



TEKNIikka JA LIIKENNE

Tietotekniikka

Ohjelmistotekniikka

INSINÖÖRITYÖ

RFID-TEKNIikka KIRJASTOISSA

Työn tekijä: Petteri Kivimäki
Työn ohjaajat: Simo Silander
Miika Vacker

Työ hyväksytty: ____ . ____ . 2010

Simo Silander
lehtori



ALKULAUSE

Tämä insinööriyö tehtiin Mikro-Väylä Oy:lle yhteistyössä Aalto-yliopiston Otaniemen kampuskirjaston kanssa. Kiitän Aalto-yliopiston Otaniemen kampuskirjastoa mahdollisuudesta insinööriyön tekemiseen kirjastolle kuuluvan työajan puitteissa sekä Mikro-Väylä Oy:tä aiheen tarjoamisesta ja siihen perehdyttämisestä. Erityisesti kiitän Mikro-Väylä Oy:n myyntipäällikkö Miika Vackeria hänen antamastaan tuesta. Kiitokset myös lehtori Simo Silanderille insinööriyön tekoon liityvästä ohjauksesta ja opastuksesta.

Lisäksi kiitän kaikkia minua insinööriyön tekemisessä tukeneita ystäviä, työtovereita ja sukulaisia. Erityisesti haluan kiittää avopuolisoani Nicolettaa koko opiskeluajan jatkuneesta tuesta ja kannustuksesta.

Helsingissä 16.10.2010

Petteri Kivimäki

TIIVISTELMÄ

Työn tekijä: Petteri Kivimäki	
Työn nimi: RFID-tekniikka kirjastoissa	
Päivämäärä: 16.10.2010	Sivumäärä: 50 s. + 9 liitettä
Koulutusohjelma: Tietotekniikka	Suuntautumisvaihtoehto: Ohjelmistotekniikka
Työn ohjaaja: lehtori Simo Silander	
Työn ohjaaja: myyntipäällikkö Miika Vacker	
<p>Tässä opinnäytetyössä tutustuttiin RFID-tekniikan toimintaperiaatteisiin ja sen käyttöön kirjastoympäristössä. Työssä käytiin läpi kirjastojen keskeisimmät tietojärjestelmät sekä niiden yhteistoiminta RFID-järjestelmän kanssa. Osana työtä kehitettiin RFID-järjestelmän ja kirjastojärjestelmän yhteiskäyttöä tehostava RFID Plugin -sovellus.</p> <p>RFID on radiotaajuuksiin perustuva langaton tekniikka tiedon etäluvuun ja tallentamiseen. RFID-järjestelmät koostuvat neljästä peruskomponentista: tunnistesta, lukijasta, väliohjelmistosta ja taustajärjestelmästä. Lukijan avulla luettu tieto siirretään eteenpäin väliohjelmiston kautta taustajärjestelmään, jossa sitä voidaan käsitellä ja analysoida.</p> <p>Kirjastojen RFID-sovellukset keskittyvät pääasiassa aineiston lainaukseen ja inventointiin, varkaudenestoon sekä kokoelmienhallintaan. Langattoman tiedonvälityksen ansiosta RFID-tekniikka tarjoaa monia etuja kirjastoissa yleisesti käytettyihin viivakoodeihin verrattuna. Kirjaston RFID-järjestelmän toiminta on tiukasti sidoksissa taustajärjestelmän toimivaan kirjastojärjestelmään.</p> <p>Työssä toteutetun RFID Plugin -sovelluksen tarkoituksena oli tehostaa RFID- ja kirjastojärjestelmän yhteiskäyttöä kirjaston asiakaspalvelussa. Tämä saavutettiin parantamalla järjestelmien välistä keskinäistä tiedonvaihtoa välittämällä tiedot kirjastojärjestelmän tapahtumista RFID-järjestelmälle. Sovellus suunniteltiin ja toteutettiin yhteensopivaksi Voyager-, PrettyLib-, ATP Origo- ja PallasPro-kirjastojärjestelmien kanssa sekä 3M Staff Pad Workstation- ja Bookmatic RFID Staff -väliohjelmistojen kanssa.</p> <p>Tämän työn pohjalta voidaan todeta, että RFID-tekniikka tarjoaa kirjastoille monia mahdollisuuksia, joiden avulla kirjastot voivat tehostaa ja kehittää palveluitaan sekä toimintaansa. Eri järjestelmien välinen puutteellinen kommunikaatio kuitenkin rajoittaa vielä tällä hetkellä koko RFID-tekniikan tarjoaman potentiaalinn hyödyntämistä kirjastoympäristössä.</p>	
Avainsanat: RFID-tekniikka, kirjastot, kirjastojärjestelmät, kirjastoautomaatio	

ABSTRACT

Name: Petteri Kivimäki	
Title: RFID Technology in Libraries	
Date: 16 October 2010	Number of pages: 50 + 9
Department: Information Technology	Study Programme: Software Engineering
Instructor: Simo Silander, Senior Lecturer	
Supervisor: Miika Vacker, Sales Manager, Mikro-Väylä Oy	
<p>The aim of this bachelor's thesis was to explore the principles of Radio frequency identification (RFID) and how it is deployed in libraries. The subject for the thesis was commissioned by Mikro-Väylä Oy. The study is based on object-oriented software development and literary research in the field of library automation and RFID.</p> <p>This study includes an overview of the characteristics of integrated library systems, the fundamentals of RFID and a description of the communication between the RFID system and the integrated library system. The RFID Plugin application that improves the communication between the two systems was also implemented as a part of the study.</p> <p>RFID is wireless system which enables data transmission via radio waves without physical or line of sight contact. An RFID system consists of a tag, a reader, a middleware and an IT system running in the background that makes use of the data the reader reads on the tag.</p> <p>Libraries use RFID for stock management, inventory, item security and circulation control. Thanks to RFID system's wireless data transmission it offers several advantages over traditional barcodes that are widely used in libraries.</p> <p>The main goal of the RFID Plugin application was to make the common use of the RFID system and the integrated library system more effective in the customer service. The goal was achieved by improving the communication between the systems by relaying the information about the integrated library system's events to the RFID system. The application is compatible with the following integrated library systems: Voyager, PrettyLib, PallasPro and ATP Origo. In addition the application is compatible with 3M Staff Pad Workstation and Bookmatic RFID Staff middleware.</p> <p>As a conclusion it can be stated that RFID offers many potential advantages that libraries can utilise in their services and routines. The current problem is that the lack of communication between various systems is preventing getting the best out of RFID in libraries.</p>	
Keywords: RFID, libraries, integrated library systems, library management systems, library automation	

SISÄLLYS

ALKULAUSE

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	1
2	KIRJASTOJÄRJESTELMÄT	2
2.1	Taustaa	2
2.2	Rakenne ja toiminnot	5
2.2.1	<i>Moduulit</i>	5
2.2.2	<i>Standardit</i>	7
2.2.3	<i>Arkkitehtuuri</i>	11
2.3	Tulevaisuus	13
3	RFID-TEKNIikka JA KIRJASTOT	14
3.1	RFID-järjestelmän komponentit	16
3.1.1	<i>RFID-tunniste</i>	17
3.1.2	<i>RFID-lukija</i>	21
3.1.3	<i>Taajuusalueet</i>	24
3.2	Standardit	25
3.3	Tietoturva	26
4	RFID-TEKNIIKAN KÄYTÖN TEHOSTAMINEN	29
4.1	Määrittely	30
4.2	Toimintaperiaate	31
4.3	Käytännön toteutus	33
4.4	Tekninen toteutus	35
4.5	Käyttäytyminen peruskäytössä	41
4.6	Testaus ja käyttöönotto	43
5	YHTEENVETO	45
	VIITELUETTELO	47

LIITTEET

Liite 1. PrettyLib-kirjastojärjestelmän lähettämät viestit

Liite 2. Listener-luokan toteutus

Liite 3. KeyStrokeListener-luokan listen-funktion toteutus

Liite 4. VoyagerHandler-luokan update-funktion toteutus

Liite 5. VoaygerListener-luokan listen-funktion toteutus

Liite 6. PallasListener-luokan mouseHookProc-funktion toteutus

Liite 7. PrettyLibListener-luokan listen-funktion toteutus

Liite 8. AntennaHandler3M-luokan update-funktion toteutus

Liite 9. RFID Plugin -sovelluksen pääohjelman toteutus

1 JOHDANTO

Tässä insinööriyössä tutustutaan RFID (Radio Frequency Identification) -tekniikan perusteisiin ja tekniikan käyttöön kirjastoympäristössä. Tämän lisäksi käydään läpi RFID-järjestelmän taustajärjestelmänä toimivan kirjastojärjestelmän toimintaperiaatteet ja keskeisimmät toiminnot. Lisäksi tutustutaan myös RFID-tekniikan ja kirjastojärjestelmän toimintaa sääteleviin keskeisiin standardeihin.

Insinööriyön tavoitteena on kehittää asiakaspalvelutilanteessa kirjastojärjestelmän lainausmoduulin ja RFID-järjestelmän välistä kommunikointia parantava sovellus. Tällä hetkellä lainaus- ja palautus-tilanteissa käyttäjä joutuu valitsemaan suoritettavan toiminnon sekä kirjastojärjestelmän lainausmoduulista että erillisestä RFID-lukijan toimintaa ohjaavasta väliohjelmistosta. Kirjastojärjestelmän lainausmoduulista aktivoidaan joko lainaus- tai palautus-toiminto, jonka lisäksi sama toiminto on aktivoitava myös väliohjelmistosta. RFID-lukijan kannalta lainaus- ja palautus-tapahtumien erona on se, että lukija muokkaa käsiteltävän aineiston sisältämän RFID-tunnisteen sisältämää suojaustietoa eli joko aktivoi tai passivoi suojauksen. Kehitettävän sovelluksen tarkoituksena on poistaa väliohjelmiston käytön aiheuttama ylimääräinen työvaihe.

Tämän insinööriyön tilaajana toimiva Mikro-Väylä Oy on 3M-kirjastotuotteiden valtuutettu jälleenmyyjä sekä johtava kirjastojen RFID-järjestelmien ja -tuotteiden toimittaja Suomessa. Heidän asiakaskuntansa koostuu eri sektoreilla toimivista kirjastoista, joiden joukkoon kuuluva Aalto-yliopiston Otaniemen kampuskirjasto tuki tämän insinööriyön tekemistä tarjoamalla mahdollisuuden työn tekemiseen kirjastolle kuuluvan työajan puitteissa.

Insinööriyön osana kehitettävä sovellus toteutetaan tiiviissä yhteistyössä Mikro-Väylä Oy:n kanssa. Sovelluksen toiminta tulee perustumaan heidän määrittelemiin tavoitteisiin ja vaatimuksiin. Insinööriyössä käydään läpi kehitettävän sovelluksen määrittely, toimintaperiaate, tekninen toteutus, käyttäytyminen peruskäytössä sekä testaus ja käyttöönotto. Lisäksi käydään lyhyesti läpi myös sovelluksen toteutuksessa käytetyt työvälineet.

2 KIRJASTOJÄRJESTELMÄT

Kirjaston ja sen koko toiminnan ytimenä on perinteisesti pidetty kirjaston kokoelmia. Kokoelmien tehokkaan käytön edellytyksenä kuitenkin on, että tiedot kokoelmien sisältämisestä teoksista ja niiden sisällöstä ovat saatavilla sellaisessa muodossa, joka mahdollistaa yksittäisen teoksen etsimisen ja paikantamisen kokoelmista. Kokoelmien tiedot on perinteisesti talletettu kirjaston aineistorekisteriin, jonka toteutustapa on muuttunut vuosien saatossa erilaisten listojen ja kortistojen kautta nykyiseen sähköiseen muotoonsa.

Nykyään aineistorekisteri on käytännössä tietokanta, joka pitää sisällään tiedot kirjaston kokoelmiin kuuluvasta aineistosta. Tietokanta on osa kirjastojärjestelmää, joka tarjoaa toiminnallisuuden tietokannan sisältämien tietojen käyttöön esimerkiksi tiedonhakuja varten. Kirjastojärjestelmän keskeisimpänä tehtävä voidaan pitää aineistorekisterin ylläpitoa, koska kaikki muut järjestelmän toiminnot rakentuvat tavalla tai toisella aineistorekisterin sisällön ympärille. Aineistorekisterin lisäksi kirjastojärjestelmissä on myös joukko muita rekistereitä, kuten esimerkiksi asiakas- ja erilaiset tapahtumarekisterit. Rekisterien ja niiden sisältämien tietojen ajantasaisuus ja virheettömyys ovat avainasemassa järjestelmän virheettömän toiminnan kannalta.

2.1 Taustaa

Kirjastojen historiaa ajassa taaksepäin katsottaessa ovat tietokonepohjaiset järjestelmät vielä kohtuullisen tuore ilmiö kirjastomaailmassa. Siirtyminen erilaisista listoista ja kortistoista digitaaliseen aikaan alkoi USA:ssa noin 40 vuotta sitten.

Kirjastojen atk-järjestelmien kehitys sai alkunsa USA:ssa 1960-luvulla ja vuosikymmenen loppuun mennessä käytössä oli jo useita toimivia järjestelmiä. Vain suurimmilla kirjastoilla oli kuitenkin varaa järjestelmien hankintaan, koska niiden käyttöön vaadittava laitteisto oli tuohon aikaan kallista. Järjestelmät olivat pääsääntöisesti paikallisesti käytettäviä ja tiettyä kirjastoa varten kehitettyjä. Tekniikan asettamista rajoituksista johtuen järjestelmät sisältsivät yleensä vain muutamia perustoimintoja, kuten esimerkiksi luetteloiden tulostus tai lainauksenvalvonta. Tietojen käsittely tapahtui määrääjain suoritettavien eräajojen avulla. Kirjaston lainaustapahtumat vietiin järjestelmään esimerkiksi päivittäin suoritettujen eräajojen kautta. [1, s. 12.]

Ensimmäiset online-tiedonhaku tarjonneet järjestelmät otettiin käyttöön 1960-luvun lopulla ja 1970-luvun alussa tarjolla oli jo useita online-hakupalveluita. Ensimmäiset hakupalvelut eivät kuitenkaan vielä kehittyvätömistä ja kustannuksiltaan kalliista tiedonsiirtoyhteyksistä johtuen toimineet reaaliajassa, vaan tietojen siirto tapahtui määräajoin toimitettujen magneettinauhujen avulla. [2.]

1970-luvulla tekniikan kehittymisen ja halpenemisen myötä kirjastojärjestelmien käyttö yleistyi myös pienempien kirjastojen keskuudessa. Myös järjestelmien toiminnallisuus kehittyi ja reaaliaikaiset järjestelmät alkoivat syrjäyttää eräajoihin perustuneita järjestelmiä. Reaaliaikaiset järjestelmät mahdollistivat muun muassa reaaliaikaisen lainaustapahtumien hallinnan sekä useiden eri kirjastojen muodostamien kirjastoverkkojen syntymisen. Reaaliaikaisten järjestelmien aiheuttamat eräajojärjestelmiä suuremmat kustannukset pyrittiin usein jakamaan usean eri kirjaston kesken muodostamalla kirjastoverkko, johon kuuluvat kirjastot käyttivät samaa suurta keskustietokonetta, jolle kirjastojärjestelmä oli asennettu. [1, s.12.]

1970-luvun lopulla USA:ssa kehitettiin myös ensimmäiset kirjastojärjestelmiin sidoksissa olleet online-luettelot. Luetteloiden käyttö tapahtui telnet-yhteyden avulla ja hakeminen suoritettiin tiettyjen ennalta määriteltyjen hakuvaihtoehtojen perusteella. Ensimmäiset luettelot olivat perustoiminta-ajatukseltaan hyvin lähellä paperikortistoja eivätkä siinä mielessä tarjonneet suuriakaan tiedonhakuun liittyviä edistysaskeleita. [3.]

Kirjastojärjestelmien kehityksen kannalta merkittävänä asiana voidaan pitää myös Yhdysvaltain kongressin kirjaston 1960-luvulla kehittämän MARC (Machine Readable Cataloging) -formaatin syntymistä. Formaatti määritteli, kuinka luettelokorttien sisältämä bibliografinen data esitetään tietokoneluetavassa muodossa, jota voidaan siirtää eri järjestelmien välillä. Päämääränä oli vähentää luettelointiin liittyvää työmäärää jakamalla luettelointitietoja kirjastojen kesken. 1960-luvun lopulla formaatin uudistettu versio MARC2 oli jo kansainvälinen standardi, jota uudet kirjastoille kehitetyt atk-sovellukset suunniteltiin tukemaan. [2.]

MARC-formaatilla oli merkittävä vaikutus sekä koko kirjastotyöhön että kirjastojärjestelmien kehitykseen, sillä sen käyttö mahdollisti luettelointitietojen jakamisen usean eri kirjaston kesken. Aikaisemmin jokaisen kirjaston oli itse

luetteloitava kaikkien kokoelmiinsa kuuluvien teosten tiedot, mutta kopioluetteloinnin avulla luettelointitietoja pystyttiin kopioimaan keskitetyistä luetteloitipalveluista tai kirjastoverkkojen sisäisistä aineistorekistereistä. Kopioluetteloinnin mahdollistaneilla järjestelmillä oli erittäin suuri vaikutus siihen, miten tietotekniikkaa ja tietoverkkoja tulitaisiin tulevaisuudessa hyödyntämään kirjastojen välisessä yhteistyössä ja tiedonvaihdossa. [4, s. 62.]

1980-luvulla tietotekniikan kehitys sekä laitteiden halpeneminen mahdollistivat entistä suurempien ja monitoimisempien järjestelmien rakentamisen. Vuosikymmenen puoliväliin mennessä kirjastojärjestelmien toiminnallisuus olikin laajentunut kattamaan kirjastotyön keskeisimmät osa-alueet. Kirjastojärjestelmät koostuivat yleensä useista eri ohjelmamoduuleista, joista jokainen piti sisällään tiettyyn kirjastotyön osa-alueen hoitamiseen tarvittavat toiminnot. Tyypillisiä ohjelmamoduuleja olivat esimerkiksi lainaus-, luettelointi-, hankinta-, tilastointi- ja näyttöluettelo-moduuli, joiden kautta järjestelmän eri rekisterien sisältämiä tietoja käsiteltiin. Tällä periaatteella toimivia järjestelmiä alettiin kutsua integroiduiksi kirjastojärjestelmiksi, joita myös nykyään käytössä olevat kirjastojärjestelmät ovat.

Integroitujen kirjastojärjestelmien keskeisiin ominaisuuksiin kuului myös kyky kommunikoida ja vaihtaa tietoja toisten kirjastojärjestelmien kanssa. Näitä ominaisuuksia hyödynnettiin esimerkiksi luettelointitietojen kopioimiseen sekä tiedonhaussa viitteiden ja tiedonlähteiden hakemiseen muista järjestelmistä. [1, s. 13.]

1980-luvulla myös asiakkaiden käyttöön tarkoitetut reaaliaikaiset näyttöluettelot alkoivat yleistyä. Niiden avulla asiakkaat pääsivät itsenäisesti tekemään hakuja kirjaston kokoelmista. Ensimmäisten näyttöluettelojen ongelmina olivat kuitenkin heikko käytettävyys sekä yhteisten tekijöiden puuttuminen eri järjestelmien välillä. Yhden hakujärjestelmän käytön oppiminen ei siis välttämättä auttanut lainkaan toisen hakujärjestelmän käytössä. [5, s. 170.]

1990-luvulla tapahtunut internetin yleistyminen ja tietokoneiden tuleminen suurten massojen ulottuville vaikutti suuresti myös kirjastoihin ja kirjastojärjestelmiin. Viimeistään tässä vaiheessa pienimmätkin kirjastot siirtyivät atk-pohjaisten järjestelmien käyttöön ja kirjastojen kokoelmaluettelot tulivat ihmisten käytettäväksi internetin kautta. Kirjastojen internetin kautta tarjoamien palvelujen määrä kasvoi myös nopeasti ja asiakkaat saattoivat suorittaa mo-

net aiemmin kirjastossa käyntiä vaatineet toimenpiteet kuten lainojen uusimisen ja varausten tekemisen omatoimisesti kirjastojen verkkopalveluissa. Kirjaston näyttöluettelon rooli muuttui, kun tiedonhakumahdollisuudet kehittyivät ja näyttöluettelon kautta tarjolla olleiden palvelujen määrä kasvoi ja monipuolistui [6, s. 3]. Kirjastojärjestelmään kohdistuvat vaatimukset kasvoivat kaikilla osa-alueilla ja myös kirjastojen käyttäjät alkoivat asettaa omia vaatimuksiaan.

Kirjastojärjestelmien kehittyessä kyky kommunikoida ja vaihtaa tietoja muiden kirjasto- ja tietojärjestelmien kanssa on noussut kaiken aikaa tärkeämpään asemaan järjestelmien toiminnassa. Erilaisten kirjastojärjestelmien sisäiseen sekä ulkoiseen toimintaan liittyvien standardien ja protokollien määrä onkin kasvanut merkittävästi viimeisten vuosikymmenten aikana. Kirjastoalan omien vaatimusten lisäksi järjestelmien on nykyään tuettava myös muiden alojen standardeja ja protokollia kuten esimerkiksi erilaisten web-palveluiden toteuttamiseen liittyvät protokollat. [7.]

2.2 Rakenne ja toiminnot

Valtaosa nykyisin käytössä olevista kirjastojärjestelmistä lukeutuu toiminta-ajatuksensa puolesta 1980-luvun puolivälissä markkinoille tulleisiin integroituihin kirjastojärjestelmiin. Näiden järjestelmien keskeisenä piirteenä on, että järjestelmät ovat laajoja ja niiden toiminnallisuus kattaa valtaosan kirjastotyön eri osa-alueista.

Integroidut kirjastojärjestelmät rakentuvat monista erilaisista ohjelmamoduuleista, joista jokainen tarjoaa jonkun tietyn kirjastotyön osa-alueen hallintaan tarvittavan toiminnallisuuden. Kaikki moduulit sisältävät kuitenkin erilaisia aineiston hakuun liittyviä toimintoja. Kukin moduuli voi olla toteutettu useana erillisenä ohjelmana tai vaihtoehtoisesti yksi massiivinen ohjelma voi pitää sisällään usean eri moduulin toiminnallisuuden. Toteutusmalli vaihtelee järjestelmästä riippuen. Tässä yhteydessä ohjelmamoduuleita käsitellään itsenäisinä kokonaisuuksina.

2.2.1 Moduulit

Tyypillisiä ohjelmamoduuleita ovat lainaus-, hankinta-, saapumisvalvonta-, luettelointi- ja ylläpitomoduli. Lisäksi myös näyttöluettelo eli OPAC (Online Public Access Catalog) luetaan usein omaksi moduulikseen. Osa kirjastojär-

jestelmistä sisältää myös niin sanotun itsepalvelumoduulin, jonka avulla asiakkaat voivat omatoimisesti lainata ja palauttaa aineistoa. Varsin usein itsepalvelumoduuli on kuitenkin toteutettu kirjastojärjestelmästä erillisellä, toisen ohjelmistotoimittajan valmistamalla ohjelmistolla, joka käyttää hyväkseen kirjastojärjestelmän tarjoamia rajapintoja.

Lainausmoduuli

Lainausmoduuli tarjoaa asiakasrekisterin ylläpitämiseen sekä kirjaston kokoelmiin kuuluvan aineiston lainaamiseen tarvittavat toiminnot. Lainausmoduulin kautta suoritettavat lainaustapahtumat tallentuvat järjestelmän tapahtumarekisteriin, jossa olevia lainaustapahtumiin liittyviä tietoja käytetään muun muassa erilaisten asiakkaille lähetettävien muistutusten ja tilastojen tuottamiseen järjestelmästä.

Luettelointimoduuli

Aineistorekisterin ylläpito tapahtuu pääasiassa luettelointimoduulin avulla. Moduulia käytetään kirjaston kokoelmiin kuuluvien teosten tietojen syöttämiseen, muokkaamiseen ja poistamiseen aineistorekisteristä. Järjestelmästä riippuen moduuli voi tarjota myös mahdollisuuden luettelointitietojen eli tietueiden hakemiseen ja kopioimiseen muista järjestelmistä.

Hankinta- ja saapumisvalvontamoduuli

Hankinta- ja saapumisvalvontamoduuli liittyvät myös aineistorekisterin ylläpitoon, sillä niitä käytetään uusien teosten tilaamiseen ja sarjajulkaisujen saapumisvalvontaan. Hankintamoduuli pitää yleensä sisällään myös laskutukseen ja kulujen hallintaan käytettävää toiminnallisuutta.

Ylläpitomoduuli

Ylläpitomoduulin kautta on mahdollista muuttaa erilaisia järjestelmään liittyviä asetuksia. Asetukset ja oikeudet muutoksiin riippuvat hyvin suuressa määrin käytettävästä järjestelmästä. Tyypillisimpiä ylläpitomoduulin kautta tehtäviä määrityksiä ovat käyttäjätunnukset ja lainattavan materiaalin laina-aikoihin liittyvät asetukset. Erilaisten määritysten ja asetusten määrä voi suurissa järjestelmissä nousta erittäin suureksi, mikä hankaloittaa järjestelmän ylläpitoa ja vikatilanteiden selvittämistä.

Näyttöluettelomoduuli

Nykyisin näyttöluettelo on poikkeuksetta internet-palvelu, jonka kautta kirjaston käyttäjät voivat muun muassa hakea kirjaston kokoelmiin kuuluvaa aineistoa, tehdä varauksia ja uudistaa lainojaan. Nykyiset näyttöluettelot ovat ajantasajärjestelmiä ja kirjaston tietokannassa tapahtuvat muutokset päivittyvät niihin välittömästi. Käytännössä tämä tarkoittaa sitä, että esimerkiksi asiakkaan kirjastoon palauttama laina siirtyy näyttöluettelossa saatavantaan heti, kun laina kuitataan palautetuksi lainausmoduulissa. Näyttöluettelo eroaa järjestelmän muista moduuleista siten, että se on suunnattu asiakkaiden käyttöön, kun muut moduulit on tarkoitettu pelkästään kirjaston henkilökunnalle.

Tiedot, joita eri ohjelmamoduulien avulla käsitellään, on tallennettu tietokantapalvelimella sijaitsevaan tietokantaan. Tietokanta koostuu yleensä kymmenistä, ellei jopa sadoista, toisiinsa sidoksissa olevista tauluista. Tietokanta sisältää yleensä kirjaston aineisto-, asiakas- ja tapahtumarekisterin.

Tietokantapalvelimen ja ohjelmamoduulien lisäksi kirjastojärjestelmä pitää sisällään myös useita muita komponentteja. Www-pohjainen näyttöluettelo tarvitsee esimerkiksi toimiakseen www-palvelimen, eikä asiakkaille sähköpostitse lähetettävien ilmoitusten, kuten myöhästymismuistutusten lähettäminen onnistu ilman sähköpostipalvelinta. Järjestelmä koostuu siis monista erillisistä toisiinsa tiiviisti sidoksissa olevista osista, jotka eivät useinkaan sijaitse fyysisesti kovin lähellä toisiaan.

2.2.2 Standardit

Kirjastojärjestelmän on kyettävä kommunikoimaan ja vaihtamaan tietoja sekä muiden kirjastojärjestelmien että monien erilaisten tietojärjestelmien kuten esimerkiksi kaukopalvelu- tai taloushallinto-ohjelmistojen kanssa. Tämä ei kuitenkaan ole mahdollista ilman yhtenäisiä standardien ja protokollien varaan rakentuvia rajapintoja, joita kaikki keskenään kommunikoivat järjestelmät tukevat. [7.]

Osa standardeista ja protokollista on yleisesti monien eri alojen sovellusten käytössä, kun osa taas on erityisesti kirjastoalaan liittyviä. Kirjastojärjestelmien tukemia yleisesti käytössä olevia standardeja ja protokollia ovat muun muassa OSI (Open Systems Interconnection Reference Model) -malli,

SGML (Standard Generalized Markup Language) -kieli, TCP/IP- ja SMTP (Simple Mail Transfer Protocol) -protokollat sekä Unicode-merkistöstandardi. [8.]

Kirjasto- ja tietopalvelualan standardien tekemisestä vastaa NISO (National Information Standards Organization) -komitea, joka toimii amerikkalaisen liittovaltion standardoinnista vastaavan ANSI (American National Standard Institute) -viraston alaisuudessa [9]. Pääsääntöisesti NISO:n standardit hyväksytään myöhemmin sellaisenaan myös ISO-standardeiksi [10]. NISO-komiteasta käytettävä koodi on Z39, minkä johdosta komitean tekemät juoksevasti numeroidut standardit ovat aina Z39-alkuisia. Suomessa tietopalvelualan standardeista vastaa puolestaan Suomen standardisoimisliiton Tietohuoltokomitea [10].

Osa kirjastoalan standardeista tunnetaan paremmin standardin nimen ja osa puolestaan standardin koodin perusteella. Keskeisimpiä kirjastoalan standardeja ja protokollia ovat MARC-formaatti, Z39.50-standardi, SIP (Standard Interchange Protocol) -protokolla, NCIP (NISO Circulation Interchange Protocol) -protokolla ja OpenURL-standardi.

MARC-formaatti

MARC-formaatti on määritelty ANSI/NISO Z39.2 -standardissa, ja se määrittää bibliografisen datan esittämisen tietokone luettavassa muodossa aneistotyyppistä riippumatta. Käytännössä tämä tarkoittaa teoksen bibliografisen datan tallentamista tietueeseen, jossa jokaista yksittäistä bibliografista tietoa kuten esimerkiksi tekijää tai nimekettä varten on oma kenttensä. Bibliografisen datan lisäksi standardia voidaan soveltaa myös varastotietojen tallentamiseen. MARC-formaatti mahdollistaa bibliografisen datan vaihdon eri kirjastojärjestelmien välillä. [11.]

MARC-formaatista on olemassa useita erilaisia versioita, jotka eroavat toisistaan sisältämiensä kenttien perusteella. Käytännössä tämä tarkoittaa sitä, että tietyn bibliografisen tiedon kuten esimerkiksi tekijän tai nimekkeen tallennuskenttä vaihtelee eri formaattien välillä. Eri formaateissa esitettyä bibliografista dataa pystytään kuitenkin siirtämään kirjastojärjestelmien välillä erilaisten konversiotyökalujen avulla. Suomessa käytettävät MARC-formaatit ovat kansallinen FINNMARC-yhtenäisformaatti ja kansainvälinen MARC 21-formaatti. [12.]

Z39.50-standardi

Z39.50 on NISO-komitean standardi sekä myös ISO-standardi ISO 23850. Standardi määrittelee asiakas-palvelin-mallin mukaisen palvelun ja protokollan, jonka avulla asiakas pystyy suorittamaan tiedonhakuja palvelimella sijaitsevasta tietokannasta. Protokolla määrittelee asiakasohjelman ja palvelimen välisen kommunikaation, mutta asiakasohjelman ja loppukäyttäjän välinen vuorovaikutus ei kuulu standardin piiriin. [13, s. 1.]

Nykyään lähes kaikki kirjastojärjestelmät tukevat Z39.50-standardia, sillä ilman sen tukea organisaatorajat ylittävä yhteistyö ja kirjastojärjestelmän liittäminen mukaan erilaisiin portaaleihin olisi erittäin vaikeaa, ellei jopa mahdotonta. Standardi mahdollistaa hakulogiikan ja -termien esittämisen laitteisto- ja ohjelmistoriippumattomassa muodossa, minkä ansiosta erilaisten portaalien ja monihakujen toteuttaminen yksinkertaistuu, kun jokainen kohdejärjestelmä soveltaa samaa hakumenetelmää. Käytännössä eri kirjastojärjestelmien Z39.50-toteutuksissa on kuitenkin vielä yhteistoimintaa haittaavia puutteita ja eroavaisuuksia. [10.]

SIP-protokolla

SIP-protokolla on alun perin 3M-yhtiön kehittämä protokolla, jonka avulla 3M:n valmistamat kirjastojen itsepalvelulaitteet kommunikoivat kirjastojärjestelmän kanssa. Protokolla on sittemmin otettu varsin laajasti käyttöön myös muiden laite- ja ohjelmistovalmistajien keskuudessa, vaikka se ei virallinen standardi olekaan. Myös suurin osa kirjastojärjestelmistä tukee SIP-protokollaa, vaikka toteutuksen yksityiskohdat saattavatkin vaihdella eri järjestelmien välillä. Protokollaa kehitetään ja ylläpidetään 3M:n toimesta ja tällä hetkellä protokollan tuorein versio on SIP 2.0. [14, s. 25.]

SIP-protokolla määrittelee joukon eri järjestelmien välisen tiedonsiirron mahdollistavia viestejä ja palveluita. Protokolla mahdollistaa asiakas- ja aineistotietojen siirtämisen sekä lainaus- että palautustoimintojen suorittamisen eri järjestelmien välillä. Käytännössä protokollaa käytetään kirjastojen itsepalvelulaitteiden ja kirjastojärjestelmän välisessä kommunikoinnissa. [15.]

NCIP-protokolla

NCIP-protokolla on määritelty ANSI/NISO Z39.83 -standardissa ja se pohjautuu osittain 3M:n SIP-protokollaan. NCIP eroaa SIP-protokollasta kuitenkin siten, että se määrittelee SIP-protokollaa monipuolisemman eri järjestelmien yhteistoiminnan mahdollistavan rajapinnan. Rajapinta muodostuu SIP-protokollan tapaan joukosta eri järjestelmien välisen tiedonsiirron mahdollistavia viestejä ja palveluja. [14, s. 26.]

Myös NCIP-protokolla mahdollistaa tietojen siirtämisen sekä lainaus- ja palautustoimintojen suorittamisen eri järjestelmien välillä. Kirjastojärjestelmän ja kirjastojen itsepalvelulaitteiden välisen kommunikoinnin lisäksi sitä voidaan käyttää myös esimerkiksi kirjastojärjestelmän ja erillisen kaukopalveluohjelmiston välisen lainaustapahtumiin liittyvän tiedonvaihdon toteuttamiseen. Aineiston paikantamisessa ja hakujen tekemisessä kirjastojärjestelmän aineistorekisteristä kaukopalveluohjelmisto hyödyntää sen sijaan Z39.50-standardin määrittelemää protokollaa. Kaukopalveluohjelmistoa käytetään kirjastojen väliseen lainaus- ja jäljennepalvelutoimintaan. [16, s. 1.]

Käytännössä sekä SIP- että NCIP-protokolla mahdollistavat kirjastojärjestelmän kanssa kommunikoinnin ilman järjestelmän sisäisen rakenteen ja toteutuksen yksityiskohtien tuntemista. Tämä puolestaan tekee mahdolliseksi sen, että myös järjestelmätoimittajista riippumattomat yritykset pystyvät tuottamaan kirjastojärjestelmän kanssa yhteensopivia laitteita ja palveluita. Kevyemmän toteutuksensa ansiosta SIP-protokolla on näistä kahdesta huomattavasti käytetympi [14, s. 26].

OpenURL-standardi

OpenURL on ANSI/NISO Z39.88 -standardi, joka mahdollistaa verkossa olevan julkaisun dynaamisen linkittämisen viitteeseen. Dynaaminen linkittäminen tarkoittaa tässä yhteydessä sitä, että käyttäjälle näytettävä verkkojulkaisun linkki ei vastaa julkaisun todellista sijaintia, vaan linkki rakennetaan julkaisun yksilöivien metatietojen perusteella. Näin linkki säilyy aina samana, vaikka julkaisun fyysinen sijainti verkossa muuttuisikin. Julkaisun todelliseen sijaintiin ja käyttöoikeuksiin liittyvät tiedot on tallennettu erilliseen linkityspalveluun. Käytännössä OpenURL-tukea käytetään esimerkiksi tiedonhakuportaalin, digitaalisten aineistojen hallintasovelluksen ja kirjastojärjestelmän yhteistoiminnassa. [17.]

Teknisesti OpenURL on kahdesta eri osasta muodostuva URL-osoite. Osoitteen alkuosa ilmaisee käytettävän linkityspalvelun osoitteen ja on sama kaikille saman linkityspalvelun piiriin kuuluville verkkojulkaisuille. Osoitteen loppuosa sisältää puolestaan julkaisun yksilöivät metatiedot ohjelmiston ymmärtämään muotoon koodattuna. [17.]

Kirjastojärjestelmien OpenURL-toteutukset voidaan jakaa kahteen osaan sen perusteella, sisältääkö järjestelmä omaa linkityspalvelua vai ei. Kummassakin tapauksessa kirjastojärjestelmä luo verkkojulkaisujen metatietoihin perustuvan OpenURL-osoitteen, mutta oman linkityspalvelun puuttuessa järjestelmä lähettää luomansa OpenURL-osoitteen ulkopuoliseen linkityspalveluun. Järjestelmän oma linkityspalvelu voi sen sijaan olla järjestelmätoimittajan rakentama erillissovellus tai sisältyä kirjastojärjestelmään. [17.]

2.2.3 *Arkkitehtuuri*

Modernit kirjastojärjestelmät ovat pääsääntöisesti hajautettuja järjestelmiä. Hajautetun järjestelmän perusajatuksena on, että järjestelmä koostuu useista fyysisesti eri paikoissa sijaitsevista komponenteista, jotka kommunikoivat keskenään verkon välityksellä [18, s. 2].

Vanhemmat kirjastojärjestelmät noudattivat niin sanottua yksitasoista mallia, jossa kaikki ohjelmisto sijaitsi palvelimella ja päätteellä oli vain pelkkä käyttöliittymä [19, s. 112]. Nykyisin käytössä olevat järjestelmät perustuvat sen sijaan joko kaksi- tai kolmitasoiseen malliin.

Kaksitasoisesta client-server-mallista on olemassa kaksi erilaista toteutusta: fat- ja thin-client. Fat-client on perinteinen client-server-ratkaisu, jossa tietokanta sijaitsee palvelimella ja ohjelmisto sekä käyttöliittymä käyttäjän työasemalla. Thin-client-toteutuksessa sekä tietokanta että ohjelmisto sijaitsevat palvelimella ja käyttöliittymä, esimerkiksi web-selain, sijaitsee käyttäjän työasemalla. [19, s. 113.]

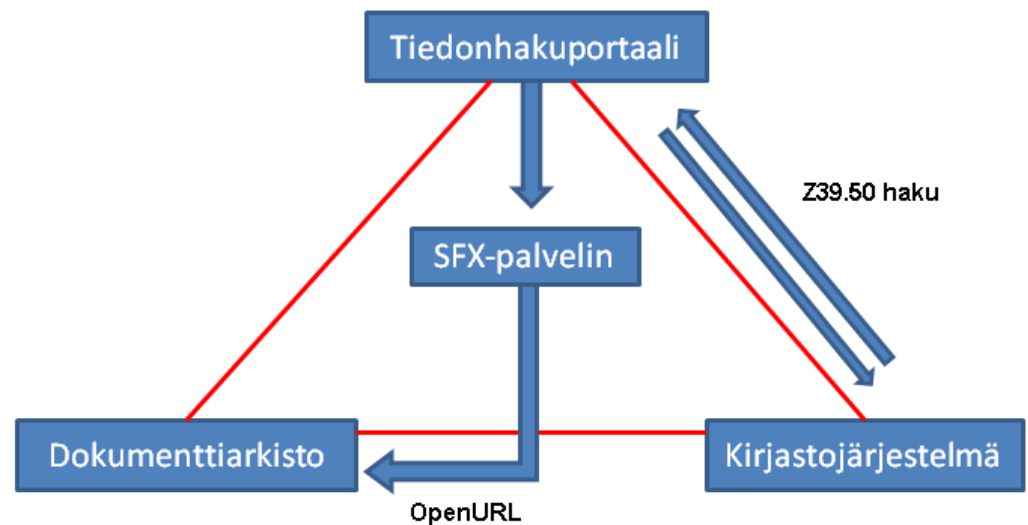
Kolmitasoisessa järjestelmässä tietokanta sijaitsee puolestaan tietokantapalvelimella ja ohjelmisto sovelluspalvelimella. Järjestelmää käytetään esimerkiksi käyttäjän työasemalla sijaitsevan web-selaimen tai muun ohjelman avulla. Kolmitasoisien järjestelmän etuna kaksitasoiseen järjestelmään verrattuna on sen parempi mukautuvuus erilaisille muutoksille ja laajennuksille.

Rakenteeltaan kolmitasoinen järjestelmä on kuitenkin huomattavasti kaksitasoista järjestelmää monimutkaisempi. [19, s. 113.]

Kirjastojärjestelmä on kirjastojen käyttämistä tietojärjestelmistä keskeisin, mutta se ei kuitenkaan yksin pysty toteuttamaan kaikkia kirjastojen tarjoamia palveluita. Monet palvelut perustuvat usean eri tietojärjestelmän yhteistoimintaan, minkä vuoksi on oleellista tutustua myös tällaisen sovelluksen arkkitehtuuriin.

Triangeli

Käytännön esimerkki usean eri tietojärjestelmän yhteistoiminnan varaan rakentuvasta sovelluksesta on Suomen yliopisto- ja ammattikorkeakoulukirjastojen käytössä oleva Triangeli. Triangelin toiminta perustuu Voyager-kirjastojärjestelmän, MetaLib-tiedonhakuportaalin ja ENCompass-dokumenttiarkiston yhteistoimintaan. Järjestelmät kommunikoivat muun muassa Z39.50- ja OpenURL-standardiin perustuvien rajapintojen välityksellä. Triangelin toimintaperiaate on esitetty kuvassa 1. [5, s. 76.]



Kuva 1. Triangelin toimintaperiaate. [5, s. 77.]

Dokumenttien bibliografinen data sijaitsee kirjastojärjestelmässä ja tekninen kuvailutieto sekä itse digitaaliset dokumentit dokumenttiarkistossa. SFX-palvelin vastaa dokumenttien OpenURL-linkkien luomisesta, jonka lisäksi sen avulla on mahdollista rajoittaa dokumenttien käyttöä. Hakujen tekeminen on puolestaan mahdollista sekä portaalin että kirjastojärjestelmän kautta. [5, s. 78.]

Käytännössä Triangeli toimii siten, että käyttäjä suorittaa tiedonhaun esimerkiksi tiedonhakuportaalin tarjoaman käyttöliittymän kautta. Tämän jälkeen tiedonhakuportaali etsii käyttäjän syöttämiä hakuehtoja vastaavia tietueita kirjastojärjestelmän aineistorekisteristä kirjastojärjestelmän tarjoaman Z39.50-protokollaan perustuvan rajapinnan avulla. Kirjastojärjestelmä palauttaa aineistorekisteristä löytyvät hakuehtoja vastaavat tietueet ja SFX-palvelin luo dokumenttien metatietojen perusteella OpenURL-linkit dokumenttiarkistossa sijaitseviin digitaalisiin dokumentteihin. Mikäli haun suorittaneella käyttäjällä on käyttöoikeus dokumenttiin, palauttaa tiedonhakuportaali SFX-palvelimen luoman OpenURL-linkin hakutulosten yhteydessä. Linkkiä klikkaamalla käyttäjä pääsee käsiksi dokumenttiarkistossa sijaitsevaan dokumenttiin.

2.3 Tulevaisuus

Tulevaisuuden kirjastopalveluista puhuttaessa käytetään usein Web 2.0 -käsitteestä johdettua Kirjasto 2.0 -käsitettä. Web 2.0 -käsite tarkoittaa karkeasti ottaen www-sivujen vuorovaikutteisuuden lisäämistä sekä käyttäjien mahdollisuutta osallistua www-sivujen sisällön tuottamiseen. Kirjasto 2.0 -käsitteen voidaan täten ajatella tarkoittavan Web 2.0:n tarjoamien mahdollisuuksien ja ominaisuuksien tuomista mukaan kirjastojen verkkopalveluihin.

Kirjasto 2.0 -käsitteeseen liittyviä palveluita, kuten esimerkiksi RSS-syötteitä ja erilaisia näyttöluetteloon liitettäviä kommentointisovelluksia on jo toteutettu monissa kirjastoissa eri puolilla maailmaa useiden vuosien ajan. Toistaiseksi nämä toteutukset ovat kuitenkin pääsääntöisesti kirjastojärjestelmän päälle tai rinnalle rakennettuja erillisiä sovelluksia, jotka on toteutettu kirjastojen toimesta ilman järjestelmätoimittajan tukea. Tulevaisuudessa tällaiset toiminnot tulevat varmasti osaksi kirjastojärjestelmän sisältämiä toimintoja ja kirjastot voivat itse päättää, missä määrin ottavat niitä käyttöönsä. [7.]

Kirjastojärjestelmän aineistorekisteri on perinteisesti ollut kirjaston kokoelman tiedot sisältävä viitetietokanta. Aineistorekisteri on sisältänyt kokoelmia koskevat kuvailu- ja sijaintitiedot, joiden perusteella aineiston hakeminen sekä paikantaminen kirjaston kokoelmista on ollut mahdollista. Käytännössä tämä on tarkoittanut sitä, että aineisto ja siihen liittyvä kuvailutieto ovat sijainneet erillisissä paikoissa. Elektronisen aineiston lisääntyminen muuttaa tässä suhteessa kirjastojärjestelmän roolia melko radikaalisti ja asettaa jär-

jestelmän toiminnalle aivan uusia vaatimuksia. Pelkän kuvailutiedon sijaan myös varsinaiselle sähköiselle dokumentille on löydettävä mahdollisimman pysyvä tallennuspaikka. Tallennuspaikka voi olla esimerkiksi kirjastojärjestelmän sisällä tai erillisessä dokumenttiarkistossa.

Erilaisten kirjastoissa käytettävien tietojärjestelmien määrän kasvaessa nousevat eri järjestelmien välisen kommunikoinnin ja tiedonsiirron mahdollistavat rajapinnat yhä tärkeämpään asemaan. Yhteneväiset rajapinnat voidaan saavuttaa vain sekä kirjastoalan että muiden alojen standardeja noudattamalla. Tulevaisuudessa kirjastojärjestelmien tulisi myös tarjota entistä paremmat mahdollisuudet niiden sisältämän datan hyödyntämiseen ja käyttämiseen muissa järjestelmissä [20].

Myös monet kirjastojen käyttämät laitteet ja teknologiat asettavat vaatimuksia kirjastojärjestelmän toiminnalle sekä päinvastoin. Tekniikan nopean kehityksen vuoksi järjestelmän oheislaitteiden ja niiden käyttämien tekniikoiden kirjo tulee tulevaisuudessa kasvamaan entisestään. Osa uusista tekniikoista on liitettävissä kirjastojärjestelmään sellaisenaan, kun osa taas edellyttää järjestelmää täydentävien lisäkomponenttien hankkimista. Oman lisähaasteensa tuo se, että uutta ja vanhaa tekniikkaa voidaan joutua käyttämään myös rinnakkain.

3 RFID-TEKNIikka JA KIRJASTOT

RFID on radiotaajuuksiin perustuva langaton tekniikka tiedon etälukuun ja tallentamiseen. Termiä käytetään yleisnimityksenä radiotaajuuksilla tapahtuvalle tunnistamiselle ja sen alle kuuluu monia erilaisia käytännön toteutuksia ja tekniikoita.

RFID ei ole mikään uusi tekniikka, sillä sen juuret liittyvät tutkan keksimiseen vuonna 1935. Ensimmäinen patentoitu RFID-tunniste rakennettiin 1970-luvun alussa amerikkalaisen Mario Cardullon ja hänen ryhmänsä toimesta. Ensimmäiset kaupalliset RFID-tekniikkaa hyödyntäneet sovellukset otettiin puolestaan käyttöön Amerikassa 1980-luvulla. [21, s. 42.]

Ensimmäisiä RFID-tekniikkaa hyödyntäneitä sovelluksia käytettiin muun muassa tiemaksujen perintään, eläinten merkitsemiseen ja ajoneuvojen jäljitykseen. Viimeisten vuosikymmenten aikana tekniikka on levinnyt lähes kaikille elämän eri osa-alueille. Nykyisin RFID-tekniikkaa käytetään muun muassa

teollisuuden prosessien seurantaan, logistiikkaan, kulunvalvontaan, eläintunnistukseen, maksamiseen, matkustamiseen, varkaudenestoon ja omaisuuden seurantaan liittyvissä sovelluksissa. [22, s. 30.]

Kirjastoissa ensimmäiset RFID-sovellukset otettiin käyttöön 1990-luvun puolivälissä, minkä myötä tekniikka alkoi vähitellen yleistyä myös kirjastomaailmassa. Kirjastojen RFID-sovellukset keskittyivät pääasiassa aineiston lainaukseen ja inventointiin, varkaudenestoon sekä kokoelmienhallintaan. Vuoteen 2005 mennessä eri puolilla maailmaa RFID-tekniikkaa käyttäviä kirjastoja oli jo useita satoja ja tekniikan piiriin kuului arviolta noin 120 miljoonaa nidettä. Tekniikan leviämistä ja käyttöönottoa kuitenkin hidastivat siihen liittyvät korkeat käyttöönottokustannukset. Kustannuksista huolimatta RFID-järjestelmä oli kuitenkin kirjastojen kannalta kiinnostava vaihtoehto, koska se pystyi tarjoamaan useita etuja kirjastoissa yleisesti käytettyihin viivakodeihin verrattuna. [14, s. 1.]

RFID vs. viivakoodit

RFID-tekniikkaa voidaan verrata viivakodeihin, sillä kumpaakin tekniikkaa käytetään asioiden yksilöimiseen kohteeseen kiinnitettävän tunnisteen avulla. RFID-tekniikkaan verrattuna viivakodeihin liittyy kuitenkin monia niiden käyttöä rajoittavia tekijöitä.

Yksi suurimmista viivakoodien rajoitteista on, että niiden lukeminen tapahtuu näkyvän lukusäteen avulla ja edellyttää täten näköyhteyttä luettavaan tunnisteseen. Lisäksi usean viivakoodin yhtäaikainen lukeminen ei ole mahdollista, vaan lukeminen on suoritettava yksi kerrallaan. Viivakoodien sisältämän datan määrä on myös erittäin rajallinen, eikä sisältöä voi muuttaa enää tulostamisen jälkeen. Lisäksi viivakoodit ovat fyysisesti alttiita erilaisille häiriötekijöille, kuten esimerkiksi lialle.

RFID-tekniikka puolestaan mahdollistaa useiden tunnisteen yhtäaikaisen lukemisen ilman näkyvää lukusädettä. Lisäksi tunnisteen lukuetaisyys voi olla huomattavasti viivakoodien lukuetaisyyttä suurempi. RFID-tekniikassa myös tunnisteseen tallennettava tietomäärä on huomattavasti viivakoodia suurempi, minkä lisäksi tunnisteen tietosisältöä voidaan muokata jälkepäin. RFID-tunnisteet ovat myös vähemmän alttiita ulkoisille häiriötekijöille kuin viivakoodit. [21, s. 41.]

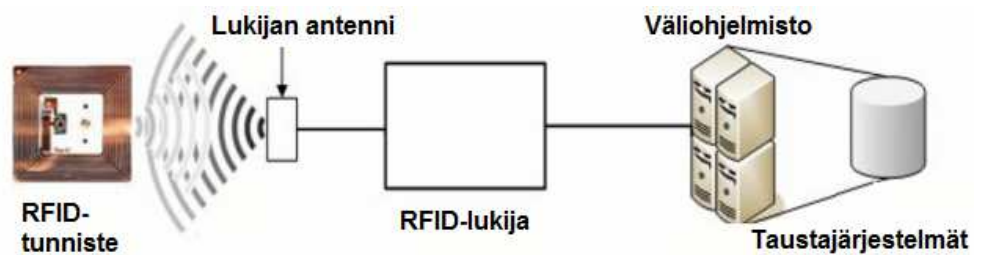
Kirjaston kokoelmiin kuuluvien niteiden yksilöimiseen ja tunnistamiseen on viimeisten vuosikymmenten aikana käytetty viivakoodia. Jokaiseen niteeseen on kiinnitetty niteen yksilöivä viivakoodi, jonka avulla fyysinen nide yhdistetään kirjastojärjestelmän aineistorekisterissä olevaan bibliografiseen dataan. Lainaustapahtumien yhteydessä niteen viivakoodi luetaan viivakoodilukijalla ja kirjastojärjestelmä tunnistaa käsiteltävän niteen ja osaa suorittaa sille tapahtuman vaatimat toimenpiteet. Viivakoodi ei kuitenkaan sisällä minikäänlaisia turvatoimintoja, jotka ehkäisivät aineiston luvaton vientiä kirjastosta.

Varkaudenestoon on perinteisesti käytetty elektromagneettista suojausta. Tässä suojausmenetelmässä jokaiseen niteeseen kiinnitetään elektromagneettinen suojanauha, jonka magnetoimiseen ja demagnetoimiseen suojaus perustuu. Kirjaston sisäänkäynneillä olevat elektromagneettiset hälytysportit havaitsevat, mikäli magnetoidun turvanauhan sisältävä nide yritetään kuljettaa porttien lävitse. Niteen viivakoodin lukemisen lisäksi on lainaustapahtumien yhteydessä siis myös suoritettava turvanauhan magnetointi tai demagnetointi tilanteesta riippuen.

RFID-tekniikkaa käyttävässä kirjastossa sekä niteen yksilöimiseen käytettävä id-numero että suojaustieto on tallennettu RFID-tunnisteelle. Lainaustapahtumien yhteydessä tunnisteiden lukeminen muuttaa myös tunnisteelle tallennettua hälytystietoa. Näin lainaustapahtuma yksinkertaistuu eikä niteeseen tarvitse kiinnittää erillistä suojanauhaa. Tämän perusteella kirjastojen RFID-sovellukset sisältävät sekä varkaudenestoon että omaisuuden seurantaan liittyviä toimintoja. [14, s. 1]

3.1 RFID-järjestelmän komponentit

RFID-järjestelmät koostuvat neljästä peruskomponentista: tunnisteesta, lukijasta, väliohjelmistosta ja taustajärjestelmästä. Järjestelmän toiminta perustuu RFID-tunnisteisiin tallennettujen tietojen langattomaan lukemiseen RFID-lukijalla. Lukijan avulla luettu tieto siirretään eteenpäin väliohjelmiston kautta taustajärjestelmään, jossa sitä voidaan käsitellä ja analysoida. RFID-järjestelmän perusrakenne on esitetty kuvassa 2. [23.]



Kuva 2. RFID-järjestelmän rakenne. [23.]

Kirjastosovelluksissa tunnisteita voidaan käyttää sekä aineiston että asiakkaiden tunnistamiseen. Tällä hetkellä yleisin toteutustapa on kuitenkin käyttää RFID-tunnisteita aineiston ja viivakoodeja asiakkaiden tunnistamisessa. Aineistoon kiinnitetyt tunnistemat käytetään samalla myös varkaudenestoon. Lukija voi puolestaan olla kiinnitetty väliohjelmiston sisältävään itsepalveluautomaattiin tai palvelupisteen työasemaan. Lukija voi olla myös itsenäinen käsiskanneri, jonka ei tarvitse olla kytkettynä työasemaan käytön aikana. Tällöin käsiskanneri itsessään sisältää lukijan lukemien tietojen vastaanottamisesta ja lukijan hallinnoimisesta vastaavan väliohjelmiston. Käytettävän lukijan tyyppistä riippumatta taustajärjestelmänä toimii kirjastojärjestelmä.

3.1.1 RFID-tunniste

RFID-tunniste koostuu vähintään kahdesta eri osasta: mikrosirusta ja antennista. Tunnisteen sisältämä tieto on tallennettuna mikrosirulle, josta on yhteys antenniin. Antennin tehtävänä on lukijan lähettämän signaalin vastaanottaminen ja tunnistella olevan tiedon lähettäminen lukijalle. [24, s. 20.]

Tunnisteiden ulkoasu vaihtelee tunnisteen ominaisuuksista, valmistajasta ja käyttötarkoituksista riippuen. Kirjastosovelluksissa käytettävät niteisiin kiinnitettävät tunnistemat rakentuvat yleensä niteeseen liimattavalle tarralle kiinnitystä mikrosirusta ja antennista [25]. Aalto-yliopiston Otaniemen kampuskirjastossa käytettävien RFID-tunnisteiden ulkoasu on nähtävillä kuvassa 3.



Kuva 3. Aalto-yliopiston Otaniemen kampuskirjaston käyttämä RFID-tunniste. [25.]

Lukijan ja tunnisteiden välinen kommunikointi perustuu takaisinsirontaan. Käytännössä tämä tarkoittaa sitä, että tunniste siroaa eli heijastaa osan lukijan lähettämästä signaalista takaisin lukijalle. Tämä on mahdollista mikrosirun sisältämän kondensaattorin ansiosta. Kondensaattori varautuu tunnisteiden vastaanottaessa lukijan lähettämän signaalin ja vastaavasti purkautuu, kun tunniste lähettää vastausignaalin lukijalle. [22, s. 38.]

RFID-tunnisteet voidaan jakaa kolmeen ryhmään ominaisuuksiensa perusteella. Nämä ryhmät ovat passiiviset, aktiiviset ja puolipassiiviset RFID-tunnisteet.

Passiiviset tunnisteet

Passiiviset tunnisteet eivät sisällä omaa virtalähdettä vaan vastauksen lähettämiseen tarvittava virta indusoituu lukijan lähettämästä signaalista. Tästä johtuen tunnisteiden lähettämä vastaus on yleensä melko lyhyt, tyypillisesti tunnisteelle tallennettu id-numero. Virtalähteen puuttumisesta johtuen passiiviset tunnisteet ovat yleensä kooltaan pieniä ja hinnaltaan muita tunnisteita edullisempia. Myös passiivisten tunnisteiden käyttöikä on virtalähteen puuttumisesta johtuen muita tunnistetyyppejä pidempi. Passiivisten tunnisteiden lukuetaisyudet vaihtelevat taajuudesta riippuen muutamista sentteistä viiteen metriin. Kirjastosovelluksissa käytetään pääsääntöisesti passiivisia tunnisteita, joiden lähettämä vastaus sisältää kirjastojärjestelmän aineistorekisteriin tallennetun niteen yksilöivän id-numeron. [22, s. 39; 26.]

Aktiiviset tunnisteeet

Aktiiviset tunnisteeet sisältävät mikrosirun ja antennin lisäksi oman virtalähteen, jonka ansiosta ne voivat itsenäisesti lähettää signaalia. Käytännössä tämä tarkoittaa sitä, että aktiiviset tunnisteeet voivat lähettää dataa myös ilman lukijan lähettämää signaalia. Oman virtalähteen ansiosta lukuetaisyys ja muistin koko ovat passiivisia tunnisteeita suurempia, mutta tunnisteeiden käyttöikä on myös rajallinen. Aktiiviset tunnisteeet ovat fyysiseltä kooltaan suurempia sekä hinnaltaan kalliimpia kuin passiiviset tunnisteeet. Aktiivisten tunnisteeiden lukuetaisyys voi olla jopa useita kymmeniä metrejä. [22, s. 40; 26.]

Puolipassiiviset tunnisteeet

Puolipassiiviset tunnisteeet sisältävät oman virtalähteen, mutta vaativat kuitenkin lukijan lähettämän herätteen aloittamaan datan lähetyksen. Lähetykseen tarvittava virta otetaan tunnisteen omasta virtalähteestä, jonka johdosta tunnisteen käyttöikä on rajallinen. Oman virtalähteensä ansiosta puolipassiivisten tunnisteeiden lukuetaisyys sekä muistikapasiteetti ovat passiivisia tunnisteeita suurempia. [22, s. 40.]

Muut ominaisuudet

RFID-tunnisteeiden tallennuskapasiteetti vaihtelee valmistajan ja käyttötarkoituksen mukaan. Yleisimpiä ovat 64, 96, 128, 256 tai 512 bittiä sisältävät tunnisteeet [24, s. 39]. Tallennuskapasiteetti vaikuttaa suoraan tunnisteen fyysiseen kokoon ja hintaan. Tallennuskapasiteetin kasvaessa myös tunnisteen fyysinen koko ja hinta nousevat. Kirjastosovelluksissa käytetään nykyisin tunnisteeita, joiden tallennuskapasiteetti on 1024 tai 2048 bittiä.

Tunnisteeiden sisältämän muistin toteutuksessa on valmistajakohtaisia eroja, mutta yleisesti ottaen muisti voidaan jakaa kahteen osaan. Suurin osa muistikapasiteetista on tarkoitettu sovellusten käyttämälle datalle, jonka lisäksi osa muistia on varattu tunnisteen käyttöön liittyville konfigurointitiedoille. Tähän osioon on tallennettu muun muassa lainaus- ja palautustapah- tumien yhteydessä muutettava tunnisteen suojaustieto.

Tunnisteeet eroavat toisistaan myös sisältämiensä luku- ja kirjoitusominaisuuksien osalta. Yksinkertaisimmillaan tunnistee voi sisältää pelkän lukuomi-

naisuuden. Tällöin tunniste sisältää vain sille valmistusvaiheessa tallennetun datan, jonka muuttaminen jälkeenpäin ei ole mahdollista. Osa tunnisteista mahdollistaa puolestaan tietojen lukemisen lisäksi yhden kirjoitusoperaation. Käytännössä tämä tarkoittaa sitä, että tunnisteelle tulevaa dataa ei tällöin tarvitse kirjoittaa vielä valmistusprosessin aikana. Tämä tekee tunnisteista pelkällä lukuominaisuudella varustettuja tunnisteita joustavampia. Monipuolisimmat tunnisteet eivät puolestaan rajoita luku- ja kirjoitusoperaatioiden määrää, vaan tunnisteiden sisältöä voidaan lukea ja kirjoittaa rajoituksitta. Tällaisten tunnisteiden käyttö on erittäin joustavaa, mutta sisältää myös tietojen muokkaamiseen ja ylikirjoittamiseen liittyviä riskejä. Tunnisteet mahdollistavat myös niille kirjoitetun datan lukitsemisen, jonka jälkeen niiden sisältöä ei voi enää muokata. Lukitseminen on mahdollista toteuttaa sekä koko tunnisteelle että vain tietylle osalle tunnistetta. Kirjastosovelluksissa käytetyt tunnisteet ovat pääsääntöisesti uudelleenkirjoitettavia tunnisteita, joiden tietosisältöä ei ole lukittu. [22, s. 42.]

RFID-tunnisteiden eri ominaisuudet ja niiden yhdisteleminen mahdollistavat tunnisteiden käytön hyvin erilaisissa toimintaympäristöissä ja järjestelmissä. RFID-järjestelmää suunniteltaessa onkin kiinnitettävä erityistä huomiota siihen, että käytettävät tunnisteet soveltuvat parhaiten juuri kyseisen järjestelmän tarpeisiin. Kirjastojen RFID-järjestelmissä käytetään niteiden yksilöimiseen pääasiassa passiivisia 13,56 MHz:n taajuudella toimivia 1024 - 2048 bitin muistikapasiteetin omaavia uudelleenkirjoitettavia tunnisteita, joiden tietosisältöä ei ole lukittu.

Tietosisältömallit

Saman RFID-järjestelmän piiriin kuuluvien tunnisteiden on noudatettava yhtenäistä tallennusmuotoa, joka määrää, mitä tietoja ja missä järjestyksessä tunnisteelle tallennetaan. Tallennusmuoto voi olla määritelty järjestelmän tai organisaation sisällä vain kyseistä järjestelmää varten tai se voi olla ulkopuolisen organisaation määrittelemä yleisessä käytössä oleva tietosisältömalli. Yleisessä käytössä olevan tietosisältömallin käyttämisen kiistattomana etuna on se, että tunnisteet ovat yhteensopivia kaikkien kyseistä mallia tukevien laitteiden ja järjestelmien kanssa. Tämän myötä järjestelmä ei ole sidottu vain tietyn laitevalmistajan tai ohjelmistotoimittajan tuotteisiin, mikä mahdollistaa järjestelmän yksittäisten osien vaihtamisen ilman kaikkien järjestelmään kuuluvien tunnisteiden uusimista.

Tunnisteelle tallennettavien tietojen määrä ja tarkkuus voi vaihdella suuresti eri toteutusten välillä. Yksinkertaisimmillaan tunnisteelle tallennetaan pelkkä niteen yksilöivä id-numero, jolloin muut niteeseen liittyvät tiedot on haettava kirjastojärjestelmästä id-numeron perusteella. Toisena ääripäänä on, että tunniste sisältää lainaus- ja varaustietoja lukuun ottamatta kaikki niteeseen liittyvät tiedot. Tämä puolestaan vähentää lukijan ja kirjastojärjestelmän välisen kommunikoinnin tarvetta ja voi tietyissä tilanteissa nopeuttaa järjestelmän toimintaa. Toisaalta tunnisteelle tallennetun tietomäärän kasvaessa tunnisteiden lukeminen hidastuu, mikä näkyy useiden tunnisteiden yhtäaikaista lukua vaativien toimintojen hidastumisena. Käytännössä tietosisältömallit eivät kuitenkaan yleensä edusta kumpaakaan ääripäätä, vaan sijoituvat niiden välille. [27.]

Niteisiin kiinnitettäviin tunnisteisiin tallennettavan datan osalta suomalaiset kirjastot noudattavat pääasiassa kansalliskirjaston tietopalvelualan verkko-standardityöryhmä KATVE:n vuonna 2005 julkaisemaa tietosisältömallia. Malli jakaa tunnisteiden tietosisällön kolmeen osaan: pakolliseen osaan sekä strukturoituun ja strukturoimattomaan lisäosaan, joiden käyttö on vapaaehtoisia. Tietosisältömalli suosittaa pakollisen osan lukitsemista turvallisuussyistä, mutta käytännön toteutuksissa lukitusta ei juurikaan käytetä. [28.]

Pakollinen osa sisältää muun muassa niteen yksilöivän id-numeron, jonka avulla nide pystytään yhdistämään kirjastojärjestelmän aineistorekisterissä olevaan tietueeseen. Lisäksi pakollisessa osassa on myös omistajakirjastoon liittyviä tietoja, kuten esimerkiksi maa- ja kirjastotunnus. Strukturoituun lisäosaan voidaan sijoittaa erilaisia lisätietoja, joiden avulla voidaan esimerkiksi ohjata itsepalvelulaitteiden toimintaa. Strukturoituun lisäosaan voidaan esimerkiksi sijoittaa niteen aineistotyyppi, jonka perusteella itsepalvelulaite pystyy tunnistamaan helposti särkyvän aineiston ja osaa käsitellä aineistoa muusta aineistosta poikkeavalla tavalla. Strukturoimattomaan lisäosaan sijoitetaan puolestaan kansainvälisestä käytännöstä poikkeavat määritykset. [28.]

3.1.2 *RFID-lukija*

RFID-lukijan tehtävänä on tiedon lukeminen ja kirjoittaminen tunnisteisiin. Tämä on mahdollista lukijan sisältämän radiolähetinvastaanottimen ansiosta. Muodostamalla sähkömagneettisen kentän lukija tuottaa passiivisten ja puolipassiivisten tunnisteiden tiedon lähettämiseen tarvitseman energian. Säh-

kömagneettisen kentän muodostaminen tapahtuu lukijan sisältämän antennin avulla. Lukija ottaa vastaan ja prosessoi tunnisteidien lähettämät tiedot, jonka jälkeen tiedot siirretään taustajärjestelmään. [22, s. 43.]

RFID-lukijan fyysinen ulkoasu voi vaihdella hyvinkin suuresti eri käyttötarkoitusten ja järjestelmien välillä. Lukijan sisältämä tekniikka ei itsessään vaadi paljoakaan tilaa, minkä johdosta lukulaite on helppo rakentaa juuri tiettyyn käyttötarkoitukseen sopivaksi. Antenni voi esimerkiksi sisältyä lukulaitteeseen tai olla sijoitettu lukulaitteesta erilleen. Kirjastosovelluksissa käytettävät yleisimmät lukulaitteet ovat kirjastojen sisäänkäynneille sijoitettuja hälytysportteja, asiakaspalvelussa ja itsepalvelulaitteissa käytettäviä levymuotoisia antennoja tai muun muassa kokoelmien inventointiin käytettäviä käsiskannereita. Kuvassa 4 on nähtävillä 3M:n Kirjastoassistentti-nimellä tunnettu käsiskanneri.



Kuva 4. 3M:n Kirjastoassistentti. [29.]

Kirjastosovelluksissa käytettävät lukulaitteet eroavat ulkoasunsa lisäksi toisistaan myös sen suhteen, miten ne ovat yhteydessä taustajärjestelmänä toimivaan kirjastojärjestelmään. Asiakaspalvelussa ja itsepalvelulaitteissa käytettävät lukijat välittävät pääsääntöisesti tietonsa väliohjelmiston kautta kirjastojärjestelmälle. Vanhemmat hälytysportit sen sijaan keskittyvät pelkäästään tunnistelle tallennetun suojaustiedon lukemiseen ja hälytyksen laukaisemiseen tarvittaessa eivätkä täten ole minkäänlaisessa yhteydessä kirjastojärjestelmään. Uusimmat hälytysportit toimivat suojaustiedon osalta täysin itsenäisesti, minkä lisäksi ne pystyvät myös tallentamaan tietoja hälytyksen

laukaisseista tunnisteista. Portit ovat yhteydessä palvelimeen, jolla sijaitsevaan tietokantaan viedään hälytyksen laukaisseelle tunnisteelle tallennettu niteen id-numero. Id-numeron perusteella palvelimelle sijoitettu ohjelmisto pystyy hakemaan niteen tarkemmat tiedot kirjastojärjestelmän aineistorekisteristä SIP2-protokollan avulla. Tällä tavoin toimivien porttien avulla pystytään esimerkiksi saamaan selville sellaisten teosten tiedot, jotka on kuljetettu hälytysporttien läpi ilman hälytyksen passivointia.

Kirjastoissa käytettävät käsiskannerit eivät yleensä ole käytön aikana suoraan yhteydessä kirjastojärjestelmään, vaan niiden tarvitsemat tiedot on erikseen ladattava kirjastojärjestelmästä käsiskannerin muistiin. Ladattavien tietojen määrä riippuu pitkälti siitä, kuinka paljon niteeseen liittyvää tietoa tunnisteelle on tallennettu ja mitä toimenpiteitä käsiskannerin avulla halutaan suorittaa. Käytännössä käsiskanneri toimii siten, että esimerkiksi kokoelmia inventoitaessa käsiskannerin muistiin ladataan kirjastojärjestelmästä inventoitavaan kokoelmaan kuuluvien niteiden id-numerot, jotka on tallennettu myös niteisiin kiinnitettyihin RFID-tunnisteisiin. Tämän jälkeen kyseiseen kokoelmaan kuuluvien niteiden tunnisteet luetaan läpi, jonka päätteeksi käsiskannerista saadaan tieto puuttuvista niteistä. Käsiskannerin toiminta perustuu tässä tapauksessa luettujen tunnisteiden id-numeroiden ja kirjastojärjestelmästä ladatun listan vertailemiseen toisiinsa. Samalla tavalla käsiskanneria voidaan käyttää myös esimerkiksi hyllyjärjestyksen tarkistamiseen tai varattujen niteiden poimimiseen hyllystä. Samoja tehtäviä pystyttäisiin toki suorittamaan myös viivakoodeilla varustetuille niteille kannettavan tietokoneen ja viivakoodinlukijan avulla, mutta tämä olisi huomattavasti työläämpää ja pelkästään siitä syytä, että jokainen nide olisi vedettävä pois hyllystä viivakoodin lukemista varten. RFID-käsiskanneria tarvitsee sen sijaan vain kuljettaa riittävän lähellä luettavien kirjojen selkämystä, jolloin se pystyy lukemaan niteisiin kiinnitettyt tunnisteet.

Lukijoita on lisäksi mahdollista kiinnittää myös esimerkiksi suoraan kirjahyllyihin [14, s. 33]. Tällaista toteutusta voidaan hyödyntää monillakin eri tavoilla. Hylly voi toimia esimerkiksi palautushyllynä, jolloin lainatun niteen palauttamiseksi ei tarvitse tehdä muuta kuin asettaa se palautushyllyyn, jolloin lukija lukee niteessä olevalle tunnisteelle tallennetun id-numeron ja välittää sen kirjastojärjestelmälle tapahtumarekisterin päivittämistä varten. Jos kaikki kir-

jaston hyllyt olisi varustettu RFID-lukijoilla, olisi puolestaan mahdollista saada täysin reaaliaikaista tietoa hyllyissä olevista niteistä [14, s. 33].

Perinteisestä viivakoodinlukijasta poiketen RFID-lukija pystyy lukemaan useita tunnisteita yhtä aikaa. Kirjastosovelluksissa tätä ominaisuutta hyödynnetään niin sanotussa pinolainauksessa. Pinolainauksen toimintaperiaate on, että lukijan kantaman piiriin tuodaan pino niteitä, joiden sisältämät tunnisteet lukija lukee yhdellä kertaa. Tämä mahdollistaa kirjapinojen lainaamisen ja palauttamisen ilman, että jokaista nidettä tarvitsee käsitellä erikseen, kuten viivakoodeja käyttävässä järjestelmässä. Pinolainausta voidaan hyödyntää niin asiakaspalvelussa kuin itsepalvelulaitteissa sekä myös aineiston inventoinnissa.

Lukijan ja tunnisteiden välisen tiedonsiirron yksityiskohdat riippuvat monista eri tekijöistä, kuten esimerkiksi lukijan ja tunnisteiden antennien koosta, rakenteesta ja sijoittelusta sekä lähetystehosta. Koska RFID-lukija ja tunniste kommunikoivat tietyllä radiotaajuudella, myös käytettävä taajuusalue vaikuttaa suuresti koko RFID-järjestelmän toimintaan. [27.]

3.1.3 Taajuusalueet

RFID-tunnisteet ja lukijat tuottavat ja heijastavat elektromagneettisia aaltoja, minkä johdosta ne luokitellaan radiolaitteiksi. Yleisesti ottaen RFID-sovellukset käyttävät niin sanottuja ISM (Industrial-Scientific-Medical)-taajuusalueita, jotka on varattu teollisten, tieteellisten ja lääketieteellisten sovellusten käyttöön. Taajuusalueiden käyttöä rajoittamalla varmistetaan, että RFID-sovellukset eivät häiritse muiden radiotaajuuksia käyttävien sovellusten, kuten esimerkiksi television, radion ja matkapuhelinten toimintaa. [30, s. 111.]

Käytettävä taajuusalue on yksi merkittävimmistä lukijan ja tunnisteiden väliseen kommunikaatioon vaikuttavista tekijöistä. Taajuusalue vaikuttaa useisiin RFID-järjestelmän ominaisuuksiin, kuten esimerkiksi tunnisteiden ja lukijan väliseen lukuetaisyteen ja tiedonsiirtonopeuteen sekä järjestelmän häiriöherkkyyteen. Passiivisten tunnisteiden käyttämät taajuusalueet ja keskimääräiset lukuetaisyydet on esitetty taulukossa 1. [22, s. 44.]

Taulukko 1. Passiivisten tunnisteiden yleisimmät taajuusalueet ja lukuetaisyydet RFID-sovelluksissa. [26.]

Taajuusalue	Nimi	Lukuetaisyys
100 KHz - 250 KHz	LF (Low Frequency)	< 0,2 m
13,56 MHz	HF (High Frequency)	< 1,5 m
865 MHz - 950 MHz	UHF (Ultra High Frequency)	~ 5,0 m

Kirjastosovellusten käyttämissä taajuusalueissa on jonkin verran maa- ja maanosakohtaisia eroja, mutta pääsääntöisesti järjestelmät toimivat kuitenkin joko HF- tai UHF-taajuusalueella. HF-taajuusalueen käyttö on yleistä Euroopassa ja Pohjois-Amerikassa, kun Aasiassa ja Australiassa käytetään myös UHF-taajuusaluetta. [27.]

Yleisesti ottaen voidaan todeta, että taajuusalue määrää lukijan ja tunnisteiden välisen tiedonsiirtonopeuden siten, että alhaisempi taajuus tarkoittaa alhaisempaa tiedonsiirtonopeutta. Alhaisemmilla taajuuksilla toimivien järjestelmien lukuetaisyydet ovat tyypillisesti lyhyitä ja häiriönsietokyky kohtuullisen hyvä. Korkeammilla taajuuksilla toimivat järjestelmät omaavat puolestaan suuremman lukuetaisyyden, mutta ovat herkempiä erilaisille ulkoisille häiriöille. Lisäksi korkeampia taajuuksia käyttävien järjestelmien antennit ja tunnisteet ovat yleensä fyysiseltä kooltaan matalia taajuuksia käyttävien järjestelmien antennejä ja tunnisteita pienempiä. [21, s. 63.]

3.2 Standardit

RFID-tekniikkaa hyödynnetään monissa erilaisissa sovelluksissa ja käyttöympäristöissä, minkä vuoksi siihen liittyy myös monia eri standardeja. Standardien julkaisemisesta vastaavat pääasiassa kansainvälinen ISO-standardisointijärjestö sekä EPCglobal-organisaatio. Osa standardeista on RFID-tekniikkaa yleisesti koskevia ja osa suoraan tiettyihin sovellusalueisiin liittyviä. Karkeasti ottaen tekniikkaan liittyvät standardit voidaan kuitenkin jakaa tiedonvälitysprotokollan ja tunnisteiden tietosisällön määritteleviin standardeihin. [31, s. 23.]

Kirjastojen RFID-sovellusten kannalta keskeisimpiä standardeja ovat ISO-standardit 15693 ja 18000. ISO 15693 -standardi määrittelee kirjastosovelluksissa yleisesti käytettävien tunnisteiden fyysiset ominaisuudet, ilmarajapinnan, törmäykseneston sekä lukijan ja tunnisteiden välisen tiedonsiirron periaatteet. Suomalaisten kirjastojen tällä hetkellä käyttämä KATVE-tietosisältömalli edellyttää, että käytettävät tunnisteet ovat ISO 15693 -standardin mukaisia. ISO 18000 -sarjassa on puolestaan seitsemän osaa, mitkä määrittelevät lukijan ja tunnisteiden välisen tiedonvälitysprotokollan eri taajuusalueille. Sarjan kolmas osa käsittelee kirjastosovellusten käyttämää 13,56 MHz:n taajuusaluetta. ISO 15693 -standardin mukaiset tunnisteet ovat yhteensopivia ISO 18000 -standardin kanssa. Käytännössä ISO 18000 -standardin mukaan toteutetuissa järjestelmissä on siis teknisesti mahdollista käyttää kumpaa tahansa standardia noudattavia tunnisteita. [32.]

Vielä työn alla oleva ISO 28560 -standardi määrittelee tietosisältömallin ni-teissä käytettävillä tunnisteilla. Standardin tarkoituksena on varmistaa yhteensopivuus eri laitteiden ja järjestelmien välillä. Suomessa käytettävällä KATVE-tietosisältömallilla ja ISO 28560 -standardilla on monia yhtymäkoh-tia, minkä ansiosta KATVE-tietosisältömallia käyttävät kirjastot voivat halu-tessaan alkaa tukemaan uutta standardia joutumatta tekemään suuria muu-toksia RFID-järjestelmiinsä. [28.]

Suoraan RFID-tekniikkaan liittyvien standardien lisäksi RFID-sovelluksissa on huomioitava myös muiden sovellusalueiden standardeja. Yksi tällaisista standardeista on RFID-sovelluksissa käytettävät kirjoitusmerkit määrittelevä ISO/IEC 10646 -merkistöstandardi [33]. RFID-tekniikkaa käyttävät itse-palvelulaitteet tukevat puolestaan yleensä 3M:n kehittämää SIP-protokollaa, jonka avulla ne pystyvät kommunikoimaan kirjastojärjestelmän kanssa [14, s. 25].

3.3 Tietoturva

Langattoman tiedonvälityksensä takia RFID-tekniikka altistuu monille ylei-sesti langattomaan tekniikkaan liittyville uhille, kuten esimerkiksi salakuunte-lulle, kopio- ja palvelunestohyökkäyksille sekä yhteyden häirinnälle. Lisäksi RFID-tekniikan tietoturvaan vaikuttaa myös se, että tunnisteiden sisältämät tiedot ovat periaatteessa kaikkien RFID-lukijoiden luettavissa. Tietojen luke-minen tapahtuu ilman fyysistä kontaktia, joten asiattomat tahot voivat hel-

posti päästä lukemaan tunnisteiden tietoja kenenkään huomaamatta. [34, s. 65.]

RFID-tekniikan sovellusalue on erittäin laaja ja riskien kriittisyys sekä tyyppi vaihtelevat suurestikin eri sovellusalueiden välillä. Järjestelmään liittyviin uhkatekijöihin vaikuttavat monet erilaiset muuttujat, jotka tulee ottaa huomioon yksittäisen järjestelmän turvallisuutta ja siihen liittyviä riskitekijöitä arvioitaessa. Erityisen kriittisiä alueita ovat maksamiseen ja henkilötietojen käsittelyyn tarkoitetut sovellukset. [35, s. 209.]

Kirjastosovelluksiin kohdistuvat periaatteessa samat uhkatekijät kuin RFID-järjestelmiin yleisestikin. Tämänhetkisiin kirjastosovelluksiin liittyvien uhkatekijöiden aiheuttamat riskit eivät kuitenkaan ole luonteeltaan kovinkaan kriittisiä. Käytännössä suurimpia uhkatekijöitä ovat niteisiin kiinnitettyjen tunnisteiden tietojen turmeleminen ja tunnisteiden luvaton lukeminen.

Luvaton lukeminen

Luvaton lukeminen tarkoittaa tunnisteelle tallennetun tiedon lukemista ilman tunnisteiden omistajan suostumusta. Tällä tavoin voitaisiin esimerkiksi saada selville yksittäisen kadulla kävelevän ihmisen laukussa olevien kirjastosta lainattujen niteiden tiedot. Käytännössä niteisiin kiinnitettyjen tunnisteiden luvaton lukeminen ei kuitenkaan aiheuta suurta tietoturvariskiä, sillä tunnisteelle tallennetun niteen yksilöivän id-numeron lukeminen on hyödytöntä ilman pääsyä kirjastojärjestelmän aineistorekisteriin. Id-numeron avulla aineistorekisteristä voidaan noutaa niteen bibliografiset tiedot, jotka ovat muutenkin julkisesti kaikkien nähtävillä kirjaston kokoelmaluettelon kautta. Id-numero ei kuitenkaan ole nähtävillä kokoelmaluettelossa, joten tiettyä id-numeroa vastaavien tietojen selvittäminen vaatii suoraa pääsyä kirjastojärjestelmän aineistorekisteriin. [27.]

Tarvittaessa tunnisteiden lukemista on mahdollista rajoittaa luvattoman lukemisen estämiseksi. Tunnisteita voidaan käyttää ns. privacy mode -tilassa, jolloin tunnisteiden lukeminen on suojattu salasanalla. Tällöin lukija lähettää tunnisteiden lukemiseksi tarvittavan salasanan kommunikointipyynnön yhteydessä. Tunniste lähettää vastauksena sen sisältämät tiedot vain siinä tapauksessa, että lukijan lähettämä salasana on oikein. [27.]

Tietojen turmeleminen

Tietojen turmeleminen tarkoittaa tunnisteele tallennetun tiedon luvaton muokkaamista tai poistamista. Niteissä käytettävien tunnisteen tietosisältöä ei yleensä ole lukittu, joten periaatteessa kuka tahansa sopivan laitteiston omaava henkilö pystyy muuttamaan tunnisteesiin tallennettuja tietoja. Tietojen muokkaamisella ei kuitenkaan ole mahdollista saavuttaa mitään käytännön hyötyä, koska keskeisin tunnisteele tallennettu tieto on niteen yksilöivä id-numero. Pahimmassa tapauksessa id-numero olisi mahdollista korvata toisen niteen id-numerolla, jolloin lainaustilanteessa niteen lainanneen asiakkaan lainaustietoihin kirjautuisi väärän niteen tiedot. Jo pelkän id-numeron muuttaminen tunnisteen muuta tietosisältöä vahingoittamatta edellyttäisi kuitenkin kirjaston käyttämän tietosisältömallin täydellistä tuntemista. Tunnisteele tallennettujen tietojen muuttaminen ilman tietosisältömallin tuntemista aiheuttaa sen sijaan tunnisteen tietorakenteen rikkoutumisen, jonka jälkeen sen tietosisältö ei ole enää kirjaston RFID-järjestelmän ymmärtämässä muodossa.

Tunnisteen tietojen luvaton muokkaaminen on mahdollista estää lukitsemalla tunnisteele tallennetut tiedot. Lukitseminen on mahdollista tehdä koko tunnisteele tai vain osalle tunnistetta. Lisäksi lukitus on mahdollista tehdä pysyvästi tai se voi olla purettavissa salasanan avulla. Tällöin tunnisteele tallennettujen tietojen muokkaaminen toimii samalla periaatteella kuin salasanalla suojattujen tunnisteen lukeminen. [27.]

Tunnisteen sisältämät tiedot on mahdollista suojata myös salausta käyttämällä [14, s. 59]. Salauksen toteuttamiseen on olemassa useita eri käytännön tekniikoita, mutta pääperiaate on kuitenkin kaikissa toteutuksissa osapuilleen sama. Tunnisteele tallennetut tiedot on salattu, eikä niiden muuntaminen selkokieliiseen muotoon onnistu ilman salaamiseen käytettyä salausavainta tuntevaa lukijaa.

Kirjastojen RFID-sovelluksissa käytettävät tunnistetyypit ehkäisevät omalta osaltaan tunnisteen luvaton lukemista ja niiden sisältämien tietojen turmelemista. Kirjastosovelluksissa käytetään pääasiassa passiivisia tunnisteesia, joiden lukuetaisyys on alle metrin. Käytännössä tämä tekee tunnisteen luvattomasta lukemisesta erittäin hankalaa ja kasvattaa kiinnijäämisen riskiä huomattavasti. [14, s.57.]

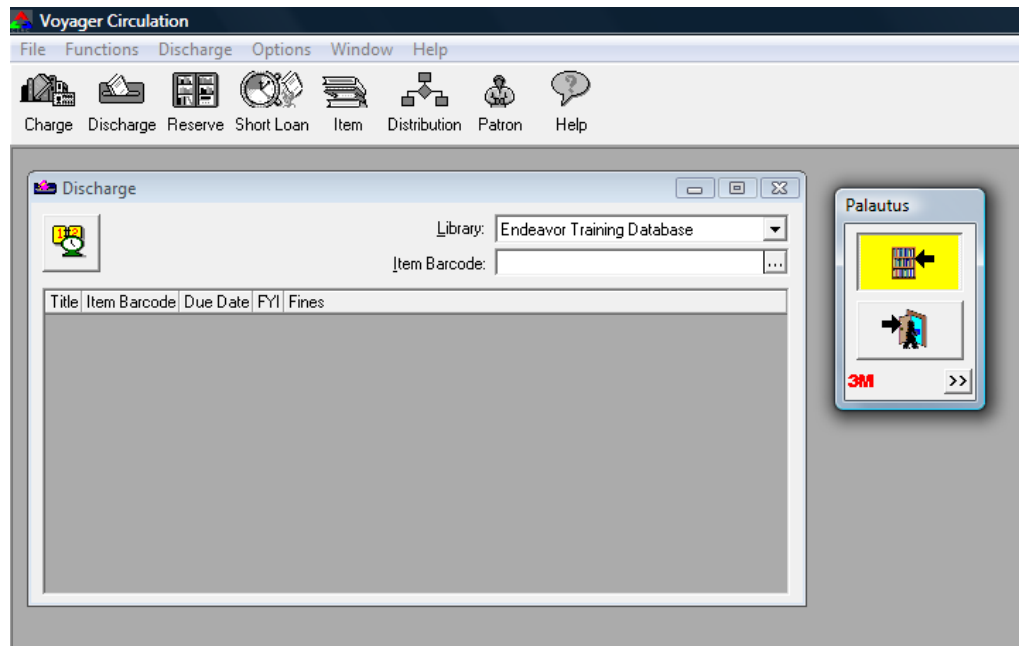
Yhteenvetona voidaan todeta, että niteisiin kiinnitettäviin tunnisteisiin ei liity oleellisia tietoturvariskejä, koska niiden sisältämä tieto on merkityksetöntä ilman pääsyä kirjastojärjestelmän aineistorekisteriin. Lisäksi tunnisteille tallennetut id-numerot ovat kirjastokohtaisia ja täten käyttökelpoisia vain niteen kotikirjaston kirjastojärjestelmässä [26]. Kirjastojen RFID-järjestelmien tietoturva ajatellen tärkeimmäksi tekijäksi nousee taustajärjestelmänä toimivan kirjastojärjestelmän tietoturva.

4 RFID-TEKNIIKAN KÄYTÖN TEHOSTAMINEN

RFID-tekniikkaa on käytetty kirjastoissa eri puolilla maailmaa jo yli kymmenen vuoden ajan. Valtaosa käytössä olevista kirjastojärjestelmistä mahdollistaa RFID-tekniikan käytön, vaikka ei suoranaisesti sisälläkään RFID-tekniikkaa tukevia ominaisuuksia.

RFID-lukijan hallinnoimisesta vastaavan väliohjelmiston ja kirjastojärjestelmän välinen kommunikaatio rajoittuu siihen, että väliohjelmisto välittää lukijan tunnisteelta lukeman niteen id-numeron kirjastojärjestelmälle viivakoodinlukijan tapaan. Id-numeron lisäksi tunnisteelle on kuitenkin tallennettu myös niteen suojaustieto, jota RFID-lukija voi muuttaa suoritetusta toiminnosta riippuen. Niteen lainauksen yhteydessä suojaustieto passivoidaan ja palautuksen yhteydessä vastaavasti aktivoidaan. Lisäksi id-numero pitää myös pystyä lukemaan suojaustietoa muuttamatta. Ongelmaksi muodostuu kirjastojärjestelmän ja väliohjelmiston välinen yksisuuntainen kommunikaatio. Väliohjelmisto välittää antennin lukeman id-numeron kirjastojärjestelmälle, mutta kirjastojärjestelmä ei välitä mitään tietoja väliohjelmistolle. Tämä on erittäin ongelmallista, koska väliohjelmiston on tiedettävä, mitä toimintoa kirjastojärjestelmässä ollaan suorittamassa, jotta se osaa ohjata lukijaa toiminnon edellyttämällä tavalla.

Kirjastojärjestelmän ja väliohjelmiston väliseen kommunikointiin liittyvät puutteet on pääsääntöisesti ratkaistu siten, että käyttäjä välittää tiedon kirjastojärjestelmässä suoritetusta toiminnosta väliohjelmistolle. Käytännössä tämä tapahtuu siten, että käyttäjä valitsee suoritettavan toiminnon sekä kirjastojärjestelmän lainausmoduulista että väliohjelmistoa edustavasta ohjelmasta. Tällöin väliohjelmiston tilan ylläpitäminen jää täysin käyttäjän vastuulle. Kuvassa 5 on esitetty näkymä Voyager-kirjastojärjestelmän lainausmoduulin ja 3M Staff Pad Workstation -väliohjelmiston yhteiskäytöstä.



Kuva 5. Voyager-kirjastojärjestelmän lainausmoduuli ja 3M Staff Pad Workstation -väliohjelmisto yhteiskäytössä.

Edellä kuvattu ratkaisu on ongelmallinen, koska se edellyttää kahden ohjelman rinnakkaista käyttöä, jonka myötä se on varsin altis erilaisille virheille. Tyypillinen virhetilanne syntyy, kun käyttäjä vaihtaa suoritettavaa toimintoa kirjastojärjestelmässä eikä muista tehdä samaa muutosta myös väliohjelmistossa. Kyseinen virhe ilmenee esimerkiksi hälytyksen laukeamisena asiakkaan kuljettaessa lainaamia niteitä hälytysporttien läpi. Niteet ovat kirjautuneet oikein lainoiksi kirjastojärjestelmän tapahtumarekisteriin, mutta tunnistajien hälytystieto on jäänyt passiivomatta väliohjelmiston väärän tilan vuoksi. Vastaavilta virhetilanteilta vältyttäisiin, jos väliohjelmisto osaisi seurata kirjastojärjestelmän tilassa tapahtuvia muutoksia automaattisesti ilman käyttäjältä vaadittavia toimenpiteitä.

4.1 Määrittely

RFID Plugin -sovelluksen tarkoituksena on automatisoida kirjastojärjestelmän tilan välittäminen väliohjelmistolle. Käytännössä sovellus välittää tiedon kirjastojärjestelmän lainausmoduulin tapahtumista väliohjelmistolle, joka muuttaa RFID-lukijan tilan lainausmoduulin tilaa vastaavaksi. Sovellus tarkkailee kirjastojärjestelmän lainausmoduulin tilaa ja siinä tapahtuvia muutoksia lainausmoduulin ollessa aktiivinen. Sovellus tunnistaa lainaus- ja palautustoiminnon aloittamisen ja lopettamisen sekä toiminnon suorittamisen ai-

kana mahdollisesti avautuvat varoitusikkunat. Sovellus tunnistaa myös toiminnot, joissa halutaan lukea tunnisteiden sisältö suojaustietoa muuttamatta. Lisäksi sovellus tukee moniosaisten tunnisteiden käsittelyä, jossa kontrolli vaihtelee kirjastojärjestelmän ja väliohjelmiston välillä tapahtuman aikana.

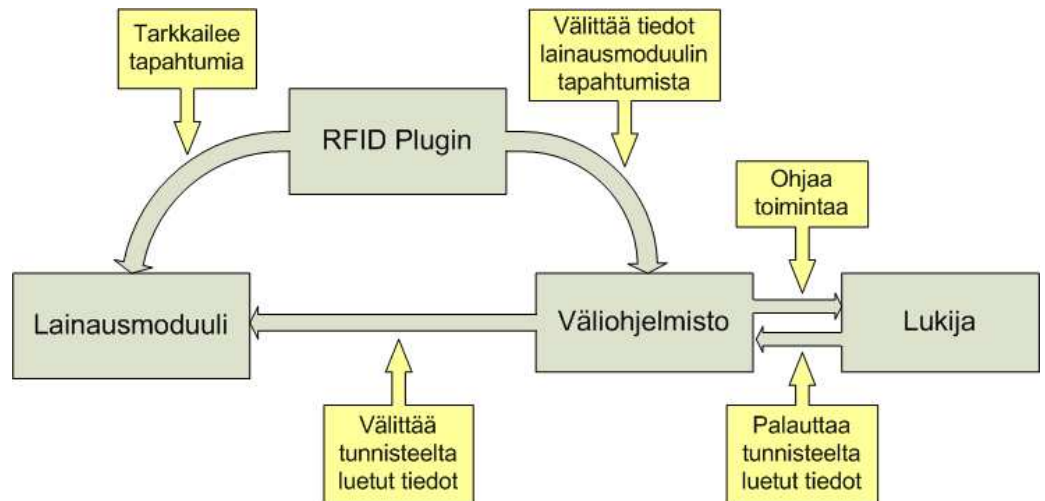
Sovelluksen toiminta on siinä mielessä käyttäjälle näkymätöntä, että sovellus toimii täysin itsenäisesti ilman käyttäjän antamia syötteitä. Käyttäjän ei myöskään tarvitse huolehtia sovelluksen käynnistämisestä tai sulkemisesta. Sovellukseen on mahdollista määrittellä pikanäppäimiä, joiden avulla voidaan avata uusia ikkunoita ja vaihtaa aktiivista ikkunaa lainausmoduulin sisällä.

Sovellus on yhteensopiva Voyager-, PallasPro-, ATP Origo- ja PrettyLib-kirjastojärjestelmien sekä 3M Pad Staff Workstation- ja Bookmatic RFID Staff -väliohjelmistojen kanssa. Lisäksi sovellus on yhteensopiva Windows XP- ja Windows 7 -käyttöjärjestelmien kanssa.

Sovelluksen toimintaan liittyviä asetuksia on mahdollista muuttaa ilman lähdekoodin muokkaamista. Käytettävä kirjastojärjestelmä ja väliohjelmisto voidaan määrittellä asennuskohtaisesti ja määrityksiä on mahdollista muuttaa myös asennuksen jälkeen. Sovelluksen asentaminen tapahtuu samoja asennustiedostoja käyttäen kirjastojärjestelmästä ja väliohjelmistosta riippumatta.

4.2 Toimintaperiaate

RFID Plugin -sovelluksen toiminta perustuu kirjastojärjestelmän lainausmoduulin tapahtumien tarkkailuun ja niiden välittämiseen lukijaa ohjaavalle väliohjelmistolle. Väliohjelmistolta saamiensa komentojen mukaisesti lukija suorittaa tunnisteelle lainausmoduulissa suoritettua toimintoa vastaavan toimenpiteen ja palauttaa tunnisteiden sisältämän id-numeron väliohjelmistolle, joka välittää sen edelleen lainausmoduulille. Sovelluksen toimintaperiaate on esitetty kuvassa 6.



Kuva 6. RFID Plugin -sovelluksen toimintaperiaate.

RFID Plugin -sovellus on riippuvainen sekä kirjastojärjestelmän lainausmoduulista että väliohjelmistosta, eikä sen käyttö ole mahdollista ilman niitä. Lainausmoduuli ja väliohjelmisto eivät sen sijaan ole riippuvaisia RFID Plugin -sovelluksesta eivätkä toisistaan, vaan kummankin käyttö on mahdollista myös erikseen.

Lainausmoduuli

Voyager-, ATP Origo- ja PallasPro-kirjastojärjestelmien toteutukset perustuvat lainausmoduulin käyttöliittymän rakenteen tuntemiseen ja käyttöliittymän tapahtumien tarkkailemiseen Windows-käyttöjärjestelmän WinAPI (Windows Application Programming Interface) -ohjelmointirajapinnan tarjoamien palvelujen avulla. Käyttöliittymän tapahtumien seuraaminen on toteutettu muun muassa ikkunoiden aukioloa ja aktiivisuutta tarkkailemalla sekä hiiritapahtumia että näppäimistön painalluksia seuraamalla.

ATP Origosta on muista tuettavista kirjastojärjestelmistä poiketen olemassa kaksi toisistaan poikkeavaa versiota: paikallisesti asennettava ja graafisen etäyhteyden yli käytettävä versio. Graafisen etäyhteyden yli käytettävä versio on RFID Plugin -sovelluksen kannalta ongelmallinen, koska Windowsin ohjelmointirajapinnan kautta ei ole mahdollista tarkkailla etäyhteyssikkunan sisällä olevien ikkunoiden tapahtumia.

Kirjastojärjestelmät voidaan jakaa kolmeen ryhmään sen perusteella, mitä menetelmää niiden tapahtumien tarkkailemiseen käytetään. Voyagerin ja ATP Origon paikallisesti asennettavien versioiden toteutukset perustuvat

käyttöliittymän ikkunoiden tarkkailuun, kun PallasPro:n ja ATP Origon graafisen etäyhteyden yli käytettävien versioiden toteutukset puolestaan perustuvat pääasiassa hiiritapahtumien ja näppäimistön painallusten seuraamiseen.

PrettyLib-kirjastojärjestelmän toteutus eroaa sen sijaan täysin kaikkien muiden järjestelmien toteutuksista, ja se on kehitetty yhteistyössä kirjastojärjestelmän kehityksestä ja ylläpidosta vastaavan suomalaisen PrettyBit Software Oy -yrityksen kanssa. PrettyLib-kirjastojärjestelmän lainausmoduuli välittää RFID Plugin -sovellukselle tietoa järjestelmän tapahtumista jaetun muistialueen avulla. Kommunikointi perustuu yhteisesti määriteltyjen viestien käyttöön siten, että PrettyLib kirjoittaa ennalta määriteltyjen lainausmoduulin tapahtumien yhteydessä tapahtumaa vastaavan koodin jaetulle muistialueelle, jota RFID Plugin -sovellus aktiivisesti tarkkailee. Yhteisesti määritellyt viestit on esitetty liitteessä 1.

Väliohjelmisto

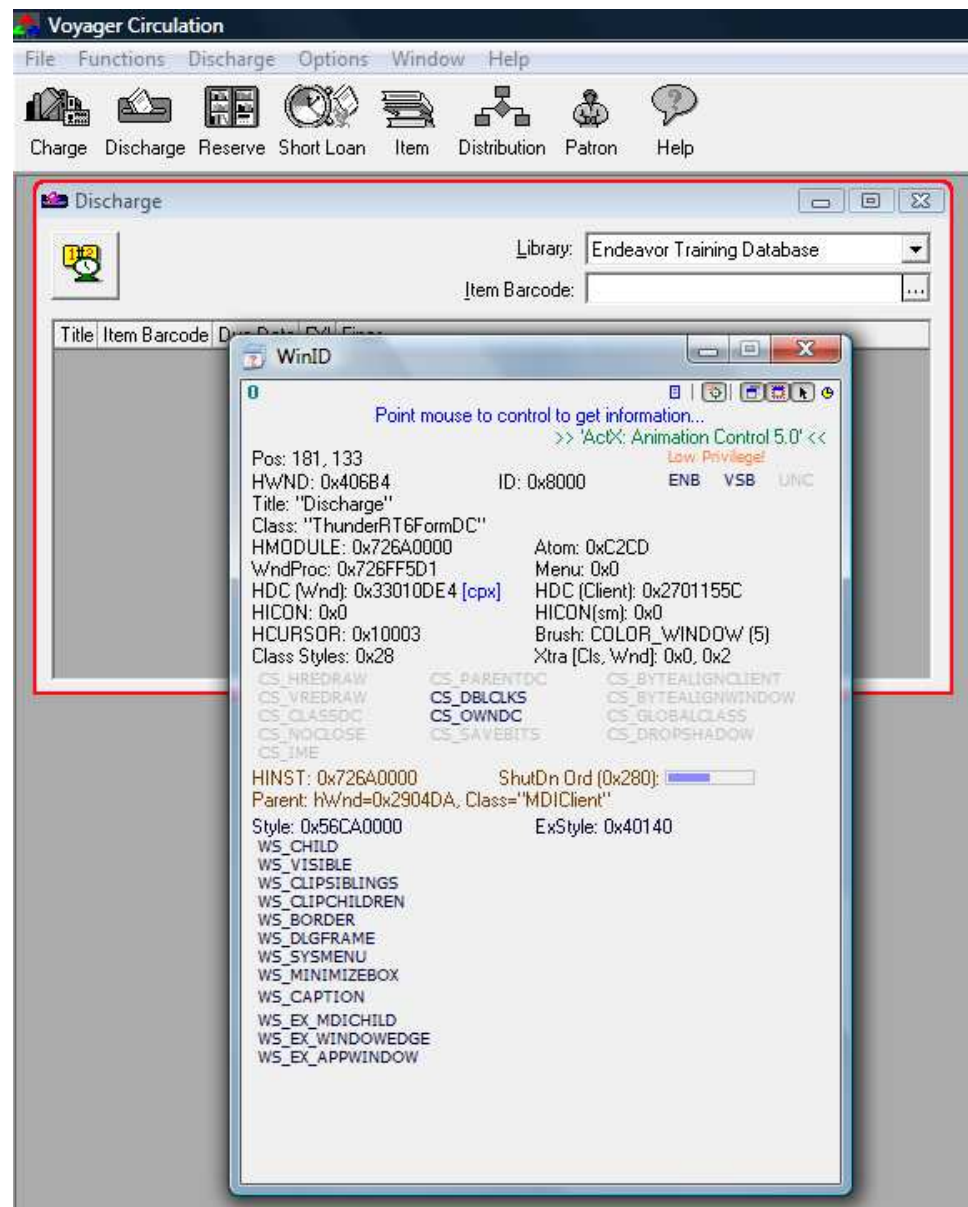
Lainausmoduulin tapahtumien välittäminen väliohjelmistolle perustuu kahden erilaiseen toteutustekniikkaan. 3M Staff Pad Workstation -väliohjelmiston kanssa kommunikointi tapahtuu näppäinten painallusten ja Bookmatic RFID Staff -väliohjelmiston kanssa jaetun muistialueen avulla. Jaetun muistialueen käyttö perustuu ohjelmiston kehityksestä ja ylläpidosta vastaavan Mikro-Väylä Oy -yrityksen kanssa yhteistyössä määriteltyjen viestien käyttöön. Viestit ovat identtisiä liitteessä 1 esitettyjen PrettyLib-kirjastojärjestelmän kanssa kommunikointiin käytettyjen viestien kanssa. Jaetun muistialueen käyttö RFID Plugin -sovelluksen ja väliohjelmiston välisessä kommunikoinnissa mahdollistaa huomattavasti näppäimistön painalluksiin perustuvaa kommunikointia monipuolisemman tiedonvaihdon.

4.3 Käytännön toteutus

RFID-plugin -sovelluksen toteutuksessa on käytetty C++-ohjelmointikieltä ja Microsoft Visual Studio 2008 -kehitysympäristöä. Sovellus on Win32 Application -tyyppinen sovellus, joka ei asennusohjelmaa lukuun ottamatta sisällä minkäänlaista käyttöliittymää. Sovelluksen asennusohjelma on luotu kehitysympäristön tarjoaman opastetun toiminnon avulla. RFID Plugin -sovelluksen asentamisen lisäksi asennusohjelma lisää sovelluksen tiedot Windows-käyttöjärjestelmän käynnistyksen yhteydessä käynnistettävät ohjelmat mää-

rittelevään rekisteriin. Näin sovellus käynnistyy automaattisesti käyttöjärjestelmän käynnistymisen yhteydessä.

Eri kirjastojärjestelmien lainausmoduulien käyttöliittymien rakenteiden selvittämisessä on käytetty WinID-ohjelmaa. Ohjelman avulla saatiin selville muun muassa käyttöliittymien ikkunoiden nimet ja luokat, joita tarvittiin lainausmoduulien tapahtumien tarkkailun toteuttamisessa. Kuvassa 7 on nähtävissä WinID-ohjelman Voyager-kirjastojärjestelmän lainausmoduulin palautusikkunasta näytettävät tiedot.

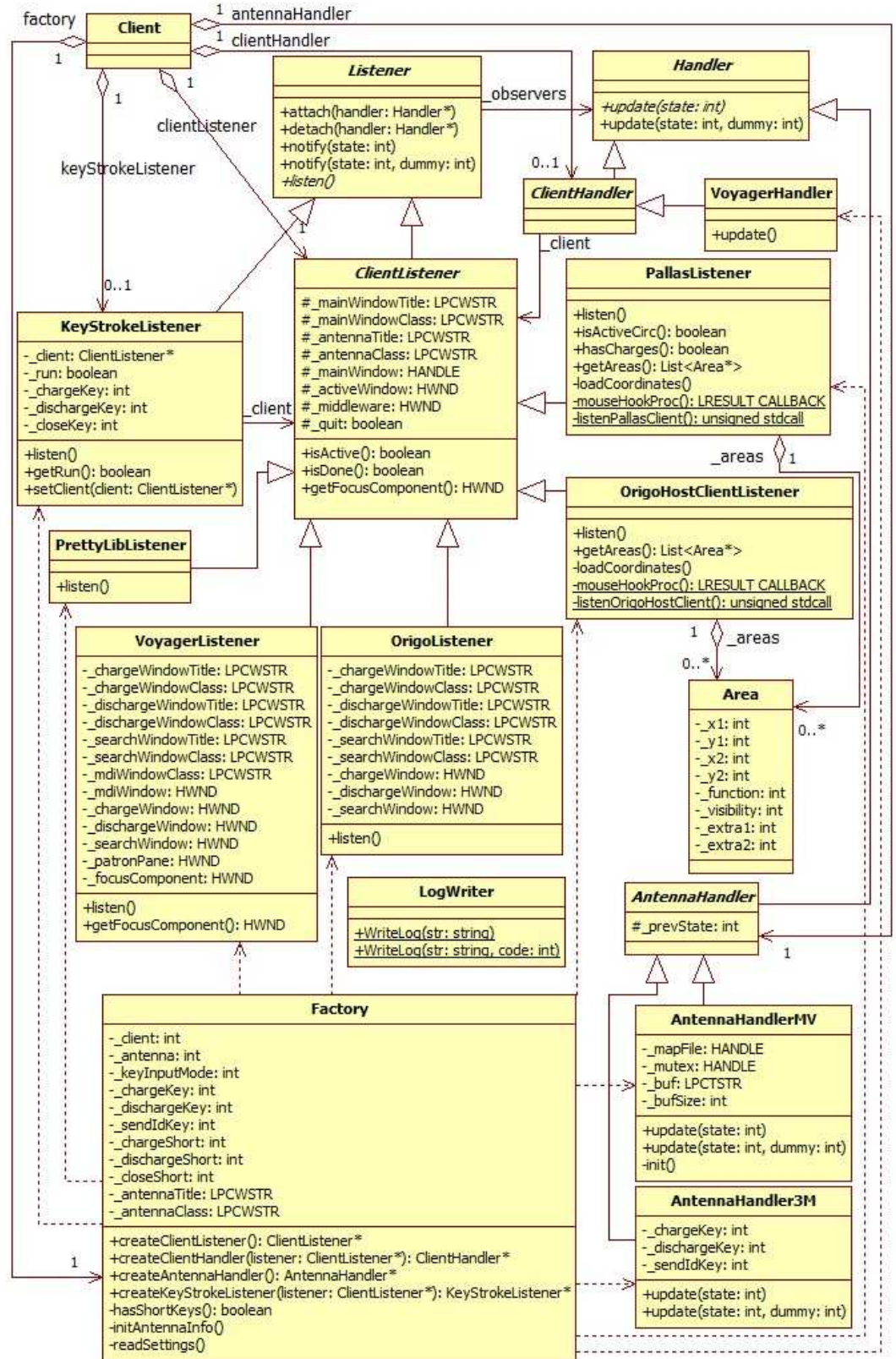


Kuva 7. Voyager-kirjastojärjestelmän palautusikkunan tiedot WinID-ohjelmassa.

4.4 Tekninen toteutus

RFID Plugin -sovelluksen sisäinen rakenne perustuu rajapintojen käyttöön. Tämä mahdollistaa sovelluksen rakentamisen siten, että sen käyttö on vai- vatonta useiden eri kirjastojärjestelmien ja väliohjelmistojen kanssa. Eri jär- jastelmiin liittyviä toteutuksia käsitellään samojen rajapintojen kautta, mutta konkreettiset toteutukset on eristetty toisistaan. Tällöin yksittäiseen järjes- telmään liittyvää toteutusta voidaan muuttaa vaikuttamatta muiden järjeste- lmiin toteutuksiin. Sovellus on myös helposti laajennettavissa, sillä uusien järjestelmien liittäminen sovelluksen piiriin ei edellytä jo olemassa olevien to- teutusten muuttamista. Sovelluksen luokkakaavio on esitetty kuvassa 8.

Sovelluksen arkkitehtuuri perustuu Observer-suunnittelumalliin. Mallin aja- tuksena on luoda rajapinta, minkä avulla yhden olion tilassa tapahtuneet muutokset aiheuttavat myös kyseisestä oliosta riippuvaisten olioiden tilojen päivittymisen [36, s. 293]. RFID Plugin -sovelluksessa abstraktit Listener- ja Handler-luokat muodostavat Observer-mallin mukaisen rajapinnan siten, että Listener edustaa tarkkailun kohdetta eli subjektia ja Handler tarkkailijaa. Lis- tener tarjoaa rajapinnan tarkkailijoiden liittämiseksi ja poistamiseksi sekä subjektin tilassa tapahtuneista muutoksista tiedottamiseksi tarkkailijoille. Subjektin sen hetkinen tila välitetään tarkkailijoille muutoksesta tiedottami- sen yhteydessä, joten tarkkailijoiden ei tarvitse tuntea subjektia. Lisäksi Lis- tener sisältää puhtaan virtuaalifunktion listen, joka kaikkien sen perivien luokkien on toteutettava. Listener-luokan toteutus on esitetty liitteessä 2. Handler tarjoaa vastaavasti rajapinnan tarkkailijoiden tilan päivittämiseksi, kun subjektin tila muuttuu.



Kuva 8. Sovelluksen luokkakaavio.

Listener-luokan perivä abstrakti ClientListener-luokka tarjoaa rajapinnan kirjastojärjestelmän lainausmoduulin tarkkailusta vastaaville luokille. ClientLis-

tener-luokka sisältää keskeisimmät aliluokkien tarvitsemat muuttujat sekä funktiot oletustoteutuksineen. Kaikki aliluokat eivät käytä kaikkia ClientListener-luokan sisältämiä muuttujia ja funktioita, mutta näin pääohjelma pystyy käsittelemään kaikkia ClientListener-luokan aliluokkia ylikuokan tarjoaman yhteneväisen rajapinnan kautta. Listener-luokassa määritellyn abstraktin virtuaalifunktion listen toteuttaminen jää lainausmoduulin tarkkailusta vastaavien luokkien tehtäväksi. Listen-funktion toteutuksen tulee sisältää lainausmoduulin tapahtumien tarkkailuun tarvittava toiminnallisuus.

Handler-luokan perivä abstrakti AntennaHandler-luokka tarjoaa rajapinnan väliohjelmiston kanssa kommunikoinnista vastaaville luokille. Rajapinta on luokkien toteutuksissa esiintyvistä eroista johtuen erittäin yksinkertainen. Toteutuksen etuna on kuitenkin se, että näin luokkia voidaan käsitellä yhdenmukaisesti AntennaHandler-luokan tarjoaman rajapinnan kautta.

Abstrakti ClientHandler-luokka tarjoaa rajapinnan lainausmoduulille viestien lähettämisestä vastaaville luokille. Viestien lähettämisellä tarkoitetaan RFID Plugin -sovelluksessa määriteltyjen pikanäppäinten toimintoja vastaavien komentojen lähettämistä lainausmoduulille. ClientHandler-luokka sisältää osoittimen ClientListener-olioon, koska joidenkin komentojen lähettäminen edellyttää tietoja lainausmoduulin sisäisestä rakenteesta.

Käytännössä rajapinnoista muodostuu kaksi subjekti-tarkkailija-paria. Toisessa parissa AntennaHandler-luokan aliluokka toimii tarkkailijana ja ClientListener-luokan aliluokka subjektina. Väliohjelmiston kanssa kommunikoinnista vastaava tarkkailija saa tiedon aina, kun subjekti havaitsee tapahtuman lainausmoduulissa. Tällä tavoin sovellus kykenee pitämään väliohjelmiston ajan tasalla lainausmoduulin tapahtumista. Toisessa parissa ClientHandler-luokan aliluokka toimii sen sijaan tarkkailijana ja Listener-luokan perivä KeyStrokeListener-luokka subjektina. KeyStrokeListener-luokka tarkkailee näppäimistön painalluksia ja välittää tarkkailijalle tiedon aina, kun pikanäppäimeksi määriteltyä näppäintä on painettu. Saatuaan tiedon näppäimen painalluksesta tarkkailija lähettää lainausmoduulille näppäimen painallusta vastaavan komennon. KeyStrokeListener-luokan listen-funktion toteutus on esitetty liitteessä 3 ja VoaygerHandler-luokan update-funktion toteutus liitteessä 4.

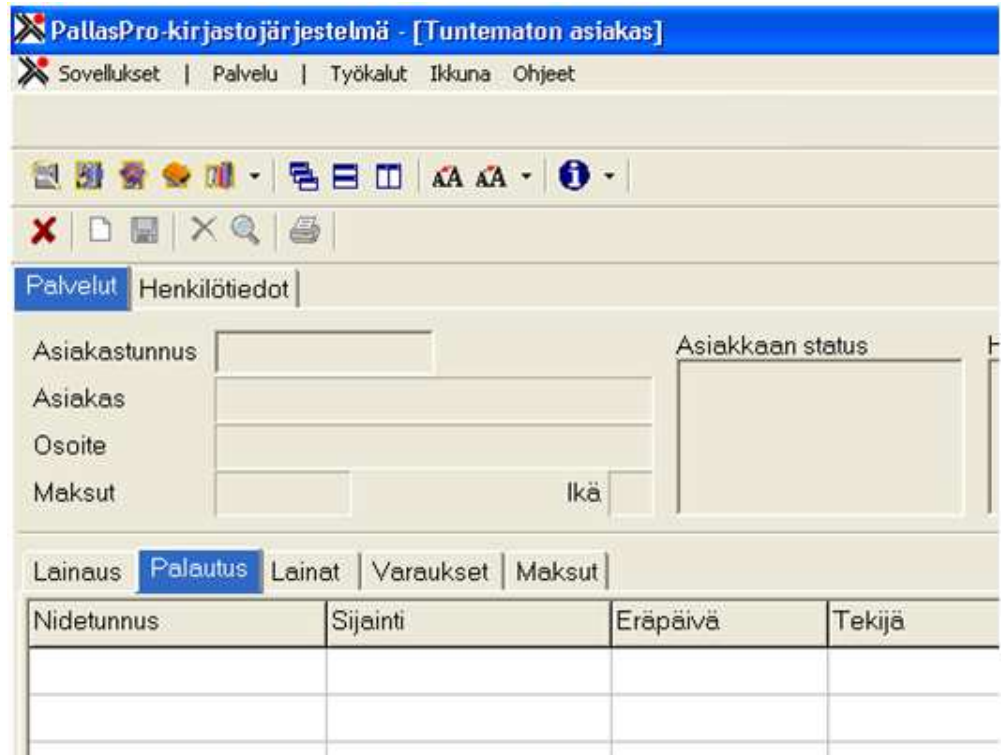
VoyagerListener- ja OrigoListener-luokat

VoaygerListener- ja OrigoListener-luokkien toteutukset pohjautuvat lainausmoduulin käyttöliittymän avointen ikkunoiden ja niiden aktiivisuuden tarkkailuun. Tämä on mahdollista Windows-käyttöjärjestelmän ohjelmointirajapinnan tarjoamien FindWindow- ja FindWindowEx-funktioiden avulla. Funktiot palauttavat osoittimen parametreinä annettuja ikkunan nimeä ja luokkaa vastaavaan ikkunaan, mikäli sellainen on avoinna. Funktiot eroavat toisistaan siten, että FindWindow-funktion avulla on mahdollista etsiä päätason ikkunoita ja FindWindowEx-funktion avulla puolestaan päätason ikkunoiden alaisia niin sanottuja lapsi-ikkunoita [37]. Tällä tavoin hankittuja lainausmoduulin ikkunoiden osoittimia voidaan verrata GetForegroundWindow-funktion palauttamaan funktion kutsuhetkellä aktiivisena olevan ikkunan osoittimeen [38]. Tietyn pääikkunan alaisten lapsi-ikkunoiden aktiivisuuden selvittäminen onnistuu puolestaan pääikkunalle SendMessage-funktion avulla lähettävän WM_MDIGETACTIVE-viestin kautta [39]. Käyttöliittymän ikkunoissa tapahtuneiden muutosten seuraaminen edellyttää ikkunoiden nykytilan lisäksi myös kahden edeltävän tilan tallentamista ja tarkkailemista. VoyagerListener-luokan listen-funktion toteutus on esitetty liitteessä 5.

PallasListener- ja OrigoHostClientListener-luokat

PallasListener- ja OrigoHostClientListener-luokkien toteutukset pohjautuvat puolestaan pääasiassa hiiritapahtumien ja näppäimistön painallusten seuraamiseen, vaikka toteutukseen sisältyy myös avointen ikkunoiden ja niiden aktiivisuuden tarkkailua. Hiiritapahtumien seuraaminen on toteutettu SetWindowsHookEx-funktion avulla asetettavan WH_MOUSE_LL-koukun kautta [40]. Tämän seurauksena kaikki hiiritapahtumat kulkevat koukun asettaneessa luokassa määritellyn staattisen mouseHookProc-funktion läpi. Funktio tutkii, onko hiiren klikkaus tapahtunut lainausmoduulin ollessa aktiivinen ja vastaavatko klikkauksen koordinaatit RFID Plugin -sovelluksen asetustiedostossa määriteltyjen alueiden koordinaatteja. Asetustiedostossa on määritelty niiden käyttöliittymän alueiden koordinaatit, joiden klikkaamisella on merkitystä sovelluksen toiminnan kannalta. Tällaisia alueita ovat esimerkiksi kuvassa 9 nähtävän PallasPro-kirjastojärjestelmän lainausmoduulin käyttöliittymän lainaus- ja palautus-toiminnon aktivoivien painikkeiden sijainnit. Jo-kaista asetustiedostossa määriteltyä aluetta kohti luodaan sovelluksen käynnistyksen yhteydessä alueen tiedot sisältävä Area-luokan ilmentymä.

PallasListener-luokan mouseHookProc-funktion toteutus on esitetty liitteessä 6.



Kuva 9. PallasPro-kirjastojärjestelmän lainausmoduulin käyttöliittymä.

Näppäimistön painallusten tarkkailu on puolestaan toteutettu GetAsyncKeyState-funktiolla, joka palauttaa funktion kutsuhetkellä painetun näppäimen arvon [41]. RFID Plugin -sovellus tunnistaa niiden näppäinten painallukset, joilla on vaikutusta lainausmoduulin tapahtumiin. Tällaisia näppäimiä ovat esimerkiksi lainaus- ja palautus-toiminnon aktivoivat näppäimet. Myös KeyStrokeListener-luokan toiminta perustuu GetAsyncKeyState-funktion käyttöön.

PrettyLibListener-luokka

PrettyLibListener-luokan toteutus perustuu jaetun muistialueen käyttöön, jonne sekä RFID Plugin -sovelluksella että PrettyLib-kirjastojärjestelmän lainausmoduulilla on pääsy. Muistialueen luominen on toteutettu CreateFileMapping-funktion avulla [42]. MapViewOfFile-funktio liittää puolestaan muistialueen funktiota kutsuvan prosessin muistiavaruuteen, mikä mahdollistaa muistialueella olevan datan käsittelyn [43]. Muistialueen synkronoinnissa on puolestaan käytetty mutex-oliota, jonka luominen tapahtuu CreateMu-

tex-funktion avulla [44]. Ennen muistialueella olevan datan käsittelemistä ohjelman on saatava mutex-olio haltuunsa WaitForSingleObject-funktiota kutsumalla. Mutex-olion haltuunotto epäonnistuu ja funktiota kutsunut prosessi joutuu odottamaan vuoroaan, mikäli toinen prosessi käsittelee muistialueella olevaa dataa [45]. Lopettaessaan muistialueella olevan datan käsittelyn prosessi kutsuu puolestaan ReleaseMutex-funktiota, mikä vapauttaa mutexin ja lähettää odottaville prosesseille tiedon mutexin vapautumisesta [46]. PrettyLibListener-luokan listen-funktion toteutus on esitetty liitteessä 7.

AntennaHandler3M-luokka

AntennaHandler3M-luokan toteutus perustuu RFID Plugin -sovelluksen ja 3M Staff Pad Workstation -väliohjelmiston asetuksissa määriteltyjen näppäinten painallusten generoimiseen keybd_event-funktion avulla. Asetuksissa on määritelty lainaus-, palautus- ja luku-toimintoa vastaavat näppäinyhdistelmät, joiden painalluksia väliohjelmisto kuuntelee. Näppäinyhdistelmät sisältävät aina ctrl-näppäimen ja jonkin asetustiedostossa määritellyn näppäimen. Keybd_event-funktio saa parametrinaan näppäimen, jonka painallus generoidaan, sekä generoitavan tapahtuman tyyppiä eli painetaanko näppäin alas vai vapautetaanko näppäin [47]. AntennaHandler3M-luokan update-funktion toteutus on esitetty liitteessä 8.

AntennaHandlerMV-luokka

AntennaHandlerMV-luokan toteutus perustuu PrettyLibListener-luokan tapaan jaetun muistialueen käyttöön. Muistialue ei kuitenkaan ole sama, jota PrettyLibListener-luokka käyttää. Bookmatic RFID Staff -väliohjelmisto tarkkailee jaettua muistialuetta ja ohjaa lukijan toimintaa muistialueelta lukemista tietojen perusteella.

VoyagerHandler-luokka

VoyagerHandler on ainoa ClientHandler-luokan aliluokka, sillä muiden kirjastojärjestelmien lainausmoduuleissa ei esiintynyt tarvetta ulkoisten pikanäppäinten määrittelyyn. VoyagerHandler-luokan toteutus perustuu AntennaHandler3M-luokan tapaan keybd_event-funktion käyttöön. Käytännössä toteutus toimii siten, että KeyStrokeListener-olion havaitessa asetustiedostossa määritellyn pikanäppäimen painalluksen se lähettää tästä tiedon VoyagerHandler-oliolle, joka lähettää Voyagerin lainausmoduulille määriteltyä pi-

kanäppäintä vastaavan näppäinyhdistelmän painalluksen. Tällä tavoin voidaan RFID Plugin -sovelluksen asetuksissa määritellä esimerkiksi lainaus-toiminnon aktivoiminen F1-näppäintä painamalla, kun Voyagerin oma näppäinyhdistelmä toiminnon aktivoimiseksi on ctrl + H. Käyttäjän painaessa F1-näppäintä RFID Plugin -sovellus generoi ctrl + H -näppäinyhdistelmän painalluksen.

LogWriter-luokka

LogWriter-luokka sisältää ainoastaan kaksi staattista funktiota, joiden avulla tiedot ohjelman ajonaikaisista virhetilanteista saadaan tallennettua erilliseen lokitiedostoon. Staattisten funktioiden käyttö ei edellytä LogWriter-luokan ilmentymän luontia, jonka ansiosta lokitiedostoon kirjoittaminen on yksinkertaista kaikista ohjelman osista. Funktiokutsun parametrinä annetaan lokitiedostoon kirjoitettava viesti sekä mahdollisesti myös käyttöjärjestelmän palauttama virhekoodi.

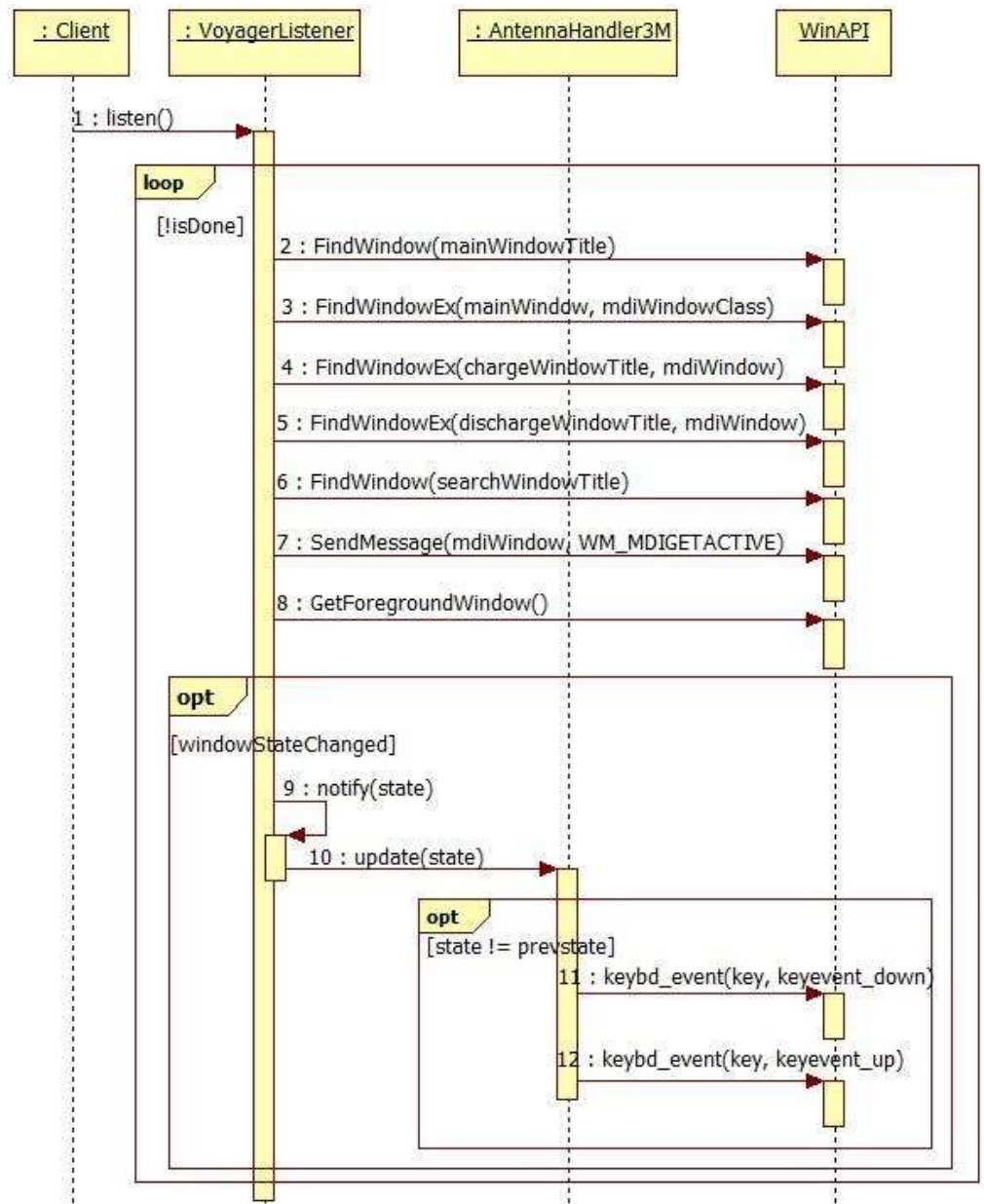
Client- ja Factory-luokat

Client-luokka sisältää RFID Plugin -sovelluksen pääohjelman, joka on esitetty liitteessä 9. Factory-luokka puolestaan vastaa pääohjelman käyttämien olioiden luomisesta asetustiedossa määriteltujen tietojen pohjalta. Pääohjelma käsittelee luotuja olioita niiden ylikuokkien tyyppisten osoittimien kautta, eikä täten ole tietoinen, minkä luokan ilmentymiä oliot todellisuudessa ovat. Toteutus mahdollistaa uusien aliluokkien lisäämisen ja olemassa olevien toteutusten muokkaamisen pääohjelmaa muuttamatta. Ohjelma käyttää kirjastojärjestelmästä riippuen ohjelman pääsäikeen lisäksi yhdestä kolmeen säiettä.

4.5 Käyttäytyminen peruskäytössä

RFID Plugin -sovelluksen toiminnan kannalta keskeisin funktio on lainausmoduulin tapahtumien seuraamisesta vastaavan ListenClient-luokan aliluokkien toteuttama listen-funktio. Listen-funktio on vastuussa lainausmoduulin tapahtumien seuraamisesta ja muutoksien raportoimisesta väliohjelmiston kanssa kommunikoimisesta vastaavalle AntennaHandler-luokan aliluokalle. Kuvassa 10 on esitetty VoyagerListener-luokan toteuttaman listen-funktion toimintaperiaate, kun väliohjelmistona toimii 3M Staff Pad Workstation.

Lainausmoduulin tapahtumien tarkkailu käynnistyy, kun sovelluksen pääohjelman sisältävä Client-olio kutsuu VoyagerListener-olion listen-funktiota. Funktion kutsuminen ei tapahdu ohjelman pääsäikeestä, vaan tarkoitusta varten on pääohjelmassa luotu oma erillinen säikeensä.



Kuva 10. Listen-funktion sekvenssikaavio VoyagerListener-luokassa.

VoyagerListener-luokassa listen-funktion toteutus perustuu lainausmoduulin käyttöliittymän sisältämien ikkunoiden tilan tarkkailemiseen Windows-käyttöjärjestelmän WinAPI-ohjelmointirajapinnan tarjoamien funktioiden avulla. Listen-funktio pyörii while-silmukassa, jonka jokaisella kierroksella haetaan FindWindow- ja FindWindowEx-funktioiden avulla osoittimet pää-, lai-

naus-, palautus- ja haku-ikkunoihin. Pääikkuna on päätason ikkuna, joten sen osoitin haetaan FindWindow-funktion avulla. Lainaus- ja palautusikkuna ovat puolestaan pääikkunan alaisia ikkunoita, joten niiden osoittimien haku tapahtuu FindWindowEx-funktion avulla. Haku-ikkuna on pääikkunan tapaan päätason ikkuna, joten siihen käytetään FindWindow-funktiota. Tämän jälkeen haetaan SendMessage-funktion avulla osoitin aktiivisena olevaan pääikkunan alaiseen ali-ikkunaan. GetForegroundWindow-funktion avulla haetaan puolestaan osoitin aktiivisena olevaan päätason ikkunaan. Lainausmoduulin ja sen ali-ikkunoiden aktiivisuudet saadaan selville vertaamalla lainausmoduulin ikkunoiden ja aktiivisten ikkunoiden osoittimia toisiinsa.

Ikkunoiden tiloissa tapahtuneiden muutosten havaitseminen perustuu ikkunoiden edellisten tilojen tallettamiseen ja vertailemiseen nykytilojen kanssa. Mikäli ikkunoiden tiloissa huomataan tapahtuneen muutoksia, kutsutaan notify-funktiota, jonka parametrina tieto havaitusta tilasta välitetään. Notify-funktion kutsuminen saa aikaan AntennaHandler3M-olion update-funktion kutsumisen, jonka parametrina tieto havaitusta tilamuutoksesta välitetään. AntennaHandler3M-olion update-funktiossa varmistetaan, että parametrinä saatu tilatieto poikkeaa edellisestä tilatiedosta, koska samojen peräkkäisten tilojen välittämisellä ei ole väliohjelmiston kannalta merkitystä. Mikäli tilatieto eroaa edeltävästä tilatiedosta, luodaan kyseistä tilaa vastaava näppäinyhdistelmän painallus keybd_event-funktion avulla. 3M Staff Pad Workstation -väliohjelmisto tarkkailee aktiivisesti näppäimistön painalluksia ja havaitessaan asetuksissa määritellyn näppäinyhdistelmän se muuttaa tilaansa kyseistä näppäinyhdistelmää vastaavaksi.

4.6 Testaus ja käyttöönotto

RFID Plugin -sovelluksen testaus voidaan jakaa kahteen eri osaan: kehitystyön aikaiseen testaukseen ja valituissa kirjastoissa tapahtuneeseen koekäyttöön. Kehitystyön aikainen testaus toteutettiin pääasiassa Voyager-kirjastojärjestelmän kanssa, kun koekäytössä oli mukana Voyager-, Pretty-Lib-, PallasPro- ja ATP Origo -kirjastojärjestelmiä käytäviä kirjastoja.

Suurin osa sovelluksen kehitystyön yhteydessä tehdystä testaamisesta tapahtui Voyager-kirjastojärjestelmän lainausmoduulin ja 3M Staff Pad Workstation -väliohjelmiston kanssa. Testauksen keskittyminen Voyager-kirjastojärjestelmään johtui puhtaasti siitä, että muiden kirjastojärjestelmien

käyttö kehitystyön yhteydessä ei ollut mahdollista. Testauksen päämääränä oli varmistaa kaikkien määrittelyssä esitettyjen vaatimusten toteutuminen sekä toteutetun toiminnallisuuden virheetön toiminta. Testauksessa havaitut virheet ja toiminnallisuuteen liittyvät puutteet korjattiin aina heti löytymisen jälkeen. Mikäli havaitut virheet vaikuttivat myös muiden kirjastojärjestelmien toteutuksiin, tehtiin korjaukset ja parannukset myös niihin. Muutosten teko muiden kirjastojärjestelmien toteutuksiin ilman välitöntä mahdollisuutta testata muutosten vaikutusta käytännössä oli varsin haastavaa, koska eri järjestelmien toteutukset eroavat toisistaan melkoisesti.

Koekäyttöön osallistuneet kirjastot saivat käyttöönsä sovelluksen testiversioita, joiden toiminnasta kirjaston henkilökunta antoi palautetta. Palautteen pohjalta korjattiin testiversioissa esiintyneitä virheitä ja täydennettiin vaja-vaista toiminnallisuutta. Koekäyttöön osallistuneet kirjastot tarjosivat mahdollisuuden testata sovellukseen tehtyjä muutoksia ja lisäyksiä käyttämänsä kirjastojärjestelmän yhteydessä, minkä avulla pystyttiin varmistamaan tehtyjen muutosten toimivuus. Testauksen suorittaminen edellytti kuitenkin aina vierailun järjestämistä kyseiseen kirjastoon.

RFID Plugin -sovelluksen koekäyttöön osallistuivat

- PallasPro-kirjastojärjestelmää käyttävä Sipoon kunnankirjasto
- ATP Origo -kirjastojärjestelmän graafisen etäyhteyden yli käytettävää versiota käyttävä Piikkiön kunnankirjasto
- ATP Origo -kirjastojärjestelmän paikallisesti asennettavaa versiota käyttävä Lahden maakuntakirjasto
- PrettyLib -kirjastojärjestelmää käyttävä Poliisiammattikorkeakoulun kirjasto
- Voyager-kirjastojärjestelmää käyttävä Museoviraston kirjasto
- Voyager-kirjastojärjestelmää käyttävä Aalto-yliopiston teknillisen korkeakoulun elektroniikan, tietoliikenteen ja automaation tiedekunnan kirjasto
- Voyager-kirjastojärjestelmää käyttävä Aalto-yliopiston teknillisen korkeakoulun tuotantotalouden laitoksen kirjasto.

Sovelluksen lopullisen version todettiin olevan valmis käyttöönottoon, kun kaikki kehitysvaiheessa ilmenneet virheet ja toiminnalliset puutteet oli korjattu. Toiminnalliset puutteet todettiin korjatuiksi, kun koekäyttöön osallistuneista kirjastoista ei esitetty enää uusia parannus- tai korjausehdotuksia.

Koekäytön aikana käyttäjiltä saadun palautteen perusteella voidaan myös todeta ohjelman käyttöönoton mukanaan tuomat hyödyt. Sovelluksen koettiin helpottavan käyttäjän työtä ja vähentävän merkittävästi asiakaspalvelussa tapahtuvia kahden ohjelman rinnakkaisesta käytöstä aiheutuvia virhetilanteita.

5 YHTEENVETO

RFID on esineiden, eläinten ja ihmisten automaattiseen tunnistamiseen käytettävä radiotaajuuksiin perustuva tekniikka. Tekniikan laajojen käyttömahdollisuuksien ansiosta sitä hyödynnetään monissa erilaisissa käytännön sovelluksissa. Kirjastoissa RFID-tekniikkaa käytetään kokoelmien hallintaan, kirjastojen itsenäiskäytön lisäämiseen sekä asiakaspalvelun tehostamiseen.

Kirjastojen RFID-toteutusten perusajatuksena on, että kirjoissa olevat viivakoodit korvataan RFID-tunnisteen sisältävillä tarroilla ja perinteiset viivakoodinlukijat RFID-lukijoilla. Tarrassa olevan RFID-tunnisteen tiedot välittyvät langattomasti antennin avulla lukijaan. Lukija konvertoi RFID-tunnisteen tiedot kirjastojärjestelmän ymmärtämään muotoon ja välittää ne kirjastojärjestelmään.

RFID-tekniikan yleistymistä kirjastojen keskuudessa on osaltaan hidastanut nykyisin käytössä olevista kirjastojärjestelmistä puuttuva tuki RFID-tekniikan käyttöön. Monet kirjastojärjestelmät kyllä mahdollistavat tekniikan käytön, mutta osa RFID-tekniikan tarjoamista mahdollisuuksista jää kuitenkin edelleen käyttämättä. Suurimpana ongelmana on RFID- ja kirjastojärjestelmän välinen yksisuuntainen kommunikaatio, joka rajoittaa RFID-tekniikan käyttömahdollisuuksia sekä hankaloittaa järjestelmien yhteiskäyttöä.

Kirjastojärjestelmät kykenevät kommunikoimaan ja vaihtamaan tietoja monien erilaisten järjestelmien kanssa eri standardeihin ja protokolliin perustuvien rajapintojen avulla. RFID- ja kirjastojärjestelmän yhteistoimintaa tehostavaa ja järjestelmien välisen kaksisuuntaisen tiedonvaihdon mahdollistavaa rajapintaa ei kuitenkaan vielä toistaiseksi ole olemassa. Osaltaan tämä johtunee erilaisten tarjolla olevien järjestelmien suuresta määrästä sekä eri järjestelmien toteutuseroista johtuvista yhteensopivuusongelmista. Tulevaisuuden kannalta tällaisen rajapinnan toteuttaminen olisi kuitenkin ensi arvoisen tär-

keää ja mahdollistaisi kaiken RFID-tekniikan tarjoaman potentiaalın hyödyntämisen kirjastoympäristössä.

RFID Plugin -sovellus tarjoaa osittaisen ratkaisun RFID- ja kirjastojärjestelmän väliseen puutteelliseen kommunikaatioon. Sovellus helpottaa RFID- ja kirjastojärjestelmän yhteiskäyttöä asiakaspalvelussa välittämällä tiedon kirjastojärjestelmän lainausmoduulin tapahtumista RFID-lukijan toimintaa ohjaavalle väliohjelmistolle.

PrettyLib-kirjastojärjestelmään ja Bookmatic RFID Staff -väliohjelmistoon toteutetut yksinkertaiset rajapinnat ovat hyvä esimerkki siitä, kuinka RFID- ja kirjastojärjestelmän yhteistoiminta tehostuu järjestelmien välistä kommunikaatiota kehittämällä. Rajapintojen toiminta perustuu liitteessä 1 esitettyjen yhteisesti määriteltyjen viestien ja jaetun muistialueen käyttöön. Järjestelmät eivät kuitenkaan kommunikoi suoraan keskenään, vaan RFID Plugin -sovellus toimii viestien välittäjänä järjestelmien välissä. Tämän ratkaisun kiistattomana etuna on se, että keskenään kommunikoivien järjestelmien ei välttämättä tarvitse tuntea toistensa käyttämiä rajapintoja. Tämä perustuu siihen, että järjestelmien välissä toimiva RFID Plugin -sovellus tuntee järjestelmien käyttämät rajapinnat ja osaa muuttaa toiselta järjestelmältä saamansa viestit toisen järjestelmän ymmärtämään muotoon. Tällä tavoin yhden järjestelmän rajapinnassa tehtävät muutokset eivät myöskään vaikuta muihin järjestelmiin vaan ainoastaan RFID Plugin -sovellukseen.

Tulevaisuudessa RFID-tekniikkaa käyttävien kirjastojen määrä tulee suurella todennäköisyydellä lisääntymään, kun uudet kirjastot ottavat tekniikkaa käyttöönsä ja tekniikkaa jo käyttävät kirjastot laajentavat sen käyttöä kokoelmiensa piirissä. Tekniikan käyttöönottokustannusten lasku ja toiminnallisuuden kehittyminen tekevät RFID-tekniikasta entistä houkuttelevamman vaihtoehdon vanhalle viivakoodeihin pohjautuvalle järjestelmälle.

VIITELUETTELO

- [1] Nuotio, Samuli, Kirjastot ja atk. Helsinki: Kirjastopalvelu Oy. 1986.
- [2] Ilva, Jyrki, Yrityksen ja erehdyksen kautta: Tieteellisten kirjastojen keskitetyn atk-suunnittelun alkuvaiheet 1970-luvulla [verkkodokumentti, viitattu 20.4.2010]. Saatavissa: http://www.kansalliskirjasto.fi/extra/tietolinja/0106/atk_alkuvaiheet.html.
- [3] Online public access catalog [verkkodokumentti, viitattu 20.4.2010]. Saatavissa: <http://en.wikipedia.org/wiki/OPAC>.
- [4] Lynch, Clifford, From automation to transformation: forty years of libraries and information technology in higher education. *Educause Review* 1 (2000), s. 60 -68.
- [5] Karppinen, Iiris - Piukkula, Juha, Kirjastot it-ympäristössä. Helsinki: BTJ Kirjastopalvelu Oy. 2005.
- [6] Akeroyd, John, Integrated Library Management Systems - Overview. *VINE* 2 (1999), s. 3 - 10.
- [7] Kirjastojärjestelmät nyt! [verkkodokumentti, viitattu 20.4.2010]. Saatavissa: http://wiki.kirjastot.fi/index.php/Kirjastoj%C3%A4rjestelm%C3%A4t_nyt!.
- [8] Suomalaisten yleisten kirjastojen atk-kirjastojärjestelmät, niiden tietovarantojen verkkokäyttöisyys ja tietotekniset valmiudet [verkkodokumentti, viitattu 14.5.2010]. Saatavissa: http://www.minedu.fi/export/sites/default/OPM/Julkaisut/2000/liitteet/opm_50_saarti1.pdf?lang=fi.
- [9] Z39.50:n historia ja nykytilanne [verkkodokumentti, viitattu 16.5.2010]. Saatavissa: <http://herkules.oulu.fi/isbn951425242X/html/x358.html>.
- [10] Standardit - välttämätön hyvä [verkkojulkaisu, viitattu 17.5.2010]. Saatavissa: <http://www.kansalliskirjasto.fi/extra/tietolinja/0104/standardointi.html>.
- [11] ANSI/NISO Z39.2-1994R2001 - Information Interchange Format. NISO. 2001.
- [12] MARC-formaatit Suomessa [verkkodokumentti, viitattu 17.5.2010]. Saatavissa: <http://www.kansalliskirjasto.fi/kirjastoala/formaatit/marcit.html>.

- [13] ANSI/NISO Z39.50-2003 - Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. NISO. 2003.
- [14] Palmer, Martin, Making the most of RFID in libraries. London: Facet. 2009.
- [15] 3M Standard Interchange Protocol [verkkodokumentti, viitattu 21.6.2010]. Saatavissa: http://multimedia.3m.com/mws/mediawebserver?mwsId=SSSSS u7zK1fslxtUm8_9m82Uev7qe17zHvTSevTSeSSSSSS--.
- [16] ANSI/NISO Z39.83-1-2008 - NISO Circulation Interchange Part 1 : Protocol (NCIP). NISO. 2008.
- [17] OpenURL: rakenne ja käyttö [verkkodokumentti, viitattu 20.5.2010]. Saatavissa: <http://www.kansalliskirjasto.fi/extra/tietolinja/0103/openURL.html>.
- [18] Coulouris, George - Dollimore, Jean - Kindberg, Tim, Distributed Systems - Concepts And Design. Boston: Addison-Wesley. 2005.
- [19] Saarti, Jarmo, Kirjastojärjestelmän hankkijan opas. Helsinki: BTJ Kirjastopalvelu Oy. 2002.
- [20] Kirjasto 2.0 ja tulevaisuuden näyttöluettelot [verkkodokumentti, viitattu 22.5.2010]. Saatavissa: <http://www.kansalliskirjasto.fi/extra/tietolinja/0206/kirjasto20.html>.
- [21] Shepard, Steven, RFID - Radio Frequency Identification. New York: McGraw-Hill. 2005.
- [22] Bhuptani, Manish - Moradroup, Shahram, RFID Field Guide - Deploying Radio Frequency Identification Systems. N.J.: Prentice Hall PTR. 2005.
- [23] RFID-tekniikan perusteet [verkkodokumentti, viitattu 30.5.2010]. Saatavissa: <http://www.rfidlab.fi/rfid-tekniikan-perusteet>.
- [24] Sweeney, Patrick J., RFID for dummies. N.J.: Wiley. 2005.
- [25] Kirjaston RFID-hanke - suunnitelmista toteutukseen [verkkodokumentti, viitattu 9.6.2010]. Saatavissa: <http://lib.tkk.fi/TietaNet/1.07/a08.html>.
- [26] RFID 301 [verkkodokumentti, viitattu 2.6.2010]. Saatavissa: <http://multimedia.mmm.com/mws/mediawebserver.dyn?6666660Zjcf6IVs6EV s66SOmcCOrrrrQ->.

- [27] RFID-teknologian käyttöönottaminen Helmet-kirjastoissa : loppuraportti [verkkodokumentti, viitattu 9.6.10]. Saatavissa: http://www.rfidlab.fi/sites/rfidlab.fi/files/pictures/HelMet_loppuraportti_Internet_version_20100204.pdf.
- [28] RFID-tietomalli kirjastoille [verkkodokumentti, viitattu 9.6.2010]. Saatavissa: <http://www.kansalliskirjasto.fi/kirjastoala/standardointi/rfid.html>.
- [29] 3M Digital Library Assistant [verkkodokumentti, viitattu 9.6.2010]. Saatavissa: http://www.mikro-vayla.fi/kirjasto/rfid/media/DLA_US.pdf.
- [30] Finkenzeller, Klaus, RFID handbook : radio-frequency identification fundamentals and applications. Chichester: Wiley. 1999.
- [31] Honkanen, Mika, RFID-standardointi kaupallisten sovellusten tukena. Helsingin ammattikorkeakoulu Stadia. Tuotantotalouden koulutusohjelma. Helsinki. 2006.
- [32] RFID 501 : RFID Standards for Libraries [verkkodokumentti, viitattu 15.6.10]. Saatavissa: <http://multimedia.3m.com/mws/mediawebserver?mwsId=66666UuZjcFSLXTtNXMyIxf2EVuQEcuZgVs6EVs6E666666-->.
- [33] HelMet-kirjastojen RFID-selvitys [verkkodokumentti, viitattu 15.6.10]. Saatavissa: <http://pandora.lib.hel.fi/julkaisut/RFID-raportti.pdf>.
- [34] Rieback, M.R. - Crispo, B. - Tanenbaum, A.S, The Evolution of RFID Security. *IEEE Pervasive Computing* 1 (2006), s. 62 -69.
- [35] DeNoia, Lynn A. - Olsen, Anne L., RFID and Application Security. *Journal of Research and Practice in Information Technology* 3 (2009), s. 209 - 221.
- [36] Gamma, Erich ym., Olio-ohjelmointi - Suunnittelumallit. Oy Edita Ab: Helsinki. 2001.
- [37] FindWindow Function [verkkodokumentti, viitattu 17.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms633499%28VS.85%29.aspx>.
- [38] GetForegroundWindow Function [verkkodokumentti, viitattu 17.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms633505%28VS.85%29.aspx>.

- [39] SendMessage Function [verkkodokumentti, viitattu 17.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms644950%28VS.85%29.aspx>.
- [40] SetWindowsHookEx Function [verkkodokumentti, viitattu 18.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms644990%28VS.85%29.aspx>.
- [41] GetAsyncKeyState Function [verkkodokumentti, viitattu 18.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms646293%28VS.85%29.aspx>.
- [42] CreateFileMapping Function [verkkodokumentti, viitattu 18.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/aa366537%28v=VS.85%29.aspx>.
- [43] MapViewOfFile Function [verkkodokumentti, viitattu 18.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/aa366761%28VS.85%29.aspx>.
- [44] CreateMutex Function [verkkodokumentti, viitattu 18.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms682411%28VS.85%29.aspx>.
- [45] WaitForSingleObject Function [verkkodokumentti, viitattu 18.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms687032%28VS.85%29.aspx>.
- [46] ReleaseMutex Function [verkkodokumentti, viitattu 18.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms685066%28VS.85%29.aspx>.
- [47] Keybd_event Function [verkkodokumentti, viitattu 19.7.10]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms646304%28VS.85%29.aspx>.

Koodi	Tila	Milloin lähetettävä
0	<i>Passiivinen</i>	
		Lainaus- ja palautus-toiminnon päättyessä eli kun lainaus- tai palautus-ikkuna suljetaan.
1	<i>Niteen id-numeron lukeminen - lainaus</i>	
		Kun lainaus-ikkuna on avoinna ja fokus siirtyy nide id - kenttään.
		Jokaisen onnistuneen lainaus-tapahtuman päätteeksi, kun fokus palaa nide id -kenttään ja voidaan lukea seuraava nide id.
2	<i>Niteen id-numeron lukeminen - palautus</i>	
		Kun palautus-ikkuna on avoinna ja fokus siirtyy nide id - kenttään.
		Jokaisen palautus-tapahtuman päätteeksi, kun fokus palaa nide id -kenttään ja voidaan lukea seuraava nide id.
3	<i>Niteen id-numeron lukeminen - pelkkä id:n lähetys</i>	
		Haluttaessa hakea niteen tiedot näkyville niteen id-numero lukemalla. Id-numeron lukemisen jälkeen lähetetään koodi 0.
4	<i>Asiakkaan id-numeron lukeminen</i>	
		Kun lainaus-ikkuna on avoinna ja fokus asiakas id - kentässä.
		Haluttaessa hakea asiakkaan tiedot näkyville asiakkaan id-numero lukemalla myös muulloin kuin lainauksen yhteydessä. Id-numeron lukemisen jälkeen lähetetään koodi 0 . (Jos ei mene johonkin muuhun tilaan kuten lainaus, jolloin lähetetään Koodi 1).
5	<i>Passiivinen - pop-up-ikkuna</i>	
		Kun kesken lainaus- tai palautus-toiminnon avautuu pop-up-ikkuna, joka keskeyttää toiminnan suorittamisen. Pop-up-ikkuna on yleensä asiakkaaseen tai niteeseen liittyvä varoitus / huomautus.

Esimerkki tilojen käytöstä lainaus-tapahtuman yhteydessä.

Järjestys	Suoritettava toiminto	Lähetettävä koodi
1.	Avataan lainaus-ikkuna, fokus siirtyy asiakkaan id -kenttään.	4
2.	Luetaan asiakkaan id, fokus siirtyy nide id -kenttään.	1
3.	Luetaan nide id, joka ei aiheuta huomautuksia / varoituksia. Niteen lainauksen jälkeen fokus palaa nide id -kenttään.	1
4.	Luetaan niteen id, jonka seurauksena avautuu pop up -ikkuna (huomautus / varoitus).	5
5.	Kuitataan pop up luetuksi, jonka seurauksena pop up sulkeutuu ja fokus siirtyy lainaus-ikkunan nide id -kenttään.	1
6.	Lainaus-tapahtuman loppuksi suljetaan / tyhjenetään lainaus-ikkuna. (Mikäli loppuksi tyhjenetään lainaus-ikkuna ja fokus siirtyy asiakas id -kenttään, siirrytään tilaan 4.)	0