



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

TILAUSTENHALLINTA

Android-sovellus

TEKIJÄ: Mika Ruotsalainen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Mika Ruotsalainen	
Työn nimi Tilaustenhallinta Android-sovellus	
Päiväys 30.4.2017	Sivumäärä/Liitteet 22 / 0
Ohjaaja(t) lehtori Jussi Koistinen, lehtori Sami Lahti	
Toimeksiantaja / Yhteistyökumppani(t) Agamania Oy	
Tiivistelmä <p>Opinnäytetyön aiheena oli valmistaa tilaustenhallintasovellus Android-tabletille. Työn toimeksiantaja oli eläinkauppa Agamania Oy.</p> <p>Sovellukseen oli tarkoitus kirjata asiakkaiden tekemät tuotetilaukset. Tavoitteena oli, että sovellukseen kirjataan tuotetilaukset, niin että tilauksia voidaan joko selata kaikki kerralla tai etsiä asiakkaan nimen avulla. Tilaukset lajiteltiin eläinryhmittäin, jotta niiden selaaminen ja etsiminen olisi helpompaa. Sovelluksella voi myös muokata tai poistaa tilauksia. Sovelluksen tarkoituksena oli korvata paperinen tilaustenhallintakäytäntö.</p> <p>Sovellus valmistettiin Android Studio kehitysohjelmalla. Kehityksen aikana sovellusta testattiin pääsääntöisesti vain fyysisillä laitteilla. Valmis tuote toimitettiin asiakkaalle mutta sitä ei tulla julkaisemaan missään sovelluskaupassa, sillä kyseessä oli vain yhdelle myymälälle erikseen suunniteltu Android-sovellus.</p> <p>Työn tuloksena kehitettiin tilaajan vaatimusten mukainen sovellus. Tilaaja oli tyytyväinen työn lopputulokseen.</p>	
Avainsanat Tilaustenhallintasovellus, Android, Java, XML	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Mika Ruotsalainen			
Title of Thesis Order Handling Android Application			
Date	30 April 2017	Pages/Appendices	22 / 0
Supervisor(s) Mr Jussi Koistinen, Lecturer, Mr Sami Lahti, Lecturer			
Client Organisation / Partners Agamania Oy			
<p>Abstract</p> <p>The purpose of this thesis was to create an Android application for order handling. The assignment came from a petshop Agamania Oy.</p> <p>The purpose of the application is to store customer orders. It had to be able to add, browse, search, edit and delete orders. Orders are categorized by different groups of animals and can be searched by the group or customer name. The application meant to replace the old paper version of order handling.</p> <p>The application was developed with the Android Studio development environment and tested with Android devices during development. The application will not be released in any application distribution platform because it was a custom built for Agamania Oy.</p> <p>The result of this thesis was the application that filled the customer's requirements. The customer was pleased with the result.</p>			
Keywords Order handling, Android, Java, XML			

ESIPUHE

Haluan kiittää Agamania Oy:tä, joka tarjosi mahdollisuuden mielenkiintoiseen opinnäytetyöhön. Kiitokset myös Savonia-ammattikorkeakoulun lehtori Jussi Koistiselle todella kärsivällisestä työn ohjauksesta.

Kuopiossa 30.04.2017

Mika Ruotsalainen

SISÄLTÖ

1	JOHDANTO	6
2	TYÖKALUT JA LAITTEET	7
2.1	Android Studio	7
2.2	Adobe Photoshop	7
2.3	Fyysiset laitteet.....	7
3	ANDROID-KÄYTTÖJÄRJESTELMÄ.....	8
3.1	Android	8
3.2	Android API	8
3.3	Aktiviteetti.....	8
3.4	Java.....	8
3.5	XML.....	9
4	SUUNNITTELU.....	10
4.1	Toimintovaatimukset	10
4.2	Ulkoasu ja käytettävyys	11
5	TOTEUTUS.....	12
5.1	Rakenne.....	12
5.2	Tilausten selaus ja poisto.....	13
5.3	Lisäys ja muokkaus	14
5.4	Varmuuskopio.....	15
5.5	Testaus.....	16
6	JATKOKEHITYS	17
6.1	Tietokanta.....	17
6.2	Asiakasrekisteri	18
6.3	Dynaaminen käyttöliittymä.....	18
6.4	Käyttäjät	18
6.5	Web-pohjainen sovellus.....	19
6.6	Rajapinta.....	20
7	YHTEENVETO	21
	LÄHTEET JA VIITTAUKSET	22

1 JOHDANTO

Työn toimeksiantaja on eläinkauppa Agamania Oy. Projektin aiheena on luoda Android-tabletilla käytettävä tilaustenhallintasovellus. Ohjelmaan kirjataan asiakkaiden tekemät tilaukset ja sillä pystytään selaamaan kaikkia tilauksia sekä etsimään tiettyä tilausta joko asiakkaan nimen tai tuoteryhmän perusteella. Tilaukset on lajiteltu eläinryhmittäin, jotta tilausten selaaminen laitteella olisi helpompaa. Ohjelmasta päädyttiin tekemään mobiiliversio, koska myymälän tiskille ei mahdu toista työasemaa ja kassatoiminnassa olevaa konetta ei voi varata muuhun käyttöön.

Sovelluksella halutaan korvata vanha paperinen tilaustenhallintamenetelmä. Sovelluksen avulla tilausten seuraaminen ja hallinta on helpompaa sekä nopeampaa. Sovellusta ei tulla julkaisemaan missään sovelluskaupassa, sillä kyseessä on vain Agamania Oy:n tarpeisiin suunniteltu sovellus.

2 TYÖKALUT JA LAITTEET

2.1 Android Studio

Työ tehtiin Android Studio editorilla. Android Studio on Googlen tekemä virallinen kehitysympäristö Android laitteille. Ensimmäinen Android Studio julkaistiin toukokuussa 2013 ja se korvasi aiemman Eclipse kehitysympäristön. (Android, ei pvm.)

2.2 Adobe Photoshop

Sovelluksen nappeihin liitetyt kuvakkeet muokattiin Adobe Photosopilla. Adobe Photosohop on Adobe Systemsin kehittämä kuvankäsittelyohjelma, joka on saavuttanut markkinajohtajuuden kaupallisessa digitaalisten kuvien muokkauksessa. (Wikipedia, ei pvm.)

2.3 Fyysiset laitteet

Sovelluskehityksessä käytettiin vain fyysisiä Android-laitteita, sillä oikeilla laitteilla näkee suoraan mitä sovellus tulee todellisuudessa näyttämään. Vaikka sovellus ja sen käyttöliittymä suunniteltiin vain tablet-laitteille, testattiin sen soveltuvuutta myös puhelimella mahdollista jatkokehitystä varten. Projektin testikäytössä olleissa laitteissa sekä yrityksen tabletissa, johon sovellus asennettiin, oli käytössä Androidin versio 4.4, API 19.

Testauskäytössä olleet laitteet olivat

- Denver 10.1" Tablet
- Samsung Galaxy S4 Mini.

3 ANDROID-KÄYTTÖJÄRJESTELMÄ

3.1 Android

Googlen omistama Android on tällä hetkellä maailman suosituin mobiililaitteiden käyttöjärjestelmä. Androidin ensimmäinen kaupallinen versio julkaistiin vuonna 2008. Android-niminen yhtiö kehitti Android-käyttöjärjestelmän vuonna 2003. Google osti Android-yhtiön itselleen vuonna 2005. (Mobiiliblogi, 2013-20-07.)

Nykyisin Androidin kehittämisestä vastaa Open Handset Alliance. Androidiin tarkoitettua koodia kirjoitetaan Java- ja xml-kielillä ja se käyttää Googlen kehittämiä Java-kirjastoja. (Wikipedia, ei pvm.)

3.2 Android API

API (Application Programming Interface) eli ohjelmointirajapinta tarkoittaa, että kuinka hyvin sovellus voi kommunikoida laitteen sisäisten toimintojen ja toiminnallisuuden kanssa. Kun API-versio kasvaa, saavat versiot lisää ominaisuuksia ja toimintoja, tosin joitakin vanhentuneita toimintoja saatetaan uusista versiosta myös jättää pois.

Sovelluksen API versiota valittaessa tulisi ottaa huomioon ainakin kaksi asiaa. Nykyinen versioiden jakauma eli kuinka moni laite pystyy sovellusta käyttämään. Jos sovellus on valmistettu versiolle 19, versiot 18 ja sitä vanhemmat eivät pysty käyttämään sovellusta. Tätä sovellusta tehtäessä 88 % markkinoilla olevista Android-laitteista tukee API versiota 19. (Android, 2017-04-03.)

Valittaessa pienempi API-versio, useampi laite pystyy käyttämään sovellusta, mutta silloin sovelluksessa on vähemmän ominaisuuksia käytettävissä ja ominaisuudet, joita haluttaisiin käyttää, ovat vaikeammin tehtävissä kuin suuremmilla versioilla.

3.3 Aktiviteetti

Aktiviteetit (engl. Activity) ovat yksittäisiä näyttöjä tai ikkunoita, jotka näkyvät yksi kerrallaan. Lähes kaikki näytöt ovat interaktiivisia, ja niihin rakennetaan sovelluksen käyttöliittymä. Sovellus voi sisältää useita näyttöjä, joita voi avata ja sulkea. Näyttöjä voi jättää taustalle auki ja niihin voi myöhemmin palata. Niiden välillä voidaan myös lähettää tietoa.

3.4 Java

Java on Sun Microsystemsin kehittämä laaja teknologiaperhe ja ohjelmistoalusta, johon kuuluu muun muassa laitteistoriippumaton oliopohjainen ohjelmointikieli sekä ajoaikainen ympäristö virtuaalikoneineen ja luokkakirjastoineen. Java-ohjelmointikieli kehitettiin 1990-luvun alussa. JDK 1.0 (Java Development kit) ilmestyi syksyllä 1995 ja se soveltui www-sivuille luotavien sovellusten tekemiseen. Nykyään Javan kehittämisestä vastaa Oracle ja Java on käytössä kaikkialla, matkapuhelimista tietokoneisiin. (Wikipedia, ei pvm)

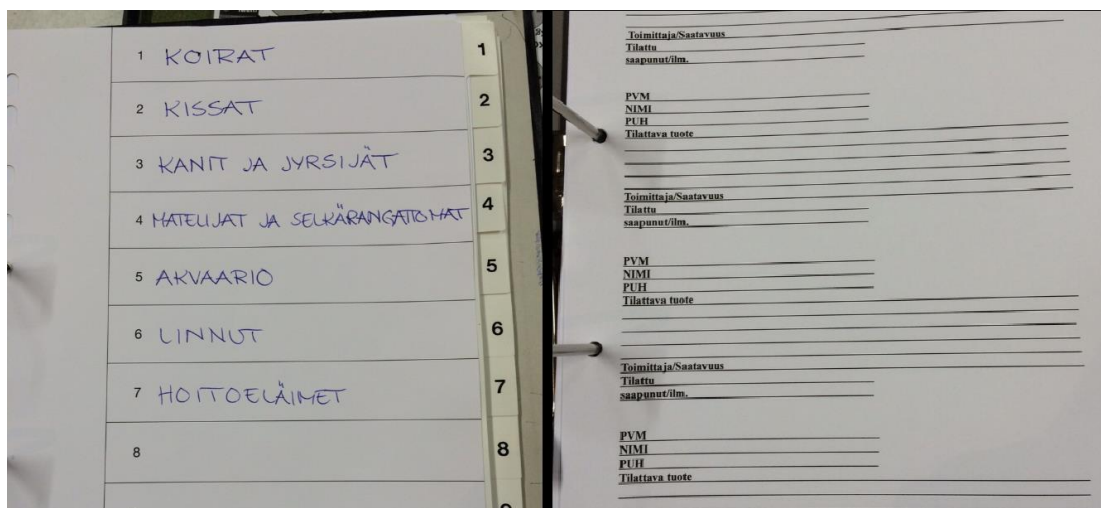
3.5 XML

XML (Extensible Markup Language) on yksinkertainen ja joustava merkintäkieli. XML suunniteltiin säilyttämään ja siirtämään dataa. Se suunniteltiin myös olemaan sekä ihmiselle ja koneelle luettava muodossa. (W3C, 2016-10-11)

4 SUUNNITTELU

Agamania Oy on vuonna 2012 perustettu Faunatar-ketjun franchising-yritys. Yritys on erikoistunut lemmikkieläinten sekä niiden hoitoon liittyvien tarvikkeiden ja ruokien myyntiin. Yritys tarvitsi yksinkertaista sovellusta, jolla voisi helposti lisätä, seurata, etsiä ja muokata asiakkaiden tekemiä tilauksia.

Sovelluksen tarkoituksena oli helpottaa tilausten hallintaa. Kuvassa 1 nähdään kuinka tilaukset aikaisemmin hoidettiin kirjoittamalla ne käsin paperille ja arkistoimalla kansioon. Tilaukset oli lajiteltu kansioon tilattavan tuotteen kategorian perusteella ja järjestelty tilauspäivämäärän mukaan, uusin ensimmäisenä. Vanhojen tilauksien selaaminen kansioista oli hankalaa, jos ei ollut tiedossa tuotteen tilauspäivämäärää.



KUVA 1. Vanha tilaustenhallintamenetelmä

4.1 Toimintovaatimukset

Ohjelmalla täytyi pystyä selaamaan avoimia tilauksia, lisäämään uusia sekä hallinnoimaan vanhoja. Sovelluksessa ei tarvinnut olla käyttäjätunnuksia, sisäänkirjautumisikkunaa tai mitään erilaisia oikeuksia käyttäjille. Yrityksessä kaikki työntekijät hoitavat asiakkaiden tilauksia, joten tilauksien muokkaus tai poistorajoituksille ei ollut tarvetta.

Ohjelman toiminnallisuudet ovat:

- tilauksien selaus ja haku
- tilauksen lisäys
- tilauksien muokkaus
- tilauksien päättäminen
- tilauksien palauttaminen takaisin avoimeksi
- tilauksien poisto
- tietojen varmuuskopiointi muistikortille.

Tilauksien täytyi olla järjestettynä lisäyispäivän mukaan, uusin ensimmäisenä, aivan kuten vanhassa järjestelmässä oli käytetty. Tilauksia piti pystyä selaamaan joko valitun kategorian mukaan, etsimään asiakkaan nimellä tai selaamaan kaikkia avoimia tilauksia kerrallaan. Listalla täytyi myös näkyä suoraan kaikki tilauksen tiedot. Koska ohjelman käyttöliittymä oli tehty vain tabletille, täytyi kaikista avoimista tilauksista saada lista erilliseen tiedostoon, jonka voi helposti siirtää tietokoneelle.

4.2 Ulkoasu ja käytettävyys

Ulkoasun tuli olla mahdollisimman yksinkertainen ja selkeä sekä värimaailman vaalea ja neutraali. Näytöllä tuli olla luettelo kaikista avoimista tilauksista. Sovelluksen pääikkunaa voidaan käyttää vain poikittaissuunnassa, jotta tilauslistalle jäisi riittävän leveä alue. Näin tilauksien lukeminen on helpompaa ja tilauksesta mahtuu enemmän tietoa yhdelle riville. Koska sovellus tilattiin tablet-laitteille, ei ulkoasun ja käytettävyyden suunnittelussa juurikaan panostettu käytettävyyteen puhelimesta, kuten näytön skaalautumiseen.

5 TOTEUTUS

Työ aloitettiin käyttöliittymän luomisella. Sovelluksessa jokainen toiminto sijaitsee omalla näytöllään ja sovelluksen kaikkien näyttöjen ulkoasu muokattiin heti alussa lähes lopulliseen muotoonsa. Tämä lähestymistapa valittiin koska työn tilaaja oli yritys, jonka toimipiste sijaitsee toisella paikkakunnalla. Yrityksen edustajan kanssa oli sovittu pidettäväksi laatupalaveri, jossa käytiin vielä toiminnot tarkemmin läpi. Palaverissa pystyi näyttämään asiakkaalle fyysiseltä laitteelta, miltä lopullinen ohjelma tulee näyttämään ja asiakas pääsi itse testaamaan sovelluksen käytettävyyttä. Kun asiakas oli hyväksynyt käyttöliittymän, aloitettiin toimintojen tekeminen. Tässä vaiheessa ei tarvinnut olla juurikaan asiakkaaseen yhteydessä, koska suunnitteluvaiheessa asiat oli selvitetty huolellisesti ja ulkoasu oli asiakkaan hyväksymä.

5.1 Rakenne

Android laitteille tyypillisesti jokainen toiminto sijaitsee omalla aktiviteetillaan. Suunnitteluvaiheessa päätettiin, että sovellusta voi käyttää tabletilla vain poikittaissuunnassa.

Sovelluksen toimintaa varten luotiin seuraavat ikkunat:

- sovelluksen latausikkuna
- tilausten selaus
- tilauksien lisäys ja muokkaus
- varmuuskopiointi.

Android-laitteissa sovelluksen ja kaikkien näyttöjen asetukset määritellään AndroidManifest.xml-tiedostossa (Kuva 2).

AndroidManifest-tiedostossa määritellään mm.

- sovelluksen nimi
- sovelluksen paketti
- sovelluksen kuvake
- sovelluksen vaatimat oikeudet
- näyttöjen orientaatiot
- aloitusikkuna.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.agamania.asiakashallinta" >
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/app_logo"
        android:label="@string/Asiakastilaukset"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/Asiakastilaukset"
            android:screenOrientation="landscape" >
        </activity>
        <activity
            android:name=".Settings"
            android:label="@string/Asetukset"
            android:screenOrientation="landscape" >
        </activity>
        <activity
            android:name=".Form"
            android:label="@string/Lomake" >
        </activity>
        <activity
            android:name=".LoadingScreen"
            android:label="@string/Asiakastilaukset"
            android:theme="@style/Theme.AppCompat.NoActionBar"
            android:screenOrientation="landscape" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

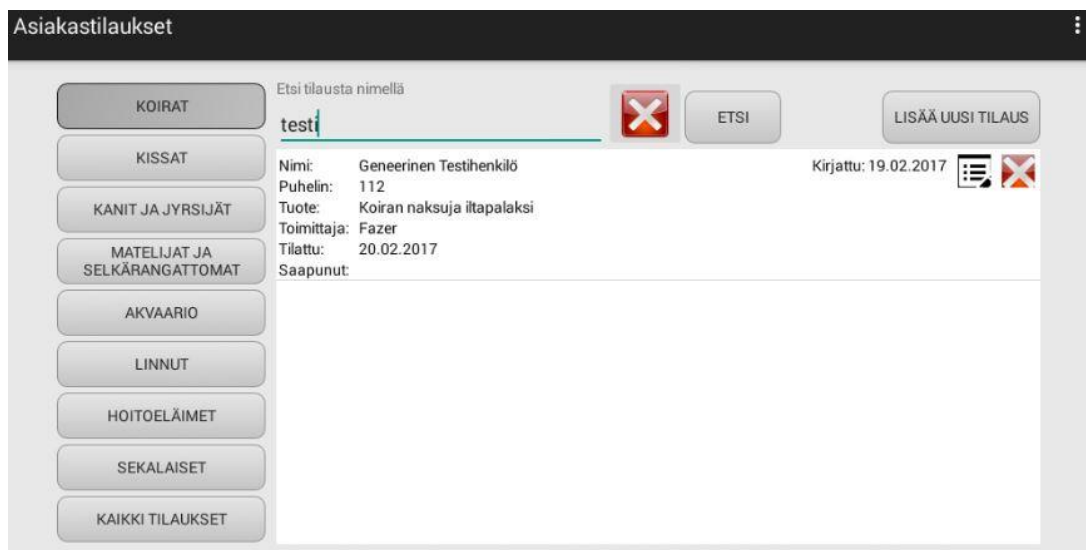
```

KUVA 2. AndroidManifest.xml tiedosto

Toimintojen saavuttamista vanhemmilla API-versioilla ei otettu huomioon ja sovelluksen pienimmäksi versioksi määriteltiin 19. Näin toimittiin koska asiakkaan senhetkinen tabletti käytti kyseistä versiota ja sovellusta ei suunniteltu julkaistavaksi yleiseen käyttöön. Laitetta vaihdettaessa uudempaan, myös API-versiot päivittyvät, joten vanhojen laitteiden tukeminen oli tässä vaiheessa turhaa. Asiasta kuitenkin keskusteltiin ensin tilaajan kanssa, voidaanko vanhempien versioiden tukeminen jättää pois.

5.2 Tilausten selaus ja poisto

Avointen ja päättyneiden tilausten selaus toteutettiin samalla näytöllä ja molemmissa on samat haku- ja ryhmittelytoiminnot (Kuva 3). Oletuksena näytöllä näkyy kaikki avoimet tilaukset. Päättyneitä tilauksia selattaessa taustan väri vaihdetaan punaiseksi, jotta vältetään sekaannus avointen ja päättyneiden tilausten välillä.



KUVA 3. Sovelluksen pääikkuna

Koska listalla täytyi näkyä kaikki tiedot asiakkaiden tilauksista, ei listakomponentin vakioulokoasu enää riittänyt, vaan sille jouduttiin luomaan oma lista-adapterinsa. Lista-adapterin avulla luodaan yksittäiselle riville oma ulkoasunsa, jotta saadaan mahtumaan kaikki tarvittava tieto yhdelle riville. Rivin ulkoasua varten tehdään oma XML-tiedosto, jota muotoillaan samalla tavalla kuin tavallisen ikkunan XML-tyylitiedostoa.

Tilauksien muokkausta ja poistoa varten muokattiin Adobe Photoshopilla pienet ikonit rivin oikeaan ylänurkkaan. Symbolista käy hyvin selväksi painikkeiden toiminta ja ne vievät vähemmän tilaa kuin teksti painikkeen sisällä. Avoimia tilauksia selattaessa muokkauspainike avaa ikkunan, jossa voi muokata ja lisätä tilauksen tietoja. Avoimissa tilauksissa poistopainike ei suoraan poista tilausta, vaan laittaa tilauksen tilan päättyneeksi ja poistaa sen avoimien tilausten listalta. Vasta päättyneeksi merkityn tilauksen poisto poistaa tilauksen kokonaan tietokannasta. Päättyneitä tilauksia selattaessa muokkausnapin toiminta muutetaan palauttamaan tilaus takaisin aktiiviseksi. Päättyneitä tilauksia ei voi muokata.

Tilausten selaaminen kategorioiden perusteella toteutettiin laittamalla jokaiselle kategorialle erillinen painike, jonka avulla sovellus hakee vain kyseiseen ryhmään luokitellut tilaukset. Tilausta nimen avulla haettaessa, hakukenttään täytyy kirjoittaa vain osa etu- tai sukunimestä, ja sovellus näyttää listalla kaikki osumat. Jos hakutuloksia löytyy paljon, voi hakutuloksiakin selata kategorioiden perusteella kuvan 3 mukaisesti.

5.3 Lisäys ja muokkaus

Tilausten lisäystä ja muokkausta käytetään samalla lomakkeella. Kuvassa 4 näkyvää lomakeikkunaa tehtäessä huomattiin, että poikittain oleva käyttöliittymä ei ollut käytännöllinen tilausta syötettäessä. Ikkunaan tehtiin mahdollisuus käyttää sovellusta myös pystysuunnassa, näin virtuaalinäppäimistö ei peittänyt liikaa lomaketta. Tilaukselle syötettävistä tiedoista pakollisia ovat nimi, puhelinnumero ja kategoria.

Lomake - MUOKKAUSTILA

Nimi: Geneerinen Testihenkilö

Puh: 112

Tilattava tuote: Koiran naksuja iltapalaksi

Toimittaja /
Saatavuus: Fazer

Tilattu: 20.02.2017

Saapunut /
ilmoitettu: _____

Kategoria: Koirat
 Kissat
 Kanit ja jyrsijät
 Matelijat ja selkärangattomat
 Akvaario
 Linnut
 Hoitoeläimet
 Sekalaiset

MUOKKAA TYHJENNÄ LOMAKE PERUUTA

KUVA 4. Tilausten lisäys- ja muokkausikkuna

5.4 Varmuuskopio

Tietokannasta voi luoda varmuuskopion muistikortille. Jos tabletissa on muistikortti paikoillaan, sovellus tarkistaa löytyykö kohteesta jo varmuuskopiota ja milloin mahdollinen varmuuskopio on luotu. Jos varmuuskopiota ei löytynyt tai edellisen varmuuskopion luonnista on kulunut yli 14 vuorokautta, luo sovellus automaattisesti uuden varmuuskopion. Varmuuskopion luonnissa oletusnimen lisäksi tarjotaan sen hetkistä päivämäärää. Nimen voi vaihtaa, mutta tallennettujen varmuuskopioiden selaamisen helpottamiseksi tarjotaan nimen liitteeksi sen hetkistä päivämäärää. Näin nimestä näkee suoraan mikä on viimeisin varmuuskopio. Sovellus listaa kaikki tallennuskohteessa olevat varmuuskopiot alavetovalikkoon, josta voi valita palautettavan version.



KUVA 5. Varmuuskopion luonti- ja palautusikkuna

Tallentamista ja lukemista varten sovelluksen täytyi päästä käsiksi ulkoiseen tallennustilaan. Android-käyttöjärjestelmässä tähän vaaditaan käyttäjän suostumus sovelluksen asennusvaiheessa. Sovelluksen vaatimat oikeudet lisätään AndroidManifest.xml-tiedostoon (kuva 6).

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

KUVA 6. Sovelluksen käyttäjältä vaatimat oikeudet

5.5 Testaus

Mitään erillistä testisuunnitelmaa ei luotu, vaan toimintoja testattiin samalla kun niitä tehtiin. Tällä tavalla varmistettiin yksittäisten komponenttien toiminnallisuus. Lopussa sovelluksen kaikkia toimintoja testattiin kokonaisuudessaan ja varmistettiin, että kaikki ominaisuudet toimivat yhdessä. Lopuksi sovellus annettiin vielä ulkopuolisille testikäyttäjille kokeiltavaksi. Kun havaitut virheet oli korjattu, eikä parannettavaa enää löytynyt, toimitettiin asiakkaalle sovelluksen lopullinen versio.

6 JATKOKEHITYS

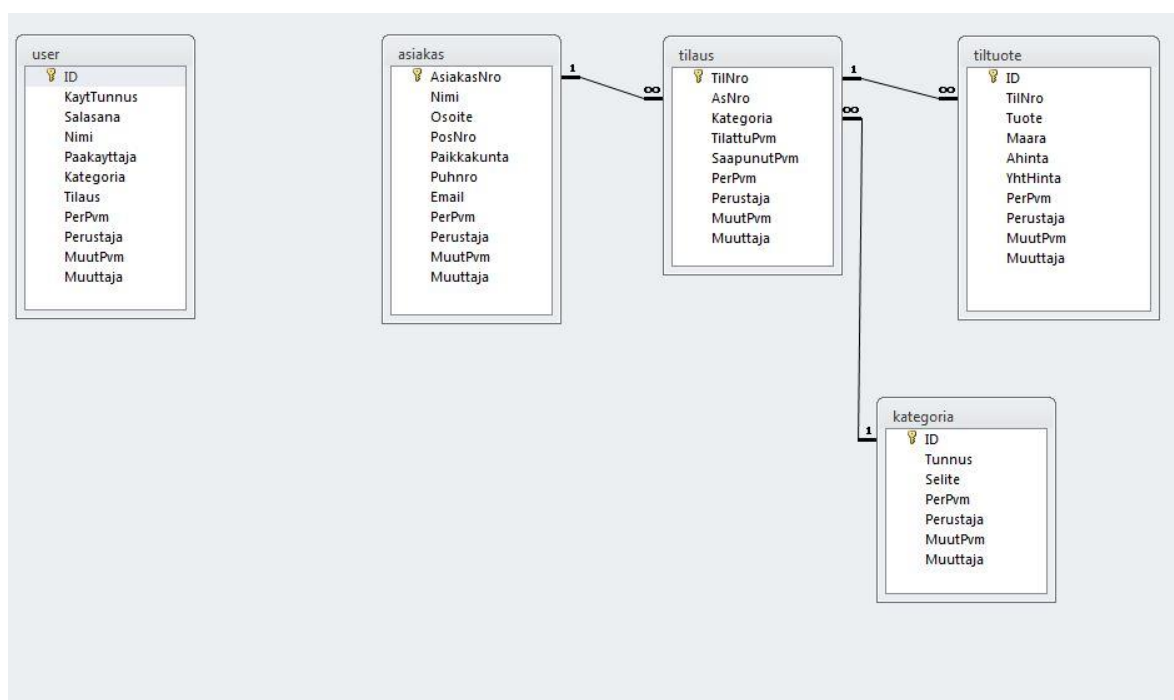
Sovellukseen jäi mahdollisuuksia jatkokehitykselle. Ominaisuuksia lisäämällä tilaustenhallinnasta tuli si parempi. Lisättäviä ominaisuuksia ovat:

- käyttäjätunnukset
- käyttäjien oikeudet
- asiakasrekisteri
- dynaaminen käyttöliittymä.

Ominaisuuksista keskusteltiin työn tilaajan kanssa mutta niitä ei kuitenkaan päätetty toteuttaa. Tämä johtui lähinnä siitä, että tilaajalla ei ollut tarvetta kyseisille lisäominaisuuksille. Lisäominaisuudet olisivat vaatineet paremman tietokannan ja tilaaja ei halunnut hankkia itselleen palvelintilaa internettistä tai pystyttää omaa palvelinta. Koska kaikki tarvittavat palvelut tulevat ketjun kautta, yritys ei halunnut hankkia toisia niiden lisäksi. Ketjun tarjoamia palveluita ei ollut mahdollista hyödyntää yrityksen omiin käyttötarkoituksiin.

6.1 Tietokanta

Tietokanta jäisi lisäominaisuuksien jälkeenkin vielä melko pieneksi mutta ominaisuuksien toteuttamista varten tarvittaisiin kuitenkin lisää tauluja. Tilaajan toivomuksesta ohjelman ensimmäisessä versiossa tiedot tallennetaan laitteen omalle levyille ja muistikortille. Vaihtoehtoisista tallennuspaikoista keskusteltiin, mutta niitä ei tällä kerralla toteutettu. Kuvassa 7 alustava suunnitelma mahdollisesta tietokannasta.



KUVA 7. Tietokannan rakenne jatkokehityksessä

Tietokannassa jokaiseen tauluun tulisi sarakkeet lisäys- ja muokkauspäivämäärää varten sekä sarakkeet niiden käyttäjien nimeä varten, jotka ovat lisäyksen tai muokkasen on tehneet. Esimerkiksi tilauspäivämäärän lisäksi jäisi myös käyttäjän nimi tilauksen tietoihin. Jos tilauksessa olisi jotain epäselvyyksiä, voisi asiaa selvittää sen työntekijän kanssa joka on tilauksen ottanut vastaan.

6.2 Asiakasrekisteri

Asiakasrekisteriä varten luotaisiin samankaltainen ikkuna kuin tilauksien lisäyksessä, jossa voisi hallinnoida kaikkia asiakkaan tietoja. Tilasta lisättäessä asiakas valittaisiin asiakasrekisteristä. Jos asiakasta ei olisi tietokannassa, syötettäisiin uuden asiakkaan tiedot tilasta tehtäessä ja tallennettaisiin sekä tilauksen, että asiakkaan tiedot samalla kerralla. Kun asiakkaat tekevät uuden tilauksen, ei heidän tietojaan tarvitsisi syöttää enää uudestaan.

6.3 Dynaaminen käyttöliittymä

Etusivulla kategoria napit ovat sovelluksen ensimmäisessä versiossa kiinteästi ohjelmoituja. Kategoriatkin voisivat olla tietokannassa mistä ohjelmaa käynnistettäessä haetaan kaikki kategoriat ja lisätään ne dynaamisesti näytölle. Kategorioita voisi lisätä ja poistaa jos uusia ryhmiä tulee muiden rinnalle tai tuotteita jää valikoimasta pois ja vanhalle tuoteryhmälle ei ole enää tarvetta. Tällä tavalla käyttäjä voisi helposti muokata ohjelman ulkoasua ja tilausten ryhmittelyä myynnissä olevien tuotteiden perusteella. Näin ei tarvitsisi ottaa tekijään yhteyttä ja muutokset tulisivat näkyviin heti ilman päivityksien asentamista.

6.4 Käyttäjät

Jos ohjelmalla hallinnoitaisiin asiakkaiden, tilauksien ja tuotteiden tietoja sekä sovelluksen ulkoasua, voisi ohjelmaan lisätä kaikille käyttäjille omat tunnukset. Vaikka tällaista ominaisuutta ei ensimmäiseen versioon haluttu, olisi jatkokehityksen kannalta hyvä luoda käyttäjille erilaisia oikeuksia. Pääkäyttäjän oikeuksilla voisi hallinnoida kaikkea ja oikeuksia voitaisiin rajoittaa muille käyttäjille. Vain pääkäyttäjän tunnuksilla voi antaa tai rajoittaa muiden käyttäjien oikeuksia. Kuvassa 8 on esitetty havainnekuva käyttäjätietojen hallintaikkunasta, jossa annetaan käyttäjille erilaisia oikeuksia.

Käyttäjätiedot

<p>Nimi: <input type="text" value="Johtaja"/></p> <p>Käyttäjätunnus: <input type="text" value="Master"/></p> <p>Salasana: <input type="password" value="••••••••"/></p>	<p>Käyttöoikeudet</p> <p>Pääkäyttäjä <input type="checkbox"/></p> <p>Käyttäjätiedot <input type="text" value="Täydet"/></p> <p>Kategoria <input type="text" value="Ei oikeuksia"/></p> <p>Tilaus <input type="text" value="Vain luku"/></p> <div style="border: 1px solid black; padding: 2px; width: fit-content;"> <p>Täydet</p> <p style="background-color: #007bff; color: white;">Vain luku</p> <p>Ei oikeuksia</p> </div>
<p>Perustettu: 25.04.2017 - Admin</p> <p>Muokattu: 30.04.2017 - Myymäläpäällikkö</p>	

KUVA 8. Käyttäjätietojen hallintaikkuna

Tavalliselle käyttäjälle eli työntekijöille voisi esimerkiksi antaa rajoitettuja oikeuksia tiettyihin paikkoihin. Käyttäjätiedoissa oikeuksia rajoitettaisiin vain omien tietojen muokkaamiseen. Toisen käyttäjän tietoja ja oikeuksia pääsee muokkaamaan vain jos on saanut kyseiset oikeudet. Myös ohjelman kategorioiden muokkausta voitaisiin rajoittaa vain tietyille henkilöille. Esimerkiksi vain myymäläpäällikkö voisi muokata ryhmiä ja työntekijöillä ei olisi mahdollisuutta edes päästä kyseiselle sivulle. Asiakkaiden ja tilaustietojen käyttörajoituksille ei välttämättä olisi suurempaa tarvetta, jotta asiakaspalvelu ei kärsisi. Korkeintaan asiakastietojen poistoa voisi käyttäjiltä rajoittaa ja antaa vain luku ja kirjoitus oikeudet. Työsuhteen päättyessä työntekijän tunnukset on helppo poistaa.

6.5 Web-pohjainen sovellus

Vaikka ohjelma suunniteltiin ainoastaan tilaajan tarpeita noudattaen, olisi koko sovelluksen voinut tehdä myös selainpohjaisena. Tätä sovellusta tehtäessä yrityksellä oli käytössään vain Android-laitteita. Kun laitteet hajoavat on uusienkin laitteiden oltava Android-käyttöjärjestelmällä varustettuja. Web-pohjaisessa sovelluksessa laitteiston käyttöjärjestelmävaatimukset poistuisivat ja laitteiston uusiminen ei aiheuttaisi minkäänlaisia yhteensopivuus ongelmia.

Web-sovelluksen etuna olisi myös se, että ohjelmaa voisi käyttää tietokoneella. Työpöytäsovellus ei ollut vaihtoehto, koska myymälän tiskille ei mahtunut uutta konetta ja siksi päädyttiin mobiiliversioon. Liiketilat voivat kuitenkin muuttua ja uusia työasemia tulla käyttöön. Näilläkin koneilla voisi tilauksia hoitaa ilman erillisiä ohjelmisto asennuksia. Nyt tietokoneella käyttö ei ole mahdollista vaikka työasemia olisi käytettävissä. Lisäksi ruuhka-aikana tilauksia voisi kirjata useilla laitteilla samanaikaisesti.

6.6 Rajapinta

Data siirtyisi ohjelman ja palvelimen välillä JSON-muodossa. Palvelimen päässä tietokantakyselyitä sekä JSON-objektin muodostamista hoitaisi PHP-kiellä ohjelmoitu rajapinta. Ohjelmasta kutsuttaisiin rajapintaa eri parametreillä joka palauttaisi tiedot JSON-objektina. Hakuparametreja olisivat asiakkaan nimi sekä tilauksen kategoria. Tietoja haettaessa lähetetään hakuparametrit palvelimelle GET-komennolla eli liitetään tieto URL-osoitteeseen ja tilausta syötettäessä tiedot siirrettäisiin POST-komennolla.

JSON-objektin rakenne olisi kuvassa 9 näkyvän esimerkin mukainen. Sekä syötettäessä että haettaessa tilausta, tiedot siirrettäisiin samanlaisella objektilla.

```
[
  {
    "tilausnro": "",
    "kirjauspvm": "",
    "tilauspvm": "",
    "ilmoituspvm": "",
    "toimittaja": "",
    "kategoria": "",
    "asiakas":
    {
      "asnro": "",
      "nimi": "",
      "puhno": "",
      "email": "",
      "osoite": "",
      "posno": "",
      "paikkakunta": ""
    },
    "tiltuotteet":
    [
      {
        "tuote": "",
        "maara": "",
        "ahinta": "",
        "yhthinta": ""
      }
    ]
  }
]
```

KUVA 9. Esimerkki JSON-tiedonsiirtomuodossa käytettävästä objektista

7 YHTEENVETO

Työn tavoitteena oli suunnitella ja luoda tilaustenhallintasovellus Android-pohjaisille tabletlaitteille, joka soveltuisi tilaajayrityksen tarpeisiin. Projekti oli mielenkiintoinen ja tarjosi sopivasti haasteita ja olen tyytyväinen lopputulokseen.

Suunnitteluvaiheessa käytiin läpi sovelluksen vaatimukset ja sovittiin alustavasta toimitusajasta. Projekti suunniteltiin hyvin joten työnteko eteni vaivattomasti. Sovellus kehitettiin Android Studio kehitysohjelmalla ja testattiin fyysisillä Android-laitteilla. Projektin aikataulu kuitenkin venyi alkuperäisestä toimitusajasta ja sovelluksen toimitus hieman viivästyi. Kaikki vaaditut ominaisuudet saatiin tehtyä ja työn tilaaja oli lopputulokseen tyytyväinen aikataulun venymisestä huolimatta. Sovelluksesta on tullut hyvää palautetta.

LÄHTEET JA VIITTAUKSET

ADOBE PHOTOSHOP. [Viitattu 2017-01-31]. Saatavissa:

https://fi.wikipedia.org/wiki/Adobe_Photoshop_Elements

ANDROID DEVELOPERS, DASHBOARD. [Viitattu 2017-04-21]. Saatavissa:

<https://developer.android.com/about/dashboards/index.html>

ANDROID DEVELOPERS, ANDROID STUDIO. [Viitattu 2017-01-31]. Saatavissa:

<https://developer.android.com/studio/intro/index.html>

ANDROID. [Viitattu 2017-01-31]. Saatavissa:

<https://fi.wikipedia.org/wiki/Android>

HYNNINEN, Teemu 2013. Androidin historia: Astrosta KitKattiin [Viitattu 2017-02-26]. Saatavissa:

<http://www.mobiiliblogi.com/2013/07/20/androidin-historia-astrosta-jelly-beaniin/>

JAVA. [Viitattu 2017-04-15]. Saatavissa:

<https://fi.wikipedia.org/wiki/Java>

W3C. Extensible Markup Language [Viitattu 2017-04-20]. Saatavissa:

<https://www.w3.org/XML/>