

SATAKUNNAN AMMATTIKORKEAKOULU



Risto Vire  
2007

INTERNET TABLET RFID-LUKIJA

Tietotekniikka Rauma  
Tietotekniikan koulutusohjelma

## INTERNET TABLET RFID-LUKIJA

Vire, Risto

Satakunnan ammattikorkeakoulu

Tekniikka Rauma

Tietotekniikan koulutusohjelma

Yritys: Codebird Oy

Valvoja: Mika Impola

Marraskuu 2007

Ohjaaja: Mikko Javanainen

UDK: 621.39

Avainsanat: etätunnistus, mobiililaitteet, ohjelmistokehitys, Linux

Opinnäytetyön tavoitteena oli kehittää ohjelmisto Nokian Internet Tablet -päätelaitteelle, jolla voidaan esitellä Codebird Oy:n valmistaman RFID-lukijan ominaisuuksia ja käyttömahdollisuuksia. Ohjelmisto mahdollistaa mobiilin RFID-tunnisteiden lukemisen sekä tietojen tallentamisen tunnisteille käyttäen langatonta lähiverkkoa tiedon välittämiseen. Ohjelmiston tuli olla mahdollisimman helppokäyttöinen, jotta sen toiminnan ymmärtäminen olisi selkeää. Sen tulisi myös antaa havainnollinen esimerkki RFID-lukijan käyttömahdollisuuksista.

Koska Nokia Internet Tablet on Linux-pohjainen laite, käytettiin ohjelmiston kehityksessä myös Linux-käyttöjärjestelmää. Tämä mahdollisti sen, että ohjelmistoa voitiin testata suoraan pöytäkoneella, mikä nopeutti ohjelmiston kehittämistä huomattavasti.

Opinnäytetyö koostuu kahdesta osasta. Ensimmäinen osa on RFID-tunnisteiden lukuohjelma nimeltä MobDemo, joka vastaa tunnisteiden lukemisesta sekä tietojen esittämisestä käyttäjälle. Toinen osa on palvelinohjelma, jonka tehtävänä on tietojen hakeminen tietokannasta sekä tallentaminen tietokantaan.

Opinnäytetyön lopputuloksena valmistunut ohjelmisto pystyy vaatimusten mukaan lukemaan ja kirjoittamaan tietoa RFID-tunnisteelle sekä näyttämään tunnisteiden tiedot käyttäjälle. Ohjelmistoa voidaan käyttää RFID-lukijan toiminnan esittelemiseen, ja se antaa selkeän kuvan RFID-lukijan käyttömahdollisuuksista.

## INTERNET TABLET RFID READER

Vire, Risto

Satakunta University of Applied Sciences

School of Technology Rauma

Information Technology

Commissioned by: Codebird Oy

Supervisor: Mika Impola

November 2007

Tutor: Mikko Javanainen

UDC: 621.39

Keywords: RFID, mobile devices, software development, Linux

The purpose of this study was to develop software for the Nokia Internet Tablet. The software should demonstrate the different uses of the RFID reader, developed by Codebird Oy, in a simple and understandable form.

The programming was done in a Linux environment. Because the Nokia Internet Tablet is a Linux-based device, the software could be tested on a desktop machine instead of a handheld device, which made the testing of the software faster and more user friendly.

The software consists of two parts. The first part of the software is the reader software, called MobDemo, which is responsible for reading the RDIF tags and displaying the information to the user. The second part is the server software, which accesses the database for reading and storing information.

The software produced as a result of this study is able to read RFID tags, display tag information to the user and save information about the tags on the server using a wireless Internet connection.

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

LYHENTEET.....	6
1 JOHDANTO.....	7
2 RADIO FREQUENCY IDENTIFICATION.....	8
2.1 RFID-tekniikka.....	8
2.2 RFID-tunniste .....	8
2.3 RFID-lukija.....	9
2.4 Käyttökohteet.....	10
2.5 RFID-tekniikan ongelmat.....	10
3 NOKIA INTERNET TABLET.....	12
3.1 Ominaisuudet.....	12
3.2 Ohjelmisto.....	13
3.3 Ohjelmointiympäristö.....	13
3.3.1 Maemo.....	13
3.3.2 Hildon Application Framework.....	15
4 YLEISESTI PROJEKTISTA.....	16
4.1 Projektin tausta.....	16
4.2 Projektin esittely.....	16
4.3 Ohjelmiston toiminnallinen määrittely.....	16
5 MOBDEMO.....	18
5.1 Nokia Internet Tabletin ohjelmistokehitystyökalut.....	18
5.1.1 Eclipse.....	18
5.1.2 Scratchbox.....	18
5.1.3 Gazpacho.....	19
5.1.4 Gimp.....	20
5.2 Suunnittelu.....	20
5.2.1 Arkkitehtuuri.....	20
5.3 Moduulit ja niiden toteutus .....	21
5.3.1 Reader-moduuli.....	21
5.3.2 Käyttöliittymä.....	22

5.3.3 Internet-yhteys.....	25
5.4 Palvelin.....	27
5.4.1 Tiedonsiirto.....	29
5.4.2 MySQL-tietokanta .....	30
5.5 Ohjelmiston käyttö .....	31
6 TESTAUS.....	34
7 JAKOTKEHITYS.....	35
8 YHTEENVETO.....	37
LÄHTEET.....	38
KUVALUETTELO.....	40

## LYHENTEET

ARM	Advanced RISC Machine. 32-bittinen prosessoriarkkitehtuuri
Java	Sun Microsystemsin kehittämä ohjelmistoalusta
PNG	Portable Network Graphics. Häviötön grafiikan tallennusformaatti
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computer. Suunnittelufilosofia
ROM	Read-Only Memory
RSS	Really Simple Syndication. Verkkosyötemuoto
SDK	Software Development Kit. Ohjelmistokehityspaketti
SQL	Structured Query Language. standardoitu kyselykieli
UDP	User Datagram Protocol. Yhteyskäytäntö
UI	User Interface. Käyttöliittymä.
WI-FI	Wireless Fidelity. Yleinen WLAN-verkko
Widget	Kolmannen osapuolen tekemä lisäosa ohjelmaan
WLAN	Wireless Local Area Network, langaton lähiverkko
WORM	Write Once Read Many
XML	Extensible Markup Language. Merkintäkieli

## 1 JOHDANTO

Tämän insinööriyön tarkoituksena on suunnitella ja toteuttaa Codebird Oy:n valmistamalle RDIF (Radio Frequency Identification)-lukijalle demo-ohjelmisto, jolla lukijan ominaisuuksia voidaan esitellä graafisessa ympäristössä. Demo-ohjelman tarkoituksena on osoittaa ajurin ja lukijan toiminnallisuus ja käyttömahdollisuuksia.

RFID-tekniikka on moniin eri käyttötarkoituksiin soveltuva langaton tunnistustekniikka, jota on käytetty jo vuosia esimerkiksi karjan sekä tuotteiden seurantaan. RFID-tekniikka sopii erinomaisesti ihmisten, eläinten ja tuotteiden tunnistamiseen, koska RFID-tunnisteet ovat pienikokoisia ja halpoja. RFID-tunnisteet on myös helppo liittää tuotteisiin jälkikäteen, joten tuotteita ei tarvitse suunnitella uudestaan pelkästään RFID-tunnisteen lisäämiseksi.

Demo-ohjelman on tarkoitus toimia Nokian kehittämässä Internet Tablet -nimisessä laitteessa, joka on kannettava multimedialaite. Laitteen taustalla toimii Linuxin päälle rakennettu käyttöjärjestelmä, joten laitteelle voi vapaasti kirjoittaa myös omia ohjelmia.

Demo-ohjelman suunnittelu alkoi ohjelmiston toiminnallisen määrittelyn tekemisestä. Tämän jälkeen valittiin toteutukseen käytettävät teknologiat ja toteutustavat. Valinta perustui aikaisempien projektien pohjalta saatuihin kokemuksiin eikä lopullisen toteutustavan valinta ollut tästä johtuen kovin hankalaa. Ohjelmointiympäristöksi valittiin Eclipse sen avoimuuden ja laajennettavuuden vuoksi. Eclipseen saa kaikki tarpeelliset lisäosat, joita Nokian Internet Tabletille ohjelmoitaessa tarvitaan.

## 2 RADIO FREQUENCY IDENTIFICATION

### 2.1 RFID-tekniikka

Radio Frequency Identification, eli RFID, on tiedon etäluku- ja tallennusmenetelmä, joka käyttää radiotaajuisista tiedonsiirtoa. RFID:n kaksi päätekijää ovat lukija ja RFID-tunniste. Lukijan tehtävä on havaita tunniste ja lukea sen sisältö. Lukija välittää tiedon RFID-tunnisteen lukemisesta eteenpäin järjestelmälle, joka käsittelee tämän tiedon. RFID-tunniste on pieni laite, joka voidaan liittää tuotteeseen liimaamalla tai jo valmistusvaiheessa. RFID-tunniste sisältää antennin, jonka avulla tunniste voi lähettää ja vastaanottaa radiotaajuisia kyselyitä. RFID-tekniikalla on ollut kaupallisia sovelluksia jo 1980-luvun puolivälistä lähtien. Tällöin tekniikkaa käytettiin tietulleissa automaattiseen tunnistukseen. Tekniikkaa on käytetty jo pitkään myös lehmien tunnistuksessa maataloudella. Suomessa ensimmäiset kaupalliset sovellukset löytyivät teollisuuden kuljetinjärjestelmistä. (Wikipedia 2007e.)

### 2.2 RFID-tunniste

RFID-tunnisteet voivat olla joko aktiivisia, passiivisia tai puolipassiivisia. Passiivisilla RFID-tunnisteilla tarkoitetaan tunnisteita, joilla ei ole omaa virtalähdettä, vaan laitteen tarvitsema pieni sähkövirta induktoituu antenniin saapuvasta radiotaajuisesta signaalista. Luettaessa passiivinen RFID-tunniste palauttaa lyhyen vastauksen, usein tunnisten ID-numeron. Puolipassiivinen RFID-tunniste sisältää virtalähteen, mutta siinä ei ole omaa lähetintä. Virtalähteen ansiosta tunnisten toimintasäde on passiivista RFID-tunnistetta suurempi, ja siihen on mahdollista tallentaa tietoja tunnisten omaan muistiin (ROM, WORM). Aktiiviset RFID-tunnisteet sisältävät virtalähteen sekä oman lähettimen. Niillä on parempi kantomatkka kuin passiivisilla RFID-tunnisteilla, ja ne voivat tallentaa tietoa. Pienimmät aktiiviset RFID-tunnisteet ovat suunnilleen kolikon kokoisia, ja niiden lukuetaisyys on kymmeniä metrejä.

Vähäisen virrankäytön ansiosta paristokesto on aktiivisilla RFID-tunnisteilla vuosia,



Kuitenkin passiiviset RFID-tunnisteet ovat huomattavasti yleisempiä RFID-tunnisteita, koska passiivisten RFID-tunnisteiden hinta on muita huomattavasti halvempi.



Kuva 1. RFID-tunnisteita

RFID-tunnisteissa on yleisesti käytössä neljä eri radiotaajuutta, jotka ovat matala taajuus (125–134 kHz), korkea taajuus (13,56 MHz), UHF, eli Ultra High Frequency, (868–956 MHz) ja mikroaalto (2,45 GHz). Näistä ainoastaan mikroaaltotaajuus on käytettävissä kansainvälisesti, sillä se on vapaasti käytettävällä ISM (Industrial, Scientific and Medical) -taajuusalueella, jonka käyttö ei vaadi erillistä lupaa. Taajuusalue on yleensä käytössä kodin elektroniikassa ja tietokoneissa esimerkiksi Bluetooth- ja WLAN-laitteissa. (Wikipedia 2007e.)

### 2.3 RFID-lukija

Lukijan tehtävä on havaita ja lukea RFID-tunniste. Lukija toimittaa luetun RFID-tunnisteen tiedot järjestelmälle, joka käsittelee tiedon. Lukija voi olla aina päällä tai se voi olla toiminnassa vain haluttuna aikana. Esimerkiksi kulunvalvonnassa käytetyt lukijat ovat jatkuvasti lukuvalmiudessa. Mobiililaitteissa on yleensä käytössä lukija, joka voidaan kytkeä tarvittaessa päälle. Tämä säästää laitteen virtalähdettä ja antaa laitteelle pidemmän käyttöajan.

## 2.4 Käyttökohteet

Matalan taajuuden RFID-tunnisteita käytetään usein eläinten tunnistukseen, tuotteiden jäljittämiseen, autojen varkauden- ja käynnistykseenestoon, sekä kulunvalvontaan. Korkean taajuuden RFID-tunnisteita käytetään kirjastoissa ja kirjakaupoissa kirjojen jäljittämiseen, kuormalavojen, lentolaukkujen ja vaatteiden jäljittämiseen sekä rakennusten kulunvalvontaan. VHF (Very High Frequency, 350–433MHz) tunnisteita käytetään konttien ja arvotavaran seurantaan. Tulevaisuudessa on suunnitteilla asentaa kaikkiin autoihin VHF-taajuudella toimiva RFID-tunniste. VHF-taajuus mahdollistaa pitkän, jopa 100 metrin, lukuetaisyyden, joten sen avulla autojen tunnistus voi tapahtua pitkän etäisyyden päästä. UHF-taajuutta käytetään lähinnä logistiikkasovelluksiin, kuten kuormalavojen ja konttien jäljitykseen. Taajuutta käytetään myös ajoneuvojen ja perävaunujen jäljitykseen rajatuilla alueilla, kuten esimerkiksi satamissa. Mikroaaltotaajuutta käytetään ajoneuvojen pitkän matkan pääsynvalvontaan. (Wikipedia 2007e.)

## 2.5 RFID-tekniikan ongelmat

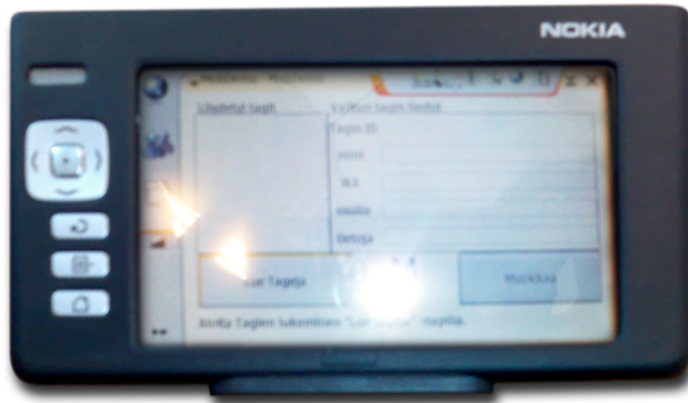
Suurin RFID-tekniikan ongelma on tunnisteiden luvaton seuraaminen. Koska RFID-tunnisteissa on hyvin vähän laskentatehoa, on tehokkaiden salausalgoritmien laskeminen käytännössä mahdotonta. RSA Security on patentoinut prototyyppilaitteen, jolla on mahdollista pysäyttää RFID-signaalit paikallisesti ja estää tunnisteiden lukeminen, jos näin halutaan.

RFID-tekniikka ei myöskään ole vapaa viruksista. Ars Technica raportoi maaliskuussa 2006 RFID puskuriylikuutiosta, joka voi vaikuttaa lentokenttien matkatavara- ja passitietokantaan. Tämän avulla on mahdollista saada luottamuksellista tietoa matkatavaroista tai passeista. Passeja yritetään tehdä turvallisemmaksi liittämällä niihin RFID-tunniste. Kuitenkin esimerkiksi Isossa-Britanniassa passien salaus onnistuttiin murtamaan alle 48 tunnissa. Sittemmin tutkijat ovat onnistuneet kloonaamaan passin tiedot kesken passin toimituksen sen omistajalle. Ennen rikollisten piti salaa avata ja sulkea kirjekuo-ri, jossa passi toimitetaan, mutta nykyään passin kloonaaminen onnistuu ilman kirjekuoren avausta, joka vähentää passisysteemin luotettavuutta. (Wikipedia 2007d.)

Washington Post uutisoi 8. syyskuuta 2007 1990-luvun puolivälissä tehdyistä laboratoriotutkimuksista, joissa RFID-implanttien oli todettu aiheuttavan syöpää laboratoriorotissa ja -hiirissä. Kasvaimet olivat useissa tapauksissa kehittyneet suoraan implanttien ympärille. Asiantuntijoiden mukaan tuloksia ei voi välttämättä soveltaa suoraan ihmisiin, mutta aihe vaatii lisätutkimuksia ennen kuin implanteja voidaan asentaa ihmisiin laajassa mitakaavassa. Yhdysvaltalaisen RFID-implanteja valmistavan VeriChipin mukaan implanteja on asennettu maailmanlaajuisesti noin 2000 ihmiseen. Koe-eläimiin implanteja on asennettu jo miljoonia 15 viime vuoden aikana, mutta yhtiön tiedossa ei ole yhtään niihin liittyvää syöpätapausta. (Lewan 2007.)

## 3 NOKIA INTERNET TABLET

### 3.1 Ominaisuudet



Kuva 2. Nokia Internet Tablet

Nokia 770 eli Internet Tablet on Nokian kehittämä Linux-pohjainen kannettava laite. Laitteen päätarkoituksena on antaa käyttäjälle mahdollisuus internetin käyttöön kaikkialla, missä on mahdollisuus ottaa yhteys internetiin käyttäen WI-FI-yhteyttä. Laitteen tärkein ominaisuus on sen suuri kosketusherkkä näyttö, jonka ansiosta internet-sivujen selaaminen on lähempänä tavallista pöytäkonetta kuin puhelimen näytöltä. Näytön resoluutio on 800 x 480 pikseliä ja se kykenee näyttämään 65 536 väriä. Näyttöä voi kontrolloida mukana tulevalla kynällä tai sormilla. Laitteen toiminta-aika on Nokian mukaan noin kolme tuntia ja valmiusaika noin seitsemän vuorokautta. Akun kestoon vaikuttaa enintään suuri näyttö, jonka tehonkulutusta voi kuitenkin hieman laskea säätämällä kirkkautta ja sallimalla näytön automaattinen sammutus. Nokia 770 osaa myös toistaa musiikki- ja videotiedostoja, minkä ansiosta laitetta voi käyttää MP3-soittimena tai kannettavana videotoistimena. Laitteen tukemat formaatit ovat yleisimmät internetistä löytyvät tiedostomuodot. Ääniformaateista MP3, WMA ja WAV sekä videoformaateista AVI, MPEG-1 ja MPEG-4. Laitteen mitat ovat 141 x 79 x 19 mm (pituus x leveys x syvyys), ja se painaa 230 grammaa. (Nokia 2007a.)

## 3.2 Ohjelmisto

Nokia 770:n mukana tulee kaikki yleiskäytössä tarvittavat ohjelmat, kuten nettiselain, joka pohjautuu suosittuun open source -selaimen nimeltä Opera, Internet-puhelu-ohjelmisto, Pikaviestin, Sähköposti, Internet-radio, RSS-lukija, median toistoon tarkoitettuja ohjelmia, tekstin- ja kuvankäsittelyyn tarkoitettuja ohjelmia sekä tiedostoselain. Laitteen mukana tulee myös ohjelmien hallintaohjelmisto, joka on toiminnaltaan monen muun Linux-jakelun kaltainen. Näiden lisäksi Nokia 770 sisältää monia pieniä ohjelmia, kuten kellon, laskimen ja pelejä.

Koska Nokia 770 on Linux-pohjainen laite, voi sille myös kirjoittaa itse ohjelmia käyttäen samoja työkaluja kuin normaalissa Linux-ohjelmoinnissa. Tämän ansioista Nokia 770:lle on saatavilla paljon käyttäjien itse kirjoittamia sovelluksia. Myös alunperin tavalliselle Linuxille kirjoitettuja ohjelmia on käännetty Nokia 770:tä varten. (Nokia 2007b.)

## 3.3 Ohjelmointiympäristö

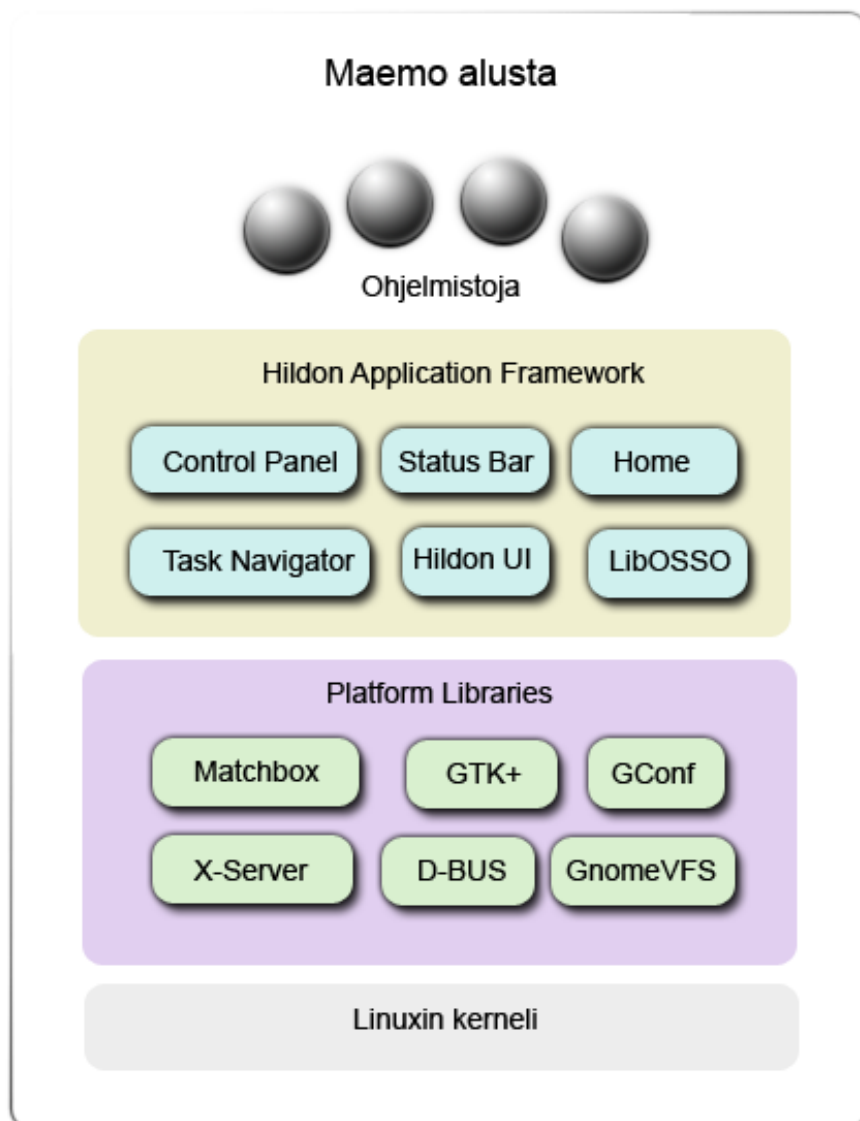
Kuten aiemmin jo mainittiin, on Nokia 770 Linux-pohjainen laite, joten ohjelmien kirjoittaminen on sille hyvin samantapaista kuin Linuxille yleensä. On kuitenkin otettava huomioon laitteen rajalliset resurssit ja kosketusnäytön tuomat rajoitteet. Linux-pohjaisuuden suurin etu on Linuxille kirjoitettujen kirjastojen suuri määrä, joita voi käyttää ohjelmoitaessa Nokia 770:lle.

### 3.3.1 Maemo

Maemo tarjoaa avoimeen lähdekoodiin perustuvan alustan ohjelmistojen suunnitteluun Nokia 770:lle. Se on rakennettu komponenteista, jotka ovat laajalti käytössä avoimissa työpöydissä ja mobiililaitteissa. Maemon tarkoituksena on olla avoin, saatavilla oleva ja käytännöllinen apu kaikille ohjelmistokehittäjille, jotka yrittävät saada kaiken irti mobiililaitteista. Maemo SDK (Software development kit) sisältää kaikki työkalut joita tarvitaan ohjelmistojen luontiin. (Maemo-yhteisö 2007a.)

Tällä hetkellä ei kuitenkaan ole olemassa kovinkaan montaa laitetta joita Maemo-alusta tukee. Nokian laitteista Maemon tukemia laitteita ovat Nokia 770 sekä sen seuraajaa N800.

Suurin hyöty Maemon käytössä on mahdollisuus testata ohjelmistoa tavallisella PC:llä, jolloin ohjelmiston ajamiseen kuluva aika pienenee ja kehitys on nopeampaa. Maemo helpottaa myös graafisten käyttöliittymien ohjelmointia ja mahdollistaa perinteisten Linux-ohjelmien kääntämisen kohdealustalle.



Kuva 3. Maemo-alustan rakenne

Kuten kuvasta 3 selviää, on Maemo-alusta jaettu välitason komponentteihin sekä käyttöliittymän komponentteja sisältävään kehikseen, jota kutsutaan Hildon Application Frameworkiksi.

Lähes kaikki välitason komponentit sisältävät avoimeen lähdekoodiin perustuvia komponentteja, kuten glibc, X-Server ja D-BUS. Jotkut komponenteista on muokattu normaaleista versioista, jotta ne vastaavat kannettavien laitteiden rajoituksia. (Maemo-yhteisö 2007b.)

### 3.3.2 Hildon Application Framework

Hildon Application Framework on osa Maemo-sovelluskehitysalustaa. Hildon Application Framework eli HAF perustuu GNOME:n perusteknologioihin kuten GTK+ widgets, joita on jatkettu ja laajennettu. Hildon UI (User Interface) koostuu GTK+:n standardiwidgeteistä, joita on laajennettu toimimaan paremmin mobiililaitteiden kanssa. Hildon widgetit sisältävät myös muutaman täysin uuden widgetin, joiden tarkoitus on palvella kannettavien laitteiden käyttäjiä.

Ohjelmointia helpottavia muita kirjastoja on mm. Glib, joka on tarkoitettu C-ohjelmoinnin helpottamiseen. Se sisältää tietotyyppejä, tietotyyppikonversioita, makroja, merkkijonoja, tiedostojen käsittelyominaisuuksia sekä muita hyödyllisiä ominaisuuksia. (Maemo-yhteisö 2007b.)

## 4 YLEISESTI PROJEKTISTA

### 4.1 Projektin tausta

Codebird Oy on vuonna 2002 perustettu langattomiin teknologioihin ja paikannusratkaisuihin keskittyvä yritys, jonka kehittämät sovellukset auttavat erityyppisiä organisaatioita esimerkiksi myynnin-, resurssien- ja tuotteidenhallinnassa. (Codebird Oy 2007.) Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa Codebird Oy:n valmistamalle RFID-lukijalle demo-ohjelmisto, jolla lukijan ominaisuuksia voidaan esitellä graafisessa ympäristössä.

### 4.2 Projektin esittely

Projektin tavoitteena on kehittää Nokia 770:lle ohjelmisto, jolla voidaan osoittaa ajurin ja lukijan toiminnallisuus sekä esitellä lukijan käyttömahdollisuuksia. Demo-ohjelmassa on graafinen käyttöliittymä, joka mahdollistaa RFID-tunnisteen lukemisen sekä tiedon muokkaamisen ja tallentamisen. Graafisen käyttöliittymän suunnittelussa on otettu huomioon Nokia 770:n fyysiset rajoitteet, kuten esimerkiksi kosketusnäytön ominaisuudet. Tämä tarkoittaa käytännössä sitä, että esimerkiksi kaikki ohjelman napit on pyritty tekemään mahdollisimman suurikokoisiksi ja selkeiksi, jotta ohjelman käyttäminen onnistuisi helposti sormeaa käyttäen, jolloin Nokia 770:n mukana tulevaa kynää ei tarvitsisi aina käyttää. Luetun tunnisteen tiedot sijaitsevat palvelimella MySQL-tietokannassa, josta projektia varten kehitetty palvelinohjelma osaa hakea kyseisen tunnisteen tiedot. Palvelinohjelma osaa myös muokata tietokannan tietoja käyttäjän näin pyytäessä. Demo-ohjelman nimeksi tuli MobDemo, joka tulee sanoista Mobiili RFID-lukija Demo.

### 4.3 Ohjelmiston toiminnallinen määrittely

Toiminnallisessa määrittelyssä on pyritty käymään läpi kaikki ohjelmiston näkymät sekä rajaamaan käyttötilanteita. Tämän tarkoituksena on luoda toteutukselle selkeät rajat



ja antaa kuva valmiin ohjelman ulkonäöllisistä ominaisuuksista. Nokia 770 aiheuttaa myös omia rajoituksia, jotka tulee ottaa huomioon.

Ongelmista mainittakoon kosketusherkän näytön tuomat rajoitukset sekä laitteen tehorojoitukset. Koska laitteen käyttäminen tapahtuu kosketusherkän näytön avulla, on tärkeää että ohjelmiston käyttö onnistuu mahdollisimman hyvin niin sormea kuin kynääkin käyttäen. Tärkeimpänä huomioitava asiana on nappien koko ja sijoittelu näytöllä, jotta käyttäjä voi sormella helposti valita haluamansa napin eikä vahingossa paina muita nappeja.

Pääideana ohjelmistossa oli, että kaikki toiminta tapahtuvat yhden käyttöliittymänäkymän ympärillä. Tunnisteen tietojen näyttäminen ja muokkaus tapahtuu siis samassa ikkunassa, joka selkeyttää ohjelmaa ja helpottaa ohjelman käytön oppimista, koska käyttäjän ei tarvitse opetella käyttämään useita eri näkymiä.

## 5 MOBDEMO

### 5.1 Nokia Internet Tabletin ohjelmistokehitystyökalut

Nokia Internet Tabletille on olemassa rajallisesti kehitystyökaluja, mutta ohjelmien kehittäminen on silti mahdollista graafisessa ympäristössä. Tässä luvussa käydään läpi projektin toteutukseen käytettyjä ohjelmistokehitystyökaluja.

#### 5.1.1 Eclipse

Eclipse on avoimen lähdekoodin ohjelmistokehitysympäristö, joka on pääasiassa kirjoitettu Javalla. Eclipse on alun perin Java-kehitysympäristö, joka koostuu Java Development toolsista ja Java-kääntäjästä. Käyttäjät voivat kuitenkin laajentaa Eclipsen käyttömahdollisuuksia asentamalla plugin laajennuksia, jotka ovat kirjoitettu Eclipse software frameworkille. Näillä laajennuksilla on mahdollista käyttää Eclipseä myös muiden kielten, kuten tässä tapauksessa C:n, kanssa.

Laajennuksista kaksi tälle projektille tärkeintä olivat CDT ja Laika. CDT mahdollistaa sovelluskehityksen C ja C++ kielillä. C ei ollut ainoa vaihtoehto, mutta koska se oli jo entuudestaan tuttu, oli se luonnollinen valinta. Toinen laajennus, Laika, on Scratchboxin graafinen käyttöliittymälaajennus Eclipselle, joka helpottaa Scratchboxin käyttöä huomattavasti sallimalla sen toimintojen kutsumisen suoraan Eclipsestä. (Wikipedia 2007a.)

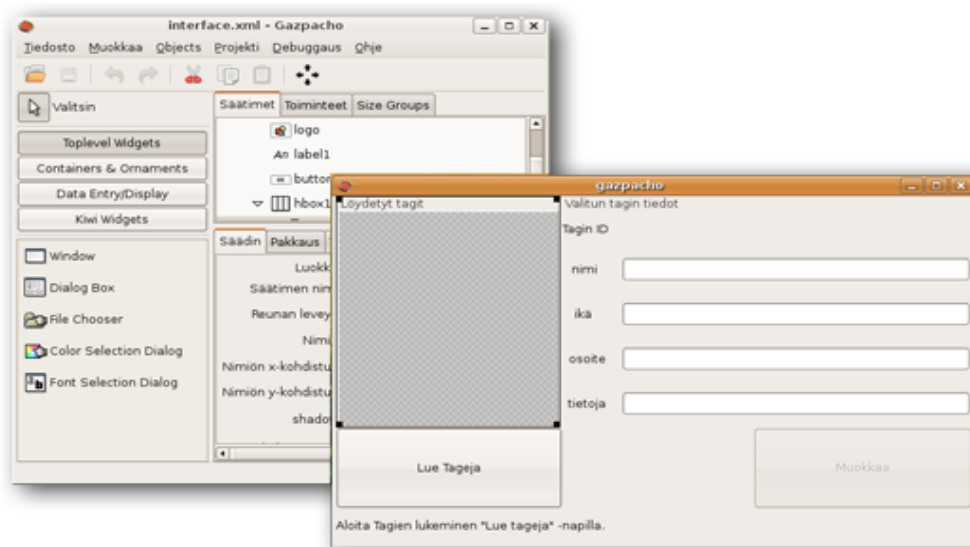
#### 5.1.2 Scratchbox

Scratchbox on Linux-järjestelmien kehitysympäristö, joka on alun perin kehitetty Suomessa. Scratchboxin tarjoaa sovellusten kehittäjille ympäristön, joka näyttää samalta ja toimii identtisesti lopullisen kohdealustan kanssa. Scratchbox on siis niin kutsuttu ristikäntäjä. Tietyille prosessorityypille kirjoitettu koodi ei toimi muissa prosessorityypeissä, jolloin tarvitaan ristikäntäjää kääntämään koodi toisen prosessorin ymmärtämään

muotoon. Nokia 770:n prosessori on ARM-prosessori, joten siinä ei toimi pöytäkoneella käännetyt ohjelmat, mutta Scratchboxin ansioista pöytäkoneella voidaan kääntää ohjelma ARM-arkkitehtuuriin sopivaksi, jolloin se toimii Nokia 770:ssa. Suurin hyöty tässä on se, että ohjelmaa voi testata pöytäkoneessa eikä sitä tarvitse aina siirtää mobiililaitteeseen, koska tämä vie jonkin verran aikaa. (Ojanperä 2007.)

### 5.1.3 Gazpacho

Gazpacho on suunnittelutyökalu, jolla voi kehittää käyttöliittymiä. Käyttöliittymät tallentuvat XML-muotoon, joten niiden tutkiminen ja muokkaus on helppoa ilman Gazpachoa. Ohjelmisto rakentaa käyttöliittymän XML-tiedoston perusteella ajon aikana. Tämä mahdollistaa käyttöliittymään tehtävät muutokset ilman ohjelman uudelleen kääntämistä, joka nopeuttaa huomattavasti pienten muutosten tekemistä. Gazpachon käyttöliittymä koostuu kahdesta ikkunasta (kuva 4): ohjelman päänäkymästä, josta käyttäjä voi valita haluamansa komponentit ja määrittellä niiden ominaisuuksia sekä ikkunasta, joka näyttää ohjelmalla luodun käyttöliittymän sellaisena kuin se näkyy itse ohjelmassa. (Gazpacho 2007.)



Kuva 4. Gazpachon käyttöliittymä

#### 5.1.4 Gimp

Gimp on avoimeen lähdekoodiin perustuva kuvankäsittelyohjelma. Ohjelma toimii useissa käyttöjärjestelmissä, kuten UNIXeissa, Windowsissa ja Mac OS X:ssä. Gimpia voi käyttää grafiikan ja valokuvien käsittelyyn. Yleisin käytötapa on grafiikan ja logojen luominen. MobDemon kaikki kuvat ja logot piirrettiin Gimpissä PNG-muotoon. PNG valittiin, jotta kuvan läpinäkyvyys säilyisi eikä kuvien ympärille jäisi reunoja, vaan ne sulautuisivat mahdollisimman hyvin käyttöliittymään. PNG on myös niin sanottu häviötön formaatti, joka tarkoittaa sitä, että kuvan tiedoista ei katoa mitään tallennuksen yhteydessä. Tästä on hyötyä varsinkin logoissa, koska kuvassa olevat yksityiskohdat pysyvät terävinä. (Wikipedia 2007b.)

### 5.2 Suunnittelu

Insinööriyön suunnittelu alkoi ohjelmiston toiminnallisen määrittelyn tekemisestä. Toiminnallisessa määrittelyssä päätettiin yhdessä Codebird Oy:n kanssa siitä, mitä ohjelman toimintaan kuului.

Kun toiminnallinen määrittely oli tehty, valittiin toteutukseen käytettävät teknologiat ja toteutustavat. Valinta perustui aikaisempien projektien pohjalta saatuihin kokemuksiin, eikä lopullisen toteutustavan valinta ollut tästä johtuen kovin hankalaa. Kun toteutustapa ja teknologiat oli päätetty, suunniteltiin käyttöliittymä, joka olisi mahdollisimman selkeä ja helppo käyttää ottaen huomioon laitteen tuomat rajoitukset. Käyttöliittymä saavutti lopullisen muotonsa kuitenkin vasta ohjelmiston valmistumisen myötä, koska joidenkin kenttien sijoittelua piti muuttaa käytössä huomattujen epäloogisuuksien vuoksi.

#### 5.2.1 Arkkitehtuuri

Jotta ohjelman koodi olisi mahdollisimman selkeää, on se jaettu useaan tiedostoon eli moduuliin, joista jokainen vastasi tiettyä ohjelman osa-aluetta. Osa-alueita ovat RFID-lukijan ajureita käyttävät funktiot, käyttöliittymän funktiot ja tiedonsiirrosta vastaavat

funktiot. Koodin jakaminen useaan tiedostoon auttaa myös ohjelman päivittämistä, koska niin kauan kun funktioiden nimet pysyvät samana, voidaan moduulit korvata uudella ilman että muihin moduuleihin tarvitsee koskea.

### 5.3 Moduulit ja niiden toteutus

Tarvittavat ohjelmistonkehitystyökalut olivat valmiiksi koneelle asennettuna, joten ohjelmiston toteutuksessa lähdettiin liikkeelle kaikkien työkalujen toiminnan testaamisesta. Kun kaikki työkalut oli saatu toimimaan, aloitettiin itse ohjelman tekeminen käyttöliittymän suunnittelusta. Käyttöliittymä suunniteltiin Gazpacho-ohjelman avulla mahdollisimman selkeäksi ja helpoksi käyttää. Kun käyttöliittymä oli saatu valmiiksi, aloitettiin itse ohjelman ohjelmointi. Käyttöliittymä tosin muuttui vielä monta kertaa ohjelmaa tehdessä.

#### 5.3.1 Reader-moduuli

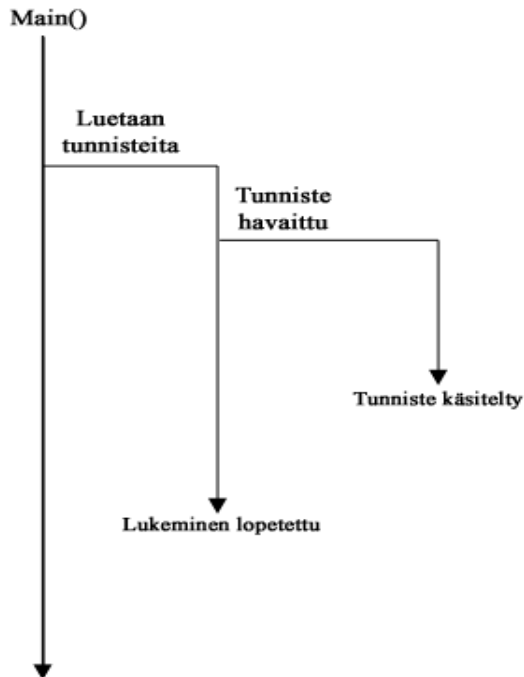
Reader-moduulin tarkoituksena on toimia rajapintana Codebird Oy:n RFID-lukijan ajureille. Moduuli sisältää funktiot lukijan alustamiselle, lukemisen aloittamiselle ja lopettamiselle sekä lukijan sammuttamiselle.

Moduulin tarkoituksena oli yksinkertaistaa RFID-lukijan ajurin käyttöä sekä mahdollistaa ajurin mahdollisesti vaihtuessa helpompi integraatio uuden ajurin kanssa. Moduuli mahdollistaa lukijan alustamisen, lukijan sammuttamisen, tunnisteen lukemisen aloittamisen sekä lukemisen lopettamisen.

Koodin lukemisen helpottamiseksi lukijan ajurin funktioiden kutsuminen tehdään omassa moduulissa. Ajurin funktiot vievät jonkin verran koodirivejä, ja niiden sijoittaminen muun koodin sekaan tuntui turhalta.

Aina, kun RFID-lukijalla halutaan lukea tunnisteita, on lukija ensin alustettava. Alustuksessa ohjelmisto määrittelee lukijan käyttäytymisen, kun lukija lukee RFID-tunnuksen. Tunnuksen lukeminen aloittaa oman säikeen, jossa lukija pyörii kunnes lukeminen py-

säytetään. Säikeessä pyöriessään lukija etsii jatkuvasti uusia tunnuksia, joiden ilmaantua lukuetaisyydelle lukija lukee ne ja lisää tunnuksen tiedot listaan, joka käydään läpi kun lukeminen päättyy.



Kuva 5. Säikeiden muodostuminen ohjelmassa

Kuvasta 5 selviää, kuinka ohjelma luo säikeitä ajon aikana. Kun RFID-lukija käynnistetään, se luo säikeen, jossa tunnisteen lukeminen tapahtuu. Aina kun lukija havaitsee tunnisteen, se luo uuden säikeen, jossa tunniste käsitellään. Kun tunniste on käsitelty, tuhoutuu tämä säie. Lukijan säie tuhoutuu, kun lukeminen lopetetaan.

### 5.3.2 Käyttöliittymä

Käyttöliittymän näkymät luodaan ohjelman suorituksen aikana XML-tiedostosta, joka luotiin Gazpacho-nimisellä käyttöliittymien suunnittelutyökalulla. Käyttöliittymän laatamiseen käytetään libglade-kirjastoa. Libglade-kirjasto muodostaa XML-tiedostosta widgettejä, jotka käyttöliittymä ottaa käyttöönsä.

Gazpachossa voidaan määritellä painikkeille kuuntelijafunktiot, joita ohjelma kutsuu,

kun kyseistä painiketta painetaan. Kuuntelijafunktion nimi voisi olla esimerkiksi btnLue(). Kun käyttäjä painaa "lue"-painiketta, ohjelmisto kutsuu btnLue() funktiota, jossa napin painaminen sitten käsitellään.

Käyttöliittymä-moduulissa on toiminnallisuuksia, joiden tarkoitus on piirtää käyttöliittymä ja sen eri osat sekä vastata käyttöliittymän kutsuihin. Käyttöliittymä-moduuli sisältää kolme alimoduulia. Nämä moduulit ovat callbacks-, list- ja interface-moduulit.

Callbacks-moduuli sisältää funktiot käyttöliittymän kutsujen käsittelyyn. Kutsuja aiheutuu nappien painamisesta. Käyttöliittymässä on vain kaksi nappia, joten callbacks-moduulin tehtävänä on lähinnä näiden kahden napin tarkkailu.

Koodiesimerkissä 1 on kuvattuna btnLue\_clicked-funktion toiminta. Funktiota kutsutaan, kun "lue tageja"-painiketta on painettu. Funktion tehtävänä on joko aktivoida tai sammuttaa RFID-lukija sekä huolehtia siitä, että painikkeiden tekstit vastaavat niiden tehtävää.

```
void btnLue_clicked( GtkAction * action, gpointer data ) {

    /* määritellään painikkeet */
    GtkWidget *gtk_button_lue;
    GtkWidget *gtk_button_muokkaa;

    gtk_button_lue = interface_libglade_get_ui_widget("btnLue");
    gtk_button_muokkaa = interface_libglade_get_ui_widget("btnMuokkaa");

    /* jos ei lueta tageja */
    if(is_reading == 0) {
        is_reading=1;
        /* lukiessa ei voi muokata tageja, joten muokkaa nappi disabloidaan */
        gtk_widget_set_sensitive(gtk_button_muokkaa, FALSE);
        /* Kerrotaan ohjelman alakulmassa lukemisen aloittamisesta */
        interface_set_label_text("label1", "Tagien lukeminen aloitettu...");
        /* Muutetaan lukunapin teksti vastaamaan sen uutta toimintoa */
        gtk_button_set_label (gtk_button_lue, "Lopeta lukeminen");
        /* Käynnistetään RFID-lukija */
        reader_start_reading_tags();
    }

    /* jos luetaan tageja */
    else {
        is_reading=0;
        /* Tehdään samat asiat kuin edellisessä vaihtoehdossa, mutta lopetetaan lukeminen */
        gtk_widget_set_sensitive(gtk_button_muokkaa, TRUE);
        interface_set_label_text("label1", "Tagien lukeminen lopetettu.");
        gtk_button_set_label (gtk_button_lue, "Lue tageja");
        reader_stop_reading_tags();
    }
}
```

```

}
/* nukutaan puoli sekuntia, jottei tulisi vahingossa tuplaklikkauksia */
usleep(500000);
}

```

Koodiesimerkki 1.

Ohjelmassa luetut tunnisteet ilmestyvät listaan, jonka funktioista vastaa list-moduuli. List-moduulin tehtävänä on lisätä haluttu tieto listaan, listata valitun rivin tiedot sekä tarkkailla, milloin valittu rivi vaihtuu.

Interface-moduuli sisältää käyttöliittymän piirtämiseen tarvittavat funktiot. Moduuli sisältää funktiot käyttöliittymän alustamiseen, ikkunoiden luomiseen ja tuhoamiseen, tietyn widgetin hakemiseen ja tekstikenttien muokkaamiseen.

Koodiesimerkissä 2 on funktio käyttöliittymän alustamiselle. Ohjelmisto kutsuu funktiota käynnistyksen yhteydessä, jolloin funktio alustaa käyttöliittymän luo tarvittavat näkymät.

```

void interface_initialize_ui( AppData * pAD ) {
    /* Tarkistetaan että pAD on olemassa */
    g_assert(pAD != NULL);

    /* Ladataan käyttöliittymä */
    if ( ! interface_libglade_load_ui(LIBGLADE_LOAD_AUTOCONNECT_ON) )
    {
        g_warning("Libglade failed to load UI.");
        g_assert(NULL);
    }

    /* määritellään uusi application */
    pAD->app = HILDON_APP ( hildon_app_new ( ) );
    /* Tarkastetaan että application saatiin luotua */
    g_assert(pAD->app != NULL);

    /* Asetetaan ohjelman otsikko */
    hildon_app_set_title ( pAD->app, _( APPLICATION_TITLE ) );
    hildon_app_set_two_part_title ( pAD->app, TRUE );

    /* Liitetään ohjelma pääikkunaan */
    pAD->main_view = HILDON_APPVIEW(hildon_appview_new( _( APPLICATION_TITLE)));
    g_assert(pAD->main_view != NULL);
    hildon_app_set_appview( pAD->app, pAD->main_view );

    /* luodaan päänäkymä ja valikko */
    interface_main_view_create( pAD );
    interface_main_view_create_menu( pAD );
}

```

Koodiesimerkki 2.



Interface-moduuli sisältää myös asetustiedoston, jossa on määritelty palvelimen osoite ja portti, sekä käyttöliittymän piirtämiseen tarvittavan XML-tiedoston, joka on luotu Gazpachossa.

Koodiesimerkissä 3 on ote XML-tiedostosta. Esimerkissä on painikkeen "btnLue" tiedot, joita ohjelmisto käyttää, kun se luo käyttöliittymän XML-tiedostosta. Esimerkissä asetetaan painikkeelle kuuntelija "btnLue\_clicked", jonka koodiesimerkki esiteltiin aiemmin.

```
<child>
  <widget class="GtkButten" id="btnLue">
    <property name="height_request">70</property>
    <property name="is_focus">True</property>
    <property name="label" context="yes" translatable="yes">Lue Tageja</property>
    <property name="visible">True</property>
    <signal handler="btnLue_clicked" name="clicked"/>
  </widget>
  <packing>
    <property name="bottom_attach">2</property>
    <property name="top_attach">1</property>
  </packing>
</child>
```

Koodiesimerkki 3.

### 5.3.3 Internet-yhteys

Nokia 770 on mahdollista yhdistää internetiin monella eri tavalla. Tässä työssä internet-yhteys muodostetaan langattoman lähiverkon kautta. Yhteys täytyy muodostaa ennen ohjelman käynnistämistä, koska ohjelma ei tässä versiossa osaa yhdistää laitetta verkkoon automaattisesti. Tiedonsiirto toteutettiin yksinkertaisesti käyttämällä UDP-protokollaa tiedon siirtämiseen palvelimen ja ohjelman välillä. Net-moduulin tehtävänä on hoitaa yhteyden alustus, jonka jälkeen moduuli osaa kysyä palvelimelta RFID-tunnuksen ID:tä vastaavat tiedot ja näyttää ne käyttöliittymässä. Moduuli osaa myös tallentaa tunnisten tiedot palvelimelle.

Koodiesimerkissä 4 on kuvattu net\_put-funktion toiminta. Funktion tehtävänä on lähettää palvelimelle tunnisten tiedot tallennettavaksi tietokantaan. Funktiossa käytetään socket.h:n sendto ja recvfrom -funktioita tiedon lähettämiseen ja vastaanottamiseen.

```

void net_put(viesti tiedot) {

    /* avataan yhteys palvelimeen */
    net_init();

    /* Asetetaan viesti */
    set_message(tiedot);

    /* Lähetetään viesti palvelimelle */
    status = sendto( sockettitunniste,
                    message,
                    strlen(message),
                    0,
                    (struct sockaddr*) &kohde_addr,
                    sizeof(struct sockaddr_in) );

    /* Odotetaan palautetta palvelimelta */
    datankoko = recvfrom(sockettitunniste,
                        &bufferi[0],
                        100,
                        0,
                        (struct sockaddr *) &hostin_osoite ,
                        &pituus);

    /* lisätään loppumerkki bufferin perään */
    bufferi[datankoko] = '\0';

    /* Suljetaan yhteys */
    close(sockettitunniste);
}

```

Koodiesimerkki 4.

Koodiesimerkissä 5 on kuvattu, kuinka UDP-yhteys palvelimeen luodaan ja kuinka tiedonsiirto palvelimelle tapahtuu. UDP-yhteyden luontiin käytetään socket.h:n socket-funktiota. Socket-funktion parametrit määrittelevät yhteyden. AF\_INET tarkoittaa IPv4 internet protokollaa ja SOCK\_DGRAM UDP:tä. Tätä socketia käytetään koodiesimerkissä 6 tiedon lähettämiseen palvelimelle.

```

struct sockaddr_in kohde_addr;

void net_init() {

    /* määritellään socket */
    sockettitunniste = socket(AF_INET,SOCK_DGRAM,0);

    /* Tallennetaan kohteen portti ja osoite */
    kohde_addr.sin_family = AF_INET;
    kohde_addr.sin_port = htons(portti);
    kohde_addr.sin_addr.s_addr = inet_addr(osoite);
}

```

Koodiesimerkki 5.

```

/* Esimerkki tiedon lähettämisestä palvelimelle */
status = sendto(
    socketitunniste,           // käytettävä socketti
    message,                  // viesti
    strlen(message),          // viestin pituus
    0,                        // liput
    (struct sockaddr*) &kohde_addr, // kohteen tiedot
    sizeof(struct sockaddr_in) // osoiterakenteen koko tavuina
);

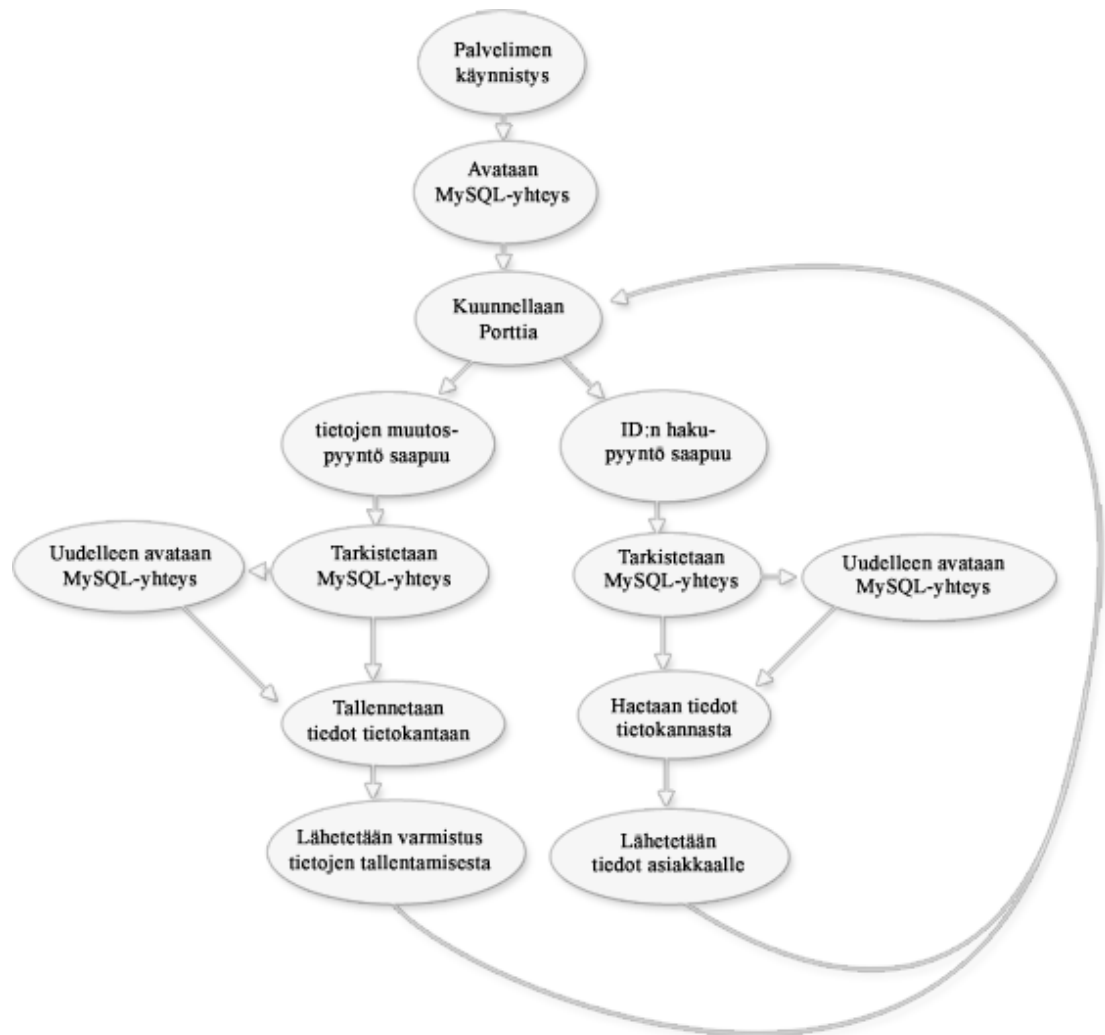
```

Koodiesimerkki 6.

## 5.4 Palvelin

Jotta ModDemo voi hakea tietoja RFID-tunnisteista, oli kehitettävä palvelin, jonne tunnistaiden tiedot tallennetaan tietokantaan. Palvelinympäristöksi valittiin Linux-palvelin, jossa on asennettuna MySQL-tietokanta sekä tuki PHP-kielelle. Palvelin-toiminnot toteutettiin käyttäen PHP:tä, koska PHP sisältää valmiiksi hyvät funktiot MySQL-tietokannan käsittelyyn sekä socket-ohjelmointiin.

Palvelin luo käynnistyessään yhteyden MySQL-tietokantaan sekä alkaa kuunnella ennalta määriteltyä porttia. Kun asiakasohjelma ottaa yhteyden palvelimeen, osaa palvelin vastata pyyntöihin eli hakea tietoja tietokannasta sekä kirjoittaa tietoja tietokantaan (kuva 6).



Kuva 6. Palvelimen toiminta

Mikäli MySQL-yhteyttä ei käytetä vähään aikaan, katkaisee MySQL siihen muodostetut yhteydet. Koska palvelimen on tarkoitus olla jatkuvasti päällä, on sen tarkistettava MySQL-yhteyden olemassaolo ennen tiedon hakua. Mikäli yhteys tietokantaan on katkenut, osaa palvelin yhdistää uudelleen tietokantaan ja vasta tämän jälkeen noutaa tiedot.

Koodiesimerkissä 7 on palvelimen getId-funktio. Funktio tarkistaa MySQL-yhteyden olemassaolon, jonka jälkeen funktio hakee id:n perustella rivin tietokannasta ja palauttaa sen.

```

function getId($id) {
    /* Tarkastetaan MySQL-yhteyden tila */
    /* Jos MySQL-yhteys ei vastaa pingiin yhteys on katkennut */
    if (!mysql_ping ($this->mysli)) {
        /* Suljetaan vanha yhteys varmuuden vuoksi */
    }
}
  
```

```

mysql_close($this->mysli);
/* Avataan uusi yhteys */
$this->mysli = mysql_connect(HOST,USER,PASS);
mysql_select_db(DB,$this->mysli);
}
/* nyt yhteys on varmasti kunnossa, joten voidaan kysyä tietoja */
$id=mysql_real_escape_string($id); /* koitetaan suojella tietokantaa injektioilta */

/* kysytään tiedot ID:n perusteella ja palautetaan rivi */
$query=mysql_query("select * from mobData where id = '$id'");
return mysql_fetch_row($query);
}

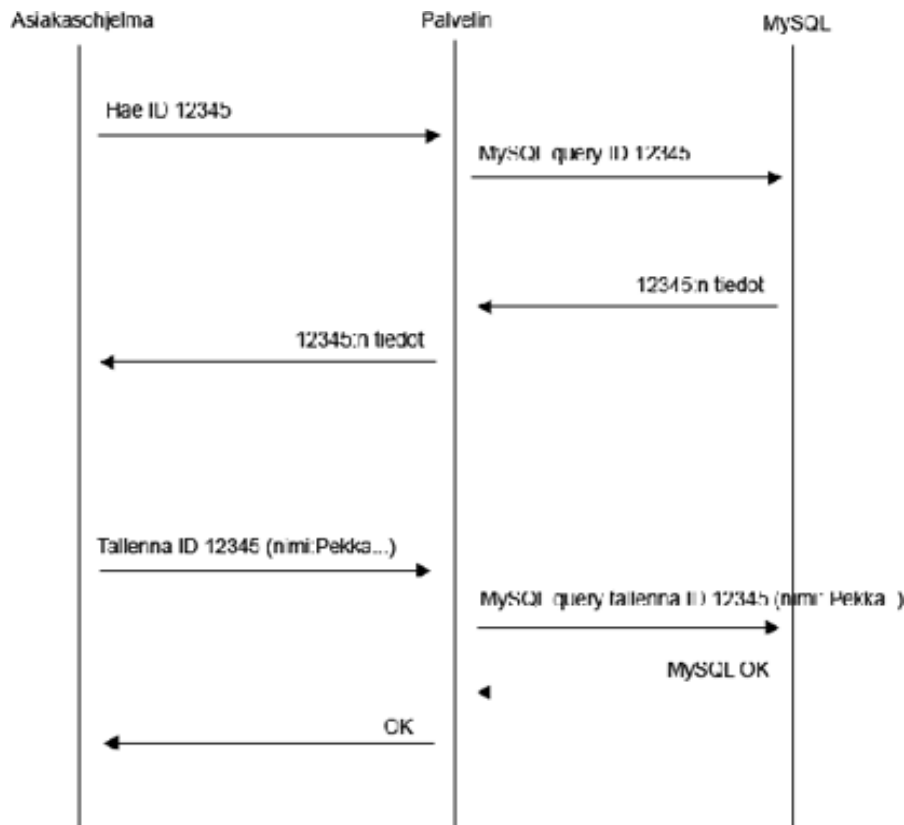
```

Koodiesimerkki 7.

#### 5.4.1 Tiedonsiirto

Tiedonsiirto asiakkaan ja palvelimen välillä tapahtuu käyttäen UDP-protokollaa. Tämä mahdollistaa miltei yhtäaikaiset kyselyt monelta laitteelta samaan porttiin, jolloin palvelin voi palvella useita laitteita samanaikaisesti.

Tieto siirtyy palvelimen ja asiakkaan välillä viestinä, jossa tiedot on eroteltu käyttäen pilkkua. Ratkaisuna tämä ei ole kaikkein paras, mutta yksinkertaisuutensa puolesta se oli ohjelmiston valmistumisaikataulu huomioon ottaen paras ratkaisu. Ongelmana tässä lähestymistavassa on sen vaikea muokattavuus. Mikäli halutaan lähettää eriävä määrä tietoja, joudutaan ohjelman koodia muokkaamaan. Parempi ratkaisu olisi käyttää XML-standardiin perustuvaa tiedonsiirtoa. Tällöin olisi mahdollista muuttaa viestin rakennetta ilman, että koodiin täytyy tehdä suuria muutoksia.



Kuva 7. Tiedonsiirto ohjelmiston ja MySQL-palvelimen välillä

Kuvassa 7 käyttäjä lukee tunnisteen, jonka ID on 12345, ja ohjelmisto lähettää palvelimelle kyselyn kyseisestä tunnisteesta. Palvelin hakee tunnisteen tiedot MySQL-tietokannasta ID:n perusteella ja palauttaa tiedot ohjelmistolle. Seuraavaksi käyttäjä muokkaa tunnuksen tietoja ja kun käyttäjä painaa "Tallenna"-painiketta, lähettää ohjelma viestin palvelimelle, jossa ohjelma pyytää palvelinta tallentamaan äsken muokatut tiedot. Palvelin tallentaa tiedot MySQL-tietokantaan ja palauttaa OK viestin ohjelmalle.

#### 5.4.2 MySQL-tietokanta

MySQL on SQL-tietokannan hallintajärjestelmä, joka on saatavilla vapaalla GNU GPL (General Public Licence) -lisenssillä tai kaupallisella lisenssillä, mikäli GPL on liian rajoittunut.

MySQL:n hallinta tapahtuu tekstipohjaisella asiakasohjelmalla, mutta sille on myös saatavilla graafisia hallintaohjelmia, kuten esimerkiksi MySQL Administrator, MySQL

Query Browser sekä phpMyAdmin. MySQL:n suosiosta kertoo se, että monet suurimmat internetsivustot pyörivät MySQL-tietokannan päällä. Tällaisia sivuja ovat muun muassa Wikipedia, Google ja Yahoo. (Wikipedia 2007c.)

MySQL-tietokanta valittiin projektiin koska sen käyttö oli ennestään jo tuttu sekä siksi, että PHP:stä, joka valittiin palvelimen ohjelmointikieleksi, löytyy suoraan tuki MySQL-tietokannoille.

MySQL tietokannan rakenne on yksinkertainen, koska se sisältää vain yhden taulun. Taulussa on kentät RFID-tunnisteen ID:lle, nimelle, iälle, osoitteelle sekä ylimääräinen data-kenttä, johon käyttäjä voi syöttää haluamansa lisätiedot.


id	nimi	ika	osoite	data
00E004010001A1D384	Pekka	12	Jokutie 5	N/A
B46DB486	Jaakko	37	Katukuja 2	Myyntipäällikkö

Kuva 8. Otos MySQL-tietokannasta

Tietokannan yksinkertaisuudesta johtuen ohjelmisto osaa tallentaa vain nämä viisi asiaa jokaisesta RFID-tunnisteesta. Ohjelman kehittyessä eteenpäin tämä tietokantarakenne ei enää tule toimimaan, mikäli tunnisteista halutaan tallentaa enemmän tietoa.

## 5.5 Ohjelmiston käyttö

Koska ohjelmisto on perimmiltään suhteellisen yksinkertainen, on ohjelmiston käytöstä pyritty tekemään mahdollisimman yksikertaista rajaamalla ikkunoiden määrä yhteen, jolloin kaikki ohjelman mahdolliset toiminnot ovat nähtävissä samasta ikkunasta jatkuvasti. Tämä helpottaa ohjelmiston käytön oppimista, koska käyttäjä tottuu nopeasti käyttöliittymään eikä joudu siirtymään useiden eri näkymien välillä.

Löydetyt tagit 00E004010001A1D384 B46DB486	Valitun tagin tiedot Tagin ID 00E004010001A1D384 Nimi <input type="text" value="Pekka"/> Ikä <input type="text" value="12"/> Osoite <input type="text" value="Jokutie 5"/> Tietoja <input type="text" value="N/A"/>
<input type="button" value="Lue tageja"/>	 <input type="button" value="Muokkaa"/>

Aloita tagien lukeminen painamalla "Lue tageja"-nappia.

Kuva 9. MobDemo-ohjelman käyttöliittymä

Ohjelmiston käyttö aloitetaan painamalla "Lue tageja" -painiketta, joka käynnistää RFID-lukijan ja aloittaa tunnisteen lukemisen. "Lue tageja" -painikkeen muuttuu "Lopeta lukeminen" -painikkeeksi, ja ohjelma ilmoittaa lukijan aktivoinnista ohjelman alalaidassa olevassa tekstikentässä sekä piirtämällä M-logon päälle vihreän ympyrän. Kun lukeminen on aloitettu, voidaan lukijalla lukea RFID-tunnisteita, jotka luettaessa ilmestyvät ohjelmistossa olevaan listaan. Listassa lukee tunnisteen ID-numero. Ohjelmisto ilmoittaa onnistuneesta tunnisteen lukemisesta myös vilauttamalla punaista ympyrää M-logon päällä.

Kun kaikki tunnistet on luettu, painetaan "Lopeta lukeminen" -painiketta, joka sammuttaa RFID-lukijan ja muuttaa M-logon takaisin alkuperäisen näköiseksi sekä muuttaa "Lopeta lukeminen" -painikkeen takaisin "Lue tageja" -painikkeeksi. Tämän jälkeen käyttäjä voi valita haluamansa tunnisteen ID:n perusteella listasta. Ohjelmisto hakee valitun tunnisteen tiedot palvelimelta ja näyttää ne oikealla olevissa kentissä, joita voi muokata valitsemalla "Muokkaa"-painike ohjelman oikeasta alakulmasta. "Muokkaa"-painike muuttuu "Tallenna"-painikkeeksi, ja ohjelmisto aktivoi tekstikentät, joita voi tämän jälkeen muokata. Kun tunnisteen tiedot on muokattu halutulla tavalla, painetaan "Tallenna"-painiketta, joka lähettää tunnisteen tiedot serverille sekä muuttaa "Tallenna"-painikkeen takaisin "Muokkaa"-painikkeeksi.

Ohjelmisto estää virheellisten valintojen tekemisen ottamalla pois käytöstä sellaiset pai-



nikkeet, joita ei voi kyseisellä hetkellä käyttää. Tämä tekee ohjelmiston käytöstä huomattavasti helpompaa ja varmempaa.

## 6 TESTAUS

Testauksella pyrittiin todistamaan MobDemo-ohjelmiston toiminta sekä varmistamaan ohjelmiston yhteensopivuus RFID-lukijan ajurin kanssa. Testiympäristönä toimivat Satakunnan ammattikorkeakoulun Tekniikka Rauman tilat sekä tilasta löytyvä WLAN-verkko. Palvelimena toimi toisaalle sijoitettu Linux-palvelin, jossa oli MySQL-tietokanta sekä tuki PHP-ohjelmointikielelle. Testin alussa tietokannassa ei ollut tallennettuna yhtään tunnistetta, vaan kaikkien tunnisteiden tiedot luotiin MobDemo-ohjelmistolla. Yhteys Nokia 770:n ja serverin välille muodostettiin käyttäen WLAN-verkkoa.

Ensimmäinen testi tehtiin lukemalla yksittäinen tunniste, jonka tietoja ei löydy tietokannasta. Kun saatiin varmistus siitä, että tietoja ei löydy tietokannasta ja ohjelmisto näyttää kaikkien kenttien olevan tyhjiä, kirjoitettiin kenttiin tietoja ja tallennettiin ne tietokantaan. Tämän jälkeen tunniste luettiin uudestaan ja varmistettiin, että äsken tallennetut tiedot löytyivät palvelimelta.

Toinen testi tehtiin lukemalla useita tunnisteita. Tunnisteiden joukossa oli myös ensimmäisessä testissä käytetty tunniste, joka voitiin havaita kun ohjelmisto haki tunnisteiden tiedot onnistuneesti. Kaikille luetuille tunnisteille kirjoitettiin tietoa ja tämän jälkeen ohjelmisto suljettiin kokonaan. Myös palvelin käynnistettiin tässä vaiheessa uudestaan. Tämän jälkeen ohjelmisto ja palvelin käynnistettiin uudestaan ja suoritettiin kolmas testi.

Kolmas testi oli hyvin samantapainen kuin toinen. Testissä luettiin useita tunnisteita ja varmistettiin, että äsken tallennetut tiedot löytyivät yhä palvelimelta ja että ne siirtyivät oikein tunnisteita luettaessa.

Testeistä selvisi, että ohjelmisto toimii vaatimusten mukaan, eli sillä voi lukea tunnisteiden tiedot sekä tallentaa tunnisteelle tietoja.

## 7 JAKOTKEHITYS

Ohjelmiston jatkokehityksessä tulisi panostaa ohjelmiston laajennettavuuteen sekä muokattavuuteen. Tärkein muutos olisi XML-muotoinen tiedonsiirto, joka sallisi muuttaa siirrettävän tiedon määrää. Tosin tämä vaatisi myös tietokannan muuttamisen niin, että se osaisi tallentaa uudet kentät. JSON (JavaScript Object Notation) voisi olla myös toinen vaihtoehto XML-muotoiselle tiedonsiirrolle. JSON on merkintäkieli, joka on nimes- tään huolimatta alustariippumaton. JSON:lle löytyy kirjastot niin C:lle kuin PHP:llekin.

Toinen jatkokehityksessä huomioitava asia on internet-yhteyden automaattinen muodostaminen, joka olisi myös hyvä saada toimimaan koska tällä hetkellä ohjelmisto ei sitä osaa tehdä. Tämä helpottaisi ohjelmiston käyttöä huomattavasti, sillä internet-yhteyden muodostaminen manuaalisesti unohtuu helposti ennen ohjelman käynnistämistä, joka aiheuttaa ohjelman kaatumisen ajon aikana.

Käyttöliittymän ohjelmointikielenä Python olisi ollut myös hyvä vaihtoehto C:lle, koska Pythonille löytyy huomattavasti enemmän esimerkkejä ja apua internetistä kuin C:lle, mutta koska Python ei ollut ennestään tuttu, jäi se harkinnan asteelle.

Tietokantapuolella kehitystä vaatii itse tietokannan rakenne. Tällä hetkellä tietokanta ei osaa tallentaa kuin ennalta määrätyt neljä asiaa RFID-tunnisteesta, jotka ovat nimi, ikä, osoite sekä tiedot. Tähän ratkaisuna voisi olla tietojen tallentaminen tietokantaan XML-muotoisena, jolloin tiedon määrää ei ole rajattu mitenkään muuten kuin tietokannan kyseisen solun koolla. Toinen vaihtoehto voisi olla uuden, otsikkokentät sisältävän taulun luominen. Tässä tapauksessa täytyisi luoda myös nämä kaksi taulua yhdistävä taulu, jonka tehtävänä olisi tallentaa itse tieto sekä yhdistää tieto tauluihin.

<b>MobData</b>	<b>MobOtsikot</b>	<b>MobOtsikko_to_MobData</b>
<u>MobID</u> MuokkausPvm	<u>Otsikkoid</u> Otsikko	<u>OtsikkoId</u> <u>MobId</u> Data

Kuva 10. Mahdollinen parannus MySQL-tietokantaan

Tämä mahdollistaisi sen, että tietokantaan voitaisiin tallentaa millä tahansa otsikolla olevaa tietoa ja kuinka paljon tietoa tahansa kutakin tunnistetta kohden. Tiedon hakeminen tietokannasta suoraan MySQL:n omilla työkaluilla onnistuisi myös tällöin otsikon tai tiedon mukaan, toisin kuin pelkästään XML-muotoon tallennetussa tiedossa.

## 8 YHTEENVETO

Projektin tavoitteena oli toteuttaa Codebird Oy:n RFID-lukijaa käyttävä ohjelmisto, jolla voidaan esitellä lukijan toimintaa sekä käyttötapoja. Ohjelmiston tuli toimia Nokia Internet Tablet -laitteella, joka mahdollistaa mobiilin RFID-tunnisteiden lukemisen.

Ohjelmiston toteuttamisessa haastavin osuus oli ohjelmointiympäristön käytön opiskelu, johon kului suurin osa projektiin käytetystä ajasta. Itse ohjelma on suhteellisen yksinkertainen, mutta koska kyseessä oli uusi ohjelmointiympäristö, oli työ haastavaa ja mielenkiintoista. Uusia asioita opittavaksi riitti lähes jokaisella osa-alueella aina käyttöliittymän luomisesta internetin kautta tapahtuvaan tiedonsiirtoon asti.

Eclipse-ohjelmointiympäristöstä oli paljon hyötyä ohjelmaa kehitettäessä, koska se sisälsi hyviä pohjia, joiden päälle ohjelma rakennettiin. Eclipse tarjosi Laika-lisäosan myötä myös graafisen käyttöliittymän Scrathbox-työkaluihin, jotka ilman Eclipseä toimivat tekstipohjaisesti konsolista. Käyttöliittymän rakentaminen XML-muotoon Gazpachossa oli myös huomattavasti helpompaa, kuin käyttöliittymän ohjelmointi käsin suoraan koodin sekaan.

Projektissa onnistuttiin toteuttamaan toimiva RFID-lukija, jonka toiminta testauksessa vastasi ohjelmistolle asetettuja odotuksia.

## LÄHTEET

Codebird Oy:n WWW-sivu [Verkkodokumentti]. [Viittaus 12.9.2007]. Saatavissa: <http://www.codebird.com/yritys.php>

Gazpacho 2007 [Verkkodokumentti]. [Viittaus 20.8.2007]. Saatavissa: <http://gazpacho.sicem.biz/>

Maemo-Yhteisö 2007a. Maemo introduction [verkkodokumentti]. [Viittaus 20.8.2007]. Saatavissa: <http://maemo.org/intro/>

Maemo-Yhteisö 2007b. Development Platform: White paper [verkkodokumentti]. [Viittaus 20.8.2007]. Saatavissa: [http://test.maemo.org/platform/docs/maemo\\_exec\\_whitepaper.html](http://test.maemo.org/platform/docs/maemo_exec_whitepaper.html)

Nokia 2007a. Nokia 770 Internet Tablet [verkkodokumentti]. [Viittaus 17.8.2007]. Saatavissa: <http://www.nokia.fi/A4312087>

Nokia 2007b. Nokia 770 User's guide [verkkodokumentti]. [Viittaus 20.8.2007]. Saatavissa: [http://nds2.nokia.com/files/support/nam/phones/guides/770\\_US\\_en.PDF](http://nds2.nokia.com/files/support/nam/phones/guides/770_US_en.PDF)

Ojanperä, V 2005. Nopeaa koodinkäännöstä [Verkkodokumentti]. [Viittaus 15.9.2007]. Saatavissa: <http://www.paivalehdenarkistosaatio.fi/arkisto/artikkelit/2005-2/PDF/TEENAIN.PDF>

Lewan, T 2007. Chip implants linked to animal tumors [Verkkodokumentti]. [Viittaus 5.9.2007]. Saatavissa: [http://www.washingtonpost.com/wp-dyn/content/article/2007/09/08/AR2007090800997\\_pf.html](http://www.washingtonpost.com/wp-dyn/content/article/2007/09/08/AR2007090800997_pf.html)

Wikipedia. 2007a. Eclipse (software) [verkkodokumentti]. [Viittaus 20.8.2007]. Saatavissa: [http://en.wikipedia.org/wiki/Eclipse\\_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))

Wikipedia. 2007b. GIMP [verkkodokumentti]. [Viittaus 15.9.2007]. Saatavissa: <http://en.wikipedia.org/wiki/GIMP>

Wikipedia. 2007c. MySQL [verkkodokumentti]. [Viittaus 15.9.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/MySQL>

Wikipedia. 2007d. Radio-frequency identification [verkkodokumentti]. [Viittaus 15.8.2007]. Saatavissa: <http://en.wikipedia.org/wiki/RFID>

Wikipedia. 2007e. RFID [verkkodokumentti]. [Viittaus 16.8.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/RFID>

## KUVALUETTELO

Kuva 1. RFID-tunnisteita

Kuva 2. Nokia Internet Tablet

Kuva 3. Maemo-alustan rakenne

Kuva 4. Gazpachon käyttöliittymä

Kuva 5. Säikeiden muodostuminen ohjelmassa

Kuva 6. palvelimen toiminta

Kuva 7. Tiedonsiirto ohjelmiston ja MySQL-palvelimen välillä

Kuva 8. Otos MySQL-tietokannasta

Kuva 9. MobDemo-ohjelman käyttöliittymä

Kuva 10. Mahdollinen parannus MySQL-tietokantaan