka on jaettu osiin. Osiin jakaminen tapahtuu käyttämällä fragment-asetteluelementtejä. Näihin osiin voidaan kuhunkin määrätä oma asettelunsa, ja niiden sisällä voidaan suorittaa toimintoja toisista osista riippumattomasti (Google Inc. 2015d).

### 3.4.2 Yhteys henkilön lisääminen

Käyttäjä voi lisätä uusia henkilöitä kahdella tavalla: suoraan Androidin osoitekirjasta tai kirjoittamalla henkilön tiedot itse (kuva 8). Henkilöt Androidin osoitekirjassa sijaitsevat tietokannassa. Tähän tietokantaan voidaan suorittaa hakuja, joilla haluttu henkilö voidaan löytää.

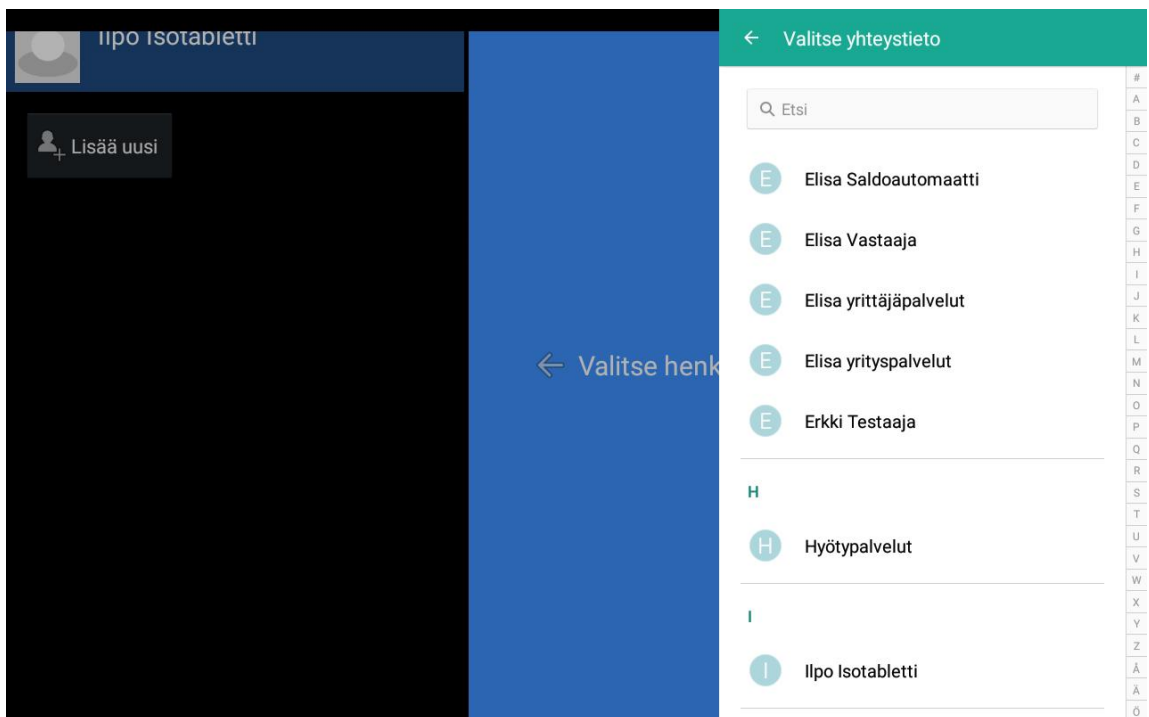


Kuva 8. Dialogi-ikkuna: uuden henkilön lisääminen.

Tietojen lukeminen edellyttää, että sovelluksella on siihen lupa (Google Inc. 2015a). Se merkitään AndroidManifest.xml-tiedostoon seuraavalla tavalla:

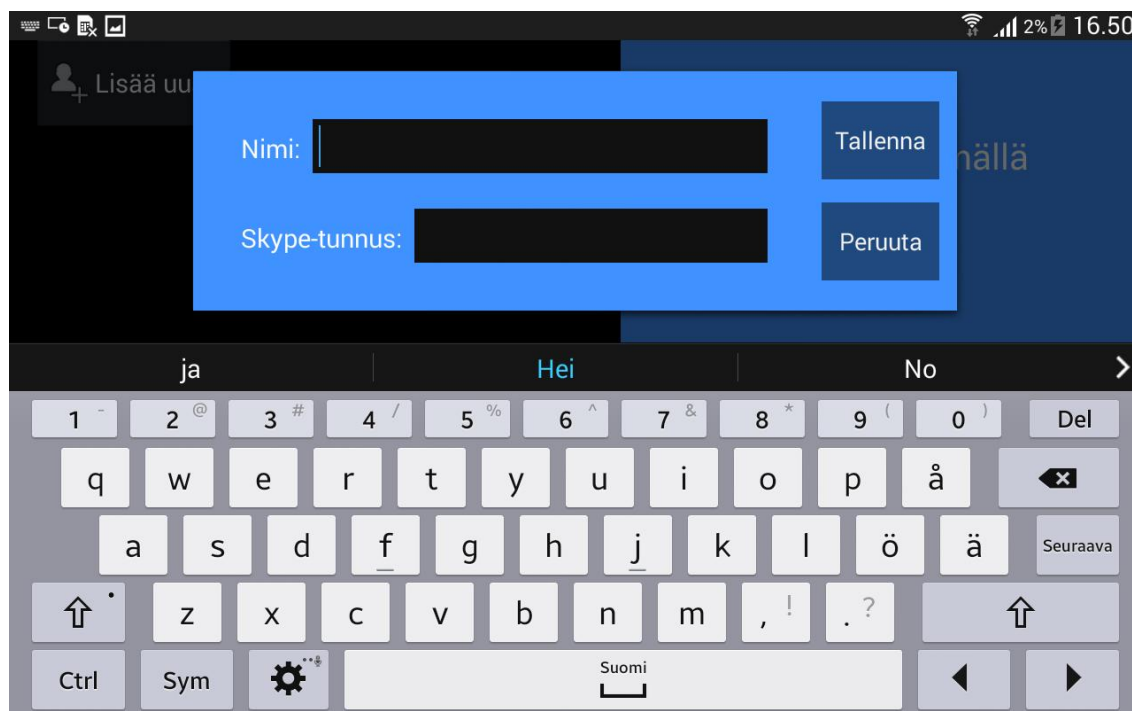
```
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

Henkilön poimiminen osoitekirjasta Four Buttonin yhteyshenkilölistalle ei kuitenkaan vaadi haun tekemistä tietokantaan, vaan henkilön tiedot saadaan onActivityResult-metodissa, kun käyttäjä on valinnut jonkin henkilön Androidin osoitekirjanäkymästä (kuva 9). Valitun henkilön tiedot voidaan sen jälkeen käsitellä halutulla tavalla ja päivittää käyttöliittymään.



Kuva 9. Androidin yhteyshenkilölista.

Jos käyttäjä haluaa lisätä yhteyshenkilön manuaalisesti, hänelle näytetään dialogi, jossa voi syöttää nimen ja Skype-tunnuksen (kuva 10).



Kuva 10. Manuaalinen yhteyshenkilön lisääminen.

Ennen lisäämistä Androidin osoitekirjaan Skype-tunnuksen esiintyminen tietokannassa tarkistetaan, jotta kopioita ei syntyisi.

Androidin tietokannassa jokaisella yhteyshenkilöllä on yksilöllinen tunnistenumero (Google Inc. 2015a). Jos käyttäjä lisää yhteyshenkilön manuaalisesti, on yhteyshenkilölle generoitava tunnistenumero. Tunnistenumerot ovat long-tyyppisiä kokonaislukuja Androidin tietokannassa, mutta generoidessa uutta tunnistetta integer-tyyppisen kokonaisluvun avaruus on riittävä.

Myös tunnistenumeron esiintyminen tietokannassa täytyy tarkistaa ennen kuin uusi henkilö voidaan kirjoittaa tietokantaan. Jos numero on jo käytössä tietokannassa, generoidaan uusi numero ja yritetään uudelleen.

### 3.4.3 Tiedostoon tallentaminen

Yhteys henkilölistan tietojen tulee säilyä myös silloin kun laite sammutetaan. Siksi tiedot varastoidaan pysyvästi laitteen sisäiseen massamuistiin kirjoittamalla ne tiedostoon.

Tietojen yksityisyyden turvaamiseksi kirjoittamisessa käytetylle Context-luokan metodille `openFileOutput(String name, int mode)` annettiin toisena parametrina vakio `Context.MODE_PRIVATE`, joka tarkoittaa kirjoittamista yksityisessä tilassa; muut sovellukset eivät voi lukea tiedostoa. Yksityinen tila ei kuitenkaan mahdollista tiedostoon kirjoittamista siten, että tietoa voidaan lisätä jo olemassa olevan tiedon perään samaan tiedostoon, vaan kaikki tieto pitää kirjoittaa tiedostoon yhdellä kertaa. (Google Inc. 2015d.)

Ratkaisuna on tallentaa kaikki yhteystiedot yhteen olioon, joka kirjoitetaan tiedostoon. Kukin yhteys henkilö sisältää neljä tietuetta:

- id: yksilöllinen tunnistenumero kokonaislukuna
- nimi: yhteys henkilön koko nimi merkkijonona
- skypeid: yhteys henkilön Skype-käyttäjänimi merkkijonona
- photo: tavutaulukoksi muunnettu yhteys henkilön valokuva

Nämä tietueet tallennetaan Person-luokasta luotuun olioon, jonka on oltava sarjallistettava (*eng. serializable*) tiedostoon kirjoittamista varten. Person-luokka sarjallistuu, kun sillä toteutetaan Serializable-rajapintaluokka (Oracle 2014). Kaikki Person-luokasta luodut oliot tallennetaan People-luokasta saatuun olioon. Myös People-luokka on sarjallistettava.

People-luokasta luotu olio voidaan kirjoittaa tiedostoon käyttämällä yksityistä tilaa. Tiedostosta lukemisen yhteydessä tiedon purkaminen tapahtuu päinvastaisessa järjestyksessä.

### 3.5 Kotisovellus

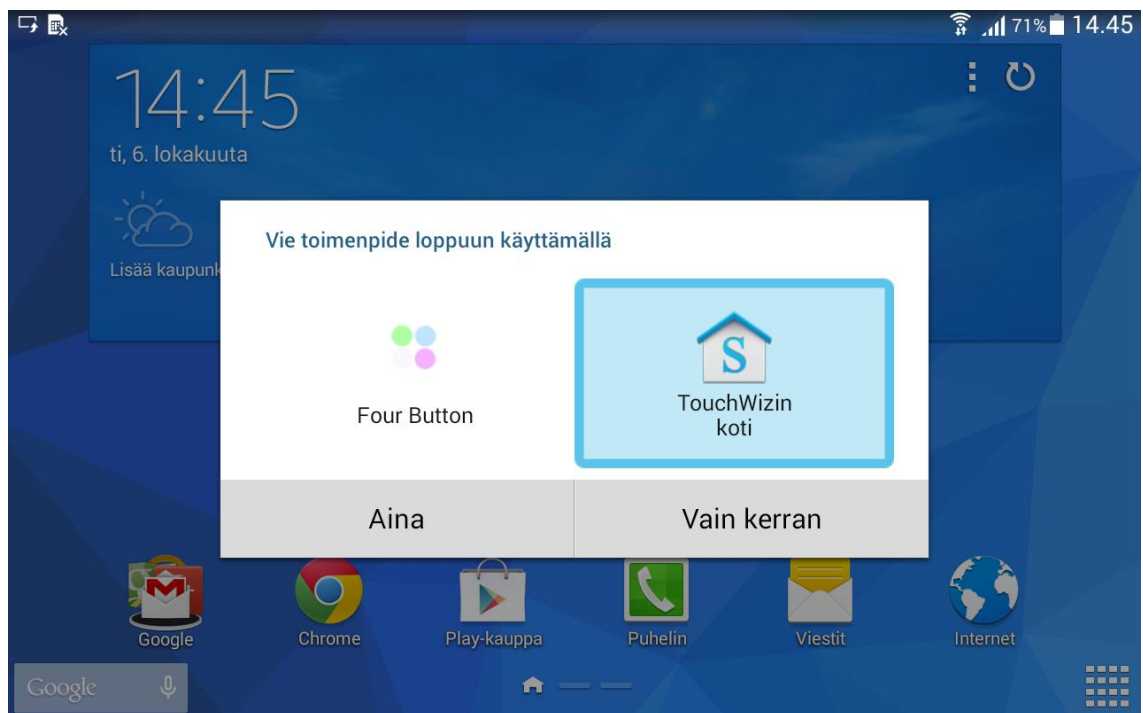
Sovelluksen saa asetettua laitteen kotisovellukseksi tekemällä pääaktiviteetille seuraavan merkinnän AndroidManifest.xml-tiedostossa:

```
<category android:name="android.intent.category.HOME" />
```

Tämä merkintä ilmoittaa käyttöjärjestelmälle, että sovellus on yhtenä vaihtoehtona käynnistettäväksi sovellukseksi, kun käyttäjä painaa kotipainiketta (Google Inc. 2015h). Jos oletusarvoista kotisovellusta ei ole vielä asetettu, näyttää käyttöjärjestelmä käyttäjälle valintadialogin (kuva 11).

Sovelluksen asettaminen kotisovellukseksi pysyvästi vaatii sovelluksen valitsemisen kyseisestä dialogista; tätä vaihetta ei voi ohittaa.

Kotisovelluksen valintadialogi voi olla ongelmallinen esimerkiksi siinä tapauksessa, että käyttäjän tarvitsee asentaa sovellus itse. Ainoa keino välttää valintadialogi on huolehtia siitä, että sovellus on asennettu etukäteen ja kotisovelluksen valinta on suoritettu.



Kuva 11. Kotisovelluksen valintadialogi.



Kuvan 11 dialogissa näkyy vaihtoehtona Four Buttonin lisäksi TouchWiz, joka on Samsung Galaxy -tableteissa esiasennettuna oleva kotisovellus (PhoneArena 2015). Valitsemalla Four Button ja painamalla "Aina" saadaan Four Button asetettua oletusarvoiseksi kotisovellukseksi, ja näin ollen korvaamaan laitteen valmistajan kotisovellus.

Tämän jälkeen Four Button tulee näkyviin kotipainiketta painaessa ja laitteen uudelleenkäynnistämisen jälkeen. Laitteen sammuttaminen tai uudelleenkäynnistäminen ei vaikuta kotisovelluksen asetettuun oletusarvoon. Oletuksena käytettävältä sovellukselta voi myöhemmin poistaa oletusstatuksen laitteen asetuksista.

### 3.6 Toisen sovelluksen käyttäminen videopuheluihin

Vaikka Four Button -sovelluksen tavoitteena on hallita laitetta, se ei kuitenkaan voi hallita muita sovelluksia, eikä suorittaa niitä ikkunassa. Toisia sovelluksia voi kuitenkin käynnistää, jos niiden tekijä on sallinut näin tapahtuvan. Toisia sovelluksia voi käynnistää käyttäen Intent-luokkaa, joka on abstrakti kuvaus suoritettavaksi tulevasta operaatiosta. (Google Inc. 2015h.)

Videopuheluiden demonstroiintiin käytettiin Skypeä. Intent-luokan setData-metodiin annetaan parametriksi Skypen omaa skeemaa käyttävä URI, joka sisältää yhteyshenkilön Skype-tunnuksen ja muut tarvittavat parametrit (kuva 12). Lopuksi kutsutaan Context-luokan startActivity-metodia, jolloin videopuhelu käynnistyy heti.

```

public void onSoitaButtonClicked(View v)
{
    if (!selectedSkype.equals("0") && !selectedSkype.equals("")){
        Intent skype = new Intent("android.intent.action.VIEW");
        skype.setData(Uri.parse("skype:" + selectedSkype + "?call&video=true"));
        startActivity(skype);
    }
    else{
        showCallRefusedWarning();
    }
}

```

Kuva 12. Koodiesimerkki: toisen sovelluksen käynnistäminen.

Yhteyshenkilölle soittaminen tapahtuu Skypen käyttöliittymässä. Näkymä, jossa puhelu tapahtuu, tulee puhelun ajaksi taustalta näkyviin. Four Button menee puhelun ajaksi taustalle.

Yleensä puhelun päättymisen jälkeen Skypen näkymä menee takaisin taustalle, ja Four Buttonin näkymä palaa päällimmäiseksi. Käytännön kokeiden perusteella näin ei käy tilanteessa, jossa Skype on ollut taustalla riittävän kauan. Jos puhelu tuo Skypen taustalta esiin ensimmäistä kertaa pitkän ajan jälkeen, jää Skypen käyttöliittymä näkyviin puhelun päätyttyä. Muun muassa tämän syyn vuoksi Skypeä voitiin käyttää vain kokeilumielessä.

Tarkkaa mittaustulosta siitä, kuinka kauan Skypen täytyy olla taustalla aiheuttaakseen em. ongelman, ei ole, mutta yleensä ongelma ilmeni kun tabletti oli ollut käyttämättömänä pöydällä vähintään puoli tuntia.

Skypen käyttöliittymän ollessa näkyvissä oli laitteen fyysisten painikkeiden hallinta mahdotonta. Tämän takia käyttäjän oli mahdollista painaa paluupainiketta Skype-puhelun ollessa käynnissä. Four Buttonin käyttöliittymä tuli taustalta esiin, mutta Skype-puhelu jäi näkyviin pieneen ikkunaan päällimmäiseksi. Ratkaisuksi ongelmaan ei löytynyt muuta kuin paluupainikkeen painamisen välttäminen Skype-puhelun aikana.

### 3.7 Oletusnäköymästä poistuminen

Sovelluksessa oli oltava huoltotoimenpiteitä varten keino poistua käyttöliittymästä esimerkiksi laitteen asetuksien muokkaamista varten. Laitteen valikkopainikkeella avautuvaan valikkoon on lisätty vaihtoehto "Admin". Sitä painamalla aukeaa dialogi, jossa kysytään salasanaa. Jos käyttäjä tietää asetetun salasanan, hänelle näytetään kotisovelluksen valintadialogi. Valitsemalla alkuperäisen kotisovelluksen voi laitetta jälleen käyttää normaalisti. Four Buttonin voi palauttaa myöhemmin oletusarvoiseksi kotisovellukseksi laitteen asetuksista.

Salasana on tallennettu selkokiekisenä Strings.xml-tiedostoon, josta sitä verrataan suoraan käyttäjän antamaan syötteeseen. Selkokiekisenä tallennetun salasanan voisi ajatella olevan tietoturvariski, mutta oikeastaan se ei ole ainakaan huomattava sellainen. Jos jollakin henkilöllä on riittävästi osaamista salasanan löytämiseen, kykenee hän ohittamaan salasanaäkyselyn helpommallakin keinolla. Sen sijaan tahattoman käyttöliittymästä poistumisen estämiseen salasanavarmistuksen toteutus on riittävän turvallinen.

### 3.8 Ammattilaisnäköymä

Four Button ei sisällä valmista ammattilaisnäköymää, koska sitä ei määritelty riittävän tarkasti, mutta valmius näköymälle on olemassa. Pääsy näköymään tapahtuu samasta valikosta kuin poistuminen Four Buttonista.

Ammattilaisnäköymän tarkoitus on mahdollistaa hoitajalle pääsy esimerkiksi potilastietojärjestelmään tarpeen niin vaatiessa. Hoitajien ei ole tarkoitus käyttää samaa käyttöliittymää kuin asiakkaiden, vaan ammattilaiset käyttävät käyttöliittymän ammattilaisversiota omalla tabletillaan. Four Buttonin ammattilaisnäköymän ominaisuuksia voidaan hyödyntää ammattilaisten tableteissa.

Tärkeänä osana ammattilaisnäkömää on QR-koodien lukija. Sen tarkoituksena on nopeuttaa tiedon syöttämistä laitteeseen ja eliminoida kirjoitusvirheiden mahdollisuus. Esimerkiksi osa asiakkaan henkilötiedoista voidaan syöttää laitteeseen lukemalla QR-koodi.

Koodien lukemiseen käytetään moduulia, joka on peräisin avoimen lähdekoodin ZXing-kirjastosta (Smith 2013). Moduuli sisältää kameran käyttöön tarvittavan käyttöliittymän eli kameranäkymän. Näkymän avulla laitteen kamera tähdätään QR-koodiin, jolloin moduuli tunnistaa koodin ja lukee sen sisällön. Sisältö on sen jälkeen käytettävissä halutulla tavalla.

## 4 LOPPUTULOKSEN ARVIOINTI

Suurimmat Androidin asettamat haasteet olivat seuraavat:

1. ilmoituspaneelin avaamisen estäminen
2. tilapalkin piilottaminen ja piilotettuna pitäminen
3. oman tilapalkin luominen
4. laitteen painikkeiden hallinta ja poistumisen estäminen sovelluksesta
5. muiden sovellusten käyttö
6. sovelluksen asettaminen kotisovellukseksi.

Onnistuin luomaan toimivan ja helppokäyttöisen käyttöliittymän haasteista huolimatta, mutta en ilman heikkouksia tai virheitä. Ilmoituspaneelin avaamisen estäminen ei onnistunut täydellisesti, mutta käyttötarkoitusta ajatellen toteutus oli tyydyttävä. Androidin versiossa 4.4 ongelma on onneksi korjaantunut, koska kokoruudun tilan säilyttäminen toimii siinä aggressiivisemmin, enkä ole ruutua pyyhkäisemällä onnistunut avaamaan ilmoituspaneelia kertaakaan.

Yhteyshenkilönäkymän jakamiseen fragmenteilla suhtaudun kriittisesti siinä mielessä, että tässä tapauksessa fragmenttien käytölle ei ole erityisiä perusteita. Yhteyshenkilösivun jaetun ruudun olisi pystynyt toteuttamaan pelkällä asettelullakin, koska samojen fragmenttien käyttöä muissa näkymissä ei luultavasti tule tapahtumaan. Toisaalta fragmenttien käytöstä ei ollut mitään haittaakaan.

Videopuhelutoiminnon toteutus Skypellä aiheutti hankaluuksia. Puhelut toimivat sinänsä hyvin, mutta Skypen käyttö vaatii Skype-tilin luomista etukäteen Skype-asiakassovelluksesta. Sama pätee kaveripyyntöjen lähettämiseen tai hyväksymiseen. Four Buttonin käyttöliittymä mahdollistaa yhteyshenkilön

lisäämisen manuaalisesti Skype-tunnuksen perusteella, mutta ei kaveripyynnön lähettämistä lisätulle henkilölle.

Vieläkin suurempi ongelma aiheutuu, jos Skype-asiakassovellus lopettaa toimintansa tai kirjaa käyttäjätilin ulos. Sisään kirjautuminen tai sovelluksen uudelleenkäynnistäminen ovat mahdottomia tehdä Four Buttonin käyttöliittymästä käsin. Tavallisesti Skype-asiakassovellus käynnistyy ja kirjautuu sisään automaattisesti laitteen käynnistämisen jälkeen ja pysyy sisään kirjautuneena niin kauan kun laite on päällä. Vikatilanteiden tai automaattisten päivitysten vuoksi tilanne voi yllättäen muuttua.

Androidin kehitystahti on nopea. Tämä raportti käsittelee versioita 4.1 – 4.3, mutta kirjoitushetkellä versio 5.0 on ollut asennettavissa jo parin kuukauden ajan, ja kehitystyökalut versiota 6.0 varten on jo julkaistu. Vaikka Androidin pääpiirteet ovat toistaiseksi säilyneet miltei muuttumattomina, on tämän raportin tieto silti pääosin vanhentunutta. Sama kohtalo lienee muillakin kirjallisilla tuotoksilla, jotka ovat jollain tavalla sidottuja Androidin versionumeroon.

## LÄHTEET

- Amadeo, R. 2014. The history of Android. Viitattu 4.10.2015 <http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/6/#1.0>
- APC 2011. Android Masterclass: Building a simple Android app. Viitattu 5.10.2015 <http://apcmag.com/building-a-simple-android-app.htm>
- Bornstein, D. 2008. Dalvik VM Internals. Viitattu 4.10.2015. <https://sites.google.com/site/io/dalvik-vm-internals>
- Carroll, J. M. & Reitman, J. 1987. Mental Models in Human-Computer Interaction. Washington D.C.: National Academy Press.
- Deep, R. 2013. What A Native Developer Should Know About Android Security? Viitattu 5.10.2015 <http://opensourceforu.ifytimes.com/2013/09/what-a-native-developer-should-know-about-android-security>
- Google Inc. 2015a Accessing Contacts Data. Viitattu 4.10.2015 <https://developer.android.com/training/contacts-provider/index.html>
- Google Inc. 2015b Best Practices for User Interface: Hiding the Status Bar. Viitattu 4.10.2015 <https://developer.android.com/training/system-ui/status.html>
- Google Inc. 2015c ContactsContract.Contacts. Viitattu 6.10.2015 <http://developer.android.com/reference/android/provider/ContactsContract.Contacts.html>
- Google Inc. 2015d Context. Viitattu 5.10.2015 <http://developer.android.com/reference/android/content/Context.html>
- Google Inc. 2015e Draw 9-patch. Viitattu 4.10.2015 <http://developer.android.com/tools/help/draw9patch.html>
- Google Inc. 2015f Fragments. Viitattu 5.10.2015 <http://developer.android.com/guide/components/fragments.html>
- Google Inc. 2015g Handling Runtime Changes. Viitattu 3.11.2015 <http://developer.android.com/guide/topics/resources/runtime-changes.html>
- Google Inc. 2015h Intent. Viitattu 5.10.2015 <http://developer.android.com/reference/android/content/Intent.html>
- Google Inc. 2015i Introduction: Application Fundamentals. Viitattu 4.10.2015 <http://developer.android.com/guide/components/fundamentals.html>
- Google Inc. 2015j Managing the Activity Lifecycle: Starting an Activity. Viitattu 4.10.2015 <http://developer.android.com/training/basics/activity-lifecycle/starting.html>
- Gypss, M. 2014. Pixel Perfect Precision Handbook 3. Surrey: Ustwo Studio Ltd. <https://ustwo.com/blog/the-ustwo-pixel-perfect-precision-handbook-3>
- Hildenbrand, J. 2015. What is Android? Viitattu 17.8.2015 <http://www.androidcentral.com/what-android>

Hooper, S. & Berkman, E. 2011. Designing Mobile Interfaces. First edition. Sebastopol: O'Reilly Media Inc.

International Data Corporation 2015. Smartphone OS Market Share. Viitattu 27.9.2015  
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

Kanaujia, V. 2011. Virtual Machines for Abstraction: The Dalvik VM. Viitattu 2.10.2015  
<http://opensourceforu.ifytimes.com/2011/06/virtual-machines-for-abstraction-dalvik-vm>

Lidwell, W.; Holden, K. & Butler, J. 2010. Universal Principles of Design, Revised and Updated: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach through Design. Second edition. Beverly: Rockport Publishers.

Lockwood, A. 2013. Handling Configuration Changes with Fragments. Viitattu 3.11.2015  
<http://www.androiddesignpatterns.com/2013/04/retaining-objects-across-config-changes.html>

Morrill, D. 2008. Inside the Android Application Framework. Viitattu 9.10.2015  
<https://sites.google.com/site/io/inside-the-android-application-framework>

Oracle 2014. Interface Serializable. Viitattu 4.10.2015  
<http://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html>

PhoneArena 2015. 15 TouchWiz features that make the day-to-day Android experience better. Viitattu 4.10.2015  
[http://www.phonearena.com/news/15-TouchWiz-features-that-make-the-day-to-day-Android-experience-better\\_id67067](http://www.phonearena.com/news/15-TouchWiz-features-that-make-the-day-to-day-Android-experience-better_id67067)

Saarelma, O. 2015. Värisokeus ja poikkeava värinäkö. Viitattu: 6.10.2015  
[http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p\\_artikkeli=dlk00347](http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=dlk00347)

Shore, J. & Warden, S. 2008. The Art of Agile Development. Sebastopol: O'Reilly Media Inc.

Smith, S. 2013. Android SDK: Create a Barcode Reader. Viitattu 3.11.2015  
<http://code.tutsplus.com/tutorials/android-sdk-create-a-barcode-reader--mobile-17162>

Yang, H. 2012. Introduction of Activity Lifecycle. Viitattu 14.11.2015  
<http://www.herongyang.com/Android/Activity-Introduction-of-Activity-Lifecycle.html>



## KUVAT

Kuva 1. Android-sovelluksen elinkaari (Yang 2012).	11
Kuva 2. Käyttöliittymän oletusnäkyvä.	12
Kuva 3. Koodiesimerkki: kokoruututilan käyttäminen.	14
Kuva 4. Koodiesimerkki: valikoiden piilottaminen.	15
Kuva 5. Koodiesimerkki: AlphaAnimation-luokka.	16
Kuva 6. Tyhjä yhteyshenkilölista.	18
Kuva 7. Yhteyshenkilölista, jossa henkilö on valittu.	18
Kuva 8. Dialogi-ikkuna: uuden henkilön lisääminen.	19
Kuva 9. Androidin yhteyshenkilölista.	20
Kuva 10. Manuaalinen yhteyshenkilön lisääminen.	21
Kuva 11. Kotisovelluksenvälintadialogi.	23
Kuva 12. Koodiesimerkki: toisen sovelluksen käynnistäminen.	24