

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Maver

UPORABA RADIOFREKVENČNE IDENTIFIKACIJE  
V KNJIŽNICAH

DIPLOMSKO DELO NA  
VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Janez Demšar

Ljubljana 2010



Št. naloge: 00458/2009

Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JURE MAVER**

Naslov: **UPORABA RADIOFREKVENČNE IDENTIFIKACIJE V KNJIŽNICAH**  
**RADIO-FREQUENCY IDENTIFICATION IN LIBRARIES**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Radiofrekvenčna identifikacija (RFID) je tehnologija, ki omogoča identifikacijo označenih objektov na določeni razdalji in zaradi večje prikladnosti postopno zamenjuje identifikacijo s črtno kodo. Tehnologija je uporabna tudi za označevanje knjižničnega gradiva in lahko omogoči preprostejšo izposajo in vračanje knjig, bistven prihranek časa pri inventurnih popisih in do določene mere tudi varstvo proti kraji knjig.

V diplomskem delu opišite tehnologijo RFID in izdelajte prototip aplikacije za uporabo RFID v knjižnicah.

Mentor:

doc. dr. Janez Demšar



Dekan:

prof. dr. Franc Solina



# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani Jure Maver,

z vpisno številko 63030175,

sem avtor diplomskega dela z naslovom:

Uporaba radiofrekvenčne identifikacije v knjižnicah.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom **doc. dr. Janez Demšarja**
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 9.3.2010

Podpis avtorja: \_\_\_\_\_



## **Zahvala**

Zahvaljujem se mentorju doc. dr. Janez Demšarju za pomoč in usmerjanje pri izdelavi diplomske naloge. Zahvaljujem se tudi podjetju LEOSS d.o.o. za posojilo tehnične opreme in podporo v času pisanja diplomske naloge. Zahvalil bi se še staršem in bratu za podporo v času študija ter puncici Nataši za popestritev zadnjih let študija in podporo pri pisanju diplomske naloge.



## **Kazalo vsebine**

Povzetek .....	1
Abstract.....	3
1 Uvod .....	5
2 Uporabljene tehnologije .....	6
2.1 Radio frekvenčna identifikacija (RFID) .....	6
2.1.1 Aktivni odzivniki RFID.....	6
2.1.2 Pasivni odzivniki RFID .....	7
2.1.3 Frekvenca delovanja sistemov RFID.....	7
2.1.4 RFID standardi .....	8
2.1.5 Odzivniki za uporabo v knjižnicah in varnostna vrata .....	11
2.1.6 RFID čitalnik .....	12
2.1.7 Programska knjižnica za komuniciranje s čitalnikom .....	13
2.2 Povezava z operacijskim sistemom .....	15
2.2.1 Pozivanje .....	15
2.2.2 Serijska vrata .....	16
2.2.3 Območje za obvestila.....	17
2.2.4 Povezava s tipkovnico .....	17
2.2.5 Podatkovna baza Microsoft Office Access.....	18
3 Izvedba programa Knjižnica.....	19
3.1 Zagon programa.....	20
3.2 Izposoja .....	22
3.3 Urejanje nalepk RFID .....	25
3.4 Simulator tipkovnice.....	26
4 Navodila za uporabo programa Knjižnica .....	28
4.1 Izposoja .....	29
4.2 Urejanje RFID nalepke .....	30
4.3 Simulator tipkovnice.....	31
4.4 Dnevnik.....	32
4.5 Napaka .....	32
5 Prihodnost tehnologije RFID.....	33
6 Zaključek .....	38
7 Literatura .....	40



## **Kratice**

<b>AFI</b>	Application Family Identifier; Naslovni prostor za identifikacijo vrste aplikacije
<b>EPC</b>	Electronic Product Code; Elektronska številka izdelka
<b>HF</b>	High Frequency; Visoka frekvenca
<b>ISO</b>	International Organization for Standardization; Mednarodna organizacija za standardizacijo
<b>LF</b>	Low Frequency; Nizka frekvenca
<b>RFID</b>	Radio Frequency Identification; Radiofrekvenčna identifikacija
<b>UHF</b>	Ultra High Frequency; Ultra visoka frekvenca
<b>UML</b>	Unified Modeling Language (Poenoteni jezik modeliranja)
<b>XML</b>	Extensible Markup Language (razširljiv označevalni jezik)



## **Povzetek**

Namen diplomskega dela je razviti aplikacijo, ki omogoča knjižnicam prehod iz označevanja knjig s črtno kodo na označevanje z odzivniki RFID. Ob tem pa je potrebno nuditi vso programsko podporo, da se njihov delovni proces ne bi bistveno spremenil in bi jim še naprej omogočal uporabo že obstoječe programske opreme.

V delu bo predstavljena zgodovina in osnove tehnologije RFID. Spoznali bomo kaj je to odzivnik RFID in vrste le-teh, ki jih poznamo. Predstavljen bo razvoj standardov na tem področju in opis posameznih standardov. Spoznali bomo strojno opremo, ki je bila uporabljena za razvoj aplikacije. Podrobneje bo predstavljen način uporabe čitalnika HF Mouse RDL5, na katerem temelji aplikacija. Predstavljen bo tudi potek samega razvoja, načini uporabe končne aplikacije, vse uporabljene tehnologije ter problemi, ki so se dogajali med razvojem.

Spoznali bomo tudi kaj lahko pričakujemo v prihodnosti na področju tehnologije RFID in težave, ki bi se lahko pojavile s širšo uporabo te tehnologije. Še bolj podrobno bo predstavljena prihodnost na področju knjižnic in načini uporabe tehnologije RFID, ki bi olajšali delo zaposlenim in hkrati poenostavi uporabo knjižničnih storitev članom. Predstavljene bodo tudi ideje, ki bi jih lahko v prihodnosti dodali aplikaciji in bi lahko nadomestile določeno obstoječo programsko opremo v knjižnicah.

**Ključne besede:** radio-frekvenčna identifikacija, tehnologija RFID, odzivnik RFID, knjižnica.



## **Abstract**

The purpose of the (diploma) thesis is to develop an application needed for libraries which want to swap from barcode tagging to the system of tagging with RFID tags. Another part of the application includes a support for the software usually used in libraries. This way their working process would remain almost the same so they could carry on using their already existing software.

The thesis presents the history and basic facts about the RFID technology. It explains what a RFID tag means and represents which of the RFID types are available. Furthermore, the development of different standards including their description is presented. All hardware used for the application is described along with the instructions for the correct usage of RFID reader HF mouse RDL5 on which the application is based on. The thesis also discusses the development of the application, the technologies which are used for it and problems which have occurred during its development.

The last part of the work presents the future of RFID technology, along with the troubles that might occur using this technology on daily basis. Here, more detailed future plans for libraries are presented. The thesis also discusses the options to use the RFID technology for the simplification of the working process for library employees and members. (In conclusion,) the ideas about next steps in development of the application are presented.

**Keywords:** radiofrequency identification, RFID technology, RFID tag, library.



## 1 Uvod

Prvi začetki radiofrekvenčne identifikacijske (RFID) tehnologije naj bi segali v čas druge svetovne vojne. Britanci, Nemci in Japonci so uporabljali radarje za zaznavo letal vendar jim sam radar ni zadostoval za podatek ali gre za sovražno ali domače letalo. Tako so Nemci ugotovili, da če pilot v letu proti domači postojanki zavrti svoje letalo se bo ta sprememba videla tudi na radarju in tako so vedeli, da ni sovražno letalo, kar v bistvu pomeni, da je to bil prvi pasivni sistem RFID.

Tudi Britanci so razvili svoj način prepoznave domačih letal, vendar na nekoliko drugačen način. Vsakemu letalu so dodali oddajnik, kateri je ob sprejemu signala iz britanske baze začel oddajati povratni signal. Na istem principu še danes delujejo sistemi RFID. Signal se pošlje sprejemniku, ta pa potem bodisi vrne nek signal (pasivni sistem) bodisi začne oddajati signale (aktivni sistem).

Po končani vojni so znanstveniki začeli z razvojem sistemov RFID in hitro se je pojavila ideja, da bi to tehnologijo lahko uporabili za oddaljeno prepoznavo stvari. Z odzivniki RFID so se začeli opremljati vrednejši artikli v trgovinah in s tem so se pojavila tudi prva enostavna vrata RFID proti kraji izdelkov v trgovinah. Pojavile so se ključavnice, ki se jih je dalo odkleniti le z ustrežno kartico RFID.

Poleg izdelkov so se odzivniki RFID kmalu začeli uporabljati tudi pri označevanju živali. Tako so prve odzivnike, kateri so bili v stekleni kapsuli, vstavili v krave s čimer so lahko sledili, katera krava je že dobila zdravilo in koliko, saj je bilo pri čredi več sto ali tisoč krav skoraj nemogoče slediti, katere krave so že dobile zdravilo in katere ne.

Pravi dvig popularnosti pa je RFID dosegel leta 1999, ko sta David Brock in Sanjay Sarma, prišla do sklepa, da bi lahko to tehnologijo uporabili v dobavnih verigah. Ideja je bila, da se razvije odzivnike RFID za identifikacijo palet izdelkov, le-te pa naj bi vsebovali le serijsko številko. Na tak način se v povezavi z podatkovno bazo, lahko sledijo vsi izdelki, ki so na določeni paleti izdelkov.

Z razvojem cenovno ugodnejših nalepk so se odprle tudi nove možnosti uporabe in tako so se začele pojavljati prve knjižnice, ki so svoje knjige poleg nalepke s črtno kodo opremile tudi z nalepkami RFID. S tem so omogočile uporabnikom 24-urni dostop do knjig, olajšale delo zaposlenim in zmanjšale stroške upravljanja [2].

Zadnji odstavek je tudi temelj za to diplomsko nalogo. Cilj je ustvariti program, ki bi bil namenjen tako začetnemu opremljanju knjižnice z nalepkami RFID, kot tudi kasnejši uporabi v kombinaciji s programsko opremo, katero knjižnice že imajo. S programom bo možno brati in pisati na odzivnike RFID, prav tako bo možno vpisovati kontrolni znak, s katerim označimo ali je knjiga na izposoji ali ne. Za lažje in bolj učinkovito delo s programom bodo potrebne tudi določene povezave, kot je povezava s čitalnikom RFID, tipkovnico in operacijskim sistemom Microsoft Windows. Predstavljene bodo tudi kasnejše možnosti uporabe nalepk RFID v knjižnicah in na drugih področjih uporabe.

## 2 Uporabljene tehnologije

Osnovo za izdelavo programa Knjižnica predstavlja tehnologija RFID. Ob tem pa so ji v pomoč tudi številne druge tehnologije, kot so podatkovna baza, povezava s tipkovnico, pozivanje in nekatere funkcionalnosti operacijskega sistema Microsoft Windows. V nadaljevanju bodo predstavljene posamezne tehnologije, ki so bile uporabljene za izdelavo programa Knjižnica.

### 2.1 Radio frekvenčna identifikacija (RFID)

Oznaka RFID je kratica za Radio-frekvenčno identifikacijo (ang. Radio Frequency Identification). Gre za identificiranje s pomočjo elektromagnetnega valovanja na področju radijskih frekvenc, glavna prednost pa je, da je to prepoznavanje lahko tudi brezkontaktno, kar pomeni, da odzivnik ne potrebuje direktnega vidnega stika s čitalnikom. Sestavni deli sistema RFID so odzivniki RFID, čitalniki RFID in antene RFID. Sami odzivniki so sestavljeni iz dveh delov. Iz majhnega integriranega vezja (čip) in radio-frekvenčne antene (Slika 1). Ob vstopu odzivnika v območje dosega čitalnika, ta preko ustrezne antene in s pomočjo radijskih valov prebere podatke, ki so zapisani v odzivniku. Glede na čitalnik je možno prebrati tudi več odzivnikov v sekundi. Ob branju je možno tudi vpisovanje novih podatkov v odzivnik, če le-ta to omogoča. Obstaja več različnih oblik odzivnikov: etikete, katerim se reče tudi pametne nalepke, kartice, obeski, steklene cevke, ploščice, diski, škatlice in podobno. Uporabljajo se za označevanje in sledenje živali, izposoji knjig, za kodiranje ključev, prepoznavanje artiklov v trgovinah, registraciji delovnega časa in drugo. Glede na prisotnost baterije poznamo aktivne in pasivne odzivnike [3, 4].



Slika 1: Prikaz antene in integriranega vezja odzivnika RFID  
(Vir: <http://www.jltrfid.com/default.asp?sec=61&langid=2>)

#### 2.1.1 Aktivni odzivniki RFID

Takšni odzivniki vsebujejo poleg antene in čipa tudi baterijo, katera ima življenjsko dobo od 5 do 10 let. S to baterijo se napaja notranje vezje odzivnika in omogoča oddajanje radijskih

valov čitalniku ter komunikacijo s čitalnikom. Zaradi dodane baterije so takšni dražji vendar omogočajo bistveno daljši domet saj jih je možno zaznati tudi do 30 metrov. Z dodatnimi baterijami, pa je možno doseči celo razdaljo 100 metrov. Največkrat se uporabljajo, ko je potreben daljši domet, za označevanje vrednejših izdelkov in v primerih, ko se uporabljajo v bližini kovin ali tekočin, zaradi katerih bi bilo oddajanje radijskih valov popačeno ali v celoti onemogočeno. Pogosto se uporabljajo tudi na vrednejših izdelkih, saj so lahko narejeni tako, da ob nasilni odstranitvi oddajo signal, ki ga zaznajo varnostne naprave in s tem opozorijo na morebitno krajo izdelka. Ker je signal tudi močnejši, ga je težje zadušiti in na tak način ukrasti izdelek. Zaradi baterije, pa niso le dražji in imajo krajšo življenjsko dobo, ampak so tudi večji, zato so neprimerni za označevanje nekaterih izdelkov. Prav tako niso primerni za vstavev v živali. Pogoji za uporabo je tudi dostop do teh izdelkov, saj je po izpraznitvi baterije le-to, ali celoten odzivnik, potrebno zamenjati [2].

### 2.1.2 Pasivni odzivniki RFID

Pasivni odzivniki ne vsebujejo baterije in se napajajo direktno preko čitalnika. Tak sistem s pasivnimi odzivniki deluje tako, da ko se odzivnik nahaja dovolj blizu antene RFID, sprejme valove, ki jih antena oddaja in si s tem zagotovi napajanje za delovanje. To mu omogoča, da po nekem standardu s čitalnikom vzpostavi komunikacijo, v okviru katere si izmenjata zelene podatke. Ker so cenovno dostopni, majhni, odporni na okolje in imajo dolgo življenjsko dobo, so bolj primerni za masovno označevanje kot aktivni. Zaznava takšnih kartic je odvisna od frekvence na kateri odzivnik deluje. Uporabljajo se za označevanje vseh manj vrednih izdelkov, za označevanje živali, palet v skladiščih, knjig v knjižnicah in drugod, kjer je potrebno brezkontaktno označevanje. Za označevanje vrednejših izdelkov niso primerni, saj jih je dokaj enostavno odstraniti ali zadušiti. Običajno je dovolj za zadušitev izdelka že zaprtje le-tega v aluminijasto vrečko.

### 2.1.3 Frekvenca delovanja sistemov RFID

Pri identifikaciji RFID moramo vedeti, da se uporabljajo različne radijske frekvence. Najnižje se začnejo pri približno 100 kHz in segajo do več GHz. Vse komponente takih sistemov so nastavljene na točno določeno frekvenco, na kateri potem sistem deluje. Odzivnik, nastavljen na frekvenco 125 kHz, ne more delovati na frekvenci 13,56 MHz. Samo frekvenco se izbere glede na namen uporabe, saj nekatere frekvence omogočajo daljše prepoznavanje odzivnikov, spet druge pa hitrejši prenos podatkov in podobno.

Običajno so odzivniki na eni od naslednjih frekvenc:

- **Nizko frekvenčne** (ang. Low Frequency - LF) 125-135 KHz
- **Visoko frekvenčne** (ang. High Frequency - HF) 13.56 MHz
- **Ultra visoko frekvenčne** (ang. Ultra High Frequency - UHF) 868-930 MHz
- **Mikrovalovi** (ang. Microwave) 2.45 GHz ali 5.5 GHz

Primerjavo med frekvencami prikazuje spodnja tabela (Tabela 1).

	<b>Nizko frekvenčne (125-135 KHz)</b>	<b>Visoko frekvenčne (13.56 MHz)</b>	<b>Ultra visoko frekvenčne (868-930 MHz)</b>	<b>Mikrovalovi (2.45 ali 5.5 GHz)</b>
<b>Pozitivne lastnosti</b>	-Predrejo skozi večino stvari -Ne sevajo -Cenovno ugodna ostala oprema	-do 1 m oddaljenosti -Tolerantno ob prisotnosti tekočin in kovin -Standardizirano	-Najdaljši doomet (do 30 m brez baterije)	-Dolg doomet -Hiter prenos podatkov -Najmanjši -Najcenejši
<b>Negativne lastnosti</b>	-Oddaljenost običajno manj od 1 m -Počasen prenos podatkov		-Hitro se odbijejo ali absorbirajo ob prisotnosti tekočin in kovin -Stalno izpostavljanje visokim frekvencam škodi zdravju -Draga ostala oprema	
<b>Primeri uporabe</b>	živali, jeklenke, pnevmatike, pivski sodi	prevozna sredstva, knjižnice, tekstilni izdelki, bankovci		

Tabela 1: Primerjava med različnimi frekvencami delovanja sistemov RFID

Poudariti je še treba, da z nižanjem frekvence pri nizko in visoko frekvenčnih odzivnikih, negativne lastnosti naraščajo. Prenos podatkov je zato pri nižjih frekvencah še nižji in doomet odzivnikov je še krajši. Tako je pri nizko frekvenčnih oddajnikih doomet le še kakšnih 10 cm ali manj. Pri ultra visoko frekvenčnih odzivnikih in odzivnikih, ki delujejo na mikrovalovih, se negativne posledice večajo z naraščanjem frekvence. Oprema pri višjih frekvencah je še dražja, valovanje je bolj škodljivo zdravju in še manj je odporno na odbijanje ali absorbiranje valov [5].

#### 2.1.4 RFID standardi

Standardi so ključnega pomena za veliko aplikacij RFID, kot so plačilni sistemi in sledenje dobrinam. Zelo pomembni so tudi pri sledenju zabožnikov, katerih vsebina se v nabavnih verigah večkrat spremeni. Zaradi tega so se v zadnjem desetletju razvili različni standardi za različne frekvence in aplikacije. Poleg že obstoječih standardov pa je še veliko predlaganih, ki so velikokrat že v uporabi, vendar niso še priznani kot standard.

Standardi predpisujejo pravila, po katerih nato odzivniki in čitalniki med seboj komunicirajo. Prav tako predpisujejo, kako so podatki na odzivnikih organizirani in formatirani. V njih je tudi zapisano, na kakšen način odzivnik ali čitalnik opravijo test, če ustrezajo določenemu standardu. In kot zadnje predpisujejo, v katerih aplikacijah se bodo uporabljali in na kakšen način. Za potrditev predpisov v posameznih standardih skrbi Mednarodna organizacija za standardizacijo oz. angleško International Organization for Standardization. Organizacija je bila ustanovljena leta 1947 in jo sestavljajo posamezne organizacije za standardizacijo iz več kot 140 držav. Standardi so prvotno objavljeni v angleškem jeziku, nato pa je vsaka

posamezna organizacija zadolžena, da ga prevede v svoj jezik. Za mednarodno oznako standardov je bila izbrana kratica ISO in z njo se začnejo vsa imena njenih standardov [11].

Prvotni standardi s področja RFID so bili namenjeni identifikaciji živine. Tako je v standardu ISO 11784 definirano, kako so podatki zapisani na odzivniku. V ISO 11785 so predpisana pravila za komunikacijo med čitalnikom in odzivniki. Zatem je bil razvit standard ISO 14443, ki se uporablja za plačilne sisteme in tako imenovane pametne kartice, ki so v določenih državah vgrajene v potne liste. Med pomembnejše standarde sodi tudi ISO 15693, ki se uporablja za identifikacijo oseb. Tako je večina večjih podjetij opremljena s takšnim sistemom za beleženje prihoda in odhoda delavcev, saj mora vsak ob prihodu ali odhodu iz podjetja, svojo unikatno kartico prisloniti k čitalniku. S tem se zabeleži čas, na podlagi katerega se lahko preveri točno prisotnost ali odsotnost posameznega delavca. Za testiranje pripadnosti določenim standardom je bil razvit ISO 18047 in ISO 18046, s katerim se testira učinkovitost tako odzivnikov kot tudi čitalnikov.

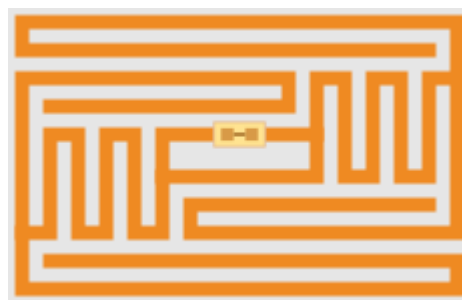
Nato pa se je določevanje standardov nekoliko zapletlo, saj je bila ustanovljena neprofitno združenje med zasebnimi podjetji in pionirji na področju razvoja infrastrukture za sledenje dobrinam z odzivniki RFID, katera se je imenovala Auto-ID Ceter. Vsak njihov odzivnik je vseboval tako imenovano elektronsko kodo izdelka, ki so jo poimenovali EPC (ang. Electronic Product Code). Predpisali so tudi svoja pravila po katerih se sledi tem izdelkom v mednarodnih nabavnih verigah. Cilj je bil ustvariti nizko cenovni sistem RFID, ker bodo odzivniki uporabljeni samo enkrat, saj izdelovalec ne more pričakovati, da bo odzivnike, ki jih je prilepil na svoje izdelke, dobil nazaj za ponovno uporabo. Sistem je moral delovati na ultra visokih frekvencah, saj je le pri tej frekvenci razdalja, na kateri se odzivnike da prebrati, dovolj dolga. Tipičen primer delovanja teh sistemov je prebiranje vsebine odzivnika, pritrjenega na paleto izdelkov, že ob samem prehodu palete skozi skladiščna vrata. Čitalnik pridobi elektronsko kodo palete, na podlagi katere se točno ve, kateri izdelki so na paleti. Ker izdelki običajno potujejo med regijami znotraj države in tudi med državami od enega proizvajalca do vmesnih in na koncu tudi do trgovin, je bilo nujno predpisati globalna pravila za branje takih odzivnikov. Zato so razvili svoj lasten protokol, ki so ga licencirali in postavili edini pogoj, da je brezplačno na voljo tako izdelovalcem kot tudi končnim uporabnikom.

Prvotno je bil načrtovan en sam protokol, ki bi se uporabljal za več različnih razredov odzivnikov. Razrede so nato oštevilčili in jih razvrstili glede na razvitost posameznih odzivnikov, ki so pripadali temu razredu.

V osnovi je bilo načrtovanih naslednjih 5 razredov:

- **Razred 1:** enostavni, pasivni odzivniki, ki komunicirajo s čitalnikom le takrat, ko ta to od njih zahteva. Podpirajo le branje podatkov, ki jih je moč vpisati le enkrat. Brisanje ali spreminjanje podatkov na odzivnikih ni mogoče.
- **Razred 2:** pasivni odzivniki, še vedno komunicirajo s čitalnikom na zahtevo. Dodano je 65 KB spomina, v katerega je podatke možno večkrat vpisati.
- **Razred 3:** pasivni odzivniki, katerim pa je poleg spomina, v katerega je možno tudi pisat, dodana še baterija, zaradi katere jih je možno prebrati na daljši razdalji.
- **Razred 4:** aktivni odzivniki z vgrajeno baterijo, ki poganja tako mikročip kot tudi anteno, s katero se signal oddaja čitalniku.
- **Razred 5:** aktivni odzivniki, ki so poleg komuniciranja z več čitalniki sposobni tudi komunicirati z drugimi odzivniki, kateri tudi pripadajo razredu 5.

Kasneje so razvili še razred 0, na katerega se že ob izdelavi vpiše nespremenljiva vrednost. Zasnovan je bil na drugačnem protokolu, kar je pomenilo, da so uporabniki morali kupiti čitalnike, ki so podpirali več protokolov, če so želeli brati odzivnike kateri so pripadali razredu 0 in 1. Oba razreda sta bila priznana kot standarda EPC. Oba standarda pa imata tudi pomanjkljivosti. Ena od pomanjkljivosti je, da posamezen standard ne more sodelovati s sistemom, ki temelji na katerem drugem razredu. Verjetno še večja pomanjkljivost pa je, da nista združljiva z standardi ISO in da jih ni možno uporabljati globalno, saj razred 0 sprejme signal na eni frekvenci, povratni signal pa pošlje na drugi, kar je v Evropi celo prepovedano. So pa takšni odzivniki zelo pogosti v Združenih državah Amerike, kjer je družba Wal-Mart Stores, ki velja za največjo verigo diskontnih trgovin in trgovskih centrov v svetu[13], zahtevala od svojih 100 največjih dobaviteljev, da vse pošiljke označijo z odzivniki EPC (Slika 2) [12].



Slika 2: odzivnik EPC, kakršnega uporabljajo v trgovski verigi Wal-Mart  
(Vir: <http://en.wikipedia.org/wiki/File:EPC-RFID-TAG.svg>)

Zaradi tega so leta 2004 začeli pri EPC razvijati nov protokol druge generacije, katerega cilj ni bil kompatibilnost z razredom 1 ali razredom 0. Želeli so ustvariti globalen standard, kateri bi se bolj približal ISO standardom, kar jim je tudi uspelo. Edina točka spora je bil naslovni prostor, v katerem bi se določilo, kateri vrsti aplikacij odzivnik pripada, kar v angleščini poimenujejo Application Family Identifier oz. s kratico znak AFI. To 8 bitno kodo oz. kontrolni znak vsebujejo vsi standardi ISO. EPC druge generacije pa v standardu nima zapisano, da je obvezen, čeprav ima 8 bitni blok, katerega se lahko uporabi tudi kot znak AFI. V primeru potrditve standarda EPC kot standarda ISO, bi ga dodali k enemu od že obstoječih standardov ISO.

Med tem so pri ISO razvili serijo standardov, kateri so bili namenjeni odzivnikom za uporabo v nabavnih verigah. Standard se imenuje ISO 18000 in je sestavljen iz 7 delov.

- **18000–1:** Opisuje generične parametre, kateri so enaki za vse globalno sprejemljive frekvence.
- **18000–2:** Pravila za komunikacijo na frekvenci 135 KHz
- **18000–3:** Pravila za komunikacijo na frekvenci 13,56 MHz
- **18000–4:** Pravila za komunikacijo na frekvenci 2,45 GHz
- **18000–5:** Pravila za komunikacijo na frekvenci 5,8 GHz
- **18000–6:** Pravila za komunikacijo na frekvenci od 860 MHz do 930 MHz
- **18000–7:** Pravila za komunikacijo na frekvenci 433,92 MHz

11. julija 2006 je le prišlo do potrditve standarda EPC druge generacije kot standarda ISO. S tem je postal del standarda ISO 18000–6.

### **2.1.5 Odzivniki za uporabo v knjižnicah in varnostna vrata**

V Sloveniji so v uporabi odzivniki, katerih arhitektura in delovanje je predpisana v združenem standardu podjetja Texas Instruments za frekvenco 13.56 MHz. Narejeni so tako da ustrezajo sistemom RFID, ki delujejo po standardu ISO/IEC 15693 (drugi in tretji del) in standardu ISO/IEC 18000 (tretji del).

Dodatne lastnosti odzivnikov:

- delovna frekvenca 13.56 MHz;
- 256 bitni uporabniški spomin, ki je razdeljen na 8 blokov velikosti 32 bitov;
- omogočajo uporabniško in tovarniško zaklepanje blokov;
- vsebujejo naslovni prostor, s katerim opredelimo odzivnik, kateri vrsti aplikacije pripada (AFI, ang. Application Family Identifier);
- predpisuje hitro in simultano prepoznavanje več odzivnikov hkrati (do 50 odzivnikov v sekundi).

Vsi odzivniki narejeni na temeljih tega standarda imajo 64 bitno unikatno serijsko številko, ki je ni možno spremeniti. Zaradi tega je dovolj, da se pri uporabi v knjižnicah vsaka unikatna serijska številka v podatkovni bazi poveže s črtno kodo, običajno pa se na samo kartico tudi zapiše črna koda knjige. Dogovorjeno je tudi, da se za uporabo odzivnikov v knjižnicah uporablja dva znaka AFI v šestnajstiškem zapisu. Na podlagi tega zapisa se odzivajo vrata proti kraji knjig. Znak C2 tako pomeni, da je knjiga izposojena. Ob prehodu uporabnika s to knjigo skozi vrata (Slika 3) bodo le-ta zaznala, da je knjiga bila izposojena in se alarm ne bo vklopil. Če pa uporabnik želi zapustiti knjižnico s knjigo, na kateri je znak AFI 07, kar pomeni, da je knjiga v knjižnici in ni bila izposojena, bodo vrata vklopila alarm ter zvočno in vizualno signalizirala, da gre za prehod knjige, katera ni bila izposojena in s tem opozorila uslužbenca knjižnice, da obstaja verjetnost kraje ali nepravilne izposoje knjige.



Slika 3: Varnostna vrata proti kraji knjig ali drugih izdelkov  
(Vir: [http://www.rfid-library.com/e\\_rf06.html](http://www.rfid-library.com/e_rf06.html))

### 2.1.6 RFID čitalnik

Podlaga pri izdelavi programa Knjižnica je bil čitalnik podjetja Deister Electronics in sicer model RDL5 – HF RFID-Mouse (Slika 4). Čitalnik sodi med najmanjše čitalnike RFID, vendar je kljub temu zelo zmogljiv. Sposoben je brati in pisati podatke na nizko in visoko frekvenčnih odzivnikih. Priklop na računalnik je enostaven s pomočjo 1.5 metra dolgega kabla USB. Čitalnik ne potrebuje dodatnega napajanja, saj mu zadostno količino energije zagotavlja že sam kabel USB. Priloženi so gonilniki, ki podpirajo tako 32-bitne kot tudi 64-bitne operacijske sisteme.

Zaradi svoje majhnosti in prenosljivosti lahko čitalnik HF Mouse v kombinaciji s prenosnim računalnikom predstavlja manjši identifikacijski center v eni od čitalnic knjižnice, kar lahko pomeni velik prihranek časa ob prvem opremljanju knjig z odzivniki RFID. Ob opremljanju več deset tisoč knjig, lahko ravno ta fleksibilnost zmanjša čas tega opremljanja za več dni, če ne celo več tednov, saj ni potrebno nositi vseh knjig k fiksnem računalniku, ampak se prenosni računalnik in čitalnik RFID prinese v bližino knjig.



Slika 4: Čitalnik RDL5 – HF RFID – Mouse  
(Vir: [http://www.deisterrfidportal.de/\\_docu\\_pdf/rf\\_rd15\\_eo.pdf](http://www.deisterrfidportal.de/_docu_pdf/rf_rd15_eo.pdf))

### 2.1.7 Programska knjižnica za komuniciranje s čitalnikom

Pri podjetju Deister Electronics, so v podporo svojemu čitalniku napisali razred DLLConnection, ki služi za povezavo med programom in čitalnikom RFID. V razredu so deklarirane vse funkcije, ki jih čitalnik podpira. Znotraj vsake funkcije je klic osnovnih funkcij, katere dejansko uporabljajo funkcionalnosti čitalnika RFID in so zajete v datoteki deBus.dll. Ker je datoteka zakodirana, nimamo vpogleda v same funkcije in jih uporabljamo kot tako imenovano črno škatlo. Prav tako so deklarirane vse spremenljivke in strukture, katere posamezne funkcije potrebujejo bodisi kot argument, bodisi jih vračajo kot rezultat. Ker je datoteka deBus.dll in funkcije znotraj nje napisane v programskem jeziku C in za svoje delovanje uporablja kazalce, pride do opozoril pri uporabi teh funkcij v programskem jeziku C#. Ker programski jezik C# uporablja tako imenovan proces čiščenja pomnilnika, se dejanske lokacije spremenljivk v pomnilniku neprestano spreminjajo. Zaradi tega je uporaba kazalcev rizična, saj lahko proces med samim izvajanjem programa, spremeni lokacijo spremenljivk in kazalec bi kazal na napačen del pomnilnika. Zaradi tega je potrebno vsako funkcijo ali del kode, kateri uporablja kazalce zajeti v blok »unsafe«. S tem programskemu jeziku C# povemo, da so tukaj uporabljeni kazalci, s čimer bo proces čiščenja pomnilnika, te spremenljivke v pomnilniku ohranil na svojih prvotnih lokacijah. Za delovanje programa kateri vsebuje blok unsafe, pa je potrebno to še eksplicitno potrditi v lastnostih samega programa, kjer je potrebno obkljukati izbiro »Allow unsafe code«.

Primer uporabe unsafe bloka

```
unsafe
{
    //nemotena uporaba kazalcev znotraj tega bloka
}
```

Za delovanje programa smo uporabili pet funkcij in strukture, katere so potrebne za delovanje le-teh.

### **2.1.7.1 UDLOpen**

Za vzpostavitev povezave med računalnikom in čitalnikom RFID je potrebno uporabiti funkcijo UDLOpen(int comport, int baudrate), ki za argument sprejme številko serijskih vrat, na katera je priklopljen čitalnik RFID in število, ki predstavlja koliko simbolov na sekundo RFID čitalnik lahko sprejme.

### **2.1.7.2 UDLPolling**

Kadar želimo pridobivati vsebino in serijsko številko odzivnika RFID avtomatsko, moramo čitalnik RFID stanje pripravljenosti. V takšnem stanju lahko preverjamo, če je v dosegu kakšen odzivnik RFID. Za preklon v takšno stanje poskrbi funkcija UDLPolling(byte address, reg POLLING polling). Za argument sprejme naslov, ki je v šestnajstiškem zapisu 3F in referenco na strukturo POLLING, katera je tudi del razreda DLLConnection. Metoda za delo uporabi določene spremenljivke znotraj te strukture, v njo pa tudi vrne podatke, katere je vrnil čitalnik RFID s pomočjo klica funkcije v datoteki deBus.dll. Struktura lahko tako sprejme podatke o serijski številki, podatke zapisane na odzivniku RFID, kontrolno številko, ki pove za katero vrsto odzivnika RFID gre, velikost podatkov in serijske številke in druge podatke.

### **2.1.7.3 UDLReadBlock**

Po zaznavi odzivnika RFID in pridobitvi njegove serijske številke, lahko preberemo tudi podatke zapisane v spominu. Za to je potreben klic funkcije UDLReadBlock(byte address, byte number, int tagType, byte blockNR, byte[] serialnumber, int length, ref READBLOCK readblock). Argumenti v vrstnem redu so: naslov čitalnika, število blokov, ki jih želimo prebrati in začetni blok, znak, za katero vrsto odzivnika RFID gre, serijska številka, dolžina serijske številke in referenca na strukturo tipa READBLOCK. Sama struktura je podobna strukturi POLLING, saj se tudi vanjo zapišejo podatki o serijski številki, vrsti odzivnika RFID, dejanski podatki zapisani na odzivniku RFID in drugi podatki.

### **2.1.7.4 UDLWriteBlock**

Če želimo v odzivnik RFID vpisati podatke ali le spremeniti že obstoječe, je potrebno uporabiti funkcijo UDLWriteBlock(byte address, byte nob, int tagTyp, byte blockNR, byte[] serial, int length, byte[] data). Za uspešen klic te funkcije, so potrebni skoraj identični argumenti kot za klic funkcije UDLReadBlock(), razlika je le v tem, da ji ne podamo reference na strukturo READBLOCK, ampak dejanske podatke, ki jih želimo vpisati v spomin odzivnika RFID. Izvrši se klic funkcije, ki pripada datoteki deBus.dll in podatki se vpišejo.

### **2.1.7.5 UDLWriteAFI**

Pripadnost odzivnika RFID posamezni vrsti aplikacije označujemo z znakom AFI. Pri knjižnicah se ta znak uporablja za signal varnostnim vratom ali je knjiga izposojena ali ne. Čitalnik RDL5 ne podpira branja tega znaka, ampak le vpis s pomočjo funkcije UDLWriteAfi(byte address, int tagTyp, byte[] serialnumber, int length, byte afi). Za argumente potrebuje naslov čitalnika RFID, vrsto odzivnika RFID, serijsko številko in njeno dolžino ter znak, ki predstavlja ali je knjiga izposojena ali ne.

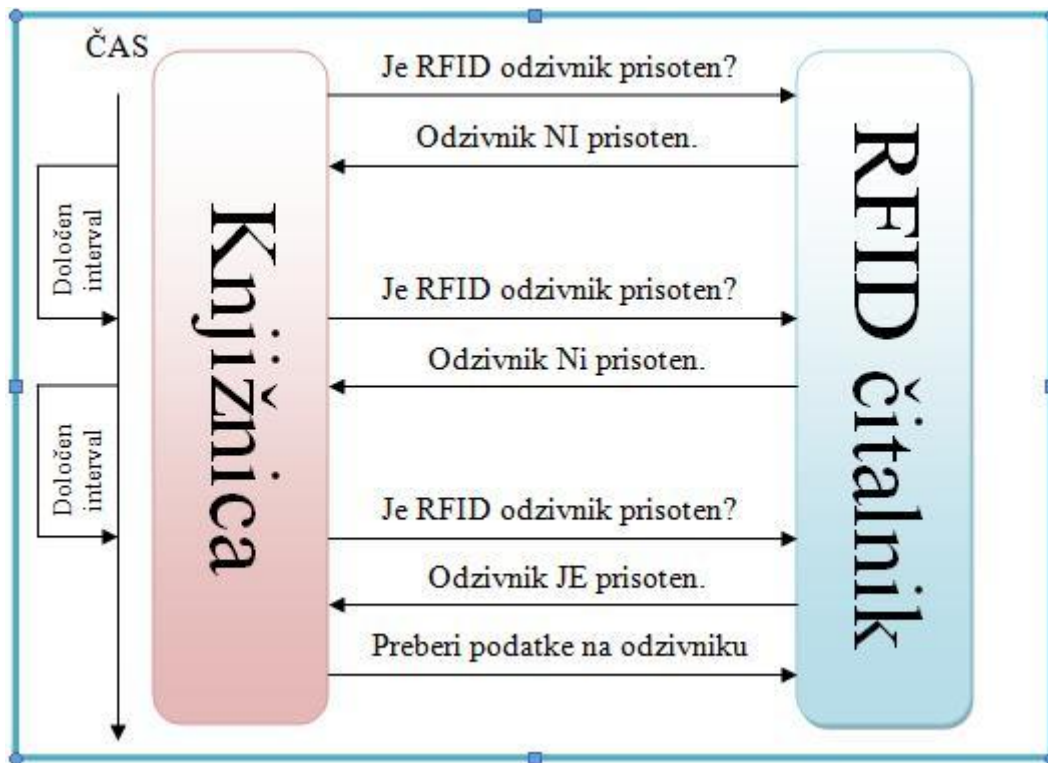
## **2.2 Povezava z operacijskim sistemom**

Da bi program Knjižnica deloval tako, kot je bil zastavljen, je bilo potrebno uporabiti tudi nekatere funkcije in lastnosti operacijskega sistema Microsoft Windows. Uporabljen je bil tudi dodaten program, in sicer Microsoft Office Access, s katerim je bila izdelana podatkovna baza. Za razširitev funkcionalnosti pa je bilo potrebno tudi ustvariti določene povezave s tipkovnico in uporaba območja za obvestila, ki je del sistema Microsoft Windows.

### **2.2.1 Pozivanje**

Pozivanje oz. s tujko polling je v računalništvu izraz za aktivno preverjanje statusa zunanje naprave s pomočjo programa, le-ta lahko teče sočasno z drugimi nalogami. Pozivanje je lahko izvedeno na dva načina. Računalnik je lahko v stanju, ko je nadaljevanje odvisno od zunanje naprave. V tem primeru je edina naloga računalnika, da preverja status zunanje naprave in v primeru, da je ta status neustrezen, počaka določen interval in preveri status še enkrat, dokler status ne ustreza in lahko nadaljuje z izvajanjem ukazov. Drugi način pa je, da je to preverjanje vzporedno z drugimi dejavnostmi. Pozivanje v takšnem primeru deluje enako kot pri prvem primeru z razliko, da računalnik ne čaka tega statusa ampak opravlja ves čas tudi druge naloge, pozivanje pa prepusti ločeni aplikaciji, ki tudi poskrbi za nadaljevanje ukazov, ki ustrezajo statusu zunanje naprave. Pozivanje je običajno realizirano na najnižjem nivoju strojne opreme. Za primer, pozivanje se odvija na paralelnih vratih za tiskalnik s čimer se preverja status znaka, ki pove ali je tiskalnik pripravljen za tiskanje ali ne [6].

V primeru aplikacije Knjižnica se izvaja pozivanje čitalnika RFID, in sicer če je v dosegu kakšen odzivnik RFID. Če čitalnik RFID dobi odgovor, da odzivnika ni v dosegu, se pozivanje ponovi, po preteku vnaprej določenega intervala. Ob prisotnosti odzivnika, se začnejo izvajati metode potrebne za prepoznavo serijske številke kartice in branje podatkov na kartici (Slika 5).



Slika 5: Prikaz delovanja pozivanja med programom Knjižnica in čitalnikom RFID

### 2.2.2 Serijska vrata

Serijska vrata predstavljajo fizični priključek, preko katerega poteka prenos bit za bitom. Značilna so bila vse računalnike, ki so podpirali IBM-ovo tehnologijo. Računalniki so jih prepoznali kot komunikacijska vrata, zaradi tega se je začela uporabljati kratica vrata COM. Na takšne priključke so bile priključene miške, tipkovnice, modemi, čitalniki črtne kode, starejši mobilni telefoni in starejši tiskalniki. Bila so namenjena tudi povezovanju dveh računalnikov in prenosu večjih datotek med njima. Serijska vrata so omogočala določene nastavitve, kot so hitrost, število bitov za posamezen znak, parnost in druge nastavitve. Brez pravilno nastavljenih nastavitev običajno naprave niso delovale[7].

S pojavom priključka USB so serijska vrata počasi začela izginjati, vendar so še vedno prisotna. Razlika je v tem, da so bila fizična serijska vrata popolnoma izpodrinjena, ostala pa so v obliki navideznih serijskih vrat, katera so v uporabi še danes. Ostala so zaradi podpore več zunanji napravam in jih je za razliko od fizičnih vrat možno poimenovati s poljubnim imenom. Prav tako je možno določiti neomejeno število virtualnih vrat. Edina omejitev je računalniška zmogljivost, saj vsaka navidezna vrata porabijo določeno količino virov. Enako kot serijska vrata tudi virtualna podpirajo vse nastavitve z razliko, da ob spremembi nastavitev običajno naprava še vedno deluje in se uspe povezati z računalnikom, vendar so podatki, ki pridejo do naprave, popačeni in zato neuporabni[8].

Za uporabo programa Knjižnica je potrebno določiti, na katera vrata je čitalnik RFID priklopljen. Ostalih nastavitev se ne spreminja, saj je čitalnik nastavljen na začetne nastavitve, vsakršno spreminjanje teh nastavitev pa pomeni prenehanje ali napačno delovanje čitalnika.

### 2.2.3 Območje za obvestila

To območje se nahaja v opravljeni vrstici in je namenjeno prikazovanju ikon programov, ki prikazujejo obvestila, in prikazovanju ure. Običajno se v območju za obvestila nahajajo ikone programov, ki niso v tistem trenutku prikazani na namizju, vendar še vedno delujejo. Določeni programi prikažejo svoje ikone v območju za obvestila že ob sami namestitvi operacijskega sistema Microsoft Windows. Takšni programi so upravljanje z zvokom, ikona s statusom omrežne povezave, ikona o stanju baterije na prenosnih računalnikih in drugi programi. Za delovanje teh programov in prikaz njihovih ikon skrbi orodje Systray.exe, ki je vključeno v sam operacijski sistem Microsoft Windows. Zaradi samega imena tega orodja, se te ikone velikokrat v angleščini poimenujejo »tray icons«, kar se prevaja kot »ikone systemske vrstice«. Razvijalci sistema Windows s tem poimenovanjem niso zadovoljni, čeprav se enako poimenovanje pojavlja tudi v nekaterih njihovih dokumentacijah. Sami predlagajo ime »ikone za obveščanje« oz. angleško »notification icons«.

Tudi del programa Knjižnica zajema delovanje v območju za obvestila. Pri takšnem delovanju je glavno programsko okno skrito in deluje v ozadju. Program takrat opravlja pozivanje in v primeru prisotnosti kartice prebere njeno vsebino. To vsebino nato pošlje računalniku na tak način, da simulira delovanje tipkovnice. S tem je možno uporabljati program Knjižnica s poljubnim programom, kadar se pojavi potreba po branju vsebine odzivnika RFID. Tako lahko osebe v knjižnici še naprej uporabljajo lastne programe za vodenje članov in izposajo knjig z edino razliko, da ne izgublja časa s prebiranjem črtnih kode in poškodovanimi nalepkami, ampak knjigo le približa čitalniku odzivnikov RFID. V primeru, da se pojavi potreba po ostalih funkcionalnostih programa Knjižnica, lahko uporabnik z dvojnim klikom na ikono priključijo glavno programsko okno na namizje.

### 2.2.4 Povezava s tipkovnico

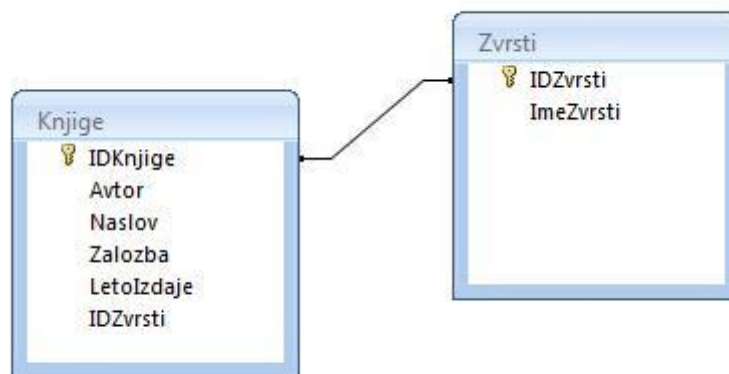
Besedna zveza povezava s tipkovnico je običajno uporabljena, kadar gre za fizično povezavo računalnika s tipkovnico. Lahko pa je uporabljena tudi takrat, ko se potrebuje povezava programa in tipkovnice. Takšnih povezav je lahko več vrst, v programu Knjižnica pa sta uporabljene dve vrsti.

Prva vrsta povezave s tipkovnico se v angleščini imenuje »keyboard hook«, kar se v slovenščino lahko prevaja kot priključitvena točka za tipkovnico. Pri tej vrsti povezave program zaznava vsako tipko na tipkovnici ki jo je uporabnik pritisnil. Povezava na tak način poteka preko sistemskih datotek operacijskega sistema Windows, ki spadajo v dinamično-povezovalno knjižnico. Takšna datoteka je knjižnica, katera vsebuje programsko kodo in podatke, kateri so na voljo več kot enemu programu istočasno. Programi povezani s tipkovnico na tak način, pa lahko predstavljajo tudi varnostno tveganje. Tak program zazna popolnoma vsako pritisnjeno tipko in je velikokrat osnova za računalniške črve, ki berejo in zapisujejo vsako uporabljeno tipko. S tem lahko lastniku črva posredujejo zaupne podatke, kot so uporabniška imena in gesla za bančne račune, račune za spletno pošto in druge zasebne podatke. Ob uporabi požarnega zidu bo le-ta opozoril na tvegano delovanje aplikacije in zahteval dovoljenje za uporabo. Tudi aplikacija Knjižnica potrebuje dovoljenje za uporabo te funkcionalnosti, saj bi v primeru blokade s strani požarnega zidu bilo onemogočeno zaznavanje bližnjic na tipkovnici, s katerimi program zazna, če je uporabnik hotel vklopiti ali izklopiti zaščito na posameznem odzivniku RFID.

Druga vrsta povezave na tipkovnico je simuliranje vnosa s tipkovnice. Pri tej povezavi aplikacija pošlje operacijskemu sistemu znake na enak način, kot jih pošlje tipkovnica. Tako se bodo ti znaki izpisali tam, kjer se trenutno nahaja uporabnik. S tem lahko združimo dve popolnoma neodvisni aplikaciji. Primer uporabe bi bil, ko se uporabnik v svojem programu za izposajo knjig postavi v polje, katero predstavlja črtno kodo ali serijsko številko knjige. Takrat približa knjigo čitalniku RFID in program Knjižnica bo vsebino odzivnika RFID poslal s pomočjo simuliranja tipkovnice operacijskemu sistemu, ki jo bo izpisal v to polje.

### 2.2.5 Podatkovna baza Microsoft Office Access

Microsoft Office Access je sistem za upravljanje relacijskih zbirk podatkov, ki združuje grafični vmesnik in orodja za razvijalce z relacijskim mehanizmom Microsoft Jet za zbirko podatkov[9]. Zaradi grafičnega vmesnika je z podatkovno bazo Microsoft Office Access enostavno ustvariti nove tabele, ustvariti povezave med tabelami in vpisati podatke, kar je tudi eden od razlogov, da je predstavitevna podatkovna baza uporabljena v programu Knjižnica narejena s tem orodjem (Slika 6).



Slika 6: Prikaz relacijskih tabel predstavitevne podatkovne baze programa Knjižnica

Poleg enostavnosti pa je na izbiro podatkovne baze v Microsoft Office Accessu vplivalo tudi to, da podpira objektno programiranje in poizvedbe s pomočjo parametrov. Za povezavo med aplikacijo in samo podatkovno bazo je bil uporabljen programski vmesnik OLEDB, ki pomeni angleško kratico za »Object Linking and Embedding Database«. Vmesnik je namenjen uporabi podatkovne baze na način, ko programerju ni potrebno vedeti vsako tehnično podrobnost in funkcionalnost, ampak ga zanimajo le podatki in poizvedbe po le-teh. V teoriji je za programerja, ki uporablja OLEDB, nepomembno, s katero tehnologijo je ustvarjena podatkovna baza, saj mu jo ta predstavi vedno na enak način, čeprav se občasno izkaže, da le ni tako in je potrebno določene poizvedbe in parametre prilagoditi posamezni podatkovni bazi.

### 3 Izvedba programa Knjižnica

Program Knjižnica je napisan v programskem jeziku C#, v veliko pomoč pri izdelavi pa je bilo okolje Microsoft Visual C# 2008 Express. Na odločitev, da bo program v programskem jeziku C#, je v veliki meri vplivalo praktično izobraževanje, katero je potekalo ravno v C# in Microsoft Visual Studio 2008 okolju. Poleg tega je na odločitev tudi vplivalo dejstvo, da je jezik zelo pogost in je na internetu zaradi tega možno najti veliko podpore, kar bistveno zmanjša čas učenja novosti in izdelave posameznih programov.

Program knjižnica je sestavljen iz štirih ločenih delov, ki so vizualno ujeti v glavno okno, dejansko pa so samostojna okna. Ta okna so:

- Izposoja;
- Uredi RFID nalepko;
- Simulator tipkovnice;
- Nastavitve.

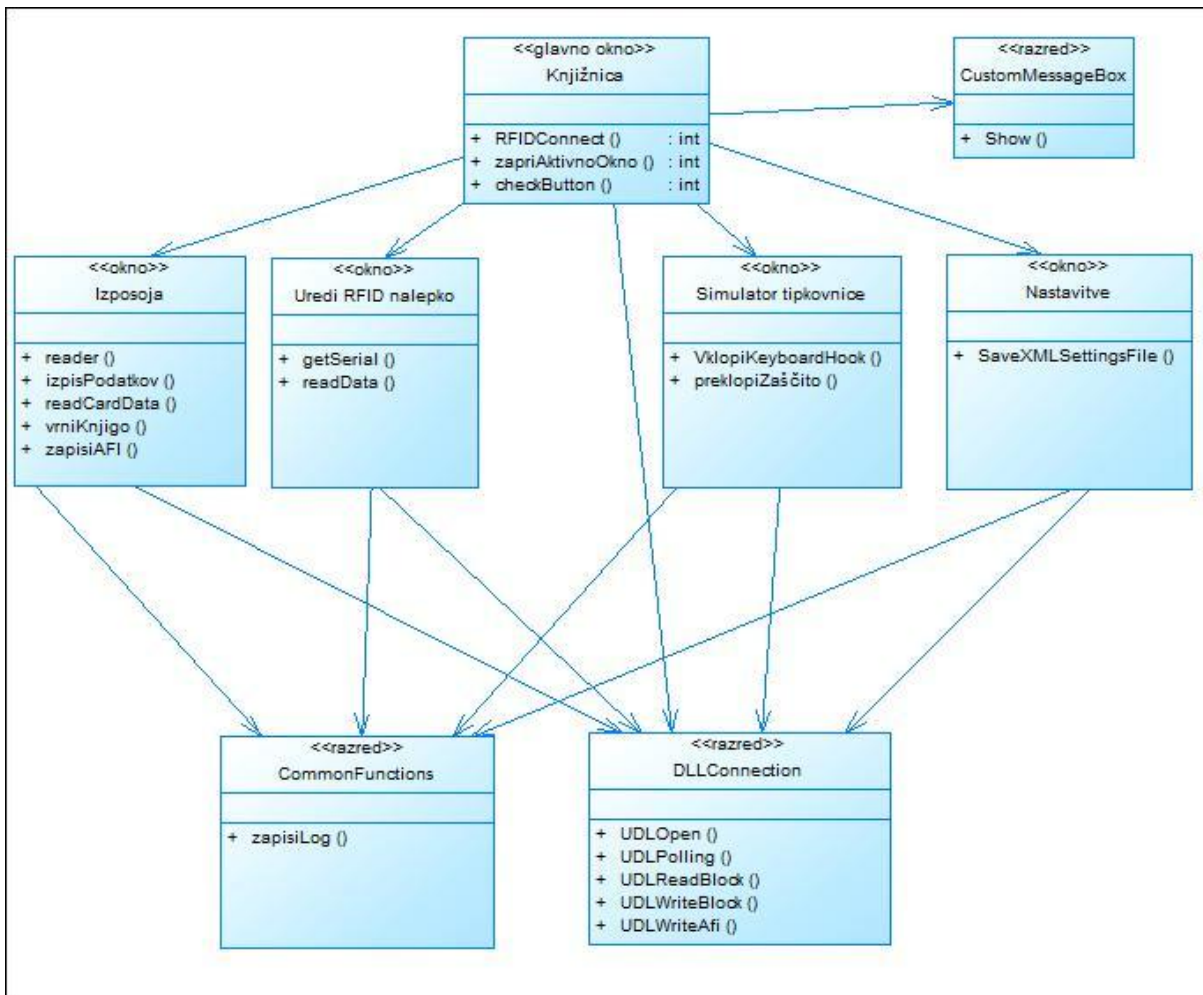
Vsako okno ima več metod, ki se izvajajo ob uporabi posameznih funkcionalnosti programa in so večinoma za vsako okno specifične.

V oknu Izposoja se nahajajo metode potrebne za prikaz podatkov o knjigi. Za to je potrebno najprej prebrati podatke na nalepki RFID. Nato se te podatke poveže s podatki v podatkovni bazi in prikaže. Tukaj je tudi metoda, katera skrbi za vklop in izklop zaščite proti kraji knjig.

Uredi RFID nalepko skrbi za vpis novih podatkov na nalepke RFID. Tako so tukaj metode za branje, vpis in brisanje podatkov na nalepki. Vse tri metode pa za svoje delo potrebujejo serijsko številko nalepke RFID, katero pridobijo v ločeni metodi.

Pri uporabi Simulatorja tipkovnice je potrebno brati uporabnikov vnos s tipkovnico, za kar se uporabi knjižnica operacijskega sistema Windows. V tem načinu je tudi potrebno brati podatke z nalepk RFID in jih izpisati na izhod tipkovnice. Prisotne so še metode za preklop zaščite proti kraji knjig.

Poleg specifičnih metod pa so tudi nekatere, kot je izpis podatkov v dnevnik, skupne. Prav tako je skupen povezovalni razred DLLConnection, ki se uporabi vedno, ko je potrebno dostopati do čitalnika RFID. Tako se preko njega izvajajo metode za pridobitev serijske številke nalepke RFID, branje vsebine pomnilnika in druge. V nadaljevanju bodo podrobnejše predstavljene metode posameznih oken, skupne metode in delovanje programa Knjižnica, za lažjo orientacijo in sledenje pa je tudi diagram UML (Slika 7).



Slika 7: diagram UML predstavljenih metod, oken in razredov programa Knjižnica

### 3.1 Zagon programa

Ob vsakem zagonu programa Knjižnica se v razredu Program kreira začetno okno. Ob tem se inicializirajo elementi okna in kreira se objekt tipa Nastavitve, okno samo pa se ne prikaže. Razred Nastavitve v svoji inicializacijski metodi najprej preveri, če v mapi, iz katere je bil program Knjižnica pogan, obstaja datoteka Settings.xml (Slika 8), v kateri so shranjene osnovne nastavitve potrebne za delovanje programa. Shranjene so v jeziku XML, saj le-ta omogoča formatiran vnos podatkov, katere je možno v programskem jeziku C# enostavno, s pomočjo ukaza ReadXml(), prebrati v objekt tipa DataSet. V primeru, da datoteka ne obstaja, se ustvari in vanjo se vpišejo začetni podatki. Prav tako se globalna zastavica, ki označuje, da nastavitve še niso vpisane, nastavi na true. S tem bo program kot prvo okno prikazal nastavitve programa. Ne glede na to ali datoteka Settings.xml obstaja ali ne, se izvede prebiranje nastavitvev, ki se zapišejo v statični razred AppSettings, od koder so dostopne celotni aplikaciji. Podatka potrebna za delovanje aplikacije sta ime virtualnih vrat na katera je priključen RFID čitalnik in pot do datoteke, katera predstavlja dnevnik aplikacije.

```

<?xml version="1.0" standalone="yes"?>
<DocumentElement>
  <Nastavitve>
    <ComPortRFID>1</ComPortRFID>
    <PotDnevnik>C:\Pot_do_dnevnika\Knjiznica.log</PotDnevnik>
  </Nastavitve>
</DocumentElement>

```

Slika 8: Prikaz primera XML datoteke, kjer so zapisane nastavitve

V inicializacijski metode se še izvede polnjenje vsebine okna Nastavitve. Tako se v polje, ki je namenjeno lokaciji dnevnik datoteke, vpiše dejanska pot, pod pogojem, da so bile nastavitve predhodno shranjene. V izbirni meni za izbiro serijskih vrat, se vnesejo vsa možna vrata, ki so na voljo. Za začetna izbrana vrata se nastavijo tista, ki so shranjena v nastavitvah. Tudi tukaj je pogoj, da so bile nastavitve že predhodno shranjene. S tem je zaključeno kreiranje in polnjenje objektov in program se nadaljuje z vzpostavljanjem povezave s čitalnikom RFID. To se izvrši s klicem metode RFIDConnect(), v kateri, poizkuša vzpostaviti povezavo s čitalnikom RFID. Metoda za povezavo s čitalnikom je UDLOpen(int comport, int baudrate) in pripada razredu DLLConnection. V primeru uspešne povezave se v status vrstici tudi izpiše, da je povezava uspela, v nasprotnem primeru, pa se izpiše, da povezava ni uspela, prikaže se gumb, s katerim uporabnik ponovi vzpostavljanje povezave, prav tako pa ga tudi opozori s pomočjo razreda customMessageBox. V primeru nepričakovane napake se poročilo o napaki prikaže tako uporabniku, kot se tudi zapiše v datoteko Napaka.log s pomočjo statične metode zapisiLog(string serial, string, podatek, string tipPodatka) v razredu CommonFunctions. Izvede se še preverjanje statusa zastavice, ki pove če nastavitve še niso obstajale, in na podlagi te se inicializacija bodisi zaključi, bodisi se prikaže okno Nastavitve, ki je bilo že predhodno kreirano. S tem je program Knjižnica pripravljen za uporabo.

### **Razred CustomMessageBox**

CustomMessageBox je pomožni razred namenjen obveščanju uporabnika. Ker je statičen, ga ni potrebno vsakič kreirati, ampak se ga pokliče z ukazom: customMessageBox.Show(besedilo za izpis, glavna forma). Metoda je bila narejena, ker osnovno sporočilo ogrođa .NET, ne podpira nastavitve očeta. Želja je bila po izpisu sporočila znotraj programa, za kar je bilo potrebno narediti nov razred. Objekti tega sporočilnega okna niso vnaprej deklarirani, ampak se kreirajo ob samem klicu. Prav tako se dodajo dogodki za gumbe. V tem primeru le dogodek za zaprtje sporočila. Pred prikazom, se še nastavi forma, ki predstavlja očeta in sporočilo se prikaže.

### **Metoda zapisiLog()**

Metoda zapisiLog(string serial, string, podatek, string tipPodatka) je namenjena vpisovanju podatkov v datoteko, ki predstavlja dnevnik uporabe in v datoteko, namenjeno shranjevanju podatkov o napakah. Odločitev v katero od datotek se bo podatek vpisal nosi spremenljivka tipPodatka. Metoda vsakič pred pisanjem podatkov, preveri, če datoteka ni večja od 1 MB. V nasprotnem primeru jo skrajša na polovico. Podatku se še doda točen datum in ura, nato pa se formatirano vpiše v datoteko.

Po končani inicializaciji programa, se prikaže začetno okno, sestavljeno iz petih gumbov, statusne vrstice in prostora, kjer se bodo prikazovala podokna. Gumbi so zajeti v kontrolo tipa ToolStrip. Vsakemu je bila določena slika, katera predstavlja posamezno aktivnost, velikost, besedilo in dogodek, kateri se zgodi ob kliku nanj. Aktivnosti, katere so na voljo uporabniku so:

- Izposoja,
- Uredi RFID,
- Simulator tipkovnice,
- Nastavitve,
- Izhod.

Dogodek, ki se sproži ob kliku na gumb izposoja, uredi RFID ali nastavitve je idejno identičen, razlika je le v tem, katero okno se bo uporabniku prikazalo. Ob kliku se izvedejo naslednje vrstice kode:

```
if (zapriAktivnoOkno("UrediRFID"))
{
    checkButton("rfidToolStripButton");
    UrediRFID UrediRFIDOkno = new UrediRFID();
    UrediRFIDOkno.MdiParent = this;
    UrediRFIDOkno.Dock = DockStyle.Fill;
    UrediRFIDOkno.Show();
}
```

Za realizacijo dogodka se uporabi klic metode `zapriAktivnoOkno(string imeOkna)` in če je rezultat resničen se izvajanje programa nadaljuje z metodo `checkButton(string gumb)`. Glede na izbran gumb se kreira okno zelene dejavnosti. Določi se mu oče v katerem se bo prikazal in ukaže se mu, naj se prikaže. Z lastnostjo `»DockStyle.Fill«` določi, da se novo kreirano okno samodejno raztegne na vso razpoložljivo površino znotraj očeta.

### **Metoda `zapriAktivnoOkno()`**

Metoda `zapriAktivnoOkno(string imeOkna)` zapre vsa aktivna okna znotraj glavnega okna. Zapiranje se ne izvede le v primeru, da je spremenljivka `imeOkna` enaka imenu trenutno aktivnega okna, kar pomeni, da je bila izbrana aktivnost, katera je že prikazana. Metoda bo zaradi tega dala negativen odgovor. S tem programu pove, da ni potrebe po odprtju novega okna, saj je zeleno okno že odprto. Ob uspešnem zaprtju aktivnih oken, metoda odgovori pozitivno in s tem, da dovoljenje za kreiranje in prikaz drugega okna.

### **Metoda `checkButton()`**

Gumbi na glavni formi ob izbiri dobijo okvirček, ki simbolizira, da je aktivnost trenutno izbrana. Ob pritisku na drug gumb, je potrebno odstraniti okvirček gumbu, ki je bil predhodno izbran, kar je tudi naloga te metode. V `checkButton(string gumb)` se v zanki `foreach` sprehodi skozi vse gumbe in vsem, razen gumbu, kateri ima isto ime, kot je zapisano v argumentu, odstrani okvir.

## **3.2 Izposoja**

Razred `izposoja` je prva izmed aktivnosti programa `Knjižnica`. Okno ima izklopljen prikaz okvirja in gumbov za upravljanje z okni, saj vedno deluje znotraj glavnega okna programa

Knjižnica in ga ročno ni možno spreminjati. Samo okno je sestavljeno iz treh logičnih delov. Prvi del je namenjen prikazu podatkov in serijske številke odzivnika RFID. V drugem delu se izpišejo podatki iz podatkovne baze, kateri pripadajo prebrani vsebini odzivnika RFID. Zadnji del pa omogoča vklop ali izklop zaščite proti kraji knjig.

### **Branje vsebine odzivnika RFID**

Samo delovanje aktivnosti izposoje je zasnovano na pozivanju čitalnika RFID za odzivniki. Da bi potekalo delovanje programa nemoteno kljub pozivanju, se že pri inicializaciji forme ustvari nova nit programa, v kateri bo delovalo pozivanje. Pred ustvarjanjem nove niti je potrebno preveriti, če ni obstoječa nit že bila ustvarjena. Če je bila, jo program zaustavi, šele nato zažene novo. Vrstica, s katero se zažene nova nit, je:

```
this.readerThread = new Thread(new ThreadStart(this.reader));
```

Ime spremenljivke je readerThread, ciljna metoda pa je metoda reader(), v kateri se odvija dejansko pozivanje.

### **Metoda Reader()**

Metoda Reader() je ujeta v neskončno zanko for (for (; ; )) in se zaključi šele takrat, ko se ustavi nit. Pred popolno porabo zmogljivosti računalnika ga omejuje ukaz Thread.Sleep(500), s čimer se ponavljanje niti ustavi za 500 milisekund. Zaradi dela z razredom DLLConnection, je večji del kode ujet v blok unsafe. Pozivanje se opravlja s funkcijo UDLPolling(). Ob prisotnosti kartice in ob pogoju, da kartica še ni bila prebrana, pride do klica metod izpisPodatkov(string podatki, string tipPodatka) in readCardData(DLLConnection.POLLING dataPolling, string serial). Pogoji, da kartica še ni bila prebrana, se beleži s pomočjo spremenljivke znotraj metode, katera se nastavi na 1 ob odsotnosti odzivnika ali na 0, ko je odzivnik bil uspešno zaznan. S tem se prepreči prebiranje istega odzivnika znova in znova, dokler ga uporabnik ne odstrani z RFID čitalnika. Ob odstranitvi odzivnika se podatki izpisani na zaslonu ne odstranijo takoj, ampak po štirih intervalih brez prisotnosti odzivnika, kar ob nastavitvi intervala 500 milisekund pomeni 2 sekundi.

### **Metoda izpisPodatkov()**

Ker se metoda reader() izvaja v ločeni niti, nima pravice dostopa do glavnega okna. Zaradi tega ni možno direktno vpisati prebrane podatke v posamezne elemente. V takih primerih je potrebno uporabiti tako imenovanega delegata. Delegat je neke vrste kazalec na metodo in je bil prvič predstavljen ravno v programskem jeziku C# kot nadomestek kazalcev na funkcije v programskem jeziku C++. Uporabi se ga vedno, ko je potrebno iz ene niti dostopati do elementov v drugi niti. Vsak element ima v lastnosti InvokeRequired zapisano, če je za dostop potreben klic delegata ali ne. Če je potreben, se najprej opravi deklaracija delegata in nato klic funkcije Invoke(delegat). Metoda, kateri smo pripisali delegata, se nato izvede v glavni in ne več v alternativni niti. V tej glavni niti pa ima metoda dostop do vseh elementov in na tak način lahko izpiše prebrane podatke na zaslon. Primer uporabe delegata:

```
private void izpisPodatkov(string podatki)
{
    if (this.Label.InvokeRequired)
    {
        delegatThreada d = new delegatThreada(izpisPodatkov);
        this.Invoke(d, new object[] { podatki });
    }
    else
        this.Label.Text = podatki;
}
```

Metoda `izpisPodatkov(string podatki, string tipPodatka)` je torej delegat za vpis podatkov iz niti, kjer se odvija metoda `reader()`, v glavno nit programa Knjižnica. V argumentih podamo podatek in vrsto podatka. Na podlagi vrste podatka se sam podatek izpiše v ustrezno polje programa Knjižnica.

### Metoda `readCardData()`

Metoda `readCardData(DLLConnection.POLLING dataPolling, string serial)` je namenjena branju vsebine na RFID odzivniku. Podano serijsko številko odzivnika je potrebno spremeniti v tabelo bajtov. Vsak bajt v tabeli predstavlja dva znaka serijske številke, zapis pa je v šestnajstiški obliki. Tabelo bajtov se nato skupaj z drugimi argumenti poda metodi `UDLreadBlock()`, ki pripada razredu `DLLConnection`. Eden od argumentov je tudi referenca na objekt tipa `DLLConnection.READBLOCK`, ki predstavlja strukturo namenjeno shranjevanju podatkov o odzivniku. V strukturo se zapišejo podatki, dolžina vseh podatkov, serijska številka in druge lastnosti odzivnika. Referenco na nek objekt se opredeli z besedo »ref«. Takšno označevanje pove metodi, da se sklicuje na isti objekt iz prejšnje metode. Vsaka sprememba na tem objektu v klicani metodi se bo poznala tudi v prvotnem objektu, ko se bo klicana metoda zaključila. Če je odgovor klicane metode pozitivno število, je bilo branje uspešno. Prebrani podatki so zapisani v bajtih, in sicer v šestnajstiški obliki, zato jih je potrebno pretvoriti v niz podatkov. Pretvorbo opravimo z naslednjo zanko:

```
string podatki = "";
for (int i = 0; i < (Convert.ToInt32(readblock.size2) * Convert.ToInt32(readblock.nob2)); i++)
{
    if (readblock.data[i] != 0) //dolžina bajta večja od 0
    {
        char c = (char)Int32.Parse(Convert.ToString(readblock.data[i], 16).PadLeft(2, '0'),
        NumberStyles.AllowHexSpecifier); //byte v char
        podatki = podatki + c;
    }
}
```

Tudi ta metoda se še vedno izvaja v ločeni niti, zato je tudi tukaj potrebno za izpis podatkov klicati metodo `izpisPodatkov(string podatki, string tipPodatka)`, ki vsebuje delegata.

### Izpis podatkov o knjigi iz podatkovne baze

Polje, v katero se izpiše črtna koda iz RFID odzivnika, ima nastavljen dogodek »TextChanged«. Tak dogodek se bo sprožil ob vsaki spremembi besedila v tem polju. Ob sprožitvi tega dogodka se najprej preveri, da besedilo v polju ni prazen niz. Nato se s klicem metode `vrniKnjigo(string IDKnjige)` v razredu `DatabaseConn` pridobijo vsi podatki, zapisani v podatkovni bazi, ki pripadajo knjigi s to črtno kodo. Zapišejo se v objekt tipa `DataTable` iz katerega se nato vpišejo v ustrezna polja na zaslonu. Sporočilo o prebrani knjigi se vpiše tudi v dnevnik. Če je pri pridobivanju podatkov prišlo do napake, je ime objekta nastavljeno na »ErrorTable«. V tem primeru se poročilo o napaki vpiše v datoteko `Napaka.log`.

### Metoda `vrniKnjigo(string IDKnjige)`

Metoda `vrniKnjigo(string IDKnjige)` je zasnovana kot univerzalna metoda za pridobivanje podatkov iz podatkovne baze. Ker bi se program Knjižnica običajno uporabljal na že obstoječih podatkovnih bazah knjižnic, bi bilo potrebno ta del programa prilagajati posamezni knjižnici in njihovi podatkovni strukturi. Zaradi tega se za pridobitev podatkov vedno kliče to metodo, v kateri pa se nato poskrbi, da se naprej poveže z metodo, katera pripada specifični

knjižnici. Trenutno edina metoda je povezava s predstavitveno podatkovno bazo in sicer metoda `dobiPodatkeIzAccessBaze(string IDKnjige)`. Ker metoda vrniKnjigo vrača podatke kot tabelo, se v primeru napake, ustvari tabela z imenom `ErrorTable`, v kateri je sporočilo o napaki zapisano v prvi celici prve vrstice.

#### **Metoda `dobiPodatkeIzAccessBaze()`**

Podatkovna baza programa Knjižnica je narejena v Microsoft Accessu, zato je potrebno klicati metodo `dobiPodatkeIzAccessBaze(string IDKnjige)`. Povezava programa s podatkovno bazo in poizvedba po njej poteka s pomočjo OLEDB programskega vmesnika. Podatki se pridobijo z izvedbo naslednje poizvedbe:

```
SELECT k.Avtor, k.Naslov, k.Zalozba, k.LetoIzdaje, z.ImeZvrsti
FROM Knjige k
LEFT JOIN Zvrsti z ON CInt(z.IDZvrsti) = CInt(k.IDZvrsti)
WHERE k.IDKnjige = @IDKnjige
```

Kljub temu, da sta polja `IDZvrsti` tako v tabeli `zvrsti` kot v tabeli `knjige` istega tipa, je ob združevanju bilo potrebno narediti pretvorbo v število. V nasprotnem primeru pride do napake. V `IDKnjige` se vpiše parameter metode. Pridobljeni podatki iz podatkovne baze se nato prepíšejo v tabelo, katero metoda tudi vrne.

#### **Vklop in izklop zaščite za knjige**

Da bi zaščitili knjige pred krajo, je v programu Knjižnica na voljo funkcionalnost za preklop zaščite. Pri preklopu se vpiše ustrezen znak v šestnajstiškem zapisu s pomočjo metode `zapisiAFI(bool vklopiZascito)`. Preklop se lahko izvrši na dva načina. Prvi način je s klikom na gumb `Vklopi` ali `Izklopi`. Drugi način pa je z uporabo tipkovnice. Okno ima deklariran dogodek, ki se sproži ob vsakem pritisku tipke na tipkovnici. Tako se ob kombinaciji tipk `ctrl+F7` kliče metoda z argumentom za vklop in ob kombinaciji `ctrl+F8` za izklop.

#### **Metoda `zapisiAFI(bool vklopiZascito)`**

Tudi ta metoda se izvaja s pomočjo razreda `DLLConnection`, točneje z metodo `UDLWriteAFI()`, kateri se poda šestnajstiško število `07` za vklop ali `C2` za izklop zaščite. Uspešen vklop ali izklop zaščite se vpiše v dnevnik, v primeru napake pa v datoteko `Napaka.log`. V primeru, da ni prisotnega nobenega RFID odzivnika, se izpiše sporočilo, v katerem se uporabnika poziva, naj približa odzivnik k RFID čitalniku.

#### **Zapiranje okna**

Pred zaprtjem okna se sproži dogodek, v kateremu se zaustavi nit v kateri se odvija pozivanje. Na ta način se sprost čitalnik RFID in s tem se na njem lahko izvajajo druge funkcije. V nasprotnem primeru, bi se pozivanje in branje odzivnikov RFID še vedno odvijalo. Poleg zasedenosti čitalnika RFID, bi ob prisotnosti odzivnika, prihajalo do napak, saj delegat ne bi imel ustreznih polj za vpis podatkov.

### **3.3 Urejanje nalepk RFID**

Zaslon za urejanje odzivnikov RFID je enostaven, sestavljen iz dveh polj za tekst in treh gumbov. Ob kliku na gumb preberi podatke se pokliče metoda `getSerial()`. V primeru, da rezultat ni prazen niz, se zgodi še klic metode `readCardData(string serial)`, obe metodi delujeta

po istem postopku kot pri izposoji. Ker branje ne poteka v ločeni niti, se izpis tako serijske kot tudi vsebine odzivnika RFID zgodi neposredno, brez vmesnih funkcij in delegatov.

Zapisovanje podatkov se izvaja v metodi `writeData(string strData, bool clearData)`. Spremenljivka `clearData` določa, ali se bo izvajalo pisanje novih podatkov ali brisanje obstoječih podatkov. Pred zapisom je treba podatke pretvoriti v tabelo bajtov. To se enostavno naredi s pomočjo sistemaškega razreda `ASCIIEncoding`.

```
System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();
byte[] data = encoding.GetBytes(strData);
```

Samo pisanje se realizira s klicem funkcije `UDLWriteBlock()`, ki je del razreda `DLLConnection`. Po brisanju ali pisanju se uporabniku izpiše, če je bilo uspešno ali ne. Prav tako se vsako pisanje ali brisanje vpiše v dnevnik.

### 3.4 Simulator tipkovnice

Ob izbiri gumba za to aktivnost, se najprej skrijejo vsa podokna nato pa še glavno okno in prikaz v opravljeni vrstici, vidna pa postane ikona v območju za obvestila. Uporabniku se tudi prikaže obvestilo, katero izvira iz same ikone, s katerim ga opozori, da aplikacija še vedno teče. Za tem se izvrši klic metode `VklopiKeyboardHook(Form1 form)`, katera pripada razredu `KeyboardHook`.

#### Metoda `VklopiKeyboardHook()`

Metoda `VklopiKeyboardHook(Form1 form)` sprejme za argument objekt, kateri predstavlja glavno okno. S tem je na najbolj enostaven način rešen dostop do te forme, ki je potreben za obveščanje uporabnika o dogajanjih. Pred začetkom, metoda vzpostavi priključitveno točko s tipkovnico, s pomočjo sistemaških knjižnic `user32.dll` in `kernel32.dll`. V knjižnici `user32.dll` uporabimo naslednje funkcije:

- `SetWindowsHookEx`: s to funkcijo ustvarimo povezavo s tipkovnico
- `UnhookWindowsHookEx` – s to funkcijo prekinemo povezavo s tipkovnico
- `CallNextHookEx` – funkcija, s katero se sporočilo pridobljeno s tipkovnice poda na naslednjo priključitveno točko s tipkovnico, katera je mogoče v lasti kakšne druge aplikacije. Ob neuporabi te funkcije, bi se podatki iz tipkovnice v tej točki izgubili in ne bi bili več dostopni.

Poleg tega uporabimo še funkcijo iz knjižnice `kernel32.dll`, `GetModuleHandler`, s katero pridobimo kodo za dostop do modula, ki ga uporablja proces.

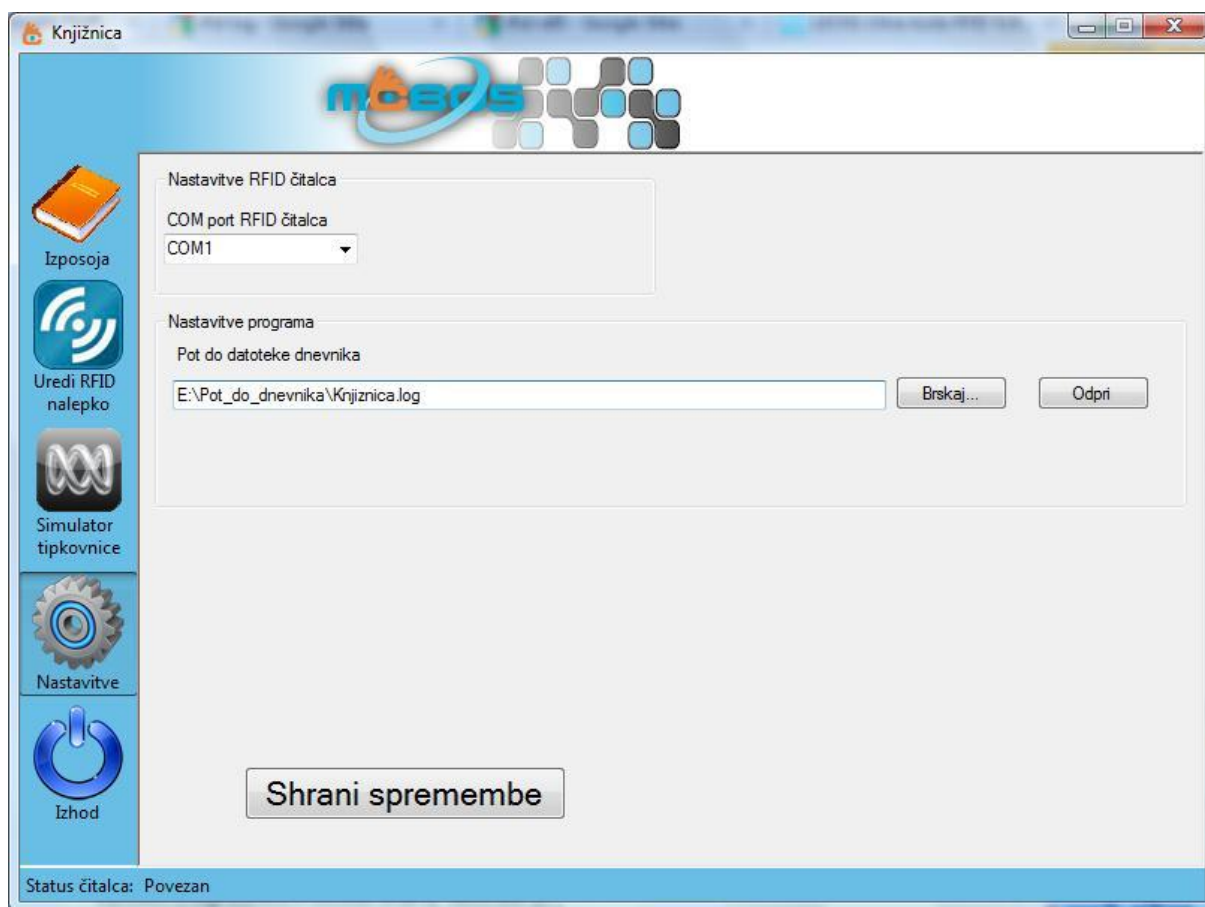
Ob vsaki pritisnjeni tipki se pokliče metoda `HookCallBack`, v kateri se preveri, če je bila pritisnjena pravilna kombinacija za vklop ali izklop zaščite na kartici. V primeru ene ali druge, se izvrši metoda `preklopiZaščito(bool vklopiZascito)`, ki najprej preveri če se izvaja pozivanje. Ker čitalnik ni sposoben in pozivati in vpisovati znaka AFI, je potrebno najprej izklopiti pozivanje. Nato se vpiše znak za vklop ali izklop zaščite, sledi pa mu zopet vklop pozivanja. Ob vsakem preklopu zaščite, se uporabnika obvesti s sporočilnim oblakom, ki se pojavi iz ikone programa v območju za obvestila, prav tako se sprememba vpiše v dnevnik. Pred končanjem metode se izvede klic funkcije `CallNextHookEx`, s čimer pošljemo prebrane tipke drugi priključitveni točki.

Poleg vklopa priključitvene točke s tipkovnico, se lahko vklopi tudi pisanje vsebine prebranega RFID odzivnika na tipkovnico. Pogoji za to je, da je obkljukana aktivnost »pisanje na tipkovnico« v mini meniju, ki se prikaže, ko uporabnik z desnim miškinim klikom izbere ikono v območju za obvestila. Za to je potrebna ločena nit, ki opravlja pozivanje in v primeru prisotnosti RFID odzivnika tudi branje vsebine in izpis le-te na tipkovnico. Pošiljanje vsebine kartice se izvede z ukazom `SendKeys.SendWait(string vsebina)`. Operacijski sistem se ob tem odzove enako, kot če bi uporabnik vpisal tekst preko tipkovnice.

Ikona programa v območju za obvestila se odziva tudi na nekatere druge dogodke. Tako pomeni dvojni klik, naj se zaključi pasivno delovanje in se prikaže zopet glavno okno programa. Ob desnem kliku nanjo se tudi izpiše mini meni, kjer je možnost vklopa ali izklopa pisanja na tipkovnico, prikaz glavnega okna ali izhod iz programa.

## 4 Navodila za uporabo programa Knjižnica

Ob prvem zagonu programa Knjižnica se prikažejo nastavitve, ki so potrebne za pravilno delovanje čitalnika RFID in programa. V nastavitvah je potrebno izbrati ustrezna serijska vrata iz menija pod napisom COM port RFID čitalca. Poleg tega je potrebno nastaviti pot do datoteke - dnevnika, kamor se bodo shranjevali podatki o uporabi programa. Pot do dnevnika je možno vpisati ročno, lahko pa se pot do datoteke izbere iz menija za izbiro mape, ki se prikaže ob kliku na gumb Brskaj. Ime datoteke »dnevnik.log« se doda avtomatsko, možno pa jo je tudi ročno preimenoovati. S klikom na gumb Odpri, se datoteka odpre v urejevalniku besedil. Spremembe je pred zaključkom potrebno še potrditi s klikom na gumb Shrani spremembe (Slika 9).



Slika 9: Prikaz možnosti nastavitve v programu Knjižnica

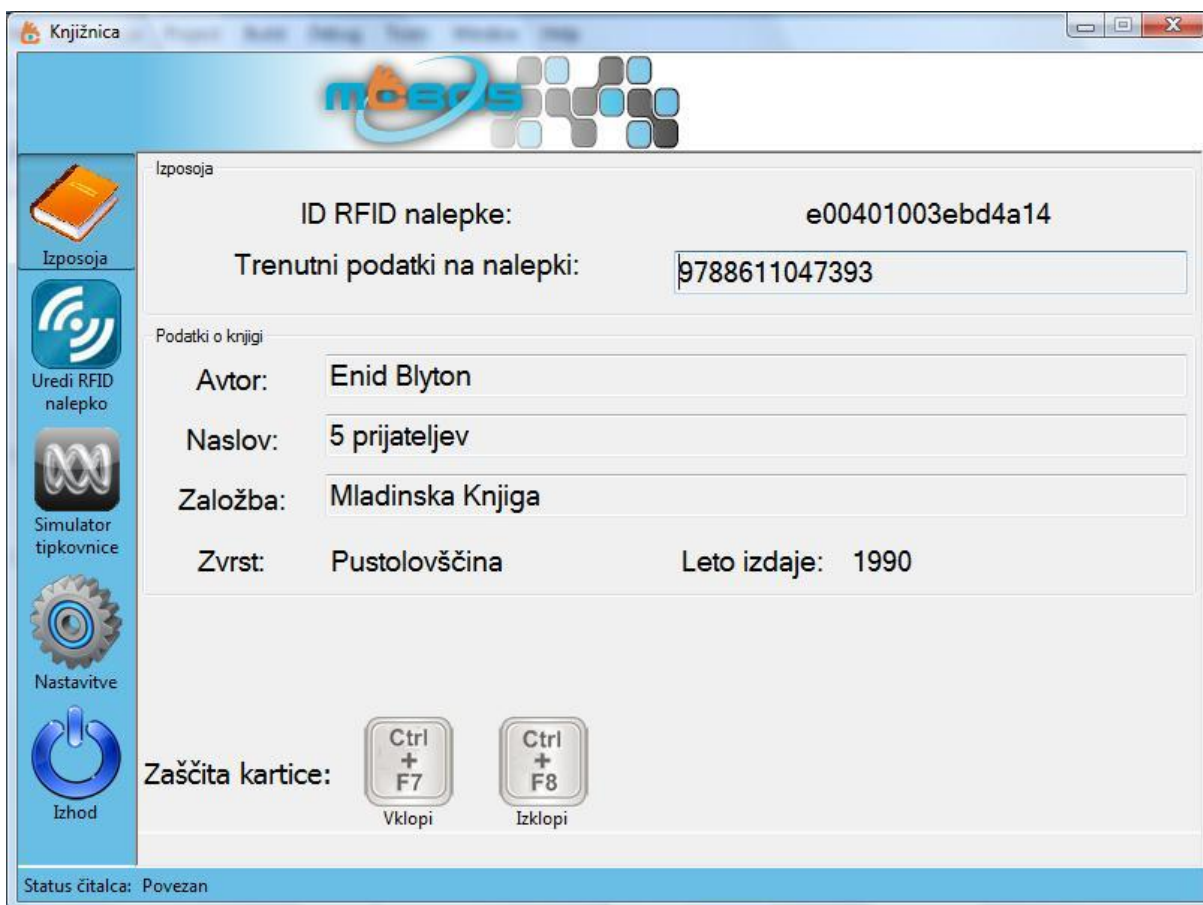
S tem so začetne nastavitve shranjene, okno nastavitve izgine in če so bila izbrana ustrezna vrata, se v statusni vrstici pri napisu status čitalca izpiše »povezan«. V nasprotnem primeru se izpiše NI POVEZAVE!, pojavi pa se tudi gumb, s katerim lahko ponovno poizkusimo vzpostaviti povezavo. Če povezava še vedno ne uspe, so bodisi izbrana napačna serijska vrata, bodisi čitalnik ni priklopljen ali pravilo nameščen.

Poleg statusne vrstice in gumba za izbiro nastavitev se na glavnem zaslonu nahajajo še štiri dodatni gumbi, kateri vsak predstavlja neko dejanje ali aktivnost. Preostali gumbi na zaslonu so:

- **Izposoja**
- **Uredi RFID nalepko**
- **Simulator tipkovnice**
- **Izhod**

#### 4.1 Izposoja

Ob izbiri gumba izposoja se znotraj glavnega okna prikaže dodatno okno, ki predstavlja izposajo in pregled podatkov na odzivniku RFID. Okno je sestavljeno iz treh delov. V prvem delu se ob približanju knjige opremljene z odzivnikom RFID izpiše serijska številka nalepke, in v nalepko RFID vpisana črtna koda te knjige. V drugem delu se na podlagi prebrane knjige izpišejo podrobnosti o tej knjigi, kot so avtor in naslov knjige, založba, pri kateri je bila knjiga izdana, leto izdaje in zvrst, v katero knjiga sodi (Slika 10).



Slika 10: Prikaz izgleda funkcionalnosti izposojanja knjig

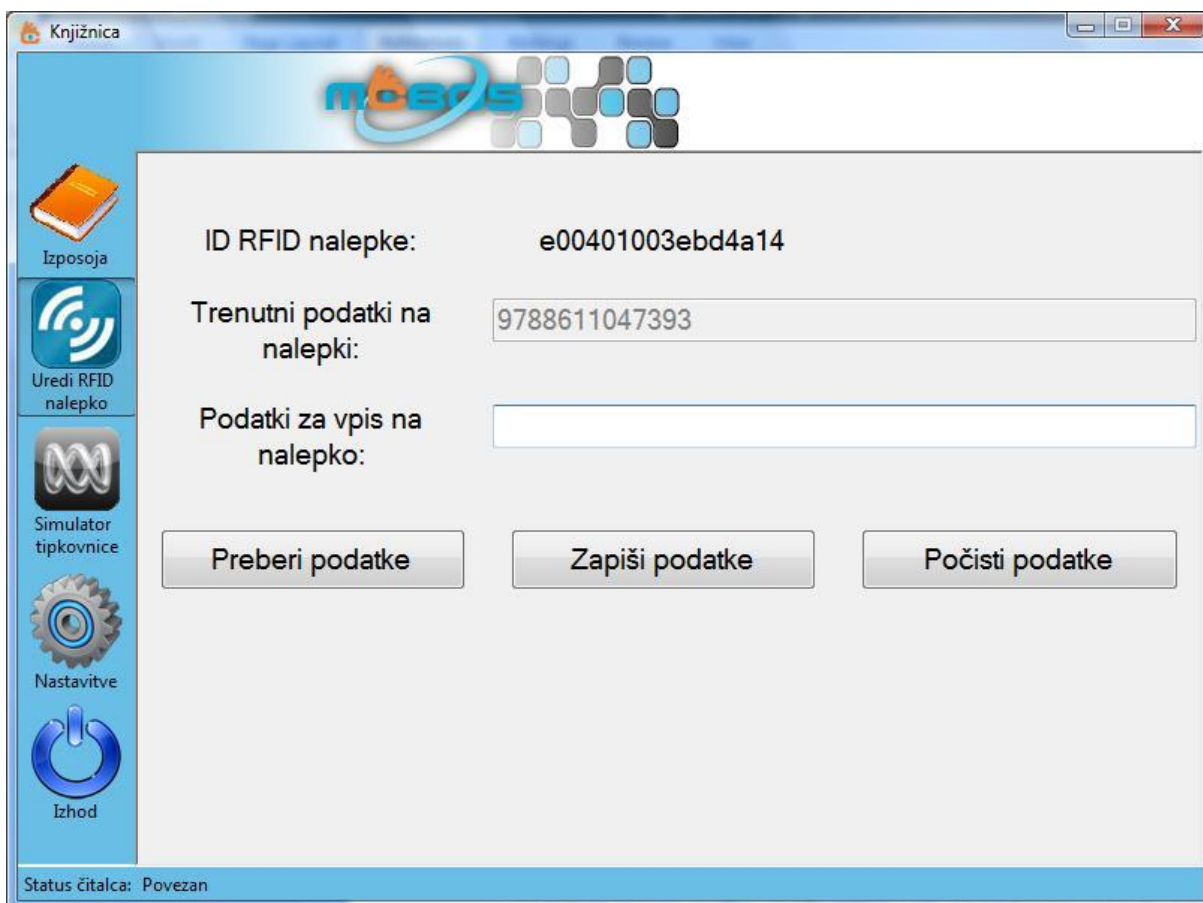
Če uporabnik želi knjigo označiti za izposojeno, mora knjigo približati čitalniku RFID in pritisniti na gumb Izklopi. S tem se na knjigi izklopi varnostna zaščita proti kraji, s čimer se ne bo vklopilo opozorilo na varnostnih vratih. Ob vračilu knjige v knjižnico se zaščita zopet vklopi z izbiro gumba Vklopi. Uporabnik lahko uporabi tudi hitre tipke in ob pravilni kombinaciji vklopi ali izklopi zaščito. Za vklop je potrebno hkrati pritisniti na gumb Ctrl in F7, za izklop pa na Ctrl in F8 (Slika 11). V opozorilo in vednost uporabniku, da je možna izbira tudi s hitrimi tipkami, sta obe kombinaciji ponazorjeni tudi z napisom na posameznem gumbu. Tako ob uspešnem kot tudi neuspešnem vklopu in izklopu zaščite, uporabnika obvesti sporočilo.



Slika 11: Gumbi za vklop ali izklop zaščite na RFID odzivniku

## 4.2 Urejanje RFID nalepke

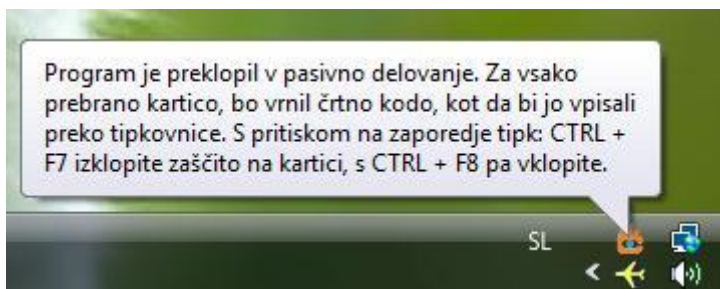
Kadar želi uporabnik zapisati podatke na novo kartico ali popraviti obstoječe podatke, je potrebno izbrati Uredi RFID nalepko. Prikaže se okno s poljem za vnos podatkov, poljem za prikaz obstoječih podatkov in serijske številke odzivnika RFID in tremi gumbi. Prvi je namenjen branju podatkov z odzivnika RFID. Ob prisotnosti le-tega, se morebitna vsebina izpiše skupaj s serijsko številko. Nato uporabnik vnese želene nove podatke v polje in izbere drugi gumb, Zapiši podatke. Običajno se na odzivnik RFID vnese že obstoječa črna koda knjige. Prebere se jo lahko z ustreznim čitalnikom črtno kodo ali pa se jo prepíše ročno. S tem se koda zapiše v spomin odzivnika RFID. Če knjiga s to črtno kodo že obstaja v lokalni podatkovni bazi, bo v oknu izposoja to tudi vidno, saj se bojo izpisale podrobnosti. Če je uporabnik vpisal podatke na odzivnik RFID, ki ni bil namenjen uporabi ali želi iz kakšnih drugih razlogov zbrisati podatke na odzivniku, uporabi gumb Počisti podatke, s katerim se iz spomina odzivnika RFID izbrišejo vsi podatki (Slika 12).



Slika 12: Izgled okna za urejanje nalepke RFID

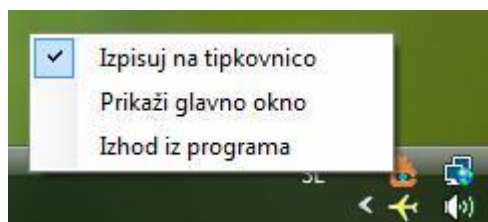
### 4.3 Simulator tipkovnice

Simulator tipkovnice je poimenovanje za aktivnosti neposredno povezane s tipkovnico. Ob izbiri se program navidezno preseli v območje za obvestila, kjer se pojavi ikona in oblaček, s katerim se uporabnika opozori, da program še vedno deluje (Slika 13). Glavno okno je mogoče kadarkoli priklicati z dvojnimi klikom na ikono. Program v takem načinu delovanja je namenjen souporabi z drugimi programi, katere knjižnica uporablja. Tako se lahko uporabnik v svojem programu, na primer namenjenemu izposoji gradiva, z miškinim kazalcem postavi v polje, kamor mora vnesti črtno kodo knjige. Nato namesto, da bi prebral knjigo s čitalnikom črtno kodo, le približa knjigo čitalniku RFID. Program Knjižnica bo takrat prepoznal odzivnik RFID nalepljen v knjigi, ga prebral in izpisal zapisano črtno kodo tako, kot da bi jo vpisali ročno preko tipkovnice ali čitalnika črtno kodo. Takšno branje je veliko hitrejše, saj se ni potrebno ukvarjati s poškodovanimi nalepkami s črtno kodo ali s pravilno oddaljenostjo čitalnika črtno kodo od nalepke. Ob tem je možno z enako kombinacijo tipk, kot pri oknu izposoja, torej z Ctrl+F7 ali Ctrl+F8, vklopiti ali izklopiti zaščito na odzivniku RFID. S tem je izposoja knjig hitrejša, predvsem pa varnejša. Ob vsaki prebrani knjigi ali vklopljeni zaščiti, se iz ikone prikaže oblaček, v katerem program uporabnika obvesti o uspešni ali neuspešni akciji.



Slika 13: Obvestilo ob preklopu programa v pasivni način delovanja

Ob desnem kliku na ikono programa se odpre meni s tremi možnostmi (Slika 14). Možnost izhoda iz programa, možnost prikaza glavnega okna in možnost izklopa simuliranja tipkovnice. Ob izklopu tega izpisovanja, program ne bo samodejno izpisoval prebranih črtnih kod ampak bo le v čakalju ukaza za izklop ali vklop zaščite.



Slika 14: Izbirni meni ob desnem kliku na ikono programa Knjižnica

#### 4.4 Dnevnik

V datoteki dnevnik, se beležijo vsi prebrani odzivniki RFID, vse spremembe vsebine in vsi vklopi ali izklopi zaščit, ob katerih je za evidenco tudi točen datum in ura, kdaj se je izvedlo. Datoteko jo je možno enostavno poiskati v raziskovalcu in jo pregledati, lahko pa se odpre tudi v možnostih nastavitvev, kjer je poleg gumba brskaj, tudi gumb za prikaz vsebine datoteke.

#### 4.5 Napaka

Za datoteko napaka ni možno spreminjati lokacije. Vanjo se zapisujejo sistemske napake programa in je v pomoč pri odpravljanju napak. Namenjena je vzdrževalcu programa in se nahaja v isti mapi kot program.

## 5 Prihodnost tehnologije RFID

Cena odzivnikov RFID z leti vztrajno pada, zato je povsem realno pričakovati, da bo v prihodnosti dosegla takšno raven, da bo uporaba odzivnikov RFID postala nekaj povsem vsakdanjega, tako kot je sedaj uporaba nalepk s črnimi kodami. S tem bi se lahko življenje, kot ga poznamo danes, povsem spremenilo. Kot najbolj pričakovano spremembo v prihodnosti, katera je povezana z uporabo odzivnikov RFID je olajšanje nakupovanja izdelkov. Tako bi lahko namesto čakanja v dolgih kolonah na blagajnah, nakupovalno košarico ali voziček enostavno postavili v območje čitalnika RFID, kateri bi v realnem času zaznal vse odzivnike RFID, v interni bazi podatkov preveril cene, jih seštel in izstavil račun. Tega bi kupec lahko poravnal na klasičen način ali pa tudi z uporabo RFID kartice, na kateri bi pred tem naložil določeno vsoto denarja. Takšni sistemi sicer že obstajajo, vendar imajo pomanjkljivosti. Glavna je, da še ni sistema, ki bi s sto odstotno zanesljivostjo prebral vse izdelke v nakupovalnem vozičku. Prav tako so vsi sistemi omejeni na neko število odzivnikov RFID, ki so jih sposobni prebrati. V primeru, da je v vozičku več izdelkov od največjega dovoljenega, bi jih spet bilo določeno število spregledanih. S tem bi prišlo do ogromnih izgub in problemov pri opravljanju inventure, ko bi se ugotovilo, da manjka velik delež izdelkov. Poleg izgub ob pomanjkanju izdelkov, pa je potrebno za vsak manjkajoči izdelek vseeno plačati davek.

Še nekoliko bolj znanstveno fantastično se sliši ideja, da bi imela vsaka oseba v sebi vgrajen odzivnik RFID, na katerem bi bili zapisani podatki o lastniku. S tem bi se izognili izdelavi osebnih dokumentov, voznških dovoljenj in podobnih dokazil, saj bi lahko vsi podatki bili zapisani na samem odzivniku RFID. Ob potrebi po identifikaciji bi se le približali odzivniku RFID in naši podatki bi se izpisali. Pri tem pa se pojavi glavna varnostna pomanjkljivost, katera je verjetno tudi razlog, da bodo tovrstne identifikacije v prihodnosti le stežka realizirane. Ta pomanjkljivost je dostop do teh podatkov nepravim osebam in s tem kraja osebnih podatkov. Če bi hotele te kartice biti enostavno uporabne in dostopne, bi s tem morali biti podatki tudi enostavno dostopni. Zelo težko bi bilo narediti takšno zaščito kartice pred vdori, katera pa bi še vedno omogočala enostavno uporabo različnim uradnim organom in to po vsem svetu. S tem, ko bi osebe imele takšne kartice, bi lahko tudi nadzirali njihovo gibanje, saj bi kartico lahko zaznal vsak čitalnik RFID kateremu se oseba približa. Tako bi lahko nadzirali, kdaj je določena oseba zapustila trgovino, kdaj in kam se je peljala z avtobusom in podobno. To bi pomenil grob vdor v zasebnost posameznikov in zato imajo odzivniki RFID vgrajeni v ljudi, kateri bi se vgrajevali na celotni populaciji malo možnosti za resnično realizacijo. Kljub temu pa so takšni odzivniki RFID ponekod že v uporabi. Verjetno najbolj znan primer je v Barceloni, kjer so v nočnem klubu za pomembne goste pripravili odzivnike RFID, katere jim vstavijo v ramo. Gosti s takšnim odzivnikom RFID, so obravnavani drugače kot ostali, saj jim ni potrebno čakati v vrsti. Ob prihodu v klub jih varnostnik identificira s pomočjo čitalnika RFID, na katerem se jim izpišejo podatki o osebi in njihova slika. Takšne osebe imajo tudi dovoljenje dostopa do posebnih sob, kjer za plačilo ne uporabljajo fizičnega denarja, ampak se jim vsota naročene pijače odšteje iz računa, kateri pripada odzivniku RFID. Natarakar mora le približati čitalnik RFID njegovi rami in plačilo je opravljeno [10].

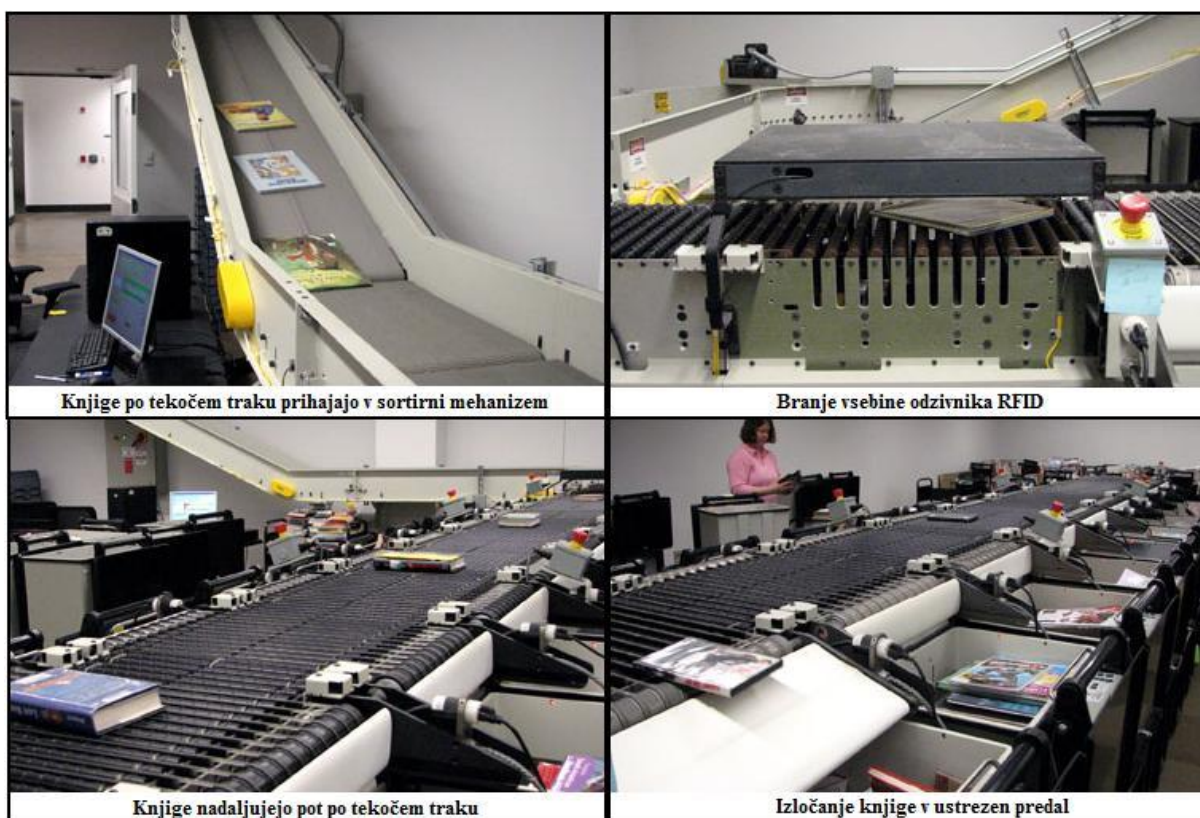
Pri uporabi odzivnikov RFID v knjižnicah pa ta prihodnost že postaja realnost. Število knjižnic, ki uporabljajo tehnologijo RFID, se vztrajno povečuje. V takih knjižnicah knjigam, katere imajo nalepljeno nalepko s črtno kodo, dodajajo še nalepke z odzivnikom RFID. S pomočjo teh nalepk se nato opravljajo vsa opravila, katera so pred tem temeljila na črtni kodi.

Ta opravila pa so z odzivniki RFID še veliko lažja in hitrejša, saj se lahko kup knjig hkrati približa čitalniku RFID, ki nato s pomočjo programa izvrši izposajo ali vračilo teh knjig. Hkrati se lahko spremeni tudi status za posamezno knjigo, na kar se bo tudi alarm na varnostnih vratih ustrezno odzval. Veliko enostavnejše je tudi iskanje določene knjige v knjižnici. Če ima knjižnica signal RFID po celotni knjižnici, je v vsakem trenutku možno enostavno razbrati, kje se določena knjiga nahaja. S tem se prihrani na času, katerega bi uslužbenci v nasprotnem primeru porabili za iskanje. Zelo enostavno pa je tudi opravljanje inventure knjig, katero je s pomočjo ročnega računalnika, v katerega je vgrajen čitalnik RFID (Slika 15), lahko opravljena v nekaj urah. S tem se, odvisno od števila knjig in velikosti knjižnice, lahko privarčujejo dnevi ali celo tedni iskanja in opravljanja inventure. Pri tem pa ni potrebno vsako knjigo ročno popisati ali prebrati črtno kodo. S pomočjo tehnologije RFID se inventuro opravi tako, da se ročni računalnik, v katerega je vgrajen čitalnik RFID, približa knjižni polici. Pri tem se premikamo od začetka do konca vsake police. Vsaka knjiga ki je v doletu čitalnika, je prebrana in s pomočjo programa na ročnem čitalniku se opravlja inventura knjig. S tem se zelo pomembno a hkrati zamudno opravilo, opravi v najkrajšem možnem času. Glede na to, da se inventura običajno opravlja vsaj enkrat letno, je s tem prihranek na času in s tem tudi denarju ob uporabi tehnologije RFID, vsako leto večji. Z ročnim računalnikom, ki vsebuje čitalnik RFID je tudi možno poiskati knjigo, katera se nahaja v knjižnici, vendar je bila postavljena na napačno polico, ali celo na napačen oddelek knjižnice. Ob uporabi samo črtnih kode, bi tako bilo potrebno pregledati naslove ali celo črtne kode vseh knjig v knjižnici, dokler ne bi našli narobe postavljene knjige. Z uporabo ročnega računalnika pa je potrebno se le sprehoditi po policah. Ko bo iskana knjiga bila zaznana, se bo čitalnik vizualno in zvočno odzval.



Slika 15: Ročni terminal Nordic ID PL3000 z vgrajenim čitalnikom RFID  
(Vir: <http://www.leoss.si>)

Z uporabo sistema RFID pa je možno knjižnico skoraj popolnoma avtomatizirati, kar je bilo sicer že možno z uporabo črtnih kod, vendar je z uvedbo odzivnikov RFID, to postalo neprimerno bolj zanesljivo in tudi hitrejše. V takem sistemu za izposajo in vračilo knjig skrbi »knjigomat«. Gre za napravo, ki popolnoma brez prisotnosti uslužbencev knjižnice opravi vračilo ali izposajo knjige. Za uporabo knjigomata, je potrebno članom knjižnice razdeliti članske kartice, ki so v bistvu odzivnik RFID. S to kartico se knjigomatu predstavijo, ko si želijo izposoditi kakšno knjižnično gradivo. Nato gradivo položijo na za to označeno mesto in knjigomat bo opravil izposajo ter izklopil zaščito. Kadar bodo želeli knjige in gradiva vrniti, jih bodo le odložili v odprtino v knjigomatu. Ta bo prebral identifikacijske številke zapisane na posameznih odzivnikih RFID in opravil vračilo knjig. Ker je knjigomat povezan s podatkovno bazo, se uporabniku ni potrebno identificirati, saj je bila izposoja posameznega gradiva vezana le na eno osebo. Prav tako se na odzivnikih zopet vklopi zaščita proti kraji. S tem so knjige pripravljene za sortiranje. Sortiranje lahko opravljajo uslužbenci ročno in se knjige le odlagajo na kup, ali pa se delno sortirajo avtomatično. Tako lahko knjigomat, ki vsebuje tudi sortirni mehanizem, knjige razvrsti po predalih, ki predstavljajo oddelke v knjižnici, ali po žanru knjige. Osebe mora nato te knjige le postaviti na police in jih ni potrebno ročno sortirati. Prikaz naprednega sortiranja knjig je predstavljen na spodnji sliki (Slika 16).



Slika 16: Prikaz delovanja sortirnega mehanizma knjig  
(Vir: [http://www.champaign.org/using\\_the\\_library/journey\\_of\\_a\\_book.html](http://www.champaign.org/using_the_library/journey_of_a_book.html))

Še ena prednost knjigomatov je, da jih je možno postaviti tudi izven knjižnic. Tako lahko knjižnica postavi svoj knjigomat za vračilo knjig sredi mesta (Slika 17), kjer je dostopen 24 ur na dan, vse dni v letu. Edina naloga knjižnice je, da poskrbi za prenos knjig iz knjigomata nazaj v knjižnico, kjer si jih je možno zopet izposoditi.



Slika 17: Knjigomat za avtomatsko vračilo knjig  
(Vir: <http://www.libraryjournal.com/article/CA6656758.html>)

Z vso avtomatizacijo knjižničnega poslovanja in prihrankom na času, se spremenijo tudi pričakovanja uporabnikov, saj imajo uslužbenci veliko več časa. Tako lahko prisluhnejo željam strank in jim pomagajo ter svetujejo pri izbiri knjig. S tem postane knjižnica bolj prijazna in posledično tudi bolj obiskana.

Uporaba odzivnikov in tehnologije RFID pa ima tudi pri uporabi v knjižnicah tudi pomanjkljivosti. Trenutno največja pomanjkljivost je še vedno cena celotnega sistema, vendar se bo v prihodnosti tudi ta zmanjšala. Vseeno pa bo ostal še problem, da so takšne nalepke RFID dokaj enostavno odstranljive. Z odstranitvijo pa se naš sistem znajde v zagati, saj varnostna vrata ne bodo zaznala kraje knjige. Prav tako bo ob izvajanju inventure prikazano, da knjige ni v knjižnici, čeprav dejansko je. Zaradi tega bi bilo smiselno uporabljati takšno obliko odzivnika RFID, katere bi se lahko vgrajevale v same platnice knjige že v tiskarnah. Knjižnica bi ob nakupu takšne knjige, knjigo le dodala v svojo bazo podatkov in glede na svoje potrebe zapisala podatke na odzivnik. S tem bi se izognili težavam pri inventuri, še vedno pa bi obstajala varnostna luknja, saj se signal odzivnikov RFID, ki se uporabljajo v knjižnicah, lahko hitro zaduši že v navadni vrečki obdani z aluminijasto folijo. Kljub temu, se je treba zavedati, da trenutne rešitve z uporabo nalepke s črtno kodo ne nudijo nikakršne zaščite pred krajo, zaradi česar je uporaba odzivnikov RFID velika pridobitev na področju varnosti.

Podobno kot pri uporabi odzivnikov RFID na ljudeh, je tudi pri uporabi na knjigah možnost zlorab zasebnosti. Tako bi lahko glede na izposojeno knjigo, tako imenovani Veliki brat, spremljal naše gibanje, saj bi vsak čitalnik RFID v trgovini, na avtobusu, parkiriščih in drugod kjer bi bil v uporabi sistem RFID zaznal naš odzivnik v knjigi. Olje na ogenj priliva tudi dejstvo, da so odzivniki RFID, ki so v uporabi v knjižnicah običajno nešifrirani. Na srečo so odzivniki, ki so v uporabi za označevanje knjig, pasivni in so kratkega dometa. Zaradi tega bi oseba s knjigo, morala le-to dobesedno približati čitalniku RFID, da bi jo lahko zaznal.

## 6 Zaključek

V sklopu diplomske naloge je bila razvita aplikacija Knjižnica. Glavni namen aplikacije je, začetno opremljanje knjižnic s tehnologijo RFID, ko je potrebno vsako knjigo opremiti z odzivnikom RFID in vanj vpisati identifikacijsko številko, ki je pred tem bila zapisana s črtno kodo. Aplikacijo je možno enostavno uporabljati tudi v kombinaciji z drugimi že obstoječimi programi v knjižnicah, saj omogoča izpis vsebine pomnilnika, ki je na odzivniku RFID, na tipkovnico. Ob branju vsebine pa hkrati omogoča še vklop ali izklop zaščite proti kraji. S tem lahko popolnoma nadomesti uporabo črtno kode in magnetnih nalepk proti kraji knjig. Sam opis programa bi verjetno bil nejasen, če ne bi poznali nekoliko teorije o tehnologiji RFID, zato smo v sklopu diplomske naloge spoznali tudi kaj sploh tehnologija RFID je, katere standarde poznamo, na katerih frekvencah sistemi delujejo in kaj lahko pričakujemo v prihodnosti na tem področju. Predstavljene so bile tudi tehnologije, ki so bile uporabljene in niso povezane s tehnologijo RFID.

Pred začetkom izdelave diplomske naloge je bilo pričakovati največje težave v povezljivosti med programom in čitalnikom RFID, vpisom in prebiranjem podatkov iz odzivnika RFID in vklop izklop zaščite proti kraji. Pri realizaciji, pa se je izkazalo, da je s pomočjo razreda DLLConnection, ta opravila mogoče opraviti dokaj enostavno. Največji problem se je pojavljal ob izvajanju klicev na čitalnik v prekratnem intervalu. V takšnem primeru je čitalnik dobil nov ukaz, še preden je uspel zaključiti prejšnjega. To se je dogajalo pri pozivanju ob prisotnosti odzivnika RFID. Čitalnik je še vedno prebiral podatke s odzivnika, ko je dobil že nov ukaz za preverjanje prisotnosti odzivnika RFID. Zaradi tega je bilo potrebno povečati interval iz 200 milisekund na 500 milisekund. Branje in pisanje podatkov ter vklop ali izklop zaščite se je izkazal za enostavno opravilo, saj dotične funkcije že vsebuje omenjeni razred. Potrebno jim je le podati ustrezne argumente.

Nepričakovane težave pa je povzročala povezava med računalnikom in tipkovnico. Ker je priključitvena točka na tipkovnico na tako nizkem nivoju, je bila težava zaznati ali je bila pritisnjena ustrezna kombinacija tipk ali ne. Težava se je tudi pojavila, ob pritisku ustrezne kombinacije za vklop ali izklop zaščite, saj se je izvajalo pozivanje. S tem je čitalnik RFID dobil več ukazov hkrati, kar je vodilo do napake. Zaradi tega je bilo potrebno vsakič, ko je zaznana pravilna kombinacija tipk, prekiniti izvajanje pozivanja, vpisati ustrezen znak AFI na odzivnik RFID ter nato ponovno vklopiti pozivanje za karticami. To pa je vodilo v novo težavo, ker se ob ponovnem začetku pozivanja, vsebina odzivnika RFID ponovno izpisala na tipkovnico, kar bi pri redni uporabi programa vodilo do problemov. Zaradi tega je ob ponovnem vklopu pozivanja dodano preverjanje, če je na čitalniku RFID še vedno isti odzivnik RFID, kot je bil pred prekinitvijo pozivanja. Na nek način je težava tudi to, da program za delovanje v pasivnem načinu potrebuje dovoljenje požarnega zidu, ob želji po ohranjanju simulacije tipkovnice.

Glede na to, da aplikacija Knjižnica predstavlja nekakšno odskočno desko za knjižnice v sistem tehnologije RFID, bi jo bilo možno še dodatno razširiti. V prvem koraku bi verjetno dodali funkcionalnost beleženja izposoje in vračila knjig. Potrebno bi bilo vzpostaviti povezavo s podatkovno bazo, kjer so shranjeni podatki o članih. Tako bi bilo možno povezati izposoje in vračilo knjige s posameznikom. Za pravilno integracijo v obstoječe knjižnično poslovanje v Sloveniji, bi bilo potrebno uskladiti program s sistemom COBISS, saj so vanj povezane praktično vse slovenske knjižnice. Aplikaciji Knjižnica, bi se lahko tudi pridružila nova aplikacija, ki bi bila na prenosnem računalniku, in bi omogočala opravljanje inventure in

iskanje knjig. Vpogled v stanje in poročilo o inventuri, pa bi bilo prikazano v aplikaciji Knjižnica. Aplikacijo bi bilo možno tudi prevesti v druge jezike. S tem bi jo bilo možno tržiti tudi izven Slovenije.

## 7 Literatura

- [1] E.C. Jones, C.A.Chung, RFID in logistics: a practical introduction, CRC Press, 2007, pogl. 2, 3, 4, 20.
- [2] (2010) The history of RFID Technology. Dostopno na: <http://www.rfidjournal.com/article/view/1338/2>
- [3] (2010) Kaj je RFID? Dostopno na: <http://www.leoss.si/index.php?vie=ctl&gr1=strSvt&gr2=&id=2006102514103146>
- [4] (2010) How RFID Works? Dostopno na: <http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid3.htm>
- [5] (2010) RFID Frequency bands. Dostopno na: [http://www.idtechex.com/research/articles/rfid\\_frequency\\_bands\\_00000040.asp](http://www.idtechex.com/research/articles/rfid_frequency_bands_00000040.asp)
- [6] (2010) Polling. Dostopno na: [http://en.wikipedia.org/wiki/Polling\\_%28computer\\_science%29](http://en.wikipedia.org/wiki/Polling_%28computer_science%29)
- [7] (2010) Serial port. Dostopno na: <http://www.computerhope.com/jargon/s/seriport.htm>
- [8] (2010) Serial port. Dostopno na: [http://en.wikipedia.org/wiki/Serial\\_port](http://en.wikipedia.org/wiki/Serial_port)
- [9] (2010) Microsoft Access. Dostopno na: [http://en.wikipedia.org/wiki/Microsoft\\_Access](http://en.wikipedia.org/wiki/Microsoft_Access)
- [10] (2010) Uporaba RFID na ljudeh. Dostopno na: <http://www.ft.com/cms/s/2/3fa0cfee-edde-11d9-98e5-00000e2511c8.html>
- [11] (2010) Mednarodna organizacija za standardizacijo. Dostopno na: [http://sl.wikipedia.org/wiki/Mednarodna\\_organizacija\\_za\\_standardizacijo](http://sl.wikipedia.org/wiki/Mednarodna_organizacija_za_standardizacijo)
- [12] (2010) Radio-frequency identification. Dostopno na: [http://en.wikipedia.org/wiki/Radio-frequency\\_identification](http://en.wikipedia.org/wiki/Radio-frequency_identification)
- [13] (2010) Wal-Mart. Dostopno na: <http://en.wikipedia.org/wiki/Wal-Mart>