

Opinnäytetyö (AMK)
Tietotekniikan koulutusohjelma
Sulautetut ohjelmistot
2012

Lauri Kare

RFID-LUKIJAA HYÖDYNTÄVÄ ANDROID-POHJAINEN ESITTELYSOVELLUS



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Sulautetut ohjelmistot

Helmikuu 2012 | 41 s.

Ohjaaja: TkL Jari-Pekka Paalassalo

Lauri Kare

RFID-LUKIJAA HYÖDYNTÄVÄ ANDROID- POHJAINEN VERKKOSOVELLUS

Tässä opinnäytetyössä suunniteltiin ja toteutettiin Nordic ID Sampo RFID -lukijaa hyödyntävä Android-verkkosovellus, verkkopohjainen käyttöliittymä ja tietokanta. Projektin tarkoituksena oli luoda esittelysovellus messutapahtumiin, jolla havainnollistetaan sitä, että Nordic ID Sampo RFID -lukijaa voidaan hyödyntää myös Android-taulutietokoneessa.

Android-verkkosovellus ohjelmoitiin Javalla ja täydennettiin XML-tekniikalla. Verkkopohjainen käyttöliittymä ohjelmoitiin käyttäen HTML-, CSS-, PHP-, JavaScript-, Ajax- ja tietokanta-tekniikoita. Tietokantarakenteesta suunniteltiin aluksi EER-malli, jonka pohjalta se toteutettiin käyttäen MySQL-relaatiotietokannan hallintajärjestelmää.

Projektissa saatiin tehtyä toimiva esittelysovellus, jota voidaan esitellä sellaisenaan tai sen kehitystä voidaan jatkaa.

ASIASANAT:

AJAX, Android, CSS, Java, JavaScript, MySQL, PHP, RFID, SQL, verkko-ohjelmointi, XML

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Information Technology | Embedded Software

2012 | 41 p.

Instructor: M. Sc Jari-Pekka Paalassalo, Lic. Sc. (Tech.), Principal Lecturer

Lauri Kare

AN ANDROID BASED WEB APPLICATION USING AN RFID READER

The subject of this thesis was to design and achieve an Android based web application that uses the Nordic ID Sampo RFID reader, a web based interface and database. The purpose of this project was to create software to demonstrate that it is possible to use the Nordic ID Sampo RFID reader with an Android tablet.

The programming language used in the implementation of the Android application was Java. The Android application layouts were created with the technologies of XML. The web based interface was programmed with the technologies of CSS, PHP, JavaScript, Ajax and database. A database structure was designed by using an EER-diagram and was later implemented with the help of the relational database management system MySQL.

The result of the project was a functional demonstration application that can be used for presentations as such or can be developed further.

KEYWORDS:

AJAX, Android, CSS, Java, JavaScript, MySQL, PHP, RFID, SQL, web development, XML

SISÄLTÖ

LYHENTEET JA SYMBOLIT	6
1 JOHDANTO	8
2 KÄYTETYT TEKNIIKAT JA TYÖKALUT	9
2.1 HTML	9
2.2 CSS	10
2.3 PHP	10
2.4 JavaScript	11
2.5 DOM	12
2.6 AJAX	13
2.7 Java	13
2.8 SQL	14
2.8.1 MySQL	14
2.8.2 PhpMyAdmin	15
2.9 Android	15
2.10 Eclipse	15
2.11 Acer Iconia Tab A500	16
2.12 Nordic ID Sampo	17
3 VAATIMUSMÄÄRITTELY JA SUUNNITTELU	18
3.1 Android-pohjainen verkkosovellus	19
3.2 Verkkopohjainen käyttöliittymä ja tietokanta	19
4 KÄYTÄNTÖ JA TOTEUTUS	22
4.1 Android-verkkosovellus	22
4.1.1 Manifest	23
4.1.2 Activity	25
4.1.3 Context	27
4.1.4 Intent	27
4.1.5 USB	28
4.1.6 WebView	29
4.1.7 VideoView	30
4.1.8 JavaScript-rajapinta	31

4.2 Tietokanta	33
4.2.1 Tuotetietokantarakenne	33
4.2.2 Tuotepalautteen tietokantarakenne	34
4.2.3 Istunnon hallinnointijärjestelmän tietokantarakenne	34
4.3 Käyttöliittymä	35
5 YHTEENVETO	39
LÄHTEET	40

KUVAT

Kuva 1. HTML-sivun käsittely.	9
Kuva 2. PHP-sivun käsittely.	11
Kuva 3. Acer Iconia Tab A500 -taulutietokone.	16
Kuva 4. Nordic ID Sampo RFID -lukija.	17
Kuva 5. Tuotteiden tietokantarakenne.	20
Kuva 6. Tuotepalautteiden tietokantarakenne.	20
Kuva 7. Istunnon hallinnointijärjestelmän tietokantarakenne.	21
Kuva 8. AndroidManifest.xml-tiedoston rakenne.	24
Kuva 9. Aktiviteetin elinkaari.	26
Kuva 10. Android USB-isäntä ja -lisälaite tilojen väliset erot.	28
Kuva 11. Käyttöliittymän tuotetieto-sivu.	36
Kuva 12. Käyttöliittymän tuotesaatavuus-sivu.	36
Kuva 13. Käyttöliittymän tuotekori-sivu.	37

TAULUKOT

Taulukko 1. Android-alustat.	22
-------------------------------------	----

KOODIESIMERKIT

Koodiesimerkki 1. Android WebView-ikkunanäkymä.	29
Koodiesimerkki 2. Android VideoView-ikkunanäkymä.	30
Koodiesimerkki 3. Videotiedostojen toistaminen Android-verkkosovelluksella.	31
Koodiesimerkki 4. Javascript-rajapinnan toteutus.	32
Koodiesimerkki 5. Käyttöliittymän index.php-tiedoston rakenne.	38

LYHENTEET JA SYMBOLIT

AJAX	Tahdistamaton JavaScript ja XML (Asynchronous JavaScript And XML) on useista verkko-ohjelmointitekniikoista koostuva joukko, jonka avulla verkkosovelluksista voidaan tehdä vuorovaikutteisempia, nopeampia ja käytettävämpiä.
API	Ohjelmointirajapinta (Application Programming Interface) on lähdekoodiin perustuva määritelmä, jonka tarkoituksena on antaa kommunikointirajapinta eri käyttöliittymien välille.
CSS	Porrastetut tyyliarkit (Cascading Style Sheets) on kehitetty verkkosovellusten sisällön, rakenteen ja tyylin kuvaavaksi säännöstöksi.
DOM	Dokumenttioliomalli (Document Object Model) on verkkosovelluksille kehitetty ohjelmointirajapinta, jonka avulla HTML- ja XML-dokumentin elementtejä tai sisältöä voidaan muokata, luoda tai poistaa.
EPC	Sähköinen tuotekoodi (Electronic Product Code) on RFID-tunnisteita varten kehitetty yksilöllinen koodausmenetelmä, jota ylläpitää EPCglobal.
HTML	Hypertekstin merkintäkieli (Hypertext Markup Language) on hypertekstin rakenteen ja ulkoasun kuvauskieli.
HTTP	Hypertekstin yhteyskäytäntö (Hypertext Transfer Protocol) on tahdistettu tiedonsiirron yhteyskäytäntö, jota verkkoselaimet ja -palvelimet käyttävät tiedonsiirtoon.
Java	Sun Microsystemsin kehittämä järjestelmäriippumaton ja täysin oliopohjainen ohjelmointikieli.
JavaScript	Netscape Communications Corporationin kehittämä asiakaspohjainen komentosarjakieli, jonka avulla verkko-sovelluksiin saadaan dynaamista ja vuorovaikutteista toiminnallisuutta.
PHP	PHP: hypertekstin esiprosessointi (PHP: Hypertext Preprocessor) on verkkosovellusten ohjelmointiin kehitetty palvelinpuolen komentosarjakieli, jolla verkkosovelluksista saadaan tehtyä dynaamisia.
RFID	Radiotaajuinen etätunnistus (Radio Frequency IDentification) on menetelmä tiedon etäkirjoittamiseen ja -lukemiseen käyttäen RFID-tunnisteita.
SDK	Ohjelmistokehityspakki (Software Development Kit) koostuu tyypillisesti ohjelmistokehitystyökaluista, joilla voidaan kehittää juuri tiettyä ohjelmistopakettia, ohjelmistoalustaa, laitealustaa tai järjestelmää.

SKU	Varastointinimike (Stock-Keeping Unit) on erilaisia tuotteita tai palveluita varten kehitetty yksilöllinen koodausmenetelmä.
SQL	Jäsentynyt kyselykieli (Structured Query Language) on maailman yleisin kysely- ja määrittämissä, jolla voidaan tehdä kyselyjä relaatiotietokantaan. Kyselyillä voidaan määrittää, muuttaa ja poistaa tietokantoja. Sillä voidaan myös käsitellä tietueita tai luoda kokonaan uusia.
XML	Laajennettavissa oleva merkintäkieli (Extensible Markup Language) on kansainvälinen yritysten ja yhteisöjen yhteenliittymän kehittämä merkintäkieli.
XHR	XMLHttpRequest on ohjelmointirajapinta verkkoselaimessa ajettavalle komentosarjakiellelle, joka mahdollistaa tahdistamattoman tiedonsiirron verkkoselaimen ja -palvelimen välille.
W3C	Kansainvälinen yritysten ja yhteisöjen yhteenliittymä (World Wide Web Consortium) ylläpitää ja kehittää World Wide Web Standardeja (W3C käyttää termiä "suositus").

1 JOHDANTO

Erilaiset radiotaajuiset etätunnistusteknologiat (engl. RFID technology) ovat nykypäivää ja niiden käyttö yleistyy erittäin nopeaa tahtia. Niille keksitään myös jatkuvasti uusia mitä innovatiivisimpia ja erilaisimpia käyttötarkoituksia. Tämän insinööriyön tarkoituksena on suunnitella ja toteuttaa Nordic ID Sampo RFID -lukijaa hyödyntävä esittelysovellus.

Esittelysovellus pohjautuu tuoteideaan, jossa kaupanalan asiakkaille voidaan kertoa tarkempaa tietoa tuotteista, joita he ovat poimineet sovitettavaksi tai tunnistettavaksi. Tuotteet tunnistetaan RFID-tunnisteiden avulla, jolloin asiakkaalle voidaan kertoa tuotteen tarkemmat tiedot, siihen mahdollisesti liittyvät alennukset ja samalla kohdistaa markkinointia asiakkaalle.

Kyseisessä tuoteideassa tuotteen tiedot kohdistetaan asiakkaalle seinään upotetulla taulutietokoneella, joka on kytketty RFID-lukijaan ja tietoverkkoon. Käyttökohteena on sovituskoppi tai sille asetettu tunnistepiste.

Vastaavia tuotteita löytyy jo markkinoilta. Tämä tuoteidea poikkeaa muista lähinnä siten, että se käyttää Nordic ID Sampo RFID -lukijaa. Myös erilaisia muunnoksia löytyy runsaasti, kuten esimerkiksi työajanseurantaan tarkoitettu kellokorttilaite tai kirjojen lainaus- ja palautuspiste.

Tässä insinööriyössä tehdään kosketusnäyttöiselle Android-taulutietokoneelle esittelysovellus, joka käyttää Nordic ID Sampo RFID -lukijaa. Kyseinen esittelysovellus koostuu Android-laitteelle suunniteltavasta Java-pohjaisesta verkkosovelluksesta ja verkkopohjaisesta käyttöliittymästä. Käyttöliittymän rakentamiseen käytetään HTML-, CSS-, PHP-, JavaScript-, AJAX- ja tietokantaohjelmointia.

2 KÄYTETYT TEKNIIKAT JA TYÖKALUT

Tässä luvussa tarkastellaan teoreettista osuutta projektissa käytetyille tekniikoille ja työkaluille. Sen tarkoituksena on avata käytettyjen tekniikoiden yhteyttä toisiinsa ja kertoa niiden ominaisuuksista.

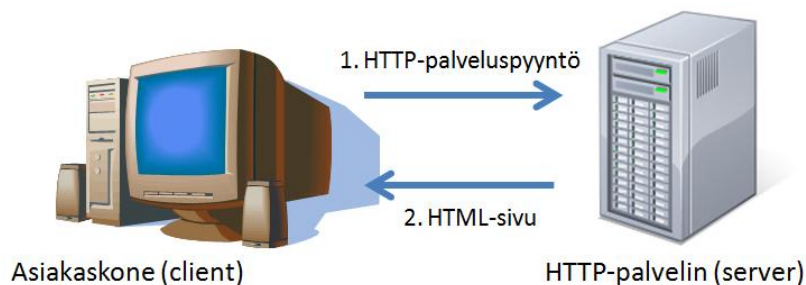
2.1 HTML

HTML eli Hypertext Markup Language (hypertekstin merkintäkieli) on hypertekstin rakenteen ja ulkoasun kuvauskieli. Kieli on avoimesti standardoitu, sitä ylläpitää ja kehittää kansainvälinen yritysten ja yhteisöjen yhteenliittymä (World Wide Web Consortium), joka tunnetaan myös nimellä W3C. [1]

Tavallisen HTML-tiedoston näyttämiseen vaaditaan sovellus (verkkosovellus tai verkkoselain), joka pystyy tulkitsemaan HTML-kieltä. Mikäli tiedostoa halutaan käyttää verkkotasolla, tarvitaan tiedonsiirtoon hypertekstin yhteyskäytäntö-palvelinohjelma (Hypertext Transfer Protocol, HTTP). [1]

Kuvassa 1 on esitetty HTML-sivun käsittely, jonka välivaiheet ovat seuraavat:

1. Asiakaskoneen verkkoselain lähettää HTTP-palveluspyynnön HTTP-palvelimelle.
2. HTTP-palvelin suorittaa pyydetyt palvelut ja lähettää vastauksen asiakkaan verkkoselaimelle, joka on yleensä HTML-tiedosto.



Kuva 1. HTML-sivun käsittely.

Projektin HTTP-palvelinohjelmana toimii Apache HTTP -palvelin, joka perustuu avoimeen lähdekoodiin. Apache on Internetin käytetyin HTTP-palvelin, ja se on erittäin helppo sekä asentaa että säätää. [2]

2.2 CSS

CSS eli Cascading Style Sheets (porrastetut tyyliarkit) on kehitetty verkkosovellusten sisällön, rakenteen ja tyylin kuvaavaksi säännöstöksi. W3C ylläpitää CSS-tasojen määritelmiä, jotka löytyvät W3C:n verkkosivuilta: <www.w3.org/TR/CSS>. [3]

CSS-säännöstöjen tarkoitus on siis kertoa sovellukselle, kuinka dokumentti tulee esittää. Kyseiset säännöstöt voidaan määrittää joko kirjoittamalla ne suoraan dokumenttiin tai erilliseen tyylitiedostoon. Tekniikan parhaimmat edut tulevat esille dynaamisten dokumenttien luonnissa, jolloin sisällön vaihtuessa ulkoasu määräytyy säännöstöjen mukaisesti.

2.3 PHP

PHP eli PHP: Hypertext Preprocessor (PHP: hypertekstin esiprosessointi) on verkkosovellusten ohjelmointiin kehitetty palvelinpuolen komentosarjakieli, jolla verkkosovelluksista saadaan tehtyä dynaamisia. Kieli on ominaisuuksiltaan kustannustehokas, sillä se on lisenssivapaa, tehokas ja käytännöllinen sekä kaikki yleisimmät järjestelmät tukevat sitä. [4]

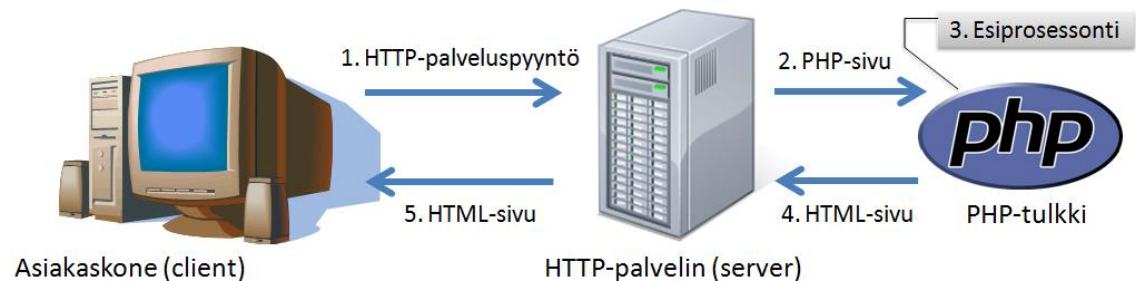
PHP-sivun viisi eri käsittelyvaihetta ovat seuraavat:

1. Asiakaskoneen verkkoselain lähettää HTTP-palveluspyynnön HTTP-palvelimelle.
2. Palvelin toteaa pyynnön koskevan PHP-tiedoston ja pyytää PHP-tulkin esiprosessoimaan tiedoston.
3. PHP-tulkki esiprosessoi (kääntää ja suorittaa) pyydetyn tiedoston, jonka pohjalta syntyy HTML-koodia.

4. PHP-tulkki palauttaa HTML-tuotoksen HTTP-palvelimelle.

5. HTTP-palvelin palauttaa HTML-koodin asiakaskoneen verkkoselaimelle.

Nämä vaiheet on esitetty myös kuvassa 2.



Kuva 2. PHP-sivun käsittely.

Dynaamisuudella saadaan verkkosovelluksiin toiminnallisuutta ja koodin uudelleenkirjoittaminen vähentyy olennaisesti. PHP:ssa on valmiina kattavat luokkakirjastot, jotka antavat esimerkiksi laajan tuen eri tietokannoille ja yhteyskäytännöille. Luokkakirjastoja voidaan laajentaa asentamalla PHP:n ohjelmalisäkkeitä (engl. extensions). Kielessä on myös rajoitettu olio-ohjelmointirajapinta, joten valmiita luokkia voidaan käyttää laajennuksina. [4, 5]

PHP:lla voidaan luoda dynaamisten verkkosovellusten lisäksi myös komentorivi-sovelluksia ja graafisia käyttöliittymiä. Palvelinpuolen komentosarjakielillä tarkoitetaan sitä, että sovelluksen lähdekoodi tulkitaan palvelimella ja lopputulos palautetaan asiakkaan verkkoselaimelle. [5]

2.4 JavaScript

JavaScript on Netscape Communications Corporationin kehittämä asiakaspohjainen komentosarjakieli, jonka avulla verkkosovelluksiin saadaan dynaamista toiminnallisuutta. Kielen nykymuoto tukee oliopohjaista ohjelmointitapaa ja se pohjautuu ECMA-262-standardiin, josta käytetään myös nimitystä ECMAScript. ECMA-262-standardi löytyy Ecma Internationalin verkkosivuilta: www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf. [6, 7]

ECMA-262-standardiin on hyvä tutustua, koska sillä on laajin verkkoselaintuki. JavaScriptille löytyy myös paljon valmiita kirjastoja, joihin kannattaa tutustua. Yksi tunnetuimmista avoimeen lähdekoodin perustuvista JavaScript-kirjastoista on jQuery. [6]

Asiakaspohjaisella komentosarjakiielellä tarkoitetaan sitä, että komentosarjoja ei suoriteta palvelimella, vaan asiakkaan verkkoselaimella. Tämän vuoksi asiakkaan verkkoselaimen tulee tukea JavaScriptiä ja sen asetuksista pitää olla JavaScript asetettuna päälle.

2.5 DOM

DOM eli Document Object Model (Dokumenttioliomalli) on kelvollisille (engl. valid) HTML-dokumenteille ja hyvin muodostetuille (engl. well-formed) XML-dokumenteille kehitetty ohjelmointirajapinta. Dokumenttioliomallilla voidaan päästä käsiksi, muokata, luoda tai poistaa HTML- tai XML-dokumenteista löytyviä elementtejä, muutamaa poikkeusta lukuun ottamatta. [8]

Dokumenttioliomallin toimintaperiaate on se, että verkkosovellus tai -selain paketoit HTML- tai XML-dokumentin dokumenttioliomalliksi, jolloin esimerkiksi JavaScriptillä päästään dokumentin elementteihin käsiksi. Verkkoselaimen tulee tukea JavaScriptiä ja XMLHttpRequest-ohjelmointirajapintaa (XHR). Tämän lisäksi JavaScript pitää olla asetettuna päälle verkkoselainasetuksista. [8]

Dokumenttioliomallissa dokumentin tieto esitetään puumaisessa elementti-tietorakenteessa, jossa puun solmut (engl. nodes) ovat hierarkiassa yksittäisiä elementtiolioita, joilla on omat ominaisuudet ja metodit. Dokumenttipuussa voidaan viitata solmun lapseen (engl. child), vanhempaan (engl. parent) tai sisarukseen (engl. sibling). [8]

2.6 AJAX

AJAX eli tahdistamaton JavaScript ja XML (Asynchronous JavaScript And XML) on useista toisiinsa liittyvistä asiakaspuolen verkko-ohjelmointitekniikoista koostuva joukko, jonka avulla voidaan tehdä vuorovaikutteisia, nopeita ja käytettävyydeltään parempia verkkosovelluksia. [9]

Ajax-ohjelmointitekniikan avulla asiakkaan verkkoselaimen ja palvelimen välinen liikenne tapahtuu tahdistamattomasti, eli verkkosivuston toiminnallisuus tapahtuu huomaamattomasti taustalla. Tekniikan etuna on myös huomattava nopeus, sillä vain tarvittava osa sisällöstä päivitetään. Verkkosovellusten käytettävyys myös paranee huomattavasti, koska käyttäjälle ei tule katkoksia sisällön vaihtuessa. [9]

Ajax-ohjelmointitekniikka vaatii verkkoselaimelta tukea JavaScriptiltä ja XHR-ohjelmointirajapinnalta. Tämän lisäksi JavaScript pitää olla asetettuna päälle verkkoselainasetuksista.

2.7 Java

Java on Sun Microsystems inc:n kehittämä järjestelmäriippumaton ja täysin oliopohjainen ohjelmointikieli. Järjestelmäriippumattomuudella tarkoitetaan sitä, että tavukoodiksi (engl. byte code) käännettyjä Java-ohjelmia voidaan ajaa kaikissa järjestelmissä, jotka tukevat Java-virtuaalikonetta. [10]

Java on kehitetty C++-ohjelmointikielestä, ja se on pyritty rakentamaan yksinkertaiseksi, helppokäyttöiseksi ja turvalliseksi. Tällä tarkoitetaan sitä, että kielestä on poistettu niin sanotut tarpeettomat, vaaralliset ja harvoin käytettävät ominaisuudet. Kyseisiä ominaisuuksia ovat esimerkiksi: struktuurit (engl. struct), osoittimet (engl. pointer), unionit (engl. union), operaattorien ylikuormitus (engl. overloading), esiprosessointi (preprocessor), funktiot ja luokkien moniperintä (engl. multiple inheritance). [10]

Javassa ei ole mahdollista käyttää osoittimia eli tietotyyppiä, joka viittaa juuri tiettyyn keskusmuistiosoitteeseen. Osoittimien käyttö johtaa helposti virheisiin, jotka ovat vaikeasti havaittavissa. Niiden käyttö on piilotettu käyttäjiltä viittausten avulla, mikä tekee ohjelmoinnista helpompaa, yksinkertaisempaa ja turvallisempaa. [10]

2.8 SQL

SQL eli jäsentynyt kyselykieli (Structured Query Language) on maailman yleisin kysely- ja määrittämissä, jolla voidaan tehdä kyselyjä relaatiotietokantaan. Kyselyillä voidaan määrittää, muuttaa ja poistaa tietokantoja. Sillä voidaan myös käsitellä tietueita (engl. records) tai luoda kokonaan uusia. [11]

Relaatiotietokanta on tietokanta, jossa tiedot esitetään taulujen ja niiden välisten yhteyksien avulla. Kyseiset taulut koostuvat riveistä, ja yhtä riviä kutsutaan tietueeksi. Taulun jokaisella tietueella on yhtä monta kenttää ja siitä pitää löytyä yksikäsitteinen pääavain (engl. primary key). Pääavaimen lisäksi taulussa voi olla useampi toissijainen avain (engl. secondary key). [11]

Relaatiotietokantataulujen väliset suhteet luodaan avainten avulla, mikä mahdollistaa tarkan tiedonhaun sekä antaa käsitteen siitä, miten tietueet ovat yhteydessä toisiinsa. [11]

2.8.1 MySQL

MySQL on maailman suosituin avoimeen lähdekoodiin perustuva SQL-tietokantojen hallintajärjestelmä, jota nykypäivänä Oracle Corporation kehittää ja levittää. MySQL-ohjelmistosta on julkaistu myös kaupallinen lisenssi. MySQL-ohjelmiston suurin käyttökohte on verkkosovellusten kehityksessä, käyttäen PHP-, Perl- tai Python-komentosarjakieliä ja Apache HTTP -palvelinta. Ohjelmistosta löytyy myös tuki muille ohjelmointirajapinnoille, kuten esimerkiksi C:lle, C++:lle, C#:lle ja Javalle. [12]

2.8.2 PhpMyAdmin

PhpMyAdmin on avoimeen lähdekoodiin perustuva MySQL-tietokannan hallintatyökalu, jota käytetään verkkoselaimen kautta. Hallintatyökalulla voidaan suorittaa SQL-kyselyjä MySQL-tietokantaan, sekä sen käyttöliittymä antaa tuen useimmille tietokantatoiminnoille. Kyseisiä toimintoja ovat esimerkiksi tietokantojen, taulujen, tietueiden, kenttien, suhteiden, hakemistojen (engl. index), käyttäjien ja oikeuksien hallinta. [13]

2.9 Android

Android on Googlen julkaisema avoimeen lähdekoodiin perustuva ohjelmistopino, joka sisältää käyttöjärjestelmän sekä väli- ja perussovelluksia. Se on alun perin suunniteltu älypuhelimille ja mobiililaitteille, mutta nykyään siitä löytyy myös muunnosversio pöytäkoneelle. Android pohjautuu GPLv2-lisensoituun Linux-käyttöjärjestelmäyttimeen. [14]

Android-käyttöjärjestelmä perustuu pääosin Java-ohjelmointirajapintaan, jolle Google tarjoaa helposti lähestyttävän ohjelmistokehityspakin (Software Development Kit). Kehityspakki sisältää kasan työkaluja, kuten esimerkiksi kääntäjän (engl. compiler), ohjelmavirheiden paikantajan (engl. debugger) ja Dalvikin laite-emulaattorin (Dalvik Virtual Machine). [14]

2.10 Eclipse

Eclipse on erittäin suosittu avoimeen lähdekoodiin perustuva ohjelmointiympäristö, jota nykyään hallinnoi ja kehittää Eclipse Foundation. Ympäristö on ohjelmoitu pääosin Javalla ja se tukee yleisimpiä käyttöjärjestelmiä. Eclipse on kehitetty alun perin Java-ohjelmointiin, mutta nykypäivänä se tukee myös esimerkiksi C-, C++- ja PHP-ohjelmointikieliä. [15]

2.11 Acer Iconia Tab A500

Projektissa käytettiin Acer Iconia Tab A500 taulutietokonetta (kuva 3) ja jonka käyttöjärjestelmänä toimi Google Android 3.2.1 Honeycomb. Taulutietokoneessa olevan 10.1-tuumaisen kosketusnäytön erottelukyky ylittää 1280 x 800 kuvapisteeseen. Projektin kannalta muita olennaisia ominaisuuksia laitteella ovat mikro-USB-portti sovelluskehitykseen, USB-portti RFID-lukijan käyttöön ja langaton verkkoyhteys verkkosivujen näyttämiseen.



Kuva 3. Acer Iconia Tab A500 -taulutietokone. (Kuvassa näkyvän käyttöliittymän Copyright Nordic ID)

2.12 Nordic ID Sampo

Nordic ID Sampo on tuoteperhe, joka tarjoaa suorituskyvyltään korkeita ja monikäyttöisiä RFID-lukijoita (kuva 4). Tuoteperheen RFID-lukijoiden lukuetaisyys voidaan säätää muutamasta senttimetristä jopa kahdeksaan metriin. Tuoteperheen kommunikointi toteutetaan USB-portin tai Ethernet-liitännän välityksellä, riippuen mallista. Sampo RFID -lukijoiden käyttövirta voidaan ottaa USB-portista, Ethernet-liitännästä tai ulkoisesta virtalähteestä. Tuoteperheen muina ominaisuuksina ovat valo- ja kosketussensorit sekä mallista riippuen liitäntäportit kolmelle ulkoiselle lisäantennille. [16]



Kuva 4. Nordic ID Sampo RFID -lukija. [16]

3 VAATIMUSMÄÄRITTELY JA SUUNNITTELU

Laaja-alainen projekti vaatii aina hyvän projektisuunnitelman, mikä taas vaatii paljon aikaa ja yksityiskohtia. Projektin alkuvaiheessa pidimme opinnäytetyön tilaajan kanssa toimeksiantopalaverin, jossa projektille määritettiin aikataulu ja sitä koskevat vaatimukset:

- Projektin aloitus 1.12.2011 ja esittely opinnäytetyön tilaajalle 13.2.2012
- Käyttöliittymä toteutetaan täysiruututilassa (Android-verkkosovellus)
 - Sovelluksesta ei voi poistua ollenkaan
 - Näyttö ei saa sammua, himmentyä tai mennä virransäästötilaan
 - Ylimääräiset ominaisuudet karsitaan, kuten esimerkiksi:
 - Tekstin valinta
 - Navigointi-, otsikko- ja toimintapalkki
- JavaScript-rajapinta, jonka avulla käytetään RFID-lukijan ohjelmointirajapintaa
 - Voi toimia vain sovelluksen kanssa
- Verkkopohjainen käyttöliittymä
 - Kuluttajanäkymä tuoteselailuun, -tilaukseen ja -palautteeseen
 - Ohjausnäkökulma tuotteiden lisäämiseen, poistamiseen ja muokkaamiseen
 - Tiedotus siitä, missä paikassa kukin taulutietokone on
 - Mahdollisuus määrittää, mille käsipäätteelle tuotetilauspyynnöt lähetetään
- Käyttää CSS-tiedostoa pohjana
 - Käyttöliittymän asemointi määritetään prosentein tai muuten laiteriippumattomasti
 - Tulee noudattaa yrityksen määrittelemää käyttöliittymän asemointimallia
- Mahdollisesti myös
 - Android-sovellusten käynnistin tyhjällä työpöydällä, joka käynnistää esittelysovelluksen automaattisesti, mikäli se ei ole jo käynnissä
 - Käyttöliittymä käsipäätteelle
 - Käsipäätteenäkymä: tuotetilausten vastaanotto
 - Toteutus esimerkiksi C#:lla
 - Toimii taustalla ja pongahtaa muiden sovellusten päälle, mikäli saadaan tuotetilaus

Vaatimusmäärittelyistä ilmeni, että kaikkea ei ehditä välttämättä sisällyttää projektiin mukaan sen laajuuden vuoksi. Projektin suunnittelu jakaantui Android-pohjaiseen verkkosovellukseen, verkkopohjaiseen käyttöliittymään ja tietokantaan.

3.1 Android-pohjainen verkkosovellus

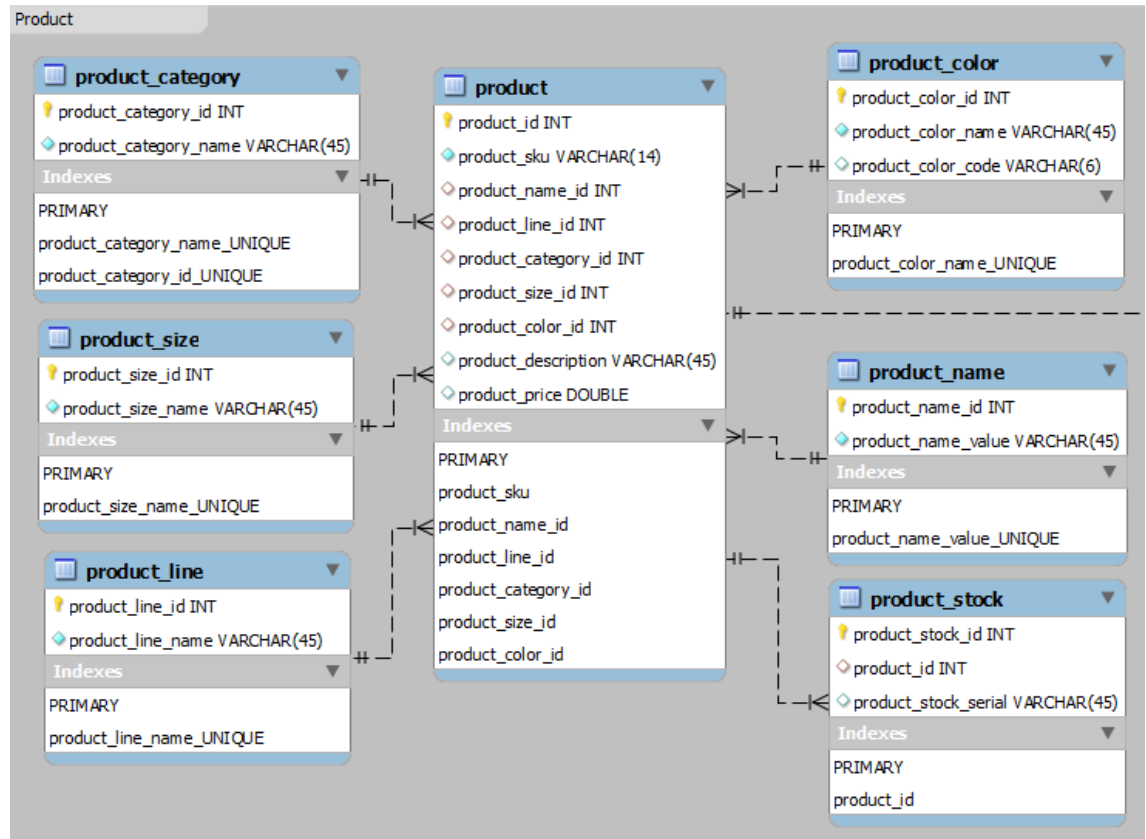
Android-pohjainen verkkosovellus ohjelmoidaan Javalla, jota täydennetään XML-tekniikalla. Vaatimusmäärittelyjen lisäksi verkkosovelluksen tulee pystyä hyödyntämään Nordic ID Sampo RFID -lukijalle ohjelmoitua Java-ohjelmointirajapintaa ja ilmoittamaan luetut RFID-tunnisteet verkkopohjaiselle käyttöliittymälle. Sen pitää pystyä myös toistamaan videotiedostoja.

3.2 Verkkopohjainen käyttöliittymä ja tietokanta

Verkkopohjaisen käyttöliittymän toteutukseen valittiin tuen ja laajennettavuuden vuoksi HTML-, CSS-, JavaScript- ja PHP-tekniikat. Jotta käyttöliittymän käytettävyys olisi vielä parempi, projektiin otettiin mukaan Ajax-ohjelmointitekniikka.

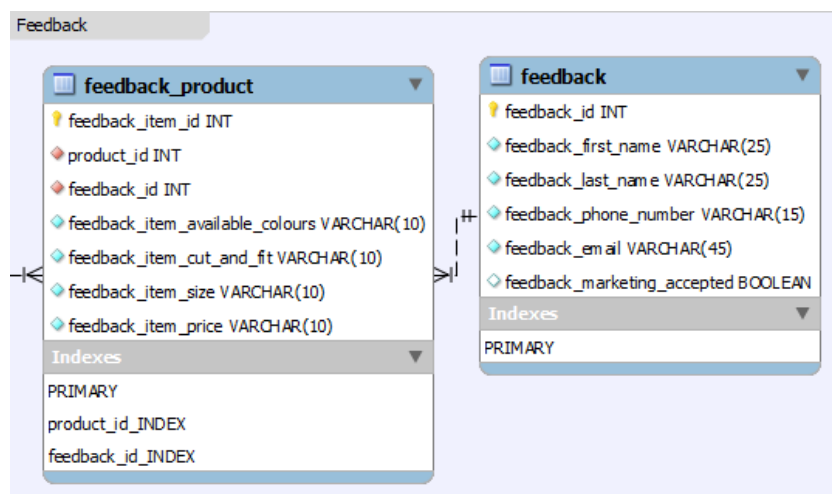
Verkkopohjaisen käyttöliittymän tulee pystyä keskustelemaan Android-pohjaisen verkkosovelluksen kanssa, käyttäen JavaScript-ohjelmointirajapintaa. Verkkosovelluksen ilmoittaessa RFID-tunnisteita käyttöliittymälle, tulee sen pystyä käsittelemään niitä. Tunnisteiden käsittelyä varten suunniteltiin PHP-luokka.

Projektiin piti myös suunnitella tietokanta, jonka toteuttamiseen valittiin MySQL-relaatiotietokannan hallintajärjestelmä. Kuvassa 5 on havainnollistettu projektin tuotteille suunniteltu tietokannan rakenne.



Kuva 5. Tuotteiden tietokantarakenne.

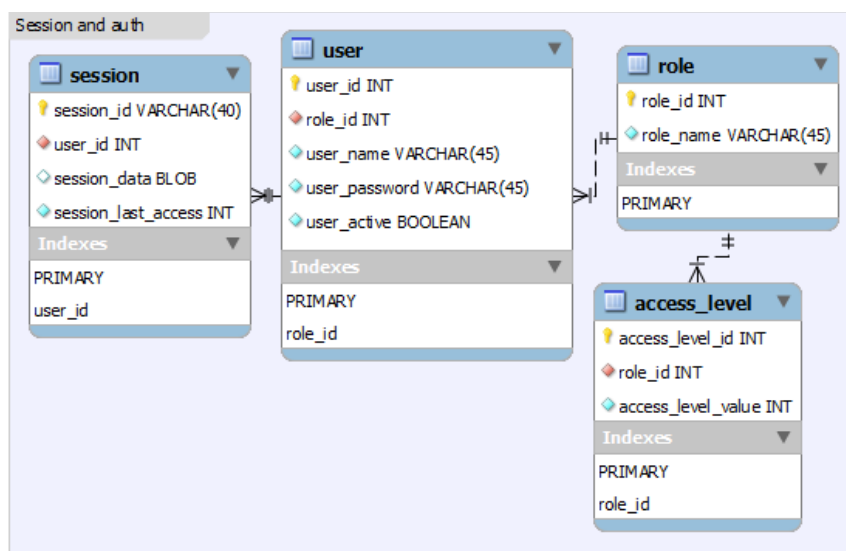
Projektin vaatimusmäärittelyjen mukaisesti haluttiin, että käyttöliittymässä on mahdollisuus antaa tuotepalautetta. Kuvassa 6 on esitetty tuotepalautteille suunniteltu tietokantarakenne.



Kuva 6. Tuotepalautteiden tietokantarakenne.

Käyttöliittymään suunniteltiin kirjautumisominaisuus, jonka avulla voidaan valvoa sitä, missä sovituskopissa kukin taulutietokone on. Kirjautumisominaisuutta voidaan hyödyntää myös esimerkiksi rajaamaan näytettävää sisältöä. Kyseinen ominaisuus suunniteltiin erilliseksi istunnon hallinnointijärjestelmäksi (Session Manager), PHP-luokkana.

Istunnon hallinnointijärjestelmä suunniteltiin käyttämään tietokantaa hyödyksi, joka koostuu käyttäjätunnuksesta, käyttöoikeuksista ja istunnon tunnisteesta. Järjestelmä mahdollistaa käyttäjäkohtaisen tiedontallennuksen istunnon ajaksi. Tämä helpottaa käyttöliittymän suunnittelua ja toteutusta. Kuvassa 7 on esitetty istunnon hallinnointijärjestelmälle suunniteltu tietokantarakenne.



Kuva 7. Istunnon hallinnointijärjestelmän tietokantarakenne.

4 KÄYTÄNTÖ JA TOTEUTUS

4.1 Android-verkkosovellus

Tämä luku käsittelee projektissa toteutetun Android-verkkosovelluksen kehittämisen kannalta tärkeimpiä kohtia ja lisäksi siinä havainnollistetaan joitakin kohtia käyttäen yksinkertaistettuja koodiesimerkkejä.

Android-alustat on nimetty koodinimillä ja niille on ilmoitettu erillinen ohjelmointirajapinnan taso (engl. API level), johon viitataan yleensä Android-tuotekehitysdokumenteissa. Taulukossa 1 on esitetty Android-alustojen koodinimet ja niille ilmoitetut ohjelmointirajapintojen tasot. [17]

Taulukko 1. Android-alustat.[17]

Alusta	Koodinimi	API-taso
Android 1.5	Cupcake	3
Android 1.6	Donut	4
Android 2.1	Eclair	7
Android 2.2	Froyo	8
Android 2.3 – Android 2.3.2	Gingerbread	9
Android 2.3.3 – Android 2.3.7		10
Android 3.0	Honeycomb	11
Android 3.1		12
Android 3.2		13
Android 4.0 – Android 4.0.2	Ice Cream Sandwich	14
Android 4.0.3		15

4.1.1 Manifest

Android-sovelluksen juurihakemistosta on löydyttävä `AndroidManifest.xml`-niminen tiedosto, jonka tarkoituksena on esitellä Android-järjestelmälle kaikki oleelliset tiedot sovelluksesta. Kyseisiä tietoja ovat esimerkiksi sovelluksessa käytettävät komponentit ja ehdot niiden käyttämiseen. [18]

Manifest-tiedostosta tulee ilmetä kaikki sovelluksessa käytettävät osat, joita ovat aktiviteetit, palvelut, yleisvastaanottajat (engl. broadcast receivers) ja sisällöntuottajat (engl. content providers). Ilman kyseisiä tietoja Android-järjestelmä ei pysty suorittamaan sovellusta. [18]

`AndroidManifest.xml`-tiedosto määrittää myös Android-sovellukselle:

- Yksilöllisen tunnisteiden, joka määräytyy sovelluksen pakkauksesta (engl. package).
- Esittelyn käyttöoikeuksille, jota sovellus tarvitsee käyttääkseen suojattuja ohjelmarajapintoja ja ollakseen vuorovaikutuksessa muiden sovellusten kanssa. Myös esittelyn käyttöoikeuksille, jotta muut sovellukset voivat käyttää kyseisen sovelluksen komponentteja.
- Listauksen työkalu-luokista (engl. Instrumentation classes), jotka tarjoavat profiloinnin ja muita tietoja ohjelman ollessa käynnissä. Kyseisiä esittelyjä saa löytyä vain, kun sovellus on kehitys- tai testausvaiheessa.
- Esittelyn vähittäistason Android-ohjelmointirajapinnalle.
- Luettelon ohjelmakirjastoista, joihin sovellus pitää linkittää.

Manifest-tiedosto rakennetaan elementeillä, joiden sisälle sallitaan vain toisia elementtejä. Kaikki tarvittavat arvot asetetaan elementtien ominaisuuksina (engl. attribute). Vain `<manifest>` ja `<application>` -elementit ovat pakollisia ja ne saavat esiintyä tiedostossa vain kerran. Kuvassa 8 on esitetty Manifest-tiedoston yleinen rakenne ja siinä sallitut elementit. [18]

```

<?xml version="1.0" encoding="utf-8"?>

<manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <supports-gl-texture />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>
        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>

        <service>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </service>

        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>

        <provider>
            <grant-uri-permission />
            <meta-data />
        </provider>

        <uses-library />

    </application>

</manifest>

```

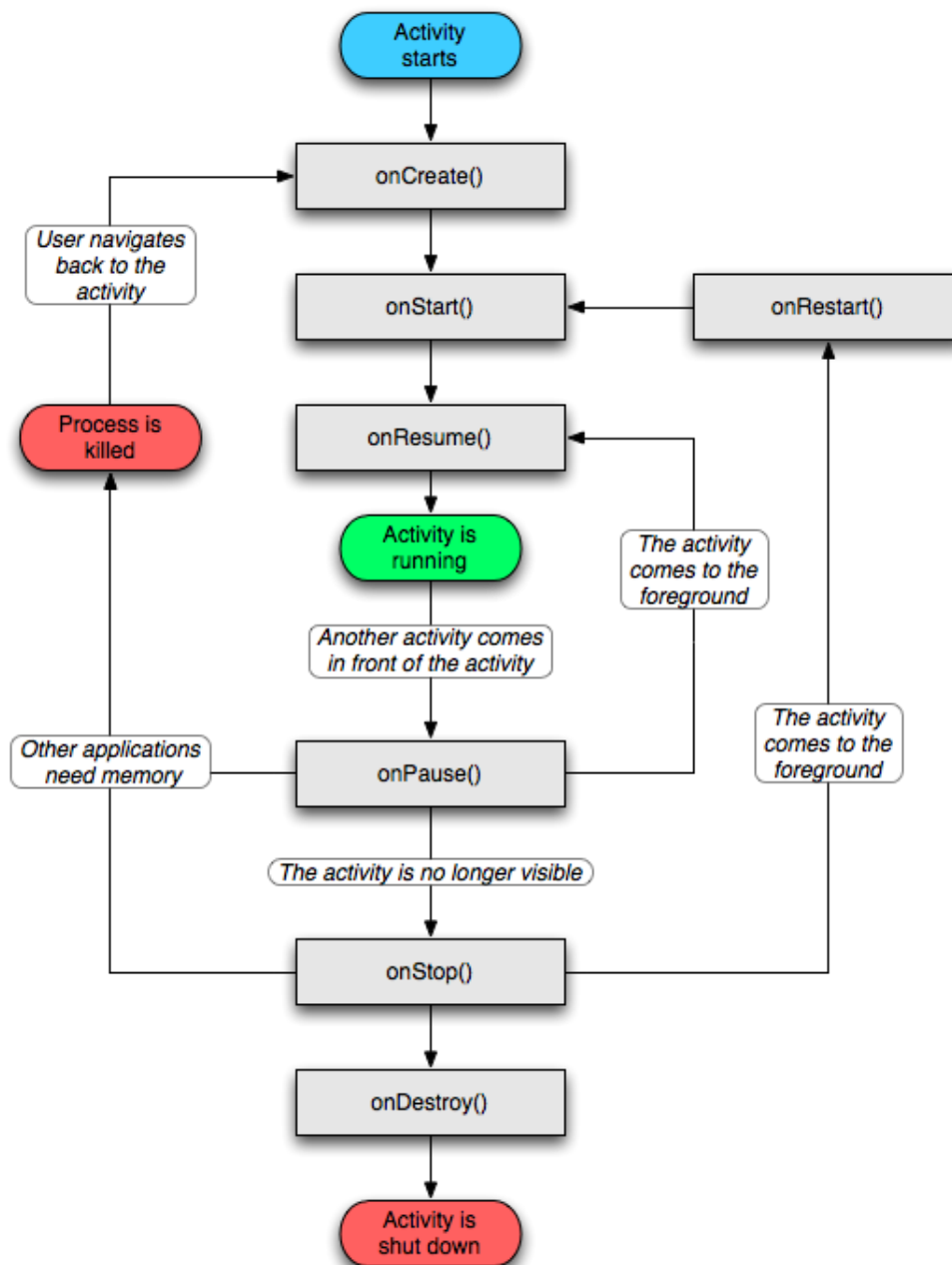
Kuva 8. *AndroidManifest.xml*-tiedoston rakenne. [18]

4.1.2 Activity

Aktiviteetti (engl. activity) on näkymä käyttäjälle, jonka kautta käyttäjä on vuorovaikutuksessa laitteiston kanssa. Android-sovelluksella voi olla useita aktiviteetteja. Aktiviteetin ominaisuudet peritään `android.app.Activity`-luokasta, josta ohjelmoitsija voi kehittää oman käyttöympäristön. [19, 20]

Oletuksena `Activity`-luokka huolehtii sovelluksen ikkunoinnista, mutta ohjelmoitsija voi halutessaan syrjäyttää (engl. override) sen ja hallinnoida sitä itse. Aktiviteetin elinkaari on esitetty kuvassa 9, josta selviää aktiviteetin perimät metodit. Kyseisiä metodeja kutsutaan seuraavan logiikan mukaisesti:

- `onCreate()`
 - Kutsutaan, kun aktiviteetti käynnistyy ensimmäisen kerran tai jos aktiviteetti on tapettu.
 - Kutsuu aina `onStart()`-metodikutsua.
- `onStart()`
 - Kutsutaan, kun aktiviteetti tulee käyttäjälle näkyväksi joko `onRestart()`- tai `onCreate()`-metodikutsun seurauksena.
 - Kutsuu `onResume()`-metodikutsua.
- `onResume()`
 - Kutsutaan, kun aktiviteetti aloittaa vuorovaikutuksen käyttäjän kanssa joko `onStart()`- tai `onPause()`-metodikutsun seurauksena.
 - Kutsuu aina `onPause()`-metodikutsua
- `onPause()`
 - Kutsutaan, kun aktiviteetti on menossa taustalle.
 - Kutsuu `onResume()`-metodikutsua, jos aktiviteettia ei ole tapettu. Muuten kutsuu `onCreate()`-metodikutsua.
- `onStop()`
 - Kutsutaan, kun aktiviteetti ei ole enää käyttäjälle näkyvässä.
 - Kutsuu `onRestart()`-metodikutsua, jos aktiviteettia ei ole tapettu. Muuten kutsuu `onCreate()`-metodikutsua.
- `onRestart()`
 - Kutsutaan `onStop()`-metodikutsun seurauksena, kun aktiviteetti palautetaan takaisin käyttäjälle näkyväksi.
 - Kutsuu aina `onStart()`-metodikutsua.
- `onDestroy()`
 - Kutsutaan, kun aktiviteettia ollaan tuhoamassa.



Kuva 9. Aktiviteetin elinkaari. [19]

Aktiviteettia käynnistettäessä ensimmäistä kertaa kutsutaan `onCreate()`-metodi-kutsua, ja kyseinen metodi tulee olla määritettynä aktiviteetissa. Metodissa tulee tehdä aktiviteetin koko elinkaareissa tarvittavat alustukset. Metodin tärkeimpänä tehtävänä on määrittää aktiviteetille näkymäasettelu (engl. layout) sekä keskustella muiden sovellusten kanssa. [19, 20]

4.1.3 Context

Konteksti (engl. context) on rajapinta sovellusympäristössä, joka tarjoaa yhteyden Android-järjestelmään. Android-järjestelmä tarjoaa rajapinnan abstraktille `android.content.Context`-luokalle, joka mahdollistaa pääsyn sovelluskohtaisiin resursseihin ja luokkiin sekä ohjelmistotason toimintoihin, kuten aktiviteettien käynnistämiseen tai selvittämään sovelluksen tai olion sen hetkinen tila. [21]

4.1.4 Intent

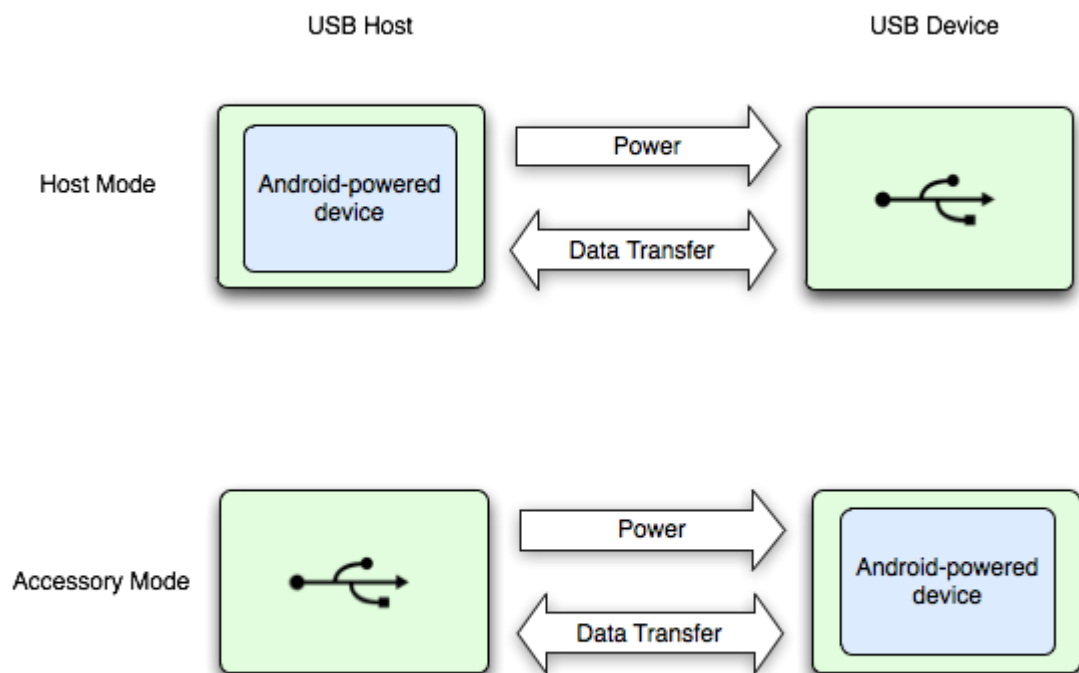
Intentit (engl. intent) ovat abstrakteja kuvauksia toiminnoista, joita Android-järjestelmä suorittaa. Kyseisten kuvausten avulla voidaan välittää tahdistamattomia viestejä aktiviteettien ja palveluiden välillä. Toisin sanoen intentti määrittelee aktiviteeteille ja palveluille suoritettavan toiminnon tai sillä saatetaan tarkentaa tiedolle miten toimia. [22]

Intentejä on kahden tyyppisiä: suora intentti (engl. explicit intent) ja epäsuora intentti (engl. implicit intent). Suoran intentin avulla voidaan nimetä tai aktivoida juuri tietty palvelu tai komponentti, esimerkiksi Java-luokka. Epäsuoralla intentillä voidaan pyytää Android-järjestelmältä tietyn tyyppistä palvelua tai komponenttia. Tällöin järjestelmä huolehtii kyseisen pyynnön tarpeista ja määrittää sille sopivan sovelluksen. Epäsuorissa intenteissa pitää antaa mahdollisimman tarkat kuvaukset siitä, millainen palvelu tai komponentti on parasta suorittaa juuri kyseiselle intentille. [22]

Android-järjestelmä saattaa kuitenkin epäsuorassa intentissä tarjota useamman palvelun tai komponentin, jolloin se näyttää käyttäjälle listan mahdollisista vaihtoehdoista, kuten esimerkiksi millä verkkoselaimella verkkosivu avataan. Epäsuora intentti vaatii myös intentti suodattimen (engl. intent filter), joka määritetään Androidin Manifest-tiedostoon. Intentti suodattimessa kuvataan, minkä tyyppisistä toiminnoista se vastaa. [22, 23]

4.1.5 USB

Android-järjestelmä tarjoaa `android.hardware.usb`-luokan, jonka avulla voidaan toteuttaa USB-laitteita hyödyntäviä sovelluksia. USB-sovelluksia voidaan kehittää joko isäntä- tai lisälaitetilojen avulla. Android-laitteen ollessa isäntätilassa se käyttäytyy USB-isäntänä ja antaa USB-väylälle virtaa. Android-laitteen ollessa lisälaitetilassa siihen kytketty lisälaitte toimii USB-isäntänä ja antaa USB-väylälle virtaa. Kuvasta 10 näkyy isäntä- ja lisälaitetilojen väliset erot. [24]



Kuva 10. Android USB-isäntä ja -lisälaitte tilojen väliset erot. [24]

Android-sovelluksen ja Nordic ID Sampo RFID -lukijan välinen USB-rajapinnan kommunikointi toteutettiin projektissa erillisenä Java-luokalla. Kyseisessä luokassa yhdistetään Nordic ID Sampo RFID -lukijalle toteutettu Java-pohjainen ohjelmointirajapinta ja Androidin tarjoama USB-isäntätilan ohjelmointirajapinta.

Projektissa tehtiin myös erillinen Java-luokka, joka hoitaa USB-laitteiden suodatuksen, rekisteröinnin järjestelmään ja yleisvastaanottimien kuuntelun.

4.1.6 WebView

WebView antaa mahdollisuuden luoda Androidille ikkunanäkymän, joka pystyy tulkitsemaan verkkosivuja. WebView-olio peritään Android-järjestelmän tarjoamasta `android.webkit.WebView`-luokasta, jota kehittämällä on mahdollista luoda jopa täysitoiminen verkkoselain. Luokka pohjautuu WebKit-selainmoottoriin. [25]

Koodiesimerkissä 1 on esiteltyä yksinkertainen WebView-ikkunanäkymästä, joka vaatii aktiviteetin sekä `AndroidManifest.xml`- ja näkymän asettelutiedoston muokkaamista.

Koodiesimerkki 1. *Android WebView-ikkunanäkymä.*

AndroidManifest.xml -tiedosto:

```
// Annetaan sovellukselle oikeudet käyttää Internettiä.
<uses-permission android:name="android.permission.INTERNET" />
```

Näkymän asettelutiedosto, `res/layout/main.xml`:

```
<WebView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
```

Aktiviteetti, `PersonalShopperActivity.class`:

```
// Esitellään WebView-olio aktiviteetin alussa.
WebView mWebView;

/* Viitataan onCreate()-metodissa aktiviteetin sisältöä kuvaavaan
näköasettelu-tiedostoon. Tässä tapauksessa main.xml. */
setContentView(R.layout.main);

/* Asetetaan onCreate()-metodissa WebView-olio näköasetteluun,
viitaten näköasettelu-tiedostossa määritettyyn "webview"
id-tunnisteeseen. */
mWebView = (WebView) findViewById(R.id.webview);

// Verkkosoite voidaan avata WebView-luokan loadUrl()-metodilla.
mWebView.loadUrl("http://www.google.com");
```

4.1.7 VideoView

VideoView antaa mahdollisuuden luoda Androidille ikkunanäkymän, joka pystyy toistamaan eri videotiedostoja. VideoView-olio peritään Android-järjestelmän tarjoamasta `android.widget.VideoView`-luokasta. Koodiesimerkissä 2 on esitetty projektissa toteutetun VideoView-ikkunanäkymän yksinkertaistettu koodiesimerkki. Tämä vaatii uuden näkymän asettelu- ja aktiviteettitiedoston luomista. [26]

Koodiesimerkki 2. *Android VideoView-ikkunanäkymä.*

Näkymän asettelutiedosto, `res/layout/video.xml`:

```
<VideoView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/VideoView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true">
</VideoView>
```

Aktiviteetti, `VideoPlayerActivity.class`:

```
// Esitellään VideoView-olio aktiviteetin alussa.
VideoView mViewVideoView;

/* Viitataan onCreate()-metodissa aktiviteetin sisältöä kuvaavaan
näkymän asettelutiedostoon. Tässä tapauksessa video.xml. */
setContentView(R.layout.video);

/* Asetetaan onCreate()-metodissa VideoView-olio näkymän
asetteluun */
mViewVideoView = (VideoView) findViewById(R.id.VideoView);

// Määritetään videotiedoston URL.
mViewVideoView.setVideoURI(Uri.parse("http://videopath.mp4"));

// Käynnistetään videotiedosto.
mViewVideoView.start();
```

Projektin vaatimusmäärittelyissä haluttiin, että Android-verkkosovellus pystyy toistamaan videotiedostoja. Sen toteuttamiseksi joudutaan muokkaamaan projektin Manifest- ja PersonalShopperActivity-tiedostoa (koodiesimerkki 3).

Koodiesimerkki 3. Videotiedostojen toistaminen Android-verkkosovelluksella.

AndroidManifest.xml -tiedosto:

```
// Mahdollistaa VideoView-aktiviteetin käynnistykseen
<activity android:name="com.google.project.VideoPlayerActivity">
    </activity>
```

Aktiviteetti, PersonalShopperActivity.class:

```
// Luodaan intentti VideoPlayerActivityyn
Intent myIntent = new Intent(PersonalShopperActivity.this,
VideoPlayerActivity.class);

// Käynnistetään VideoPlayerActivity halutussa kohdassa
PersonalShopperActivity.this.startActivity(myIntent);
```

4.1.8 JavaScript-rajapinta

Android WebView-olion `addJavascriptInterface()`-metodi mahdollistaa Java-luokan liittämisen haluttuun JavaScript-rajapintaan, jolloin luokan metodeja voidaan kutsua JavaScriptillä. `AddJavascriptInterface()`-metodille annetaan parametreina halutun Java-luokan ilmentymä (engl. instance) ja rajapinnan nimi. JavaScriptissä voidaan viitata ilmentymään käyttäen rajapinnan nimeä. [27]

Koodiesimerkissä 4 on esiteltyä aktiviteetin ja JavaScript-rajapinnan liittäminen toisiinsa, joka vaatii aktiviteetin muokkaamista. Tämän lisäksi tarvitaan Java-luokka, johon JavaScript-rajapinta liitetään. Projektissa luotiin kyseistä liitääntä varten `JSInterface`-luokka.

Koodiesimerkki 4. Javascript-rajapinnan toteutus.

Aktiviteetti, PersonalShopperActivity.class:

```
// JavaScript-rajapinnan liittäminen Java-rajapintaan
mWebView.addJavascriptInterface(new JSInterface(this), "Android");
// Asetetaan WebView-oliolle JavaScript päälle
mWebView.getSettings().setJavaScriptEnabled(true);
```

JSInterface.class:

```
import android.content.Context;
import android.widget.Toast;

public class JSInterface {
    private Context mContext;

    JSInterface(Context c) {
        mContext = c;

        public void showToast(String toast) {
            Toast.makeText(mContext, toast,
                Toast.LENGTH_SHORT).show();
        }
    }
}
```

Android-verkkosovelluksen JavaScriptInterface-luokkaan määriteltyjä metodeja voidaan kutsua JavaScriptillä. Esimerkiksi showToast()-metodia voidaan kutsua aktiviteetissa sidotulla "Android"-merkkijonolla:

```
Android.showToast('Hello Android!');
```

Seuraava funktio on määritelty verkkosivun JavaScript-tiedostossa:

```
/* Esimerkki JavaScript-funktiosta, joka näyttää halutun
elementin kahden sekunnin ajan tai määritetyllä ajalla. */
function showNotify(id,ms) {
    if(document.getElementById(id) != undefined) {
        var new_ms;
        (ms != undefined) ? new_ms = ms : new_ms = 2000;
        document.getElementById(id).style.display='block';

        setTimeout(function() {
            document.getElementById(id).
                style.display = 'none'; }, new_ms);
    }
}
```


Android-verkkosovelluksen kautta voidaan kutsua JavaScript-funktioita, jotka löytyvät avoimna olevalta verkkosivulta. Tähän käytetään WebView-luokan `loadUrl()`-metodia:

```
mWebview.loadUrl("javascript: showNotify('pagenotfound')");
```

4.2 Tietokanta

Projektin tietokanta jakaantuu tuotteiden, tuotepalautteiden ja istunnon hallinnointijärjestelmän tietokantarakenteisiin. Kyseisten tietokantarakenteiden EER-kuvat ovat esiteltyinä aiemmin luvussa 3.

4.2.1 Tuotetietokantarakenne

Tuotetietokantarakenteen tärkeimpänä tauluna on `product`-taulu, jonka tärkeimpänä kenttänä on 14-merkkinen varastointinimike, `product_sku`-kenttä. Kyseinen varastointinimike on ilmoitettu heksadesimaalina ja se on koodattu viidestä eri osasta:

- kuusi ensimmäistä merkkiä määrittelevät tuotteelle kokoelman nimen
- kaksi seuraavaa merkkiä määrittelevät tuotteelle linjan
- kaksi seuraavaa merkkiä määrittelevät tuotteelle kategorian
- kaksi seuraavaa merkkiä määrittelevät tuotteelle koon
- kaksi viimeistä merkkiä määrittelevät tuotteelle värin.

Varastointinimike-kentän lisäksi `product`-taulussa on määriteltynä tuotteelle `product_price` (hinta) ja `product_description` (selite) -kentät. Taulusta löytyy myös erilliset kentät varastointinimikkeeseen merkityille tunnisteille:

- `product_name_id`, kokoelman nimi

- product_line_id, linja
- product_category_id, kategoria
- product_size_id, koko
- product_color_id, väri.

Edellä mainitut tunnisteet ovat muutettuina 10-järjestelmän desimaaliluvuiksi ja niille jokaiselle löytyy oma taulunsa. Kyseisten taulujen pääavaimet vastaavat edellä mainittuja desimaalilukuja. Pääavaimen lisäksi tauluissa on varastointinimikkeen osaa kuvaava nimi-kenttä ja product_color-taulussa on lisäksi heksadesimaalinen värikoodi-kenttä.

4.2.2 Tuotepalautteen tietokantarakenne

Tuotepalautteen tietokantarakenne koostuu feedback- ja feedback_product-tauluista. Näistä ensimmäinen on tuotepalautteen antajan tietojen tallennusta varten ja jälkimmäinen on tuotekohtaista palautetta varten. Tuotekohtaista palautetta varten kerätään tuotetunniste, palautteen antajaan viittaava tunniste ja tuotetta koskevat arvioinnit. Tuote arvioidaan sen saatavuuden, leikkauksen ja sopivuuden, koon ja värin mukaan. Palautearviona näille voidaan antaa joko hyvä, neutraali tai huono arvosana.

4.2.3 Istunnon hallinnointijärjestelmän tietokantarakenne

Istunnon hallinnointijärjestelmän tietokantarakenne koostuu neljästä taulusta:

- Session
 - istuntoon viittaava ja satunnaisesti tuotettu merkkijono (session_id)
 - istunnon käyttäjä (user_id)
 - istunnosta tallennu tieto (session_data)
 - tieto siitä, milloin istunto on viimeksi ollut aktiivisena (session_last_access)
- User
 - käyttäjätunnus (user_name)
 - tieto siitä, onko käyttäjätunnus aktiivinen (user_active)
 - mikä on käyttäjän rooli (role_id)

- salasana (user_password)
- Role
 - roolin nimi (role_name)
 - oikeustaso (access_level)
- Access_level
 - oikeustason arvo (access_level_value)
 - roolin tunniste, jota oikeustaso koskee (role_id).

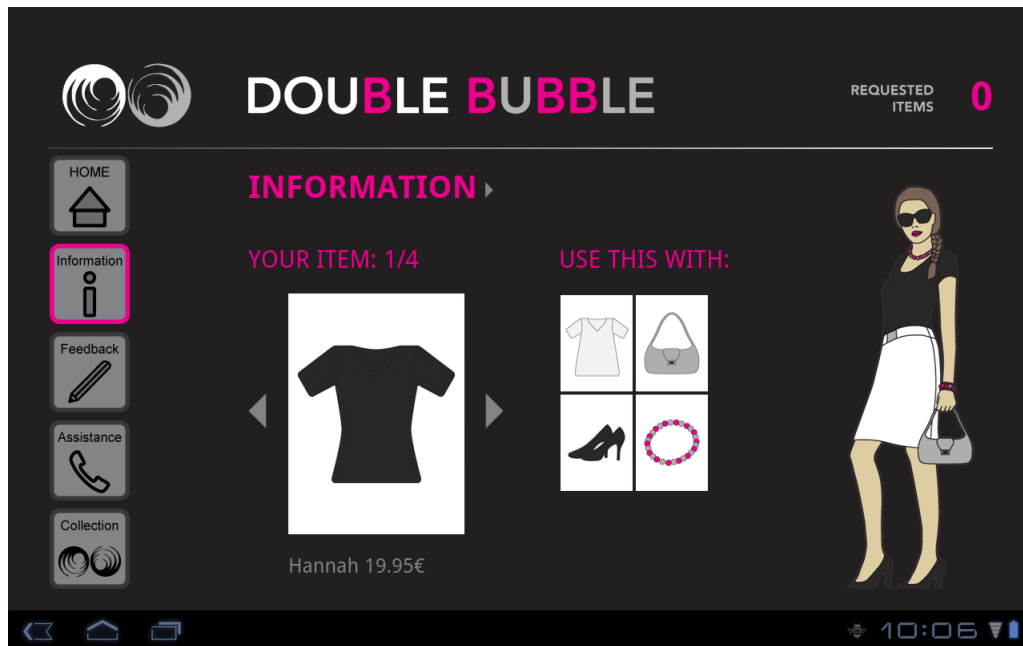
Edellä mainittujen taulujen pohjalta projektiin on luotu mahdollisuus sisään kirjautumiseen, tiedon tallennukseen istunnon ajaksi ja käyttöoikeustasojen määrittelyyn käyttäjäkohtaisesti.

4.3 Käyttöliittymä

Projektin käyttöliittymä jakaantuu Android-pohjaisen verkkosovelluksen käyttöliittymään ja verkkopohjaiseen käyttöliittymään. Käyttöliittymän ulkoasu suunniteltiin ja toteutettiin Nordic ID, ja tässä opinnäytetyössä esiintyvien käyttöliittymäkuvien tekijänoikeudet omistaa Nordic ID. Verkkosovelluksen käyttöliittymä on toteutettu Android-järjestelmän WebView-ikkunanäkymänä, josta on karsittu turhia ominaisuuksia pois. Kyseisiä ominaisuuksia ovat esimerkiksi otsikkopalkki, osoiterivi ja tekstin valinta.

Verkkopohjainen käyttöliittymä on toteutettu käyttäen HTML-, CSS-, PHP-, JavaScript- ja Ajax-tekniikoita. Käyttöliittymässä tarvittavat käyttäjä- ja tuotetiedot haetaan tietokannasta, joka toteutettiin projektia varten.

Kuvassa 11 on esimerkki tilanteesta, jossa Nordic ID Sampo RFID -lukija on löytänyt neljä RFID-tunnistetta ja asiakaskäyttöliittymän tuotetieto-sivu (information page) on esillä. Tuotetieto-sivun tarkoituksena on näyttää käyttöliittymään lisättyjen tuotteiden nimi, hinta, kuva ja neljä tuotetta, joita suositellaan käyttämään kyseisen tuotteen kanssa.



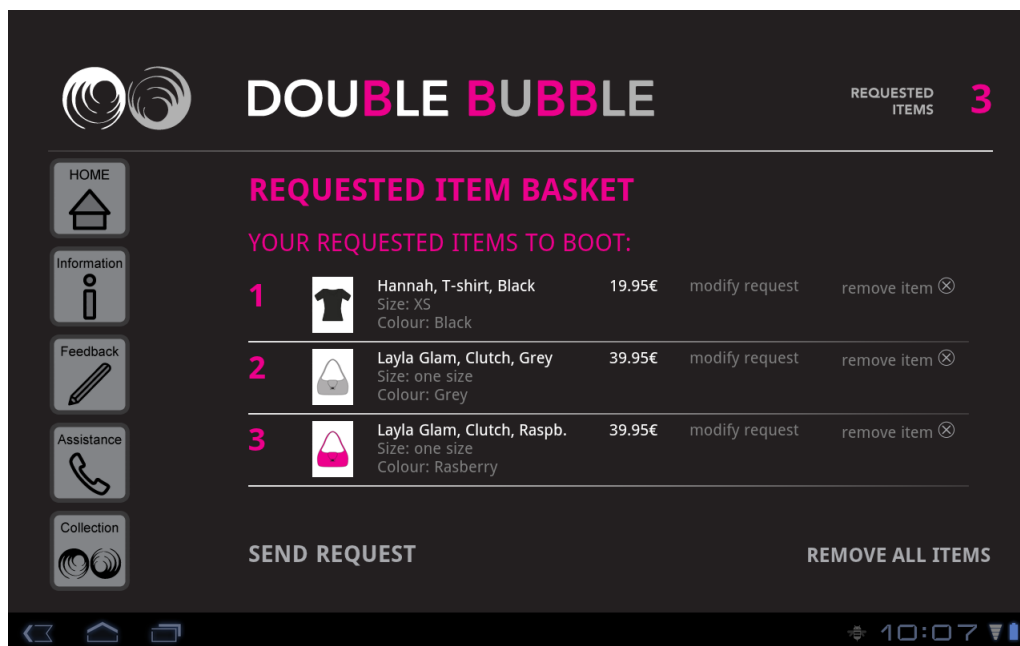
Kuva 11. Käyttöliittymän tuotetieto-sivu. (Copyright Nordic ID)

Käyttöliittymässä esiintyvien tuotekuvien napsautus ohjaa näkymän tuotesaatavuus-sivulle (information availability page), jossa tarkastellaan, löytyykö tuotteelle muita värejä tai kokoja. Kuvassa 12 on esimerkki tilanteesta, jossa tuotekuvaa on napsautettu.



Kuva 12. Käyttöliittymän tuotesaatavuus-sivu.

Käyttäjän on mahdollista napsauttaa tuotesaatavuus-sivulla (kuva 12) tarkasteltavan tuotteen väriä, joka esittää tuotteen erivärisenä. Käyttäjän on mahdollista napsauttaa myös tuotteen kokomerkintää, mikä lisää tuotteen tuotekoriin. Kuvassa 13 on esimerkki käyttöliittymän tuotekori-sivusta (requested item basket page), josta ilmenee että tuotekoriin on lisätty kolme tuotetta.



Kuva 13. Käyttöliittymän tuotekori-sivu.

Tuotekori-sivulle (kuva 13) päästään napsauttamalla käyttöliittymän oikeassa ylänurkassa sijaitsevaan "requested items"-tekstikenttään tai avustaja-sivun (assistance page) "request other styles"-tekstikenttään.

Projektin vaatimusmäärittelyissä haluttiin kuluttajanäkymä tuotetilaukseen, joka on käytännössä tuotekori-sivu (kuva 13). Tuotetilauksella tarkoitetaan tapahtumaa, jossa sovittaja pyytää myyjän tuomaan tuotekorissa olevat tuotteet sovituskoppiin. Tuotetilaustoimintoa ei projektissa toteutettu, sen aikataulun rajoittuneisuuden vuoksi.

Verkkopohjainen käyttöliittymä on jaettu otsikon, navigointivalikon ja sisällön mukaisesti. Käyttöliittymän koko sisältö on toteutettu div-elementein ja niille jokaiselle on määritelty yksilöivä id-ominaisuus, joita hyödynnetään PHP-, CSS- ja JavaScript-tiedostoissa. Koodiesimerkissä 5 on esitetty käyttöliittymän index.php-tiedosto yksinkertaistettuna.

Koodiesimerkki 5. Käyttöliittymän index.php-tiedoston rakenne.

```
...
<link href="default.css" rel="stylesheet" type="text/css">
...
<div id="wrapper">
    <div id="header"><?php include('header.php'); ?></div>
    <div id="header-separator"></div>
    <div id="nav"><?php include('nav.php'); ?></div>
    <div id="content"><?php include('content.php'); ?></div>
</div>
...
```

Verkkopohjaisessa käyttöliittymässä hyödynnetään myös Ajax-tekniikkaa, jotta vältytään turhilta sivunpäivityksiltä. Esimerkkinä navigointivalikon koti-nappula:

```
<li id="home" onclick="changePage('home');" ></li>
```

Napsauttaessa navigointivalikon nappulaa kutsutaan changePage()-JavaScript-funktiota, joka aluksi himmentää askeleittain content-elementin sisällön. Seuraavaksi funktio luo XMLHttpRequest-olion ja käskee sitä hakemaan funktiokutsussa pyydetyn sivun sisällön. Uuden sisällön saatuaan olio korvaa content-elementin vanhan sisällön uudella. Lopuksi funktio palauttaa content-elementin näkyviin askeleittain ja asettaa painetun nappulan aktiiviseksi.

5 YHTEENVETO

Tässä insinööriyössä suunniteltiin ja toteutettiin Nordic ID Sampo RFID -lukijaa hyödyntävä Android-verkkosovellus, verkkopohjainen käyttöliittymä ja tietokanta. Työssä saavutettiin haluttu tulos eli toimiva Android-pohjainen esittelysovellus, joka hyödyntää Nordic ID Sampo RFID -lukijaa.

Esittelysovelluksesta saadusta palautteesta voidaan tulla siihen johtopäätökseen, että projektin kaltaista tuotetta kaivataan markkinoille. Tämä signaali on positiivinen, sillä esittelysovellusta vastaavien tuotteiden toteutus kasvattaisi Nordic ID Sampo RFID -lukijan myyntiä sekä työllistäisi sovelluskehittäjiä ja sisällöntuottajia.

Työn vaatimusmäärittelyissä haluttua käyttöliittymän toteuttamista käsi-pääteelle ja määrittelyissä vaadittua ohjausnäkyvän toteuttamista tuotteiden poistamiseen, lisäämiseen ja muokkaamiseen ei ehditty tehdä aikataulusyistä. Myös vaatimusmäärittelyissä vaadittua toimintapalkin poistamista ei voitu toteuttaa, koska projektissa käytetyn Android-talutietokoneen käyttöjärjestelmän versio ei tukenut ominaisuuden poistamista.

Tässä projektissa opin uusina asioina Android- ja Ajax-sovelluskehitystä. Tämän lisäksi projekti oli mielestäni opettavainen, haasteellinen ja mielekäs, sillä siinä yhdisteltiin useita eri sovelluskehitystekniikoita keskenään.

LÄHTEET

- [1] "HyperText Markup Language", [Verkkodokumentti]. Saatavilla: <http://w3.org/MarkUp> (luettu: 12.12.2011)
- [2] "Apache HTTP Server". [Verkkodokumentti]. Saatavilla: <http://httpd.apache.org> (luettu: 12.1.2012)
- [3] "Cascading Style Sheets (CSS)". [Verkkodokumentti]. Saatavilla: <http://w3.org/Style/CSS> (luettu: 13.12.2011)
- [4] "PHP". [Verkkodokumentti]. Saatavilla: <http://fi2.php.net/manual/en/faq.php> (luettu: 15.12.2011)
- [5] Kolehmainen, K., "PHP & MySQL - Teoriasta käytäntöön", Readme.fi, 2006, 503 s.
- [6] "About JavaScript". [Verkkodokumentti]. Saatavilla : https://developer.mozilla.org/en/About_JavaScript (luettu: 5.1.2012)
- [7] "Standard ECMA-262 - ECMAScript Language Specification". [PDF-dokumentti]. Saatavilla: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf> (luettu: 5.1.2012)
- [8] "The Document Object Model (DOM) Introduction". [Verkkodokumentti]. Saatavilla: <http://w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/introduction.html> (luettu: 12.1.2012)
- [9] "Ajax: A New Approach to Web Applications". [Verkkodokumentti]. Saatavilla : <http://adaptivepath.com/ideas/ajax-new-approach-web-applications> (luettu: 5.1.2012)
- [10] Kosonen, P., Peltomäki, J. ja Silander, S., "Java 2 – Ohjelmoinnin peruskirja", WSOY, 2005, 642 s.
- [11] Date, C. J., "SQL and Relational Theory: How to Write Accurate SQL Code (2nd edition)", O'Reilly Media, Inc., 2011, 428 s.
- [12] "What is MySQL?". [Verkkodokumentti]. Saatavilla : <http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html> (luettu: 2.2.2012)
- [13] "phpMyAdmin". [Verkkodokumentti]. Saatavilla : <http://phpmyadmin.net> (luettu: 15.1.2012)
- [14] "What is Android?". [Verkkodokumentti]. Saatavilla : <http://developer.android.com/guide/basics/what-is-android.html> (luettu 2.2.2012)
- [15] "What is Eclipse and the Eclipse Foundation". [Verkkodokumentti]. Saatavilla : <http://www.eclipse.org/org/#about> (luettu 7.2.2012)
- [16] "Nordic ID Sampo". [Verkkodokumentti]. Saatavilla : <http://nordicid.com/en/products/nordic-id-sampo.html> (luettu 27.2.2012)
- [17] "Android platform versions". [Verkkodokumentti]. Saatavilla : <http://developer.android.com/resources/dashboard/platform-versions.html> (luettu 20.2.2012)

- [18] "Android Manifest". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/guide/topics/manifest/manifest-intro.html> (luettu 23.1.2012)
- [19] "Android Activity". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/reference/android/app/Activity.html> (luettu 20.2.2012)
- [20] "Android Activities". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/guide/topics/fundamentals/activities.html> (luettu 20.2.2012)
- [21] "Android Context". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/reference/android/content/Context.html> (luettu 23.1.2012)
- [22] "Android Intent". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/reference/android/content/Intent.html> (luettu 20.2.2012)
- [23] "Android Intent-filters". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/guide/topics/intents/intents-filters.html> (luettu 20.2.2012)
- [24] "Android USB". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/guide/topics/usb/index.html> (luettu 20.2.2012)
- [25] "Android WebView". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/reference/android/webkit/WebView.html> (luettu 17.2.2012)
- [26] "Android VideoView". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/reference/android/widget/VideoView.html> (luettu 20.2.2012)
- [27] "Android WebView". [Verkkodokumentti]. Saatavilla :
<http://developer.android.com/guide/webapps/webview.html> (luettu 17.2.2012)