

Miikka Nordlund

# OSTOSLISTA ANDROID-SOVELLUKSEN KEHITTÄMINEN

Tietotekniikan koulutusohjelma  
2016

# OSTOSLISTA ANDROID-SOVELLUKSEN KEHITTÄMINEN

Nordlund, Miikka  
Satakunnan ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Toukokuu 2016  
Ohjaaja: Ekholm, Ari  
Sivumäärä: 27  
Liitteitä: 0

Asiasanat: Android, Ostoslista, Ohjelmistosuunnittelu

---

Android on tällä hetkellä suosituin mobiilikäyttöjärjestelmä, ja sitä käytetään erilaisissa laitteissa kuten puhelimissa, taulutietokoneissa ja televisioissa. Androidin etu muihin mobiilikäyttöjärjestelmiin nähden on sen ohjelmistokehityksen helppous ja edullisuus. Lisäksi tarjolla on ilmaiset kehitystyökalut sekä tuki ohjelmiston tuomiseksi maailmanlaajuisille markkinoille.

Tässä opinnäytetyössä tutkitaan case-menetelmällä ohjelmistokehitystä Android-ympäristössä. Työn aikana kehitetään sovellus nimeltä Ostoslista – Säästä rahaa kaupoilla. Tarkoituksena on kehittää sovellus, jonka avulla käyttäjä pystyy syöttämään sovellukseen tuotteiden hintoja eri kaupoissa, jolloin ostoslistaa tehdessä sovellus suosittelee käyttäjälle edullisinta kauppaa. Sovelluksessa on asetus, jolla hinnat ja kaupat voi kytkeä pois käytöstä, jolloin sovellusta voi käyttää tavallisena ostoslistana. Sovelluksessa on myös tuki Android Wear -älykellolle, jonka avulla käyttäjä näkee ostoslistan suoraan kellostaan, mikä vapauttaa kädet ostosten tekemiseen.

Työn tuloksena saatiin osoitettua, että Android sopii erinomaisesti mobiililaitteiden ohjelmistokehitykseen. Kehitetty ohjelmisto toimii, ja se on saatavilla Google Play Kaupasta. Ainoa ulkoisesti maksettu korvaus on 25 USD rekisteröitymisestä viralliseksi sovelluskehittäjäksi.

# DEVELOPING AN ANDROID APPLICATION CALLED SHOPPING LIST

Nordlund, Miikka

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information Technology

May 2016

Supervisor: Ekholm, Ari

Number of pages: 27

Appendices: 0

Keywords: Android, Shopping list, Software development

---

Android is currently the most popular mobile operating system, which is used in various devices such as phones, tablet computers and televisions. Advantage of Android over other mobile operating systems is the ease and affordability of software development. In addition, there are free development tools and support to bring the developed software to global market.

This thesis explores software development for Android-platform with a case-study. The case is developing an Android application called ShoppingList – Save money when shopping. The aim is to develop an application that allows the user to enter product prices in different stores, so the application can recommend a store that saves money. The application has a setting which can turn off product prices and shops, so it can also be used as a standard shopping list. The application has support for Android Wear smartwatch, which allows the user to see the shopping list directly on a watch, freeing the user's hands while shopping.

The results show that Android is ideal for software development. The developed application works, and is available from the Google Play Store. The only expense was 25 USD for registering as an official software developer for Android.

## TERMIT JA LYHENTEET

ADT	Android Development Tools, eclipse ohjelmointiympäristöön lisättävä lisäosa, jotta eclipsellä voidaan ohjelmoida Android sovelluksia.
API	Application Programming Interface, Ohjelmointirajapinta, jonka mukaan eri ohjelmat voivat keskustella keskenään
APK	Android application package, Android sovelluksen asennuspaketti
AVD	Android Virtual Device, Virtuaalinen Android emulaattori
NDK	Native Development Kit, Työkalu jonka avulla voi käyttää C ja C++ ohjelmointikieltä Android ohjelmoinnissa
NFC	Near Field Communication
Widget	Minimaalinen sovellus, joka voidaan upottaa muihin sovelluksiin, kuten aloitusnäyttöön

# SISÄLLYS

1	JOHDANTO.....	6
2	ANDROID KÄYTTÖJÄRJESTELMÄ .....	6
2.1	Mikä on Android? .....	6
2.2	Androidin synty .....	7
2.3	Versiohistoria ja suurimmat uudistukset.....	7
3	SOVELLUSTEN KEHITTÄMINEN ANDROID KÄYTTÖJÄRJESTELMÄLLE .....	8
3.1	Sovellusten perusteet .....	8
3.2	Sovelluksen elinkaari .....	9
3.3	Dynaamisen käyttöliittymän toteutus .....	10
3.4	Käytettävät ohjelmointikielet.....	11
3.5	Kehitysympäristö .....	12
3.6	Sovelluksen ulkonäön suunnittelu .....	12
3.7	Sovelluksen API-tason valinta.....	14
3.8	Sovelluksen testaaminen .....	15
4	OSTOSLISTASOVELLUKSEN KEHITTÄMINEN .....	16
4.1	Ostoslista sovellus.....	16
4.1.1	Suunnitteluperiaatteet .....	16
4.1.2	Sovelluksen käyttö.....	17
4.2	Älykello sovellus .....	22
5	SOVELLUKSEN JULKAISU .....	24
6	LOPPUTULOS .....	26
	LÄHTEET.....	27

## 1 JOHDANTO

Suomessa tuotteiden hinnat vaihtelevat suuresti eri kauppojen välillä, ja ihmisten on vaikeaa säästää rahaa ostoksia tehdessään, koska heillä ei välttämättä ole tietoa samankaltaisten tuotteiden hintaeroista eri kaupoissa. ”Merkille pantavaa on, että saman kaupungin sisällä hinnat voivat vaihdella kauppaketjusta riippuen hyvinkin merkittävästi.” (Jakosuo, 2015)

Ostoslistasovelluksia on tarjolla monia kymmeniä sovelluskaupoissa, mutta yksikään ei käsittele tuotteiden hintoja eri kauppojen välillä. Tässä opinnäytetyössä kehitetty sovellus pyrkii ratkaisemaan tämän ongelman antaen käyttäjän syöttää hintoja eri kaupoissa samalle tuotteelle. Ostoslistaa tehdessään sovellus suosittelee edullisinta kauppaa. Tavoitteena on kehittää yksinkertainen ja helppokäyttöinen, mutta silti monipuolinen sovellus.

Sovellukseen tulee myös mahdollisuus ottaa hinta-ominaisuus pois käytöstä, jolloin sovellusta voi halutessaan käyttää tavallisena ostoslistana.

Toinen osa kehitettävää sovellusta oli Android-älykellolle tehtävä sovellus. Tämän avulla käyttäjä voi seurata ostoslistaa suoraan omasta kellostaan ja merkitä tuotteen listalta ostetuksi suoraan kellosta ottamatta puhelintaan esille. Älykello-sovellus siirtää tiedon käyttäjän puhelimeen, joten tuotetta ei tarvitse enää erikseen merkitä puhelimesta ostetuksi.

## 2 ANDROID-KÄYTTÖJÄRJESTELMÄ

### 2.1 Mikä on Android?

Android on hakukonejätti Googlen kehittämä Linux kerneliin perustuva käyttöjärjestelmä. Android on kehitetty pääosin käytettäväksi kosketusnäytöltä, ja se kehitettiin alun perin vain puhelimelle. Nykypäivänä Android-käyttöjärjestelmä

löytyy monista eri valmistajan puhelimista, televisioista, autoradioista, tableteista, kannettavista tietokoneista ja älykelloista. (Rouse, 2008)

Android on täysin avoimeen lähdekoodiin perustuva käyttöjärjestelmä, joten kuka hyvänsä voi käydä hakemassa lähdekoodin ilmaiseksi, tehdä muutoksia siihen ja julkaista oman versionsa. Androidin lähdekoodiin ja ohjeistuksiin voi käydä tutustumassa source.android.com- sivustolla. (Hildenbrand, 2015)

## 2.2 Androidin synty

Android syntyi lokakuussa 2003, kun Andy Rubin, Rich Miner, Nick Sears, ja Chris White perustivat startup yrityksen nimeltä Android Co. Tavoitteena oli kehittää matkapuhelin, joka olisi tietoinen käyttäjän sijainnista ja tottumuksista. (Brachmann, 2014)

Google osti Android Co:n elokuussa 2005, jonka jälkeen se on ollut Googlen omistuksessa kokonaisuudessaan. (Brachmann, 2014)

## 2.3 Versiohistoria ja suurimmat uudistukset

Android on ensimmäisiä käyttöjärjestelmiä, joissa kehityssykli muistuttaa enemmän verkkosivua kuin käyttöjärjestelmää. Esimerkiksi Windows käyttöjärjestelmän uusi versio tulee kolmen – viiden vuoden välein, mutta Androidin kehityssykli on huomattavasti nopeampi. (Amadeo, 2014)

Taulukko 1. Androidin eri versiot ja niiden tuomat uudistukset (The Android Story, 2016)

<b>Versio</b>	<b>Koodinimi</b>	<b>Julkaistu</b>	<b>Tärkeimmät uudistukset</b>
<b>1.6</b>	Donut	15.9.2009	Tuki erikokoisille näytöille
<b>2.1</b>	Eclair	26.10.2009	Google maps navigointi; kotinäytön personointi; puheesta tekstiksi

<b>2.2</b>	Froyo	20.5.2010	Ominaisuuksien aktivointi puheella; mahdollisuus käyttää puhelinta wi-fi reitittimenä
<b>2.3</b>	Gingerbread	6.12.2010	API tuki peleille; NFC tuki
<b>3.0</b>	Honeycomb	22.2.2011	Tablet-laitteille suunniteltu versio; Fyysisten näppäinten vaihto System Bar-nimiseen toimintoon;
<b>4.0</b>	Ice Cream Sandwich	18.10.2011	Sama käyttöjärjestelmä tableteille ja puhelimille; Koti-näytön kustomointi; Mobiilidatan käytön rajoitus
<b>4.1</b>	Jelly Bean	9.7.2012	Tuki monelle käyttäjälle yhdellä laitteella; Google Nyt-palvelu
<b>4.4</b>	Kit Kat	31.10.2013	Koko näytön tila sovelluksille; Puhetunnistuksen aktivointi puheella sanomalla ”OK Google”
<b>5.0</b>	Lollipop	12.11.2014	Material Design -niminen tyyli kaikille sovelluksille; Tuki televisioille
<b>6.0</b>	Marshmallow	5.10.2015	Sovellusten oikeuksien hallinta käyttäjille; Älykäs akunhallinta

Näiden suurimpien päivitysten välillä on vielä monta pienempää päivitystä. Yllä olevasta taulukosta voidaan kuitenkin todeta, että versiokehitys on todella nopeaa.

### 3 SOVELLUSTEN KEHITTÄMINEN ANDROID KÄYTTÖJÄRJESTELMÄLLE

#### 3.1 Sovellusten perusteet

Jokainen sovellus suoritetaan omassa suljetussa virtuaalisessa ympäristössä, joten mikään muu sovellus ei pysty kommunikoimaan kehitetyn sovelluksen kanssa, ellei kehittäjä itse tee tarvittavia yhteyksiä. (Application Fundamentals, 2016)



Sovellus koostuu neljästä eri rakennuspalikasta, jotka ovat: Aktiviteetit, palvelut, sisällöntuottajat ja lähetysten vastaanottajat. (Application Fundamentals, 2016)

Aktiviteetti on yksittäinen näyttö sovelluksessa. Esimerkiksi sähköpostisovelluksessa yksi aktiviteetti on lista, joka näyttää saapuneet viesti. Toinen aktiviteetti on näyttö, jossa kirjoitetaan sähköpostia ja niin edelleen. (Activities, 2016)

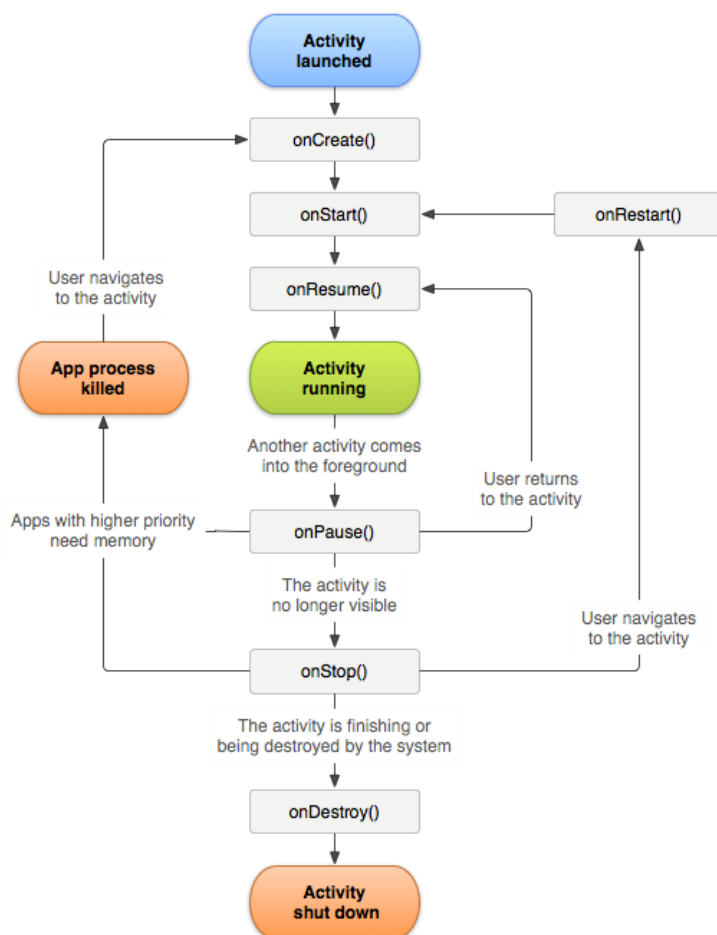
Palvelu on komponentti, joka toimii taka-alalla ja jota käyttäjä ei koskaan näe. Esimerkiksi palvelu soittaa sovelluksesta musiikkia, kun käyttäjä on toisessa sovelluksessa. (Services, 2016)

Sisällöntuottajalla voidaan toteuttaa linkki esimerkiksi toisiin sovelluksiin, internetissä säilytettyyn tietoon ja puhelimessa sijaitsevaan tietokantaan. (Content Providers, 2016)

Lähetysten vastaanottaja vastaa järjestelmän laajuisesti lähetettyyn ilmoitukseen. Näitä ilmoituksia on monenlaisia, joista esimerkkejä ovat ”näyttö on kytketty pois päältä”, ”akku on vähissä” tai myös jokin lähetys, jonka sovelluksen kehittäjä on itse tehnyt. Lähetysten vastaanottajaan määritellään, mitä viestejä sovellus vastaanottaa ja toteutetaan toiminnallisuus näiden lähetysten sattuessa. (Application Fundamentals, 2016)

### 3.2 Sovelluksen elinkaari

Jokaisessa sovelluksessa, joka näkyy käyttäjän näytöllä, on yksi tai useampia aktiviteetteja. Jos sovelluksessa on enemmän kuin yksi aktiviteetti, niin AndroidManifest.xml-tiedostoon määritellään yksi pääaktiviteetti, joka käynnistyy sovelluksen käynnistyessä.



Kuva 1 Aktiviteetin elinkaari (Activities, 2016)

Kuvassa 1 on esiteltyä aktiviteetin elinkaari ja kaikki kutsut, joita aktiviteetti elinkaarensa aikana ajaa. Funktiot onCreate(), onStart(), onResume(), onStop(), onRestart() ja onDestroy() ajetaan aina sovelluksen elinkaaren aikana. Sovelluksen kehittäjä voi lisätä omaa koodiansa jokaiseen näistä funktiokutsuista. Sovelluksessa voidaan esimerkiksi tallentaa jonkin tarvittavan tiedon samalla, kun aktiviteetti suljetaan.

### 3.3 Dynaamisen käyttöliittymän toteutus

Androidin versiossa 3.0 esiteltiin käyttöliittymäsuunnittelua helpottava moduuli nimeltä fragmentti. Fragmentti on osa aktiviteettia, jonka voi vaihtaa dynaamisesti joutumatta vaihtamaan aktiviteettiä. Näin ei jouduta kirjoittamaan koko käyttöliittymää erikseen puhelimelle ja tabletille. (Fragments, 2016)



Kuva 2 Fragmenttien sijoittaminen taulutietokoneella ja puhelimella (Fragments, 2016)

Vasemmalla kuvassa taulutietokoneella ovat fragmentit A ja B aktiviteetissa A. Puhelimella fragmentti A on aktiviteetissa A ja fragmentti B aktiviteetissä B. Ennen fragmenttien tuloa Androidiin olisi pitänyt kirjoittaa molemmille laitteille erikseen fragmenttien A ja B toiminnallisuus. Fragmenteilla toteutettuna kaikki sovelluksen toiminnallisuus kirjoitetaan kerran fragmentteihin A ja B, joita eri laitteiden aktiviteetit kutsuvat.

### 3.4 Käytettävät ohjelmointikielet

Androidin natiivisti tukema ohjelmointikieli on Java. Käyttämällä Android NDK-kirjastoa voidaan ohjelmointi tehdä C ja C++ ohjelmointikielillä. Ohjeistus kuitenkin on, että Android NDK kirjaston tuoma tuki C ja C++ ohjelmointikielille ei tuo hyötyä suurimmalle osalle sovelluksista. Sitä ei tulisi käyttää vain sen takia, että ohjelmoija haluaa käyttää enemmän jotakin muuta ohjelmointikieltä kuin Java (Sims, 2016)

Muitakin ohjelmointikieliä voi käyttää kolmansien osapuolien ohjelmointikirjastojen avulla. Näitä ovat esimerkiksi LUA ja HTML/CSS. Ohjeistus kuitenkin on, että aina kun mahdollista, kaikki ohjelmointi Androidille tulisi tehdä Javalla ja käyttöliittymän suunnittelu XML-ohjelmointikielillä. Mitään muuta ohjelmointikieltä ei tulisi käyttää kuin vain silloin, kun se on välttämätöntä sovelluksen toimimisen kannalta. (Sims, 2016)

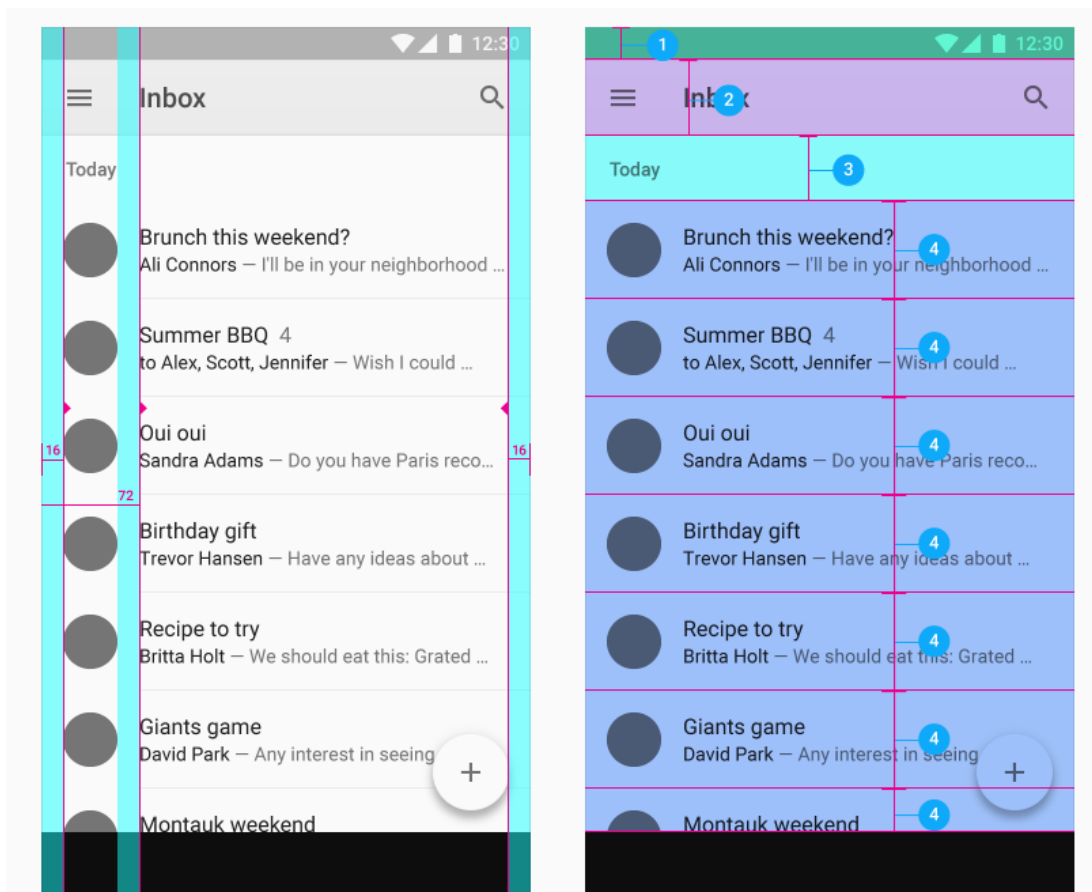
Android tukee myös tietokantojen käyttöä natiivisti ohjelmointikielellä SQLite. Tämä tehdään Javalla käyttäen hyväksi ohjelmointiluokkaa SQLiteOpenHelper. (Saving Data in SQL Databases, 2016)

### 3.5 Kehitysympäristö

Kehitysympäristöksi voi valita montaa eri ohjelmointikieltä tukevan Eclipsen, jolloin siihen pitää asentaa ADT-lisäosa. Toinen mahdollisuus on valita ainoastaan Android-ohjelmointiin tarkoitettu Android Studio. Ennen kuin Android Studio julkaistiin, Eclipse oli ainoa vaihtoehto. Nykyisin on suositeltavaa käyttää Android Studiota. (Meet Android Studio, 2016)

### 3.6 Sovelluksen ulkonäön suunnittelu

Samalla kun Android-versio 5.0 julkaistiin, julkaistiin myös ohjeistukset sovellusten ulkonäön suunnitteluun nimellä Material Design. Tämän tavoitteena on luoda yhtenäinen kokemus koko käyttöjärjestelmästä, jolloin kaikkien sovellusten ulkonäöt muistuttaisivat toisiaan. Material Designin julkaisun jälkeen kaikkien sovelluskehittäjien tulisi noudattaa näitä ohjeistuksia. Ohjeistukset löytyvät [developer.android.com](http://developer.android.com)-sivustolta, Design välilehdeltä. (Material Design for Android, 2016)



Kuva 3 Material Design -käyttöliittymän ohjeistuksia (Material Design for Android, 2016)

Yllä olevassa kuvassa on ohjeistukset käyttöliittymän eri osioiden kokojen tekemiseen.

Grey	
500	#9E9E9E
50	#FAFAFA
100	#F5F5F5
200	#EEEEEE
300	#E0E0E0
400	#BDBDBD
500	#9E9E9E
600	#757575
700	#616161
800	#424242
900	#212121

Kuva 4 Material Design -väripaletti esimerkki (Material Design for Android, 2016)

Kuvassa 4 on esimerkki sovelluksessa käytettävistä väreistä, jos pääväriksi on valittu harmaa. [developer.android.com/design-sivustolla](http://developer.android.com/design-sivustolla) on enemmän ohjeistuksia värien käyttöön, ja milloin mitäkin sävyä väreistä tulisi käyttää.

Tällä hetkellä suurin osa sovelluksista käyttää jo Material Design -tyyliä. Tämä tuo yhteneväisyyttä sovellusten välillä ja luo yhteneväisen käyttökokemuksen koko käyttöjärjestelmään.

### 3.7 Sovelluksen API-tason valinta

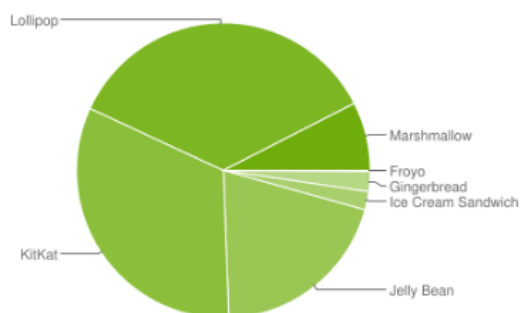
Jokainen Android-versio on tuonut mukanaan uuden API-tason, mikä on tuonut mukanaan uusia käyttötapoja ja toimintoja. Sovellukseen määritetään minimi API-taso, jolla sovellus toimii. Jos käyttäjällä on alemman tason Android-versio, hän ei voi asentaa sovellusta, tai edes löytää sovellusta laitteellaan Google Play Kaupasta.

Vanhemmilla tasoilla käyttäjiä on kuitenkin vielä paljon, joten sovelluksen kehittäjän tulee valita haluaako hän enemmän uusia toimintoja vai enemmän mahdollisia käyttäjiä. Jos sovellukseen on välttämätöntä käyttää esimerkiksi API-tason 18 toimintoja, niin sovellukseen pitää määrittää minimitasoksi 18. API-taso määritetään Android Studioissa build.gradle-tiedostoon.

```
<uses-sdk android:minSdkVersion="integer"
          android:targetSdkVersion="integer"
          android:maxSdkVersion="integer" />
```

Kuva 5 API-tason määrittely Android Studioissa

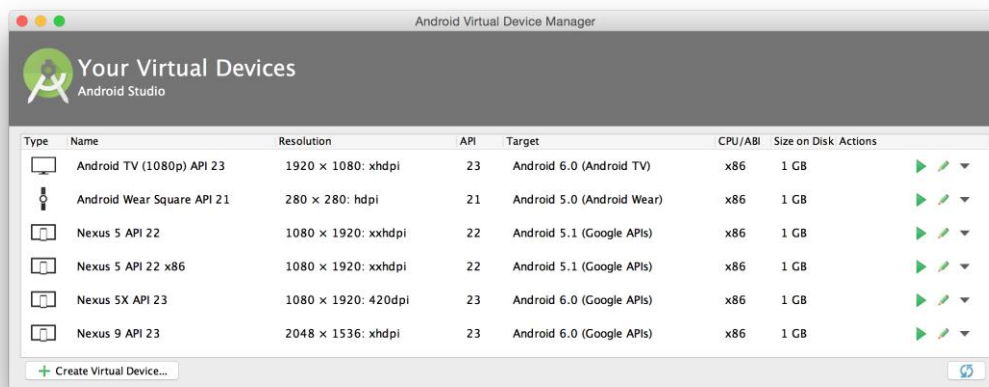
Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%



Kuva 6 Android-versioiden käyttöaste, kuvassa oleva mittaus on päivätty 2.5.2016 (Platform Versions, 2016)

### 3.8 Sovelluksen testaaminen

Sovellusten testaamiseen kannattaa ensiksi käyttää Android Studioissa olevaa AVD-ympäristöä. AVD-ympäristössä voi luoda tietokoneelle Android-laitteita, joihin voi valita haluamansa Android-version ja näyttökoon. Tällöin sovelluksen toimivuuden testaaminen eri Android-versioilla ja näyttökoilla on helppoa.



Kuva 7 AVD-ympäristöön asennettuja virtuaalilaitteita

AVD toimii aivan samalla tavalla kuin oikea Android-laite, joten tällä tavalla testaaminen on todella hyvä tapa todeta sovelluksen toimivuus mahdollisimman monissa eri laitteissa.

## 4 OSTOSLISTASOVELLUKSEN KEHITTÄMINEN

### 4.1 Ostoslista sovellus

Ostoslista – Säästä rahaa ostoksilla on Android-käyttöjärjestelmälle kehitetty sovellus, jonka avulla käyttäjä voi säästää rahaa ruokaostoksia tehdessään. Käyttäjä pystyy syöttämään hintoja tuotteille eri kaupoissa, ja kun käyttäjä tekee ostoslistaa, sovellus näyttää kokonaishinnat eri kaupoissa.

#### 4.1.1 Suunnitteluperiaatteet

Sovelluksen suunnittelussa lähdettiin liikkeelle ajatuksella, että lopullisesta sovelluksesta tulee tarpeeksi helppokäyttöinen, selkeä ja varmatoiminen, jotta käyttäjät jatkavat sen käyttämistä. Sovellus tehtiin myös suomeksi ja englanniksi. Sovellus on automaattisesti suomenkielinen, jos käyttäjän laitteen kieleksi on valittuna Suomi.



Sovelluksen ulkonäkö ja toiminnallisuus suunniteltiin ja toteutettiin Androidin Material Design -suunnitteluperiaatteiden mukaisesti, jolloin värit ja eri osien koot ovat kaikki yhdenmukaisia. Sovellus suunniteltiin käyttäen hyväksi fragmentteja.



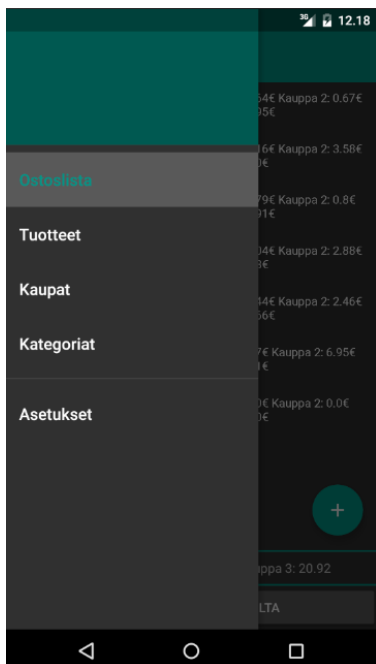
Kuva 8 Aktiviteetti ja fragmentti

Kuvassa 8 on esitetty punaisella aktiviteetin koko ja oranssilla fragmentin koko. Sovelluksessa käytettiin hyväksi fragmentteja, jotta laitteen tarvitsee luoda vain yksi aktiviteetti: käyttäjän siirtyessä ikkunasta toiseen, laitteen ei tarvitse tuhota ja luoda kokonaista aktiviteettia uudelleen. Samalla laitteen ei myöskään tarvitse luoda yläpalkkia uudestaan jokaisen ikkunan vaihdon yhteydessä

#### 4.1.2 Sovelluksen käyttö

Sovelluksen käyttöönotto on helppoa. Ensin lisätään kaupat, joiden hintatiedot halutaan. Sitten lisätään tuotteet. Kun tuote on lisätty tuotevalikkoon, niin sen voi jatkossa lisätä ostoslistalle.

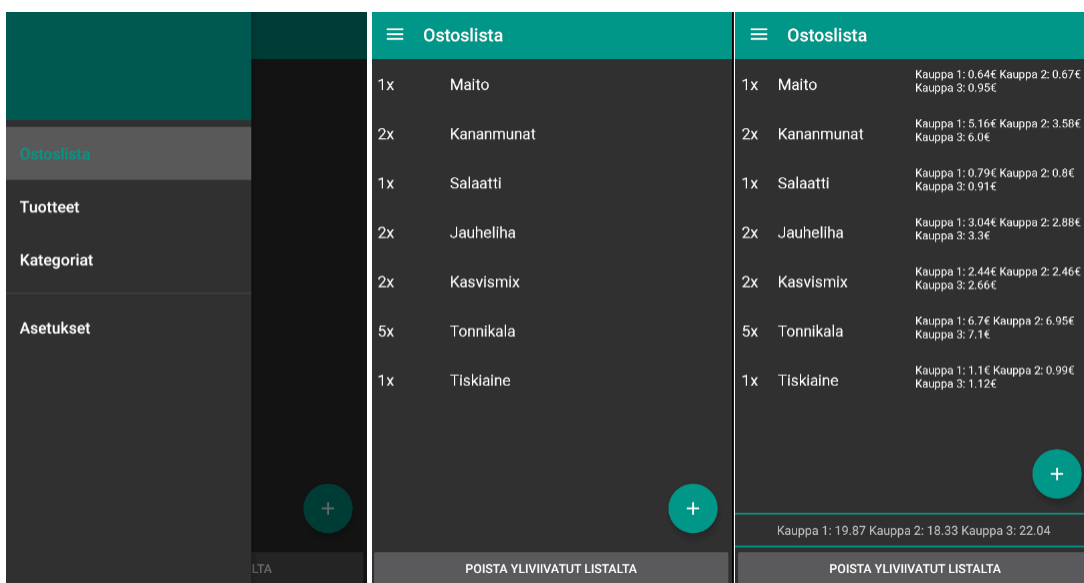
Sovelluksessa liikkuminen paikasta toiseen toteutettiin käyttäen hyväksi Androidin omaa `android.support.v4.widget.DrawerLayout`-luokaa, jolla luodaan sivulta avattava navigointipalkki. Käyttäjä voi avata navigointipalkin joko painamalla sovelluksen vasemmassa yläkulmassa olevaa kolmea viivaa, tai liikuttamalla sormen laitteen vasemmasta reunasta keskelle.



Kuva 9 Sovelluksen navigointipalkki

Navigointinäkymään tehtiin lisäksi toiminnallisuus, jolla käyttäjä näkee missä ikkunassa hän on kullakin hetkellä.

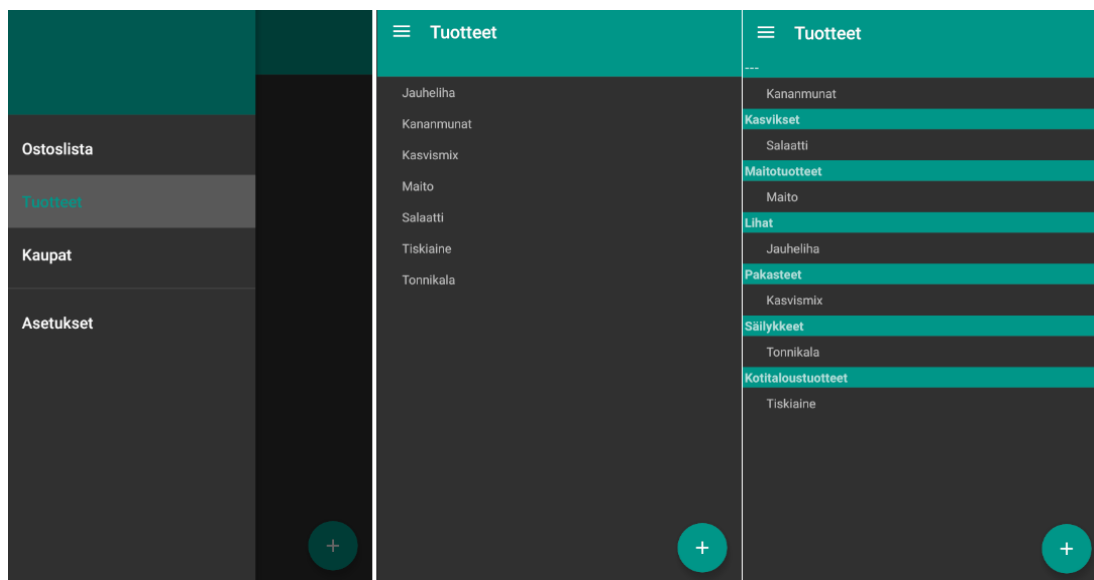
Sovelluksen ensimmäisellä käynnistyskerralla käyttäjälle näytetään asetuksetvälilehti, jossa käyttäjä voi valita haluaako hän käyttää tuotteiden hintoja ja haluaako hän eritellä tuotteet toisistaan eri kategorioilla.



Kuva 10 Hinnat asetuksen vaikutus

Kuvassa 10 nähdään ero, jos käyttäjä valitsee että ei halua käyttää hintoja. Navigointipalkista katoaa valinta ”Kaupat”, ja ostolistalla ei näy hintoja. Oikealla

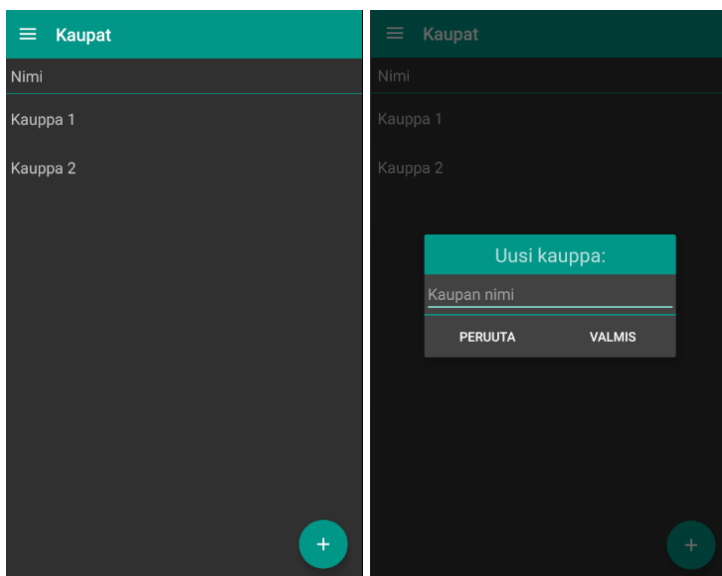
kuvassa 10 on ostoslista, jossa käyttäjällä on hinnat valittuna. Jos käyttäjä valitsee, että ei käytä hintoja, niin sovellus asettaa tietokantaan jokaiselle tuotteelle kaikissa kaupoissa hinnaksi 0. Näin sovellus toimii myös myöhemmin, vaikka aluksi ei hintoja olisi käytettykään.



Kuva 11 Kategoriat asetuksen vaikutus

Kuvassa 11 nähdään ero, jos käyttäjä valitsee että ei halua käyttää kategorioita. Navigointipalkista katoaa valinta ”Kategoriat”, ja tuotteet on järjestetty aakkosittain, ilman erottelevia kategorioita. Oikealla kuvassa 11 on tuotteet-näkymä, jossa käyttäjällä on kategoriat valittuna. Jos käyttäjä valitsee, että ei käytä kategorioita, niin sovellus sijoittaa kaikki uudet tuotteet kategoriaan yksi. Näin sovellus toimii myös myöhemmin, vaikka aluksi ei kategorioita olisi käytettykään.

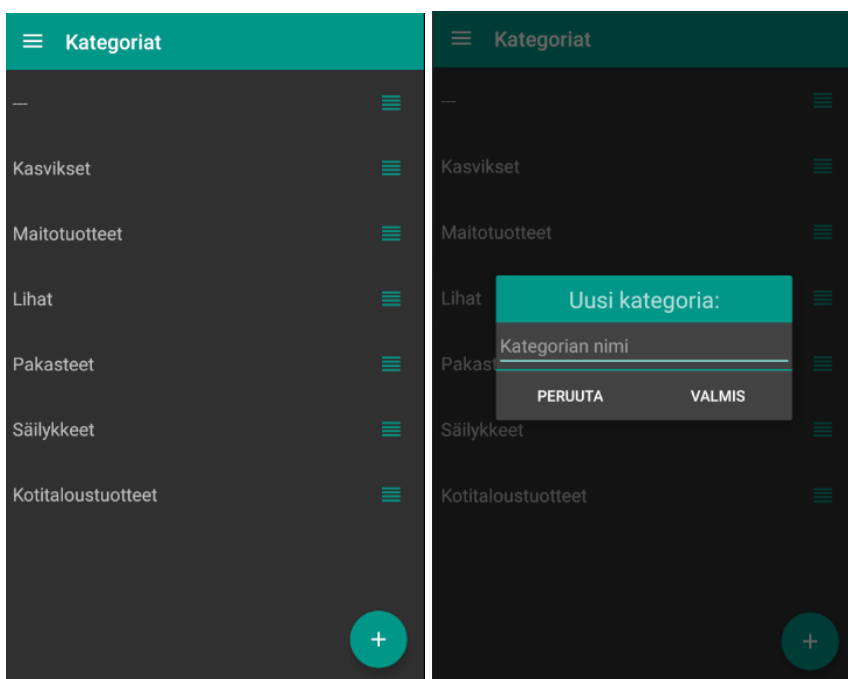
Sovelluksen käyttö aloitetaan siirtymällä kaupat-näkymään, jossa käyttäjä voi syöttää kauppojen nimiä niin monta kuin haluaa.



Kuva 12 Kaupat-näkymä

Kauppojen nimiä lisätään painamalla oikeassa alakulmassa olevaa + nappulaa. Tämän jälkeen aukeaa kuvassa 12 oikealla nähtävä ikkuna, johon käyttäjä syöttää kaupan nimen. Painamalla kuvassa 12 vasemmalla näkyvää kaupan riviä, käyttäjä voi poistaa kaupan tai nimetä kaupan uudelleen.

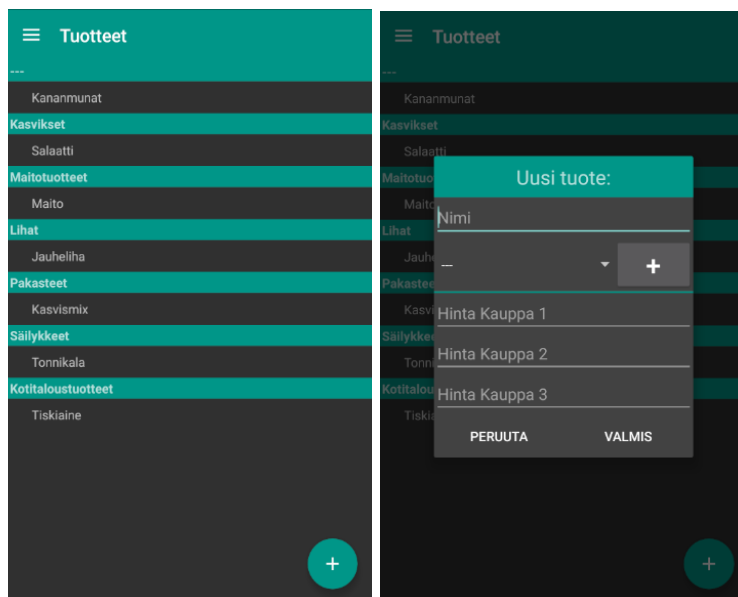
Kun halutut kaupat on lisätty, siirrytään Kategoriat-näkymään, johon käyttäjä voi syöttää haluamiaan kategorioita. Näitä ovat esimerkiksi maitotuotteet, säilykkeet tai muu nimettävä kategoria, jolla käyttäjä haluaa erotella tuotteet toisistaan.



Kuva 13 Kategoriat-näkymä

Kategorioita lisätään painamalla oikeassa alakulmassa olevaa + nappulaa. Tämän jälkeen aukeaa kuvassa 13 oikealla nähtävä ikkuna, johon käyttäjä syöttää kategorian nimen. Painamalla kuvassa 13 vasemmalla näkyvää kategoriariviä, käyttäjä voi poistaa kategorian tai nimetä sen uudelleen. Käyttäjällä voi myös vaihtaa kategorioiden paikkoja keskenään kategoriarivillä näkyvästä kolmesta viivasta. Tämä tapahtuu laittamalla sormi kategorian kohdalle ja vetämällä se halutulle paikalle. Sovelluksessa Ostoslistanäkymässä oleva ostoslista järjestetään Kategoriat-näkymällä olevan järjestyksen mukaisesti.

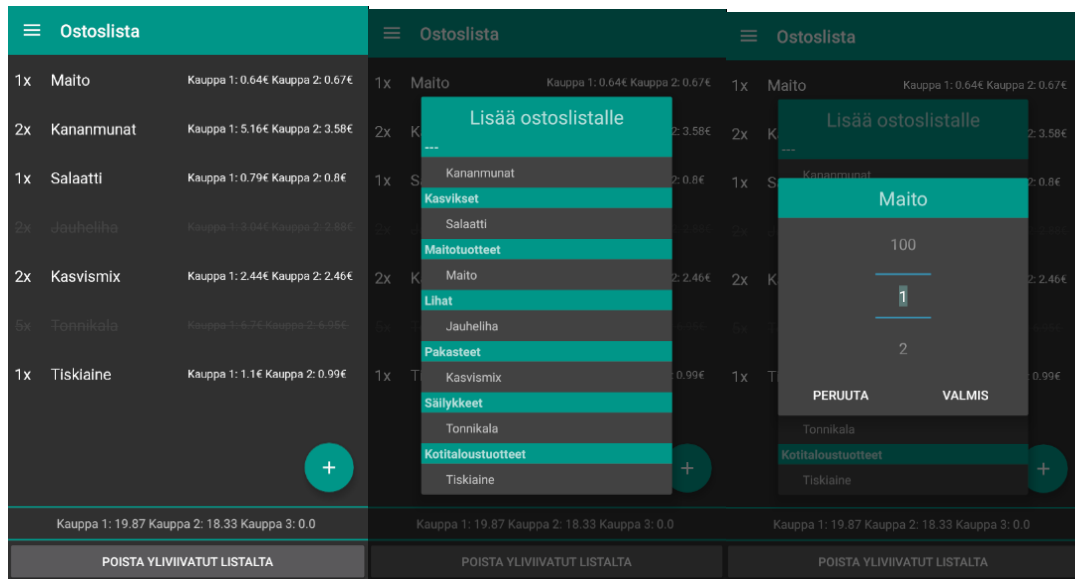
Kun halutut kategoriat on syötetty, käyttäjä siirtyy Tuotteet-näkymään, johon käyttäjä luo tuotteet ja syöttää niille hinnat.



Kuva 14 Tuotteet-näkymä

Tuotteita lisätään painamalla oikeassa alakulmassa olevaa + nappulaa. Tämän jälkeen aukeaa kuvassa 14 oikealla nähtävä ikkuna, johon käyttäjä syöttää tuotteen nimen. Samalla käyttäjä valitsee vetovalikosta tuotteen kategorian ja syöttää hinnat kuvassa näkyviin kirjoituskenttiin. Painamalla kuvassa 14 vasemmalla näkyvää tuoteriviä käyttäjä voi poistaa tuotteen, nimetä sen uudelleen, muokata hintoja ja vaihtaa tuotteen kategoriaan.

Näiden asetusten jälkeen sovellus on valmis käytettäväksi. Jatkossa Ostoslistanäkymästä lisätään tuotteet ostoslistalle.



Kuva 15 Ostoslista-näkymä

Ostoslistalle lisätään tuotteita painamalla oikeassa alakulmassa olevaa + nappulaa. Tämän jälkeen aukeaa kuvassa 15 keskellä nähtävä ikkuna, josta käyttäjä valitsee listalle haluttavan tuotteen. Tämä tapahtuu painamalla halutun tuotteen riviä. Rivin painalluksen jälkeen aukeaa kuvassa 15 oikealla nähtävä ikkuna, josta käyttäjä valitsee montako kappaletta kyseistä tuotetta lisätään ostoslistalle.

Ostoslistalta voi yliviivata tuotteita painamalla Ostoslista-näkymästä tuotteen riviä. Painallus vaihtaa rivin tekstin tumman harmaaksi ja yliviivaa tuotteen. Painamalla kuvassa 15 vasemmalla alhaalla näkyvästä nappulasta sovellus poistaa ostoslistalta kaikki ylivedetyt tuotteet.

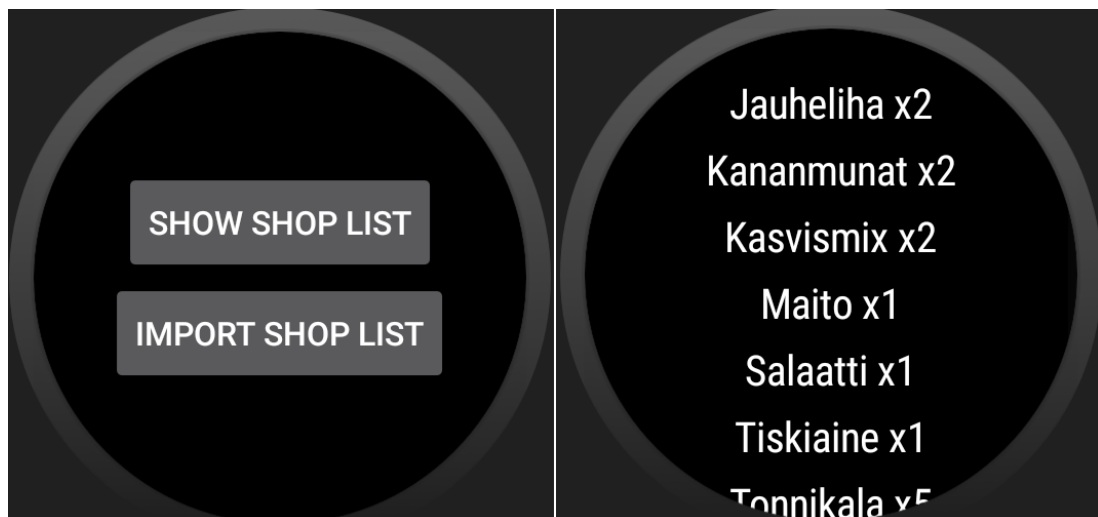
## 4.2 Älykello-sovellus

Älykello-sovellukseen oli tarkoitus tehdä kosketustoiminnallisuus, jolloin käyttäjä voisi merkitä tuotteen ostetuksi suoraan kellosta. Sovellusta tehdessä tuli vastaan ongelma: jos haluaa tehdä älykelloon listan, josta voisi merkitä tuotteen ostetuksi, niin pitää käyttää Androidin Wearable UI-kirjastosta löytyvää WearableListView-luokkaa. Siinä listan koko on valmiiksi määritetty, joten listalla näkyisi kellossa vain kolme tuotetta kerrallaan.



Kuva 16 Android Wear WearableListView (Creating Lists, 2016)

Tuon valmiiksi määritellyn listan koon takia päädyttiin käyttämään toista ratkaisua, jossa ei käytetä kosketusta. Käyttäjä näkee tämän ansiosta listalla tuotteita enemmän kuin edellisessä toteutustavassa. Samalla päädyttiin tekemään älykello-sovellukseen toiminto, jolla käyttäjä voi siirtää ostoslistan kellon muistiin. Kellon täytyy olla yhteydessä käyttäjän Android-laitteeseen vain silloin, kun ostoslista siirretään. Eli ostoslistan siirtämisen jälkeen käyttäjä näkee ostoslistan kellostansa, vaikka se ei olisi yhteydessä puhelimeen.



Kuva 17 Älykello-sovelluksen käyttöliittymä

Kun älykello-sovelluksen käynnistää, aukeaa kuvassa 17 vasemmalla näkyvä käyttöliittymä, josta käyttäjä voi ”Show Shop List”-painikkeella siirtää ostoslistan

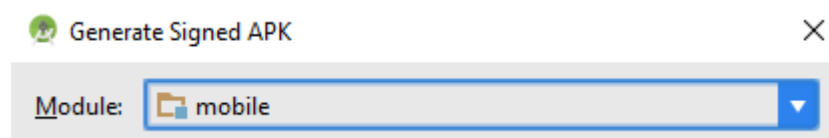
yhdistetystä Android-laitteesta. Siirron jälkeen sovellus siirtyy automaattisesti oikean puoleiseen käyttöliittymään.

”Show Shop List”-painikkeella käyttäjä pääsee oikeanpuoleiseen käyttöliittymään, vaikka älykello ei olisi yhdistettynä puhelimeen. Jos ostoslista on tyhjä, sovellus ilmoittaa oikean puoleisessa käyttöliittymässä tekstillä ”Shoplist is empty”. Ostoslistakäyttöliittymä on toteutettu ScrollView-luokalla: listalla pystyy siirtymään kosketuksen avulla, joten listalla voi olla vaikka 100 tuotetta.

## 5 SOVELLUKSEN JULKAISU

Kehittäjän pitää rekisteröityä viralliseksi sovelluskehittäjäksi Google Play Developer Consolessa osoitteessa <https://play.google.com/apps/publish/signup/>, jotta hän voi julkaista sovelluksia Goole Play Kaupassa. Tällä hetkellä kehittäjäksi rekisteröityminen maksaa 25 USD, mikä on elinikäinen kertamaksu.

Ensimmäinen vaihe sovelluksen julkaisuun on luoda Android Studiossa sovelluksesta APK-tiedosto. APK-tiedoston versionumerointi määritetään build.gradle-tiedostoon riveille versionCode ja versionName. Muuttuja versionName voi olla mitä tahansa, mutta versionCode pitää olla joka versiolle yhden numeron suurempi: esimerkiksi ensimmäinen versio on 1, toinen on 2, ja niin edelleen.

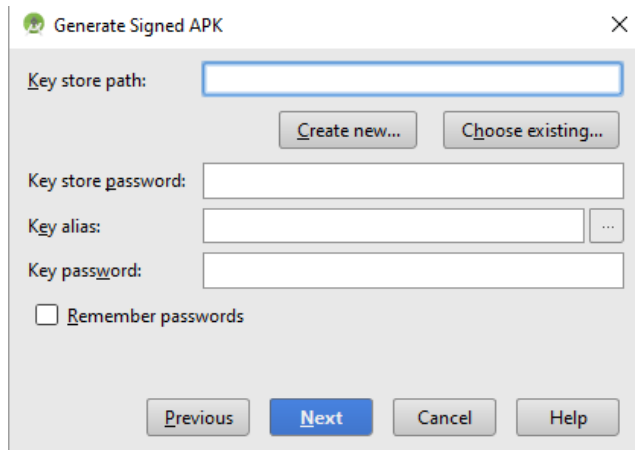


Kuva 18 APK-tiedoston luonti

Kun tämän sovelluksen tapauksessa tehtiin tuki Android-älykellolle, oli mahdollista luoda erikseen Android-laitteen ja älykellon APK-tiedostot. Tämä on tarkoitettu kuitenkin vain sovelluksen testaamista varten. Mobile-paketti sisältää molemmat APK-tiedostot.



APK-tiedostoa luodessa pitää myös luoda salausavain APK-tiedostoille. Vaikka joku saisikin kehittäjän sähköpostitunnuksen ja sen salasanan, niin hän ei voi tehdä julkaistuille sovelluksille mitään haittaa.



Kuva 19 APK-tiedoston salausavaimen luominen

Kun APK-tiedostot on luotu, siirrytään Google Play Developer Consoleen ja painetaan luo uusi sovellus-nappulaa. Ensimmäinen vaihe on antaa sovellukselle nimi ja valita sovelluksen alkuperäinen kieli. Tämän jälkeen kirjoitetaan sovelluksen tiedot, jotka näkyvät käyttäjille Google Play kaupassa. Tarvittavat tiedot, jotka pitää täyttää ovat: sovelluksen nimi, lyhyt kuvaus, täysi kuvaus, 512x512 pixeliä kooltaan oleva sovelluksen kuvake, 1024 x 500 pixeliä kooltaan oleva esittelykuva, vähintään yksi näyttökuva sovelluksesta, sovelluksen tyyppi, sovelluksen kategoria, sovelluksen sopivuus ikäluokille, kehittäjän yhteystiedot, sovelluksen hinta ja maat jossa sovellus on saatavilla. Kun kaikki nämä tiedot on täytetty, voidaan sovellus lähettää käsiteltäväksi ja käsittelyn jälkeen sovellus julkaistaan Google Play Kaupassa.

Sovellus julkaistiin versionumerolla 0.1 15.5.2016 aluksi vain Suomessa. Uusien sovellusten löytäminen Google Play Kaupasta on hankalaa, sillä sovelluksia on paljon. Lisäksi Googlen hakualgoritmit suosivat suosituimpia sovelluksia. Tämän sovelluksen löytää helpoiten syöttämällä hakukenttään: “pub:Nuge”, joka hakee sovelluksen kehittäjänimellä.

## 6 LOPPUTULOS

Tavoitteena oli luoda sovellus, jolla käyttäjä voi hinnoitella tuotteita eri kaupoissa saadakseen suosituksen edullisimmasta kaupasta ostoslistaa tehdessä. Ostoslista tulee olla luettavissa myös Android Wear -älykellosta. Tässä onnistuttiin käytännön testien perusteella kohtuullisen hyvin. Sovellus ottaa eri kauppojen tuotteiden hintoja vastaan ja suosittelee ostolistan perusteella kauppa. Testien mukaan suositukset lasketaan oikein, ja myös älykello-sovellus toimii moitteitta.

Tämän hetkisessä sovelluksessa haittana on tarvittavien tietojen syöttämisen työmäärä. Jotta sovelluksesta on säästämismielessä hyötyä, joutuu kaikki tuotteet ja kaikki hinnat syöttämään käsin. Hintojen oikeellisuudestaan ei välttämättä ole tietoa. Sovellus tosin on vasta kehityksensä alussa, ja tähän pyritään etsimään ratkaisuja.

Mahdollisia muutoksia sovellukseen on jo suunnitteilla paljon. Suurimpana tavoitteena on saada eri kaupoilta hintatiedot suoraan kauppojen palvelimilta, mutta tämä on kaukana tulevaisuudessa. Toisena suurena ideana on luoda sovellukselle palvelinyhteys, jotta kaikki tuotetiedot ja ostoslistat olisivat tallessa verkossa, ja näitä voisi jakaa haluamilleen henkilöille.

Tämänhetkinen sovellus kuitenkin toimii hyvin pelkkänä ostoslistana. Lisäksi se laskee ostosten hintatiedot, jos käyttäjä on syöttänyt tarvittavat tiedot.

## LÄHTEET

Jakosuo, K. 2015. Kuluttajan kauppakassi 2015. Kuluttajaliitto 2.7.2015. Viitattu 5.5.2016. <http://www.kuluttajaliitto.fi>

Rouse, M 2008. Definition Android OS. Viitattu 16.5.2016. <http://searchenterprise.linux.techtarget.com>

Hildenbrand, J. 2015. What is Android?. Viitattu 16.5.2016. <http://www.androidcentral.com>

Amadeo, R. 2014. The history of Android. Viitattu 16.5.2016. <http://arstechnica.com>

Brachmann, S. 2014. A Brief History of Google's Android Operating System. IPWatchdog 26.11.2014. Viitattu 9.5.2016. <http://www.ipwatchdog.com>

The Android Story. 2016. Viitattu 11.5.2016. <http://www.android.com>

Application Fundamentals. 2016. Viitattu 16.5.2016. <https://developer.android.com>

Activities. 2016. Viitattu 16.5.2016. <http://developer.android.com>

Services. 2016. Viitattu 16.5.2016. <http://developer.android.com>

Content Providers. 2016. Viitattu 16.5.2016. <http://developer.android.com>

Fragments, 2016. Viitattu 16.5.2016. <http://developer.android.com>

Sims, G. 2016. I want to develop Android Apps – What languages should I learn? Viitattu 17.5.2016. <http://www.androidauthority.com>

Saving Data in SQL Databases. 2016. Viitattu 17.5.2016. <http://developer.android.com>

Meet Android Studio. 2016. Viitattu 17.5.2016. <http://developer.android.com>

Material Design for Android. 2016. Viitattu 17.5.2016. <http://developer.android.com>

Platform Versions. 2016. Viitattu 11.5.2016. <http://developer.android.com>

Creating Lists. 2016. Viitattu 15.5.2016. <http://developer.android.com>