

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tilen Turnšek

**TESTNI RFID SISTEM ZA SPREMLJANJE
IN SORTIRANJE LETALIŠKE
PRTLJAGE**

Diplomsko delo na visokošolskem strokovnem študiju

Mentor: doc. dr. Patricio Bulić

Somentor: doc. dr. Mira Trebar

Ljubljana, 2011

Št. naloge: 00061/2011

Datum: 01.02.2011



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

Kandidat: **TILEN TURNŠEK**

Naslov: **TESTNI RFID SISTEM ZA SPREMLJANJE IN SORTIRANJE
LETALIŠKE PRTLJAGE**
**AN RFID TESTING SYSTEM FOR IDENTIFICATION AND SORTING OF
AIRPORT LUGGAGE**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Zasnujte in izdelajte sistem za identifikacijo in sortiranje letališke prtljage na osnovi radiofrekvenčne identifikacije (RFID). Preučite tehnologijo RFID sistemov in izberite najprimernejšo rešitev za zahtevano nalogo pri uporabi v letališkem okolju. Sistem naj omogoča sledenje prtljage na tekočem traku v sortirnici z namenom poenostavljenega ročnega sortiranja prtljage. Testiranje naj bo izvedeno v laboratoriju in v letališki sortirnici. Na osnovi dobljenih rezultatov podajte potrebne parametre RFID sistema (ojačenje, pozicija anten, ipd.), ki omogočajo njegovo optimalno delovanje v realnem okolju.

Mentor:

doc. dr. Patricio Bulić

Dekan:

prof. dr. Nikolaj Zimic

Somentor:

doc. dr. Mira Trebar



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Tilen Turnšek,

z vpisno številko 63040400,

sem avtor/-ica diplomskega dela z naslovom:

TESTNI RFID SISTEM ZA SPREMLJANJE IN SORTIRANJE LETALIŠKE PRTLJAGE

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)
doc. dr. Patricio Bulić
in somentorstvom (naziv, ime in priimek)
doc. dr. Mira Trebar
- 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 _____ Podpis avtorja/-ice: _____

Zahvala

Zahvaljujem se mentorju doc. dr. Patricio Buliću in somentorici doc. dr. Mira Trebar, za pomoč in usmerjanje pri izdelavi diplomske naloge. Zahvaljujem se tudi staršem za podporo, ki sem je bil deležen v času študija in izdelave diplomske naloge. Prav tako se zahvaljujem podjetju Aerodrom Ljubljana d.d., ki mi je omogočilo testiranje na Letališču Jožeta Pučnika Ljubljana, še posebej gospodoma Klemnu Denši in mag. Robertu Rauchu.

Kazalo

Povzetek.....	1
Abstract.....	2
1. Uvod.....	3
2. Sortirni sistem.....	4
2.1. RFID sortirni sistem.....	4
2.2. Radio frekvenčna identifikacija - RFID.....	5
2.3. Čitalnik RFID.....	5
2.4. RFID značka.....	6
2.5. Frekvenčna območja RFID sistemov.....	7
3. Analiza in implementacija.....	8
3.1. Sortirnica B.....	8
3.2. Izbira RFID sistema.....	9
3.3. Implementacija RFID sortirne postaje.....	11
3.3.1. Modul za upravljanje čitalnika RFID.....	11
3.3.4. Modul za upravljanje z bazo.....	15
3.3.5. Izdelava glavnega programa.....	18
3.3.6. Izdelava ostalih programov.....	21
4. Predstavitev aplikacije in testiranj.....	23
4.1. Postavitve anten.....	24
4.2. Testiranje na letališču.....	27
5. Sklepne ugotovitve.....	31
Dodatek A.....	33
Dodatek B.....	34
6. Seznam uporabljenih virov.....	35

Seznam uporabljenih kratic in simbolov

Kratica	Pomen
RFID	Radio-frequency identification
IFF	Identification friend or foe
XML	Extensible Markup Language
XSD	XML Schema Definition
LLRP	Low Level Reader Protocol
UID	Unique Identification Number
SDK	software development kit

Povzetek

Namen diplomskega dela je predstavitev testnega sistema za sortiranje prtljage, ki temelji na RFID tehnologiji. Sistem smo preizkusili v laboratoriju in v sortirnici B, ki se nahaja na letališču Jožeta Pučnika Ljubljana. Sistem bere podatke iz RFID značk, ki označujejo prtljago in jih povezuje s podatki v bazi. Te podatke nato prikaže na zaslonu sortirne postaje in s pomočjo operaterja sortira prtljago.

Delo vsebuje predstavitev RFID tehnologije, pregled njenega razvoja in njene uporabe danes. Prav tako je opisan razvoj aplikacije, njeno delovanje, zahteve okolja in načini testiranja. Aplikacija je razvita v orodju Microsoft Visual Studio 2010 v programskem jeziku C#. Za podatkovno bazo smo izbrali Microsoft SQL Server 2008 R2 Express. Na koncu smo predstavili nekaj idej za nadgradnjo in izboljšavo izdelanega sistema.

Ključne besede: radio frekvenčna identifikacija, čitalnik RFID, RFID značka, sortiranje, SQL baze

Abstract

This thesis is a presentation of a experimental baggage sorting system based on RFID technology. The system was tested in the laboratory and in sorting facility B, located at the Ljubljana Jože Pučnik Airport. The system reads data from RFID tags which mark baggage and associates it with data in the database. This data is then displayed on the screen in the sorting facility and the system sorts baggage with the aid of an operator.

Included in this publication is a presentation of RFID technology, its development and its uses today. Also described are application's development, its operation, environment requirements and testing procedures. The application was developed with the product Microsoft Visual Studio 2010 and is written in the programming language C#. Microsoft SQL Server 2008 R2 Express was chosen as the database. The thesis is concluded with a few ideas for upgrading and enhancing the created system.

Keywords: radio-frequency identification, RFID reader, RFID tag, sorting, SQL databases

1. Uvod

Radio frekvenčna identifikacija (ang. Radio-frequency identification) je hitro razvijajoča se tehnologija, ki vedno bolj prodira v naš vsakdan in jo srečamo že skoraj na vsakem koraku. Radio frekvenčni identifikacijski sistem, krajše RFID, nam omogoča branje podatkov iz RFID značk s pomočjo čitalnika RFID. Lahko se uporablja za različne namene, za označevanje izdelkov v trgovini, za kontrolo dostopa ali kot karte za avtobuse, vlake, muzeje, knjižnice [1, 2].

Ob tako hitrem razvoju in prodiranju RFID tehnologije v naš vsakdan, moramo zraven dobrih lastnosti nujno omeniti tudi slabe. Ena izmed večjih slabih lastnosti RFID tehnologije je vprašanje zasebnosti. RFID značke namreč delujejo tudi po tem, ko te niso več v svojem primarnem okolju. Na primer, ko kupec odnese izdelek iz trgovine, lahko nekdo še vedno prebere podatke iz izdelkove RFID značke in to brez njegove vednosti. RFID tehnologija se lahko zlorablja za različne namene, kot so sledenje osebam, razne analize navad in želj ljudi. Zato velja velika pazljivost pri načrtovanju RFID sistemov [1].

Cilj diplomske naloge je načrtovati in izdelati prototip sistema, ki temelji na RFID tehnologiji. Sistem se uporablja za sortiranje prtljage. Postavljen bo v laboratoriju in kasneje preizkušen v realnem okolju sortirnice B, ki se nahaja na letališču Jožeta Pučnika Ljubljana in jo upravlja podjetje Aerodrom Ljubljana d. d. Glavni cilji sistema so:

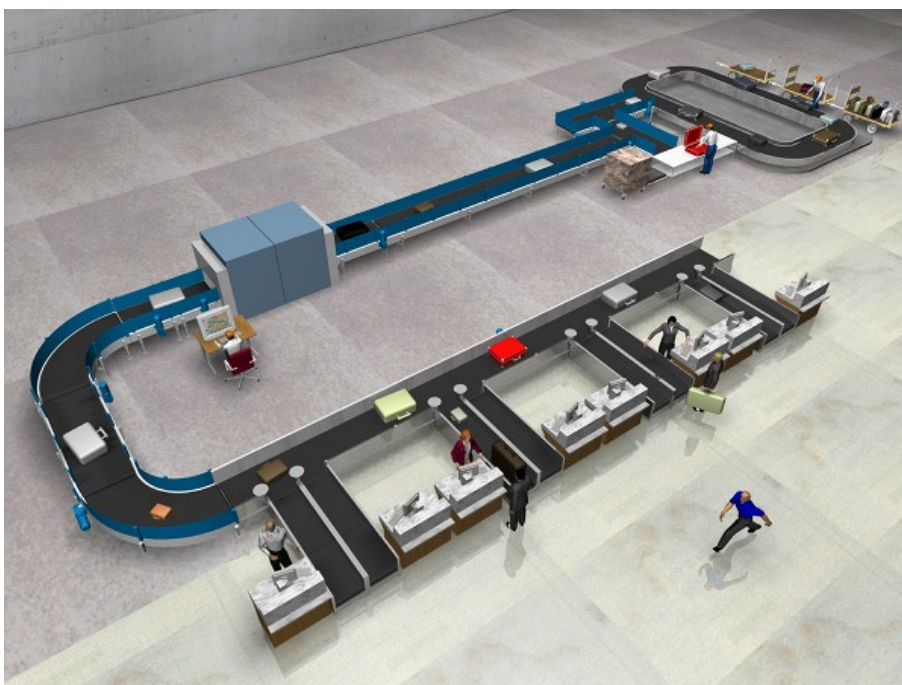
- Branje podatkov iz RFID značk, ki označujejo prtljago.
- Iskanje podatkov o prtljagi s pomočjo prebranih podatkov po podatkovni bazi.
- Pregledovanje, urejanje in uveljavljanje nastavitvev čitalnika RFID neposredno iz aplikacije.
- Jasen in enostaven uporabniški vmesnik.
- Upoštevanje zahtev za delovanje v sortirnici B. Zahteve kot so hitrost sortiranja in prostor, ki ga pri tem lahko zasede.

Za doseganje ciljev je potrebno premagati sledeče ovire:

- Ustrezna izbira čitalnika RFID in RFID značk.
- Izbira ustrezne podatkovne baze in njenega podatkovnega modela.
- Preučitev programskega vmesnika čitalnika RFID.
- Izdelava uporabniškega vmesnika, ki je za uporabnika jasen in enostaven.
- Preučitev trenutnega delovanja sortirnice B.

2. Sortirni sistem

Sortirni sistem je eden izmed ključnih sistemov na letališču (Slika 1). Letališčem omogoča hitro, varno in učinkovito razporejanje prtljage. Sortirni sistemi se morajo ob nenehnem naraščanju števila potnikov neprestano razvijati in posodabljati. Tako so današnji sortirni sistemi v večini že popolnoma avtomatizirani. Osnovni gradniki, ki sestavljajo sortirni sistem, so tekoči trakovi, bralniki črtnih kod, razne pregledovalne naprave in robotske roke, ki preusmerjajo prtljago. Eden od pomembnejših delov so bralniki črtnih kod. Ti so odgovorni za krmiljenje robotskih rok na tekočih trakovih in s tem igrajo glavno vlogo pri usmerjanju prtljage do njenega cilja. Na manjših letališčih, kjer robotskih rok ne uporabljajo, bralniki končni cilj prtljage izpišejo na zaslon in s tem obveščajo operaterje, kaj morajo storiti [3].



Slika 1: Manjši sortirni sistem

2.1. RFID sortirni sistem

Bralnike črtnih kod žal pestijo določene pomanjkljivosti. Uspešnost identifikacije cilja prtljage je odvisna od uspešnosti branja nalepke z črtno kodo. Do neuspešnega branja lahko pride, ko je nalepka v napačnem položaju, ko je raztrgana ali zabrisana. Takšne nalepke sistem ne more prebrati ali huje, prebere jo napačno. Slednje je nekaj, česar si zaradi varnosti, bolj kot zaradi ugodja potnika, v letalski industriji ne želi nihče. Zato letalska industrija danes išče zanesljivejše in hitrejše metode identifikacije prtljage.

Odgovor je letalska industrija našla v RFID tehnologiji. Ker tehnologija uporablja radijske valove, je ta veliko manj občutljiva od položaja RFID značke. Prav tako RFID značka ne more biti zabrisana, lahko je manjših dimenzij in se zato težje strga. Čitalnik RFID bo v vseh

Zgoraj naštetih primerih kodo uspešno prebral in pravilno usmeril prtljago. RFID sortirni sistemi tako dodatno izboljšujejo in dopolnjujejo obstoječe sortirne sisteme. Njihovo število raste iz dneva v dan, v času pisanja te diplomske naloge so RFID sisteme uporabljala sledeča letališča:

- Hong Kong International Airport, Hong Kong, Kitajska [6]
- McCarran International Airport, Las Vegas, ZDA [4]
- Malpensa Airport, Milan, Italija [5]
- Aalborg Airport, Aalborg, Danska [7]
- Copenhagen Airport, Copenhagen, Danska [8]

2.2. Radio frekvenčna identifikacija - RFID

Začetki RFID tehnologije segajo nazaj vse do leta 1915. Takrat so Britanci izdelali IFF (Identification friend or foe) transponder. IFF transponder je služil kot identifikacijska naprava za letala. Ko je letalo vstopilo v območje radarja, se je ob določeni frekvenci vklopil transponder in začel oddajati identifikacijsko številko. Tako so v času 2. svetovne vojne angleži ločevali svoja letala od sovražnih [1].

Eden izmed kasnejših izumov je bila Ruska vohunska naprava, ki so jo razvili leta 1945. Šlo je za pasivno prisluškovalno napravo. Ko je sprejemnik v bližini prisluškovalne naprave začel oddajati elektromagnetne valove, je ta začela oddajati zvočni signal preko radijskih valov in tako omogočila prisluškovanje. Tehnično tukaj ne gre za identifikacijsko napravo, gre pa za eno izmed prvih naprav, ki so energijo za svoje delovanje pridobivale iz elektromagnetnih valov [1].

Začetki moderne RFID tehnologije pa segajo v sedemdeseta leta devetnajstega stoletja. Takrat se prvič pojavi pasivni radio oddajnik s spominom. Oddajnik se je ob določenem signalu vklopil in začel oddajati podatke, ki so bili zapisani v njegovem spominu. Ti prvi sistemi so delovali na frekvenčnem območju 915 Mhz in so oddajali 12 bitne oznake. Tako so nastali prvi moderni RFID sistemi [1].

Uporaba RFID tehnologije za identifikacije je danes zelo razširjena. Uporabljena je v trgovinah, zdravstvu, živinoreji, javnem prometu, varnosti, knjižnicah. Lahko rečemo, da se uporablja povsod, kjer želimo povečati hitrost in zanesljivost identifikacije predmetov ali osebkov. Glavna dela RFID sistema sta:

- čitalnik RFID in
- RFID značka.

RFID sisteme lahko dalje razdelimo še po načinu napajanja RFID značke, po frekvenčnem območju, v katerem RFID sistem deluje, in po načinu sklopitve med čitalnikom RFID in RFID značko.

2.3. Čitalnik RFID

Naloga čitalnika RFID je komunikacija z RFID značko. Čitalnik preko antene pošilja signal, s

katerim aktivira in komunicira z RFID značko. Ko RFID značka zazna signal, začne oddajati svoje podatke, ki jih nato prebere čitalnik RFID. Pri enostavnih RFID sistemih signal čitalnika deluje le kot stikalo za vklop ali izklop RFID značke. Signali pri naprednejših sistemih pa lahko vsebujejo podatke ali ukaze, s katerimi lahko spreminjamo vsebino RFID značk. Starejši čitalniki RFID so bili zasnovani tako, da so lahko delovali samo na eni frekvenci, danes pa je trend drugačen in večina čitalnikov RFID podpira več frekvenc delovanja. Poznamo več vrst čitalnikov RFID, ki so najrazličnejših oblik in velikosti. Tako imamo čitalnike, ki za delovanje potrebujejo povezavo z računalnikom, avtonomne ročne čitalnike, čitalnike RFID, ki delujejo kot samostojni krmilniki in nenazadnje čitalnike, ki jih lahko vgradimo v mobilne telefone (Slika 2) [9].



Slika 2: Različna čitalnika RFID

2.4. RFID značka

RFID značke najdemo v najrazličnejših oblikah. Lahko so v obliki nalepk, kartic, obeskov, škatlic (Slika 3). Podatki na njih lahko predstavljajo identifikacijske številke, lahko vsebujejo navodila, sporočila ali podatke o osebah, izdelkih. Zaradi tipa in zahtevnosti okolij, v katerih RFID sistemi delujejo, napajanje RFID značk neposredno iz vzbujalnega signala večkrat ni dovolj. V teh primerih opremimo RFID značke z dodatnim virom napajanja v obliki baterij. S tem jim izboljšamo tako zanesljivost kot doseg komunikacije [9, 1].



Slika 3: Različne RFID značke

RFID sisteme lahko ločimo po:

- načinu napajanja RFID značk na aktivne in pasivne (Tabela 1),
- po frekvenčnem območju delovanja.

Tabela 1: Prednosti in slabosti aktivnih in pasivnih RFID sistemov

Tip sistema	Aktivni	Pasivni
Prednosti	<ul style="list-style-type: none"> • Zanesljivo delovanje tudi v neprijaznih okoljih • Večji domet komunikacije 	<ul style="list-style-type: none"> • Dolga življenjska doba • Majhnost in priročnost • Energijo za delovanje dobijo neposredno od vzbujalnega signala
Slabosti	<ul style="list-style-type: none"> • Kratka življenjska doba zaradi baterijskega napajanja • Velikost in ne praktičnost 	<ul style="list-style-type: none"> • Kratak domet komunikacije • Nezanljivo delovanje v neprijaznih okoljih

2.5. Frekvenčna območja RFID sistemov

RFID sistemi delujejo na zelo širokem frekvenčnem območju (Tabela 2). To se začne pri približno 100kHz in sega do več GHz. Za potek komunikacije med čitalnikom RFID in značko je nujno potrebno, da oba delujeta na enaki frekvenci. Frekvenco, v kateri bo RFID sistem deloval, se izbere glede na njegovo delovno okolje. Delovno okolje lahko od RFID sistema zahteva hitro komunikacijo in hiter prenos podatkov, lahko zahteva izredno dolg domet komunikacije ali pa visoko zanesljivost in natančnost ob raznih neprijaznih dejavnihikih (Tabela 2). Neprijazno okolje definirajo prisotnost vode, kovin in ostalih radijskih naprav, njihov vpliv pa je odvisen od frekvence RFID sistema. Medtem ko nam višje frekvence omogočajo daljše domete in hitrejšo komunikacijo, so te zelo občutljive na prisotnost kovinskih ovir. Prav tako jih ovira prisotnost tekočin in ostalih visoko frekvenčnih naprav. Stalna izpostavljenost visokim frekvencam naj bi prav tako škodovala zdravju ljudi [1, 9].

Tabela 2: Primerjava prednosti in slabosti RFID frekvenčnih območij

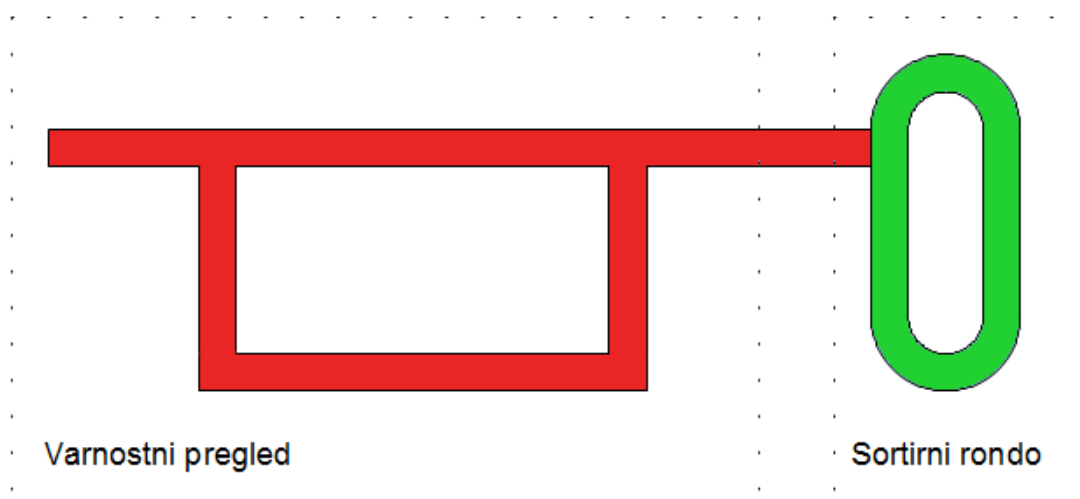
Frekvenčno področje	Frekvence	Območje delovanja	Prednosti	Slabosti	Tipična uporaba
Nizke frekvence (LF)	125KHz - 135 KHz	Fizični kontakt ali maksimalno do pol metra	Dober odziv v okolju z veliko vode in kovinami	Kratek domet, počasna hitrost branja	Nadzor dostopa Označevanje izdelkov Označevanje živali
Visoke frekvence (HF)	13,56 MHz	Do maksimalno enega metra	Nizka cena značk	Kratek domet, počasna hitrost branja	Knjižnice Skladišča Pametne kartice Sortiranje prtljage
Ultra visoke frekvence (UHF)	860MHz do 930MHz	V dobrih pogojih tudi do 10 metrov	Osnova EPC standardov, hitro branje podatkov	Slabši odziv v okolju z vodo in kovinskimi predmeti	Cestninjenje Pobiranje parkirnine

3. Analiza in implementacija

Za dodatno razumevanje predvsem manjših sortirnih sistemov smo se odpravili na letališče Jožeta Pučnika Ljubljana, kjer smo si ogledali sortirnici A in B in se spoznali z njihovim načinom sortiranja. Najzanimivejša in hkrati tudi najbolj obremenjena je bila sortirnica B, zato se bomo v tej diplomi osredotočili nanjo. Na koncu smo si ogledali še načrte za nov potniški terminal, ki bo vseboval povsem novo sortirnico. Novo sortirnico bo možno nadgraditi tako, da bo lahko uporabljala RFID tehnologijo.

3.1. Sortirnica B

Sortirnico B sestavljata dva tokokroga, ki sta realizirana s tekočimi trakovi (Slika 4). Naloga prvega so varnostni pregledi vse prtljage, naloga drugega pa sortiranje. Prvi tokokrog je delno avtomatiziran, saj v primeru neuspešnega varnostnega pregleda prtljage sam poskrbi za izločitev in ponovni varnostni pregled. Ponovni pregled prtljage lahko zahteva tudi operater varnostnega pregleda, vendar se tega redko poslužuje. Pregledana prtljaga nato potuje na drug tokokrog, kjer se izvaja sortiranje. Vsa prtljaga, ki se nahaja v sortirnici B, je označena z črtno kodo. Ko se prtljaga pripelje do sortirne postaje, operater ročno s pomočjo brezžičnega bralca črtnih kod prebere njeno črtno kodo in jo pošlje računalniku. Če je program na sortirni postaji nastavljen na prebrano destinacijo, se ta izpiše na ekranu in operater prtljago prestavi s traku na voziček. V nasprotnem primeru pa jo pusti, da nadaljuje pot do naslednje postaje, kjer se celoten postopek ponovi.



Slika 4: Shema sortirnice B na letališču Jožeta Pučnika Ljubljana

Slabosti takšnega načina sortiranja so:

- večja poraba prostora ob sortirni postaji za varno in nemoteno delo operaterjev,
- hitrost sistema je odvisna od hitrosti operaterjev,
- sistem je lahko postavljen samo v ravnini, saj mora vsa prtljaga, ki je na njem, biti na doseg roke operaterja.

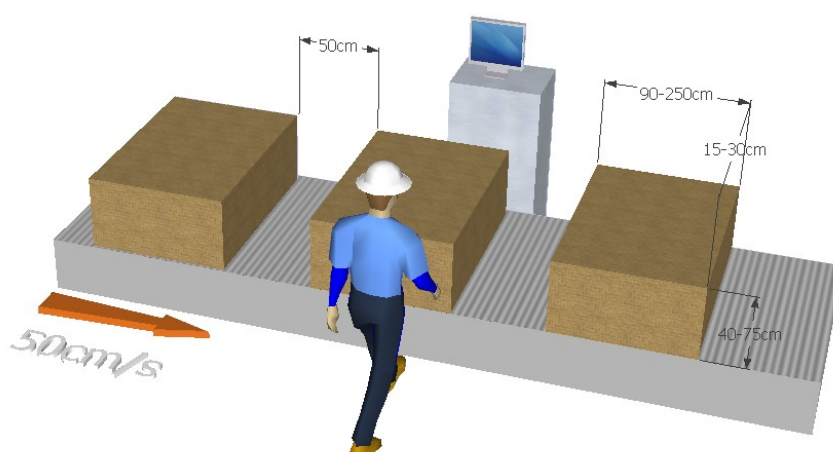
Kljub temu ima tak sistem tudi nekaj prednosti. Zaradi ravninske postavitve je lažje odpravljanje napak, kot je zagozdena prtljaga in preobremenjen trak, lažje je tudi vzdrževanje. Prav tako lahko eno sortirno postajo uporablja več operaterjev hkrati, kar pride prav ob večjih količinah prometa.

Naloga diplomske naloge je izdelava testne aplikacije, ki omogoča lažje sortiranje prtljage z uporabo RFID tehnologije. Aplikacija bo sama prebrala identifikacijsko kodo prtljage in poiskala njeno destinacijo zapisano v bazi. S tem bo razbremenila operaterja, saj mu ne bo več potrebno ročno branje črtnih kod s prtljage.

3.2. Izbira RFID sistema

Tekoči trak, ki predstavlja sortirni rondo, se premika s hitrostjo 0.5m/s. Razmak med dvema prtljagama naj bi bil 0.5m, vendar je ta v praksi manjši zaradi nihanja hitrosti traku med delovanjem in zatikanjem prtljage na ovinkih. Prtljaga, ki se sortira v sortirnici B (Slika 5), mora ustrezati sledečim merilom [10]:

- dolžina med 901 in 2500 mm,
- širina med 150 in 300 mm,
- višina med 400 in 750 mm,
- teža med 10 in 60 kg.



Slika 5: Sortirna postaja

Z uporabo zgornjih podatkov lahko postavimo zahteve, ki jih mora imeti izbran RFID sistem. Te so:

- RFID značke morajo biti majhne, v obliki nalepke. Prav tako morajo biti pasivne in poceni,
- RFID komunikacija mora biti hitra in zanesljiva,
- domet RFID sistema mora biti večji od enega metra.

Odločili smo se za uporabo čitalnika RFID Speedway revolution, ki ga izdeluje podjetje Impinj (Slika 6). Deluje v UHF frekvenčnem območju, njegov domet lahko preseže dolžino enega metra. Omogoča priklop in uporabo največ štirih anten. Za diplomsko nalogo smo uporabili dve anteni (Slika 6). Čitalnik RFID se z računalnikom povezuje preko ethernet mrežnega kabla in komunicira po TCP/IP protokolu. Izbrane RFID značke so pasivne in v obliki nalepk (Slika 7). Podatki na njih predstavljajo enoumno identifikacijsko številko UID (Unique Identification Number), ki je sestavljena iz 24 alfa numeričnih znakov. To številko uporabljamo kot identifikacijo prtljage[17].



Slika 6: Čitalnik RFID in anteni



Slika 7: RFID značka

3.3. Implementacija RFID sortirne postaje

Implementacijo RFID sortirne postaje smo začeli z razdelitvijo programa na funkcionalne bloke – module. Modularna zasnova omogoča, da programi za enake naloge uporabljajo enake module. Tak pristop omogoča postopno reševanje problemov in zelo pospeši razvoj dodatnih program, ki vsebujejo podobno funkcionalnost. Dodatna prednost je lažje vzdrževanje, saj popravki na modulih hkrati vplivajo na vse programe, ki te module uporabljajo. Programska koda je s tem veliko bolj pregledna in obvladljiva. Program smo razdelili na sledeče module:

- modul za upravljanje čitalnika RFID,
- modul za upravljanje z bazo,
- glavni program - RFID sortirna postaja.

Programsko kodo smo napisali v programskem jeziku C#. K njegovi izbiri so prispevale osebne izkušnje iz programiranja v njem in močna integracija .NET tehnologije, ki močno olajša razvoj ter dejstvo, da gre za visoko nivojski programski jezik. Knjižice za interakcijo s čitalnikom RFID prav tako podpirajo .NET tehnologijo, zato z izbiro programskega jezika tudi kasneje nismo imeli težav.

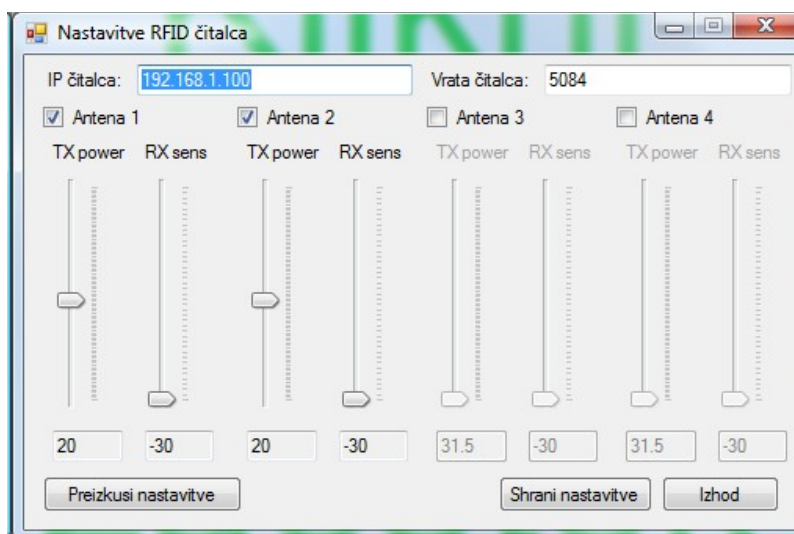
3.3.1. Modul za upravljanje čitalnika RFID

Implementacije modula za upravljanje čitalnika RFID smo se lotili z dvema projektoma. Prvi je prestavljal modul sam, drugi pa je služil za njegovo testiranje. Glavne funkcije modula so:

- povezovanje s čitalnikom RFID,
- shranjevanje in spreminjanje nastavitev čitalnika RFID,
- branje podatkov iz RFID značk.

Komunikacija med čitalnikom RFID in računalnikom poteka po TCP/IP protokolu. Čitalnik RFID je povezan z računalnikom preko mrežnega kabla. Na programskem nivoju za komunikacijo skrbi knjižica *Impinj.OctaneSdk*, ki je del Octane SDK (software development kit). Octane SDK je razširitev LLRP (Low Level Reader Protocol) toolkita, ki implementira LLRP komunikacijske protokole, podatkovne strukture in branje in ustvarjanje LLRP sporočil. Octane SDK podpira programska jezika C# in C++ in močno poenostavlja interakcijo s čitalnikom RFID [18, 19]. Za uspešno povezavo s čitalnikom RFID je potrebno poznati njegova vrata in IP naslov. Da uporabniku teh podatkov ne bi bilo potrebno vnašati ob vsakem zagonu programa, smo v modul vgradili podporo za uporabo konfiguracijske datoteke (Slika 8). Konfiguracija se vrši z uporabo uporabniškega vmesnika in zraven osnovnih nastavitev ponuja še sledeče:

- vklop in izklop posamezne antene,
- nastavev oddajne in sprejemne moči posamezne antene,
- preizkus povezave s čitalnikom,
- shranjevanje konfiguracije v datoteko.



Slika 8: Uporabniški vmesnik za vnos nastavitve čitalnika RFID

Nastavitve shranimo v datoteko v XML (Extensible Markup Language) obliki. Za uporabo XML formata smo se odločili zaradi mnogih knjižic, ki nam olajšajo delo z XML datotekami in so implementirane v programski jezik C#. Tako lahko s pomočjo serilizacije hitro in enostavno beremo in shranjujemo nastavitve (Koda 1). Nastavitve preberemo v podatkovno strukturo, ki se nahaja v pomnilniku [14].

```
RFIDNASTAVITVE rfNas = null;
FileStream fs = null;

fs = new FileStream("Nastavitve.xml", FileMode.Open, FileAccess.Read);
XmlSerializer xmlSer = new XmlSerializer(typeof(RFIDNASTAVITVE));

rfNas = (RFIDNASTAVITVE)xmlSer.Deserialize(fs);
```

Koda 1: Branje nastavitve v podatkovno strukturo

Struktura, v katero *XmlSerializer* kopira podatke mora, biti popolno enaka strukturi XML datoteke. Pri izdelavi strukture si lahko pomagamo z orodjem, ki je del Microsoftovega Visual Studia. Orodje se imenuje XSD (XML Schema Definition Tool) [11]. To orodje lahko izdela C# podatkovno strukturo na dva načina. V prvem načinu uporabi shematsko opisano strukturo XML datoteke, ki jo izdelamo sami in jo shranimo kot datoteko z xsd (XML Schema file) končnico. V drugem načinu lahko uporabimo že obstoječo XML datoteko in izdelamo shematski opis XML datoteke kar z XSD orodjem. V zadnjem koraku orodju XSD kot parameter podamo xsd datoteko in ta nam izdela ustrezno podatkovno strukturo v obliki C# datoteke (Koda 2).

```

C:\test>xsd Nastavitve.xml
Microsoft (R) Xml Schemas/DataTypes support utility
[Microsoft (R) .NET Framework, Version 4.0.30319.1]
Copyright (C) Microsoft Corporation. All rights reserved.
Writing file 'C:\test\Nastavitve.xsd'.

C:\test>xsd Nastavitve.xsd /c
Microsoft (R) Xml Schemas/DataTypes support utility
[Microsoft (R) .NET Framework, Version 4.0.30319.1]
Copyright (C) Microsoft Corporation. All rights reserved.
Writing file 'C:\test\Nastavitve.cs'.

C:\test>

```

Koda 2: Primer uporabe XSD orodja za ustvarjanje podatkovne strukture

Po uspešni implementaciji podpore za konfiguracijo smo se lotili implementacije povezovanja s čitalnikom RFID in branja podatkov z RFID značk.

Preden poskusimo vzpostaviti povezavo s čitalnikom RFID, vedno preverimo, ali je morda že nismo (Koda 3). Ko se uspešno povežemo, začnemo nastavljanje čitalnika RFID z nastavitvami iz konfiguracijske datoteke. Nastavljanje se začne z resetiranjem čitalnika RFID na tovarniške nastavitve in konča s pošiljanjem nastavitvev zapisanih v konfiguraciji.

```

public static void ConnectToReader()
{
    try
    {
        if (!isConnected)
        {
            ReadSettings(false);

            RfidReader.Connect(nas.IP, nas.Port);

            PrepareReader();

            isConnected = true;
        }
    }
    catch
    {
        isConnected = false;
    }
}

```

Koda 3: Funkcija za povezovanje s čitalnikom RFID

Potreba po resetiranju nastavitvev čitalnika RFID izhaja iz težave, ker konfiguracijska datoteka ne vsebuje vseh nastavitvev čitalnika RFID, ampak samo potrebne. Tako smo lahko prepričani, da so nastavitve čitalnika RFID točno takšne, kot si jih želimo. Programski vmesnik čitalnika RFID omogoča dva načina branja podatkov z RFID značk. Prvi deluje po načinu dogodka,

drugi kot seznam prebranih RFID značk v nekem časovnem razponu [18].

Pri uporabi prvega načina čitalnik RFID za vsako prebrano RFID značko sproži dogodek (Koda 4). Dogodek vsebuje rezultat v obliki seznama. Seznam ne vsebuje vseh RFID značk v območju, ampak vsebuje eno RFID značko v obliki seznama anten, ki so jo prebrale. Izdelava logike, ki te podatke naknadno združi v seznam vseh prebranih RFID značk, je na strani uporabnika. Pri tem se lahko hitro pojavi težava, saj čitalnik sproži dogodke samo, ko prebere RFID značko, ne sproži pa jih, ko RFID značka zapusti območje branja.

```
static void OnTagsReported(object sender, TagsReportedEventArgs args)
{
    // sprehod cez vse prebrane RFID odzivnike, več jih je v primeru
    // uporabe vecih anten ...
    foreach (Tag tag in args.TagReport.Tags)
    {
        Console.WriteLine("EPC koda: {0} Antena : {1}",
            tag.Epc, tag.AntennaPortNumber);
    }
}
```

Koda 4: Primer uporabe dogodkovnega načina

Ta način delovanja je primeren za alarmne naprave ali za ugotavljanje vstopa v prostore, ni pa primeren za našo aplikacijo, saj želimo seznam vseh prebranih RFID značk v območju. Zato smo se odločili za uporabo drugega načina.

Drugi način branja vrača rezultat v obliki seznama prebranih RFID značk v določenem časovnem obdobju. V primeru uporabe več anten lahko seznam večkrat vsebuje iste podatke. Slednjo lastnost seznama lahko uporabimo za izboljšave zanesljivosti ali hitrosti branja. Če želimo večjo zanesljivost, potem zahtevamo, da vse antene preberejo enako RFID značko. Tak seznam vsebuje podatke samo ene RFID značke in to tolikokrat, kolikor anten je v uporabi. Če pa želimo izboljšati hitrost branja, pa zahtevamo, da vsaj ena antena prebere RFID značko. Tak seznam mora vsebovati najmanj eno prebrano RFID značko. V kolikor jih vsebuje več, morajo te biti enake, drugače ne moremo zagotovo trditi, za katero značko gre (Koda 5).

```
public static string ReadTags()
{
    TagReport report = RfidReader.QueryTags(1);

    switch (report.Tags.Count)
    {
        case 1:
            return report.Tags[0].Epc;
            break;
        default:
            return null;
            break;
    }
}
```

Koda 5: Uporaba drugega načina branja, brez podpore uporabe večih anten

Za našo nalogo smo izbrali drugi način branja. Pri tem smo se odločili, da je v primeru uporabe več anten pomembnejša zanesljivost kot hitrost branja. Funkcija ob vračanju podatkov z RFID značk prav tako sporoča podatke o uspešnosti oziroma neuspešnosti branja. Čas branja smo določili s testiranjem v laboratoriju. Tekoči trak in razdalje med prtljagami smo simulirali z gibanjem osebe po laboratoriju. Oseba je nosila dve prtljagi na različnih medsebojnih razdaljah in jih ob različnih hitrostih premikala čez antene čitalnika RFID. Preizkušali smo različne moči oddajanja anten, različne dolžine bralnih časov in različne razdalje med prtljagami. Tak način testiranja in umerjanja nastavitvev čitalnika RFID je hitro pokazal svoje slabosti, predvsem kako simulirati hitrost tekočega traku na omejenem prostoru laboratorija. Tudi sama postavitvev anten je bila v laboratoriju zelo idealizirana, saj so bile antene zelo skupaj in njihova razdalja od tekočega traku je bila praktično minimalna, zato je bila potrebna moč anten tudi minimalna. Zaradi tako velikih približkov sam se odločili, da bomo v laboratoriju določili le najoptimalnejši bralni čas čitalnika RFID. Tako smo preizkušali čase od 100 milisekund do 1 sekunde v korakih po 100 milisekund. Za vsak preizkusni čas smo opravili 30 preletov ene same prtljage in preverjali, kolikokrat smo uspešno prebrali RFID značko. Uspešnost v časih, manjših od 500 milisekund, je bila izredno slaba, ponavadi pod 60 procenti. Za uspešnost večjo, od 80 procentov, smo morali uporabljati čase, večje od 800 milisekund. Za najkrajši in 100 odstotno uspešni način branja se je izkazal čas 1 sekunde. Na koncu smo preizkusili tudi čase, večje od 1 sekunde. Njihova uspešnost je bila 100 odstotna, vendar so ti časi branja predolgi in neuporabni v realnem okolju.

3.3.4. Modul za upravljanje z bazo

Implementacije modula za upravljanje z bazo smo se lotili z izbiro podatkovne baze. Odločili smo se za uporabo MS SQL 2008 R2 Express, saj je njena uporaba brezplačna in ne vsebuje nobenih funkcionalnih omejitev, ki bi lahko vplivale na našo nalogo [12]. Sledila je izdelava podatkovnega modela in ustrezne programske kode. Programska koda je bila prav tako spisana v jeziku C#.

Shranjevanja konfiguracije, potrebne za dostop do baze, smo v tem modulu implementirali na drug način. Odločili smo se za uporabo .NET knjižice, namenjene za delo s konfiguracijskimi datotekami. Gre za knjižico *ConfigurationManager*, ki je del imenskega prostora (ang. namespace) *System.config*. Slednja bere nastavitve iz datoteke *app.config* in še dodatno zmanjša programsko kodo, ki je potrebna za branje konfiguracijskih datotek (Koda 6). Kljub temu ima eno večjo pomanjkljivost. Večje konfiguracijske datoteke lahko hitro postanejo neberljive, saj je potrebno za posebne tipe vnosov izdelati njihove opise, ki se morajo nahajati v isti datoteki. Kadar uporabljamo znane vnose, kot so recimo povezovalne besede, (ang. *connection string*) teh težav ni (Koda 7) [13].

```
public static string GetConnStr()
{
    return
        ConfigurationManager.ConnectionStrings["RFID"].ConnectionString;
}
```

Koda 6: Branje zapisa iz konfiguracijske datoteke s pomočjo knjižice *ConfigurationManager*

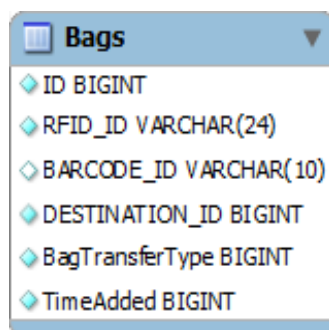
```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="RFID" connectionString="Data Source=tilen-
pc\sqlexpress;Initial Catalog=RFID;Integrated Security=True"
    providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>

```

Koda 7: Primer konfiguracijske datoteke, ki vsebuje podatke za dostop do baze

Podatkovni model smo sestavili iz dveh šifrantov in ene tabele (Slika 12). Tabela Bags služi kot seznam prtljage in vsebuje podatke, ki so potrebni za sortiranje (Slika 9).



Column Name	Data Type
ID	BIGINT
RFID_ID	VARCHAR(24)
BARCODE_ID	VARCHAR(10)
DESTINATION_ID	BIGINT
BagTransferType	BIGINT
TimeAdded	BIGINT

Slika 9: Tabela Bags

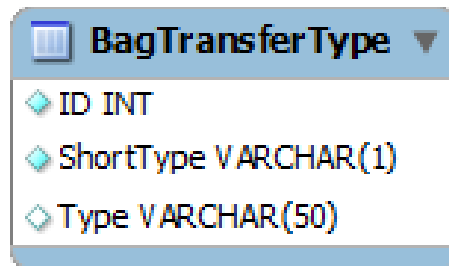
Tabela vsebuje dve identifikacijski številki prtljage. Prva in najpomembnejša je *RFID_ID*. Gre za podatek, sestavljen iz 24 alfa numeričnih znakov, ki ga preberemo iz RFID značke. Sledi mu številka črtna kode, ki jo shranimo v polje *BARCODE_ID*. Črtna koda je sestavljena iz 10 števil, vendar jo shranjujemo v obliki besedila zaradi možnosti, da se črna koda začne z ničlami. Številčna oblika shranjevanja bi te ničle odrezala in s tem bi izgubili pravi podatek. Naslednja podataka sta podataka o destinaciji prtljage in tipu poleta. Ta podataka sta zapisana kot povezava do njunih šifrantov. Podatek *TimeAdded* uporabljamo za možnost večkratne uporabe iste identifikacijske številke. Z njim sortiramo vse iskalne rezultate in v primeru, ko imamo več prtljag z enako identifikacijsko številko, vedno izberemo tisto, ki ima čas dodajanja največji. Tako vemo, katera prtljaga je bila dodana zadnja, torej je najnovejša. Tabela Airports je šifrant destinacij (Slika 10). Destinacije so letališča, ki so v tabeli opisana z dvema mednarodnima kodama, ICAO in IATA in z opisnim imenom.



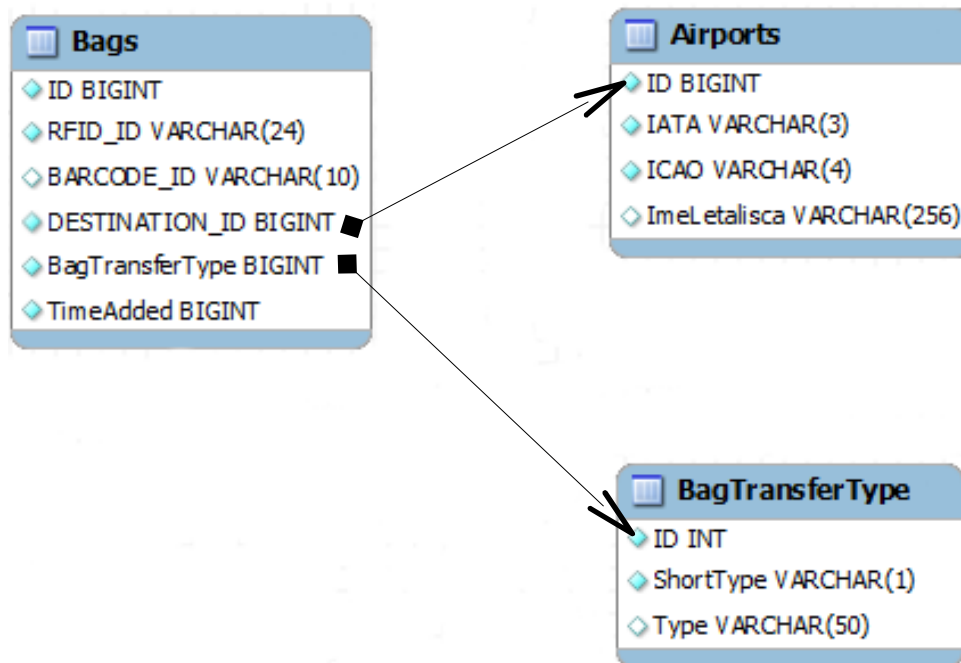
Column Name	Data Type
ID	BIGINT
IATA	VARCHAR(3)
ICAO	VARCHAR(4)
ImeLetalisca	VARCHAR(256)

Slika 10: Tabela Airports

Tabela BagTransferType predstavlja šifrant tipa poleta, na katerega je namenjena prtljaga (Slika 11). V naši nalogi uporabljamo dva tipa. To sta N, kot normalen let in T, kot prestopni (transfer) let. Spodnja slika prikazuje tabelo BagTransferType, ki je sestavljena iz enočrkovne kratice in njenega opisa.



Slika 11: Tabela BagTransferType



Slika 12: Podatkovni model

Za lažjo interakcijo modula z bazo smo iskanje prtljage po RFID UID ali črtni kodi sprogramirali kot shranjene procedure (ang. *stored procedure*). Tako smo logiko iskanja prenesli na bazo in zmanjšali pretok podatkov med bazo in modulom. Istočasno smo si olajšali vzdrževanje, saj nam v primeru spremembe tabel ni potrebno prevesti celotnega programa, ampak je dovolj, da popravimo procedure v bazi. Ostale procedure, kot so dodajanje in brisanje prtljage iz baze, smo implementirali neposredno v modulu. To smo naredili zato, ker gre za podporne funkcije, ki jih bomo uporabljali samo v programih, ki nam bodo olajšali testiranje in se kot take praktično ne spreminjajo [15].

Iskalni funkciji *GetBagByRFID(String rfid)* in *GetBagByBAR(string barcode)* pošljeta iskalne podatke v shranjeni proceduri. Rezultate nato preverita in pretvorita v podatkovno strukturo *Bag*. Če prtljaga ni najdena, funkciji vrneta vrednost *null*. Spodnja koda prikazuje shranjeno proceduro *GetBagByRFID*.

```

SELECT TOP 1 Bags.ID, Bags.RFID_ID, Bags.BARCODE_ID,
Bags.DESTINATION_ID, Airports.IATA, Airports.ICAO,
Airports.ImeLetalisca, BagTransferType.ShortType
FROM Bags
INNER JOIN Airports
ON Bags.DESTINATION_ID=Airports.ID
INNER JOIN BagTransferType
on Bags.BagTransferType=BagTransferType.ID
Where Bags.RFID_ID = @RFID
order by Bags.TimeAdded desc

```

Koda 8: SQL procedura za iskanje destinacije prtljage

Procedura išče prtljage, ki vsebujejo podano RFID UID številko. Najdeni prtljagi doda podatke iz šifrantov in rezultate sortira po datumu padajoče. S tem so rezultati urejeni od najmlajšega do najstarejšega zapisa. Ker funkcija zahteva samo najnovejšo dodano prtljago, procedura vrne samo prvi zapis. Rezultat se nato pretvori v podatkovno strukturo *Bag* (Koda 9). Podatke iz podatkovne strukture *Bag* nato prikažemo v glavni aplikaciji, sortirni postaji.

```

public class Bag
{
    public long BagID { get; set; }
    public long DestinationID { get; set; }

    public string RFID_ID { get; set; }
    public string BAR_ID { get; set; }

    public string DEST_ICAO { get; set; }
    public string DEST_IATA { get; set; }
    public string ImeLetalisca { get; set; }

    public String SortType { get; set; }
}

```

Koda 9: Podatkovna struktura bag

3.3.5. Izdelava glavnega programa

Po izdelavi modulov smo začeli izdelovati glavno aplikacijo. Naloga slednje je ugotavljanje katera prtljaga se nahaja pod senzorji in izpis njene destinacije na zaslon. Prav tako mora med delovanjem omogočati nastavljanje parametrov čitalnika RFID in izbiro destinacij, ki se bodo prikazovale na ekranu. Za dostop do potrebnih podatkov mora uporabljati že izdelana modula, njen uporabniški vmesnik pa mora biti kar se da enostaven in intuitiven.

Razvoj smo pričeli z izdelavo uporabniškega vmesnika. Prostor na ekranu smo razdelili na polja, ki prikazujejo informacije (Slika 13). Velikosti polj smo izbrali glede na pomembnost informacije, ki jo prikazujejo. Sledila je izdelava shematske risbe in implementacija.

Največji prostor na ekranu zaseda polje, v katerem se prikazuje destinacija prtljage. Zraven destinacije prtljage se v oklepaju pod njo prikazuje tudi njen tip poleta. Zraven informacij o

destinaciji se v to polje prav tako izpisujejo napake, ki se pojavijo med branjem RFID značk. Polja, ki vsebujejo lučke, se uporabljajo kot pomoč glavnemu polju. Tako s pomočjo grafike še dodatno razbremenimo operaterja pri prepoznavanju prtljage.

Desno spodaj se izpišejo vse izbrane destinacije. S klikom na seznam se odpre okno za izbiro destinacij. V levem spodnjem kotu se nahaja dnevniško (log) okno. Slednje izpiše vse, kar se dogaja med delovanjem programa in operaterju služi kot, indikator ali program deluje. Na vrhu je postavljen glavni meni, katerega naloga je dostop do nastavitve čitalnika RFID.

Glavni meni		
Zelena lučka	Glavno okno, izpisuje destinacijo prtljage	Rdeča lučka
Izbrane destinacije		Dnevniško okno

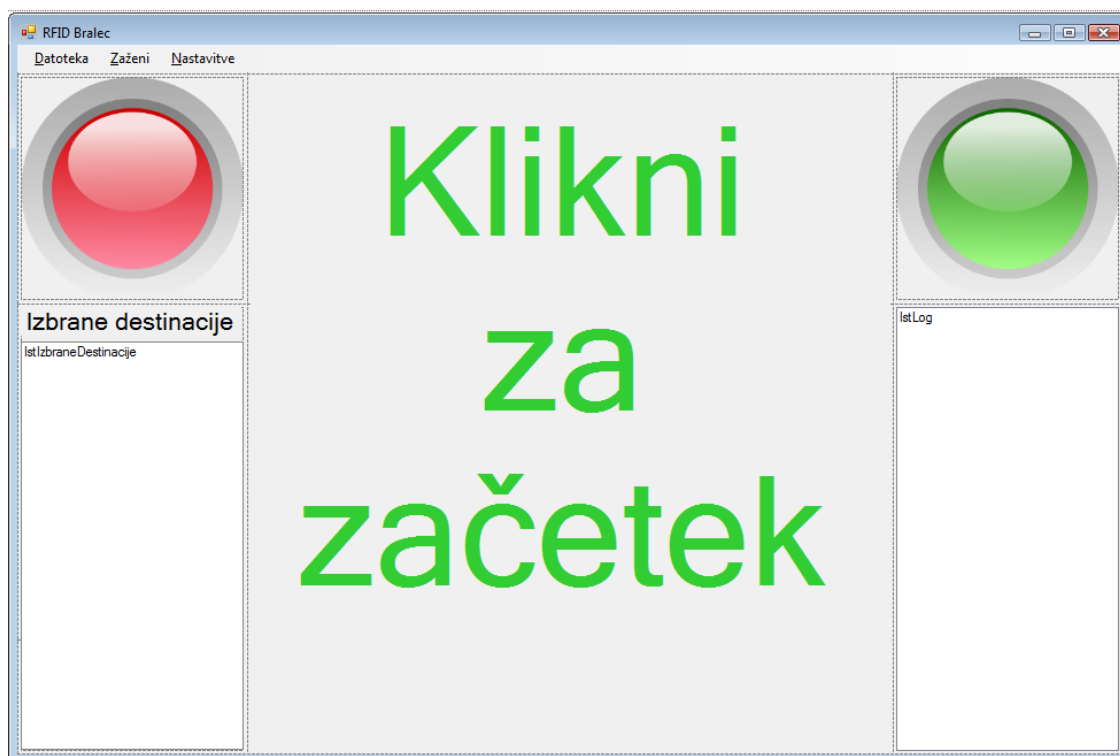
Slika 13: Shematska slika uporabniškega vmesnika

Za implementacijo uporabniškega vmesnika smo uporabili kontrolo *TableLayoutPanel*. Slednja ob souporabi kontrole *panel* omogoča izdelavo vmesnika na podlagi tabele. Kontrola *TableLayoutPanel* omogoča razdelitev ekrana na vrstice in stolpce. Njihove dimenzije so lahko podane v obliki števila pikslov, v obliki procentov celotnega ekrana ali samodejno. Načini se lahko uporabljajo kombinirano [16].

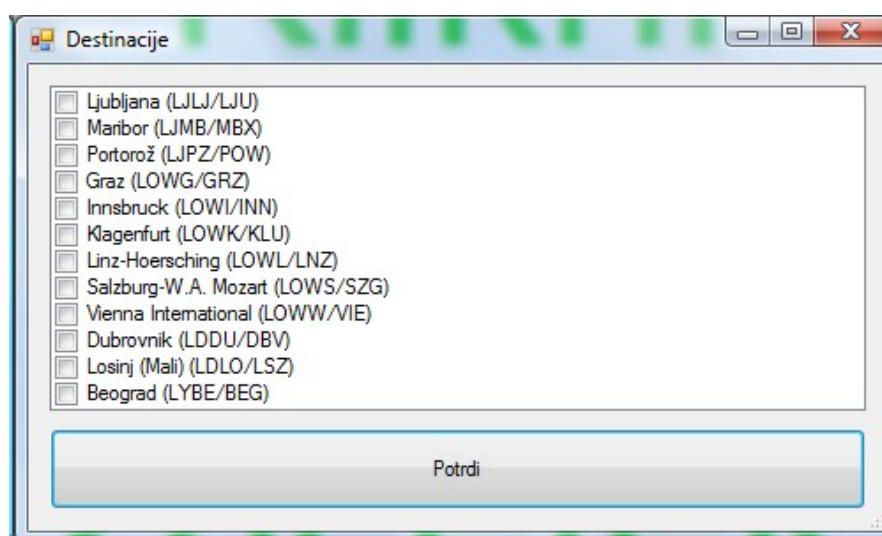
Izdelali smo kontrolo *TableLayoutPanel* s tremi stolpci in vrsticami. Velikost prvega in zadnjega stolpca je fiksna, širina srednjega, največjega, pa je samodejna. Fiksna širina onemogoča širjenje in krčenje stolpcev ob povečevanju ali krčenju okna in tako omogoča celovitost grafične slike. Velikost prve vrstice je fiksna, saj je takšna tudi velikost glavnega menija. Velikost druge vrstice je prav tako fiksna, saj je takšna tudi velikost slike rdeče in zelene lučke. Velikost zadnje vrstice je samodejna, da lahko zasede formo do njenega spodnjega konca. Ker na sredini želimo samo eno vrstico, smo vse vrstice združili s pomočjo kontrole *panel* in njenih lastnosti *ColumnSpan* in *RowSpan*. Tako smo v celoti implementirali

vmesnik, kot smo si ga zamislili na shematski sliki (Slika 13) [17].

Ker smo uporabniški vmesnik za nastavitve čitalnika RFID naredili ob implementaciji modula za upravljanje čitalnika RFID, smo slednjega samo dodali v glavni meni. Tako nam je ostala le še izdelava uporabniškega vmesnika za izbiranje destinacij (Slika 15). Slednji je sestavljen samo iz seznama destinacij in gumba za potrditev izbire. Pri izdelavi tega uporabniškega vmesnika ni bilo potrebno poseči po uporabi kontrole *TableLayoutPanel* in *panel*.



Slika 14: Izdelan uporabniški vmesnik aplikacije za sortiranje prtljage



Slika 15: Uporabniški vmesnik za izbiranje destinacij prtljage

Sledila je implementacija branja podatkov z RFID značk s pomočjo modula za upravljanje čitalnika RFID. Slednje se mora izvajati periodično in ne sme onemogočiti izrisovanja uporabniškega vmesnika. Vso branje podatkov z RFID značk izvajamo v ločeni niti. Pri tem si pomagamo s kontrolo *BackgroundWorker*. Branje izvajamo periodično s pomočjo *Timer* kontrole, ki se proži vsakih 100 milisekund. Preden zaženemo nit, ki bere podatke, vprašamo kontrolo *BackgroundWorker*, ali je ta prosta (Koda 10). To je potrebno zato, ker branje in poizvedovanje v bazi traja najmanj eno sekundo, sprožimo pa jo vsakih 100 milisekund.

```
if (!bgRfidIterogator.IsBusy)
{
    bgRfidIterogator.RunWorkerAsync();
}
```

Koda 10: Preverjanje ali je kontrola BackgroundWorker prosta

Branje v ločeni niti smo naredili zaradi njegove dolžine. V nasprotnem primeru bi čakanje na rezultate ustavilo risanje grafičnega vmesnika in uporabnik bi dobil občutek, da se je aplikacija ustavila. Z uporabo kontrole *BackgroundWorker* se temu izognemo. Ob tem se pojavi težava sporočanja podatkov med dvema nitima. Zato moramo uporabiti *delegate* in *Invoke* klice, da glavno okno aplikacije obvestimo o rezultatih iz niti za branje (Koda 11).

```
String prebrano = String.Empty;
RFIDMsg status = RFIDHelper.ReadTags(out prebrano);

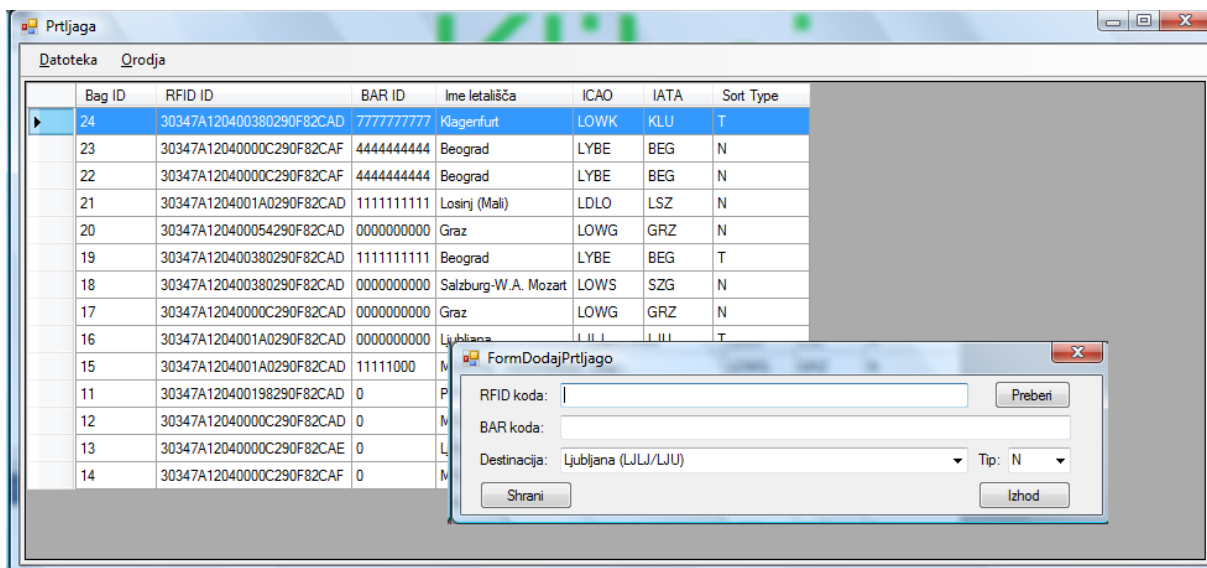
this.Invoke(new FormMain.ReportReadedTagsDel(ReportReadedTags), prebrano,
status);
```

Koda 11: Sporočanje podatkov iz bralne niti v glavno okno aplikacije

Povezovanje prebrane identifikacije s podatki iz baze izvajamo v glavnem oknu. Uporaba kontrole *BackgroundWorker* v tem primeru ni smiselna, saj je branje iz baze hitro in ga uporabnik ne opazi. Prav tako je lažje razhroščevanje, ker se funkcija izvaja samo v eni niti. Podatki, ki jih sporoči nit za branje, se prav tako izpišejo v dnevniško okno. Uporabnik tako ima občutek, da aplikacija deluje, saj neprestano izpisuje, kaj počne.

3.3.6. Izdelava ostalih programov

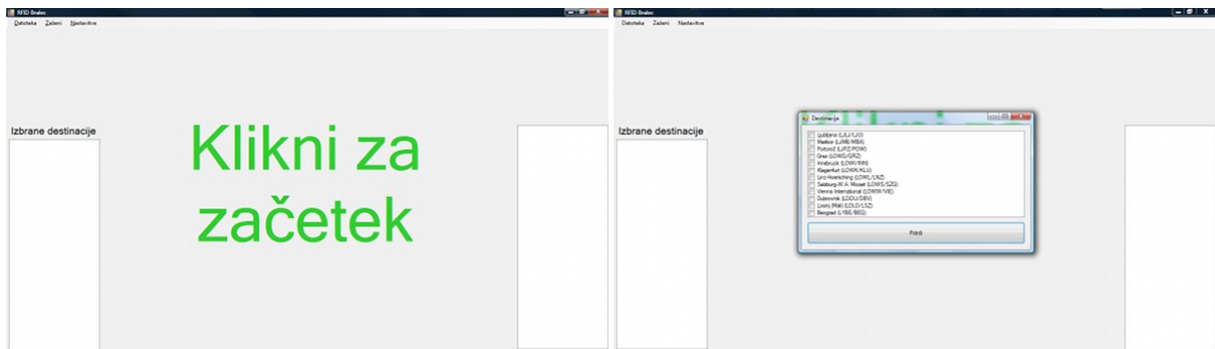
Za lažje testiranje smo izdelali dodaten program, s katerim polnimo tabelo, ki vsebuje podatke o prtljagi (Slika 16). Tako nam ni potrebno neposredno vnašati podatke v bazo. Program nam omogoča tako branje podatkov z RFID značke kot uporabo opisnih imen iz šifrantov. Največji del izdelave programa je predstavljala izdelava uporabniškega vmesnika. Funkcionalnost, potrebno za branje RFID značk in interakcije s podatkovno bazo, pa smo dobili iz že izdelanih modulov. Spodnja slika prikazuje uporabniški vmesnik programa za dodajanje in urejanje prtljage.



Slika 16: Program za dodajanje in urejanje prtljage

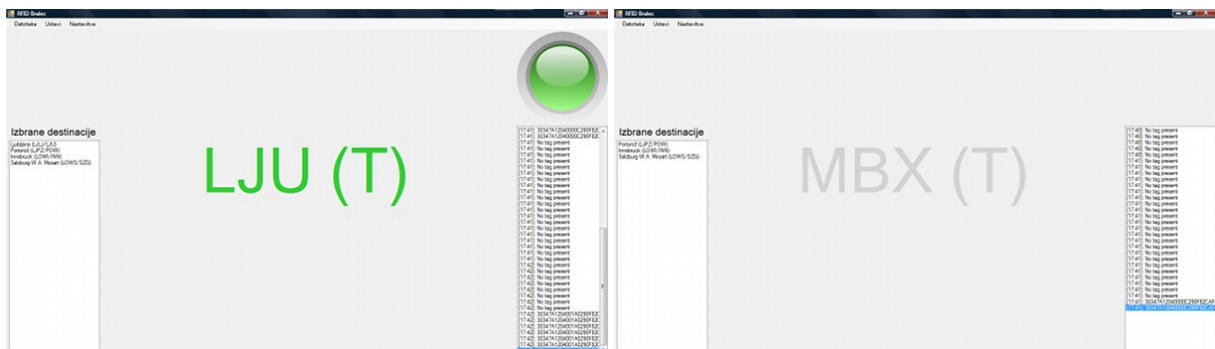
4. Predstavitev aplikacije in testiranje

Končano aplikacijo smo najprej preizkusili v laboratoriju. Preizkušali smo kombinacije z različnim številom in postavitvami anten. Tako smo ugotovili njihove prednosti in slabosti in se bolje pripravili za preizkušanje na letališču. Spodnje slike prikazujejo različna stanja med delovanjem programa.



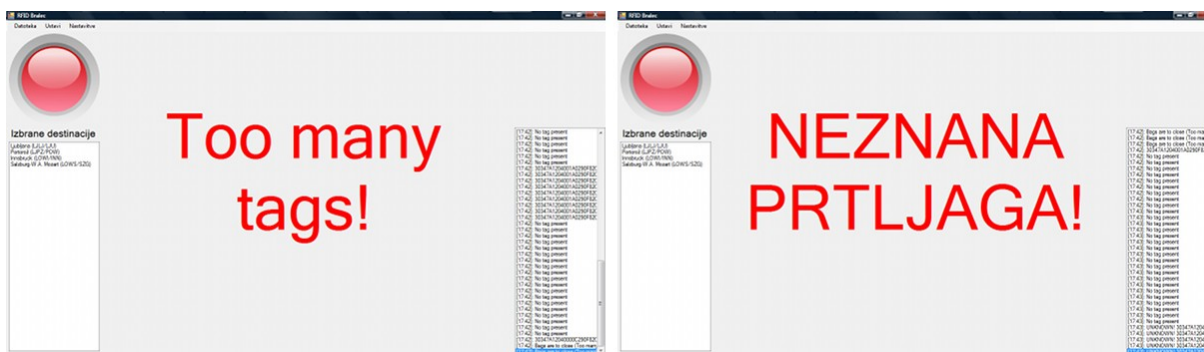
Slika 17: Izgled aplikacije ob zagonu in začetku dela z njo

Slika 17 prikazuje izgled uporabniškega vmesnika takoj po zagonu programa. Dnevniško okno je prazno, prav tako ni izbrana nobena destinacija. Ko želi uporabnik pričeti z delom, klikne na zelen napis in odpre se okno za izbiro destinacij. Ob potrditvi izbire se ta izpiše v seznam izbranih destinacij in program prične z delovanjem.



Slika 18: Izgled aplikacije ob prebranju nastavljenih in nenastavljenih destinacij

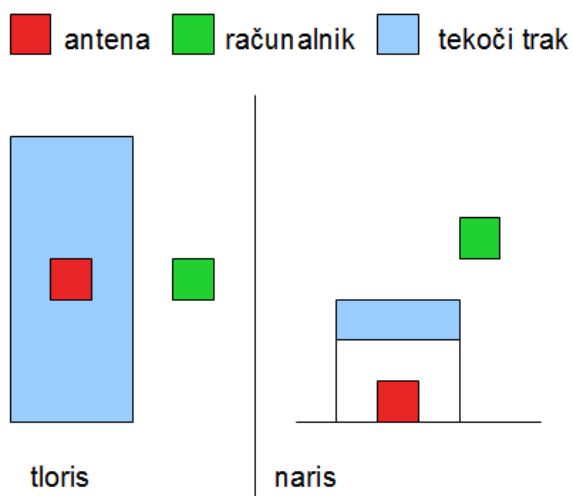
Podatke o prepoznani prtljagi program prikazuje na dva načina. Slika 18 levo prikazuje aplikacijo, ko prebere destinacijo, ki je med izbranimi. Slika 18 desno pa prikazuje aplikacijo, ko prebere destinacijo, na katero ni nastavljena. Slika 19 prikazuje izgled aplikacije v primeru napak. Leva slika prikazuje primer, ko aplikacija prebere podatke iz več kot ene RFID značke, desna pa, ko aplikacija ne uspe najti podatkov o prebrani prtljagi v bazi. Slika 19 prav tako prikazuje izgled podatkov v dnevniškem oknu. Ti se polnijo periodično in vsebujejo rezultate, ki jih vrne modul za upravljanje s čitalnikom RFID.



Slika 19: Izgled aplikacije ob sporočanju napak

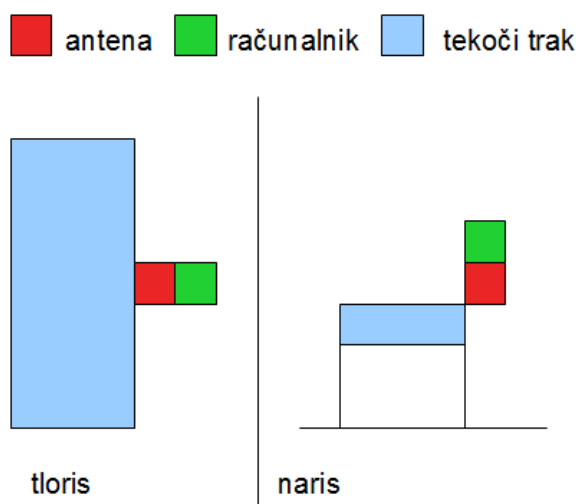
4.1. Postavitve anten

V prvem preizkusu smo uporabili eno anteno, ki smo jo postavili pod tekoči trak (Slika 20). Gre za osnovno konfiguracijo, s katero smo preizkusili splošno delovanje aplikacije in s katero smo opazovali vpliv ovir na potrebno oddajno moč antene. To konfiguracijo smo uporabili tudi pri aplikaciji za dodajanje podatkov o prtljagi v testno bazo. Za uspešno branje podatkov z RFID značk je morala oddajna moč antene biti najmanj 21,75dbm, kar predstavlja zgornjo tretjino največje oddajne moči.



Slika 20: Postavitev ene antene, postavljene pod tekoči trak

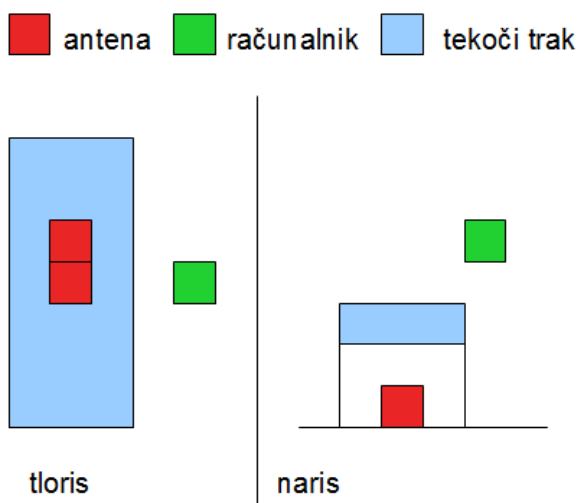
Sledila je postavitve pod računalnik (Slika 21). S tem smo se izognili oviram, ki sestavljajo tekoči trak.



Slika 21: Konfiguracija z eno anteno, postavljeno ob tekoči trak

Ta konfiguracija potrebuje oddajno moč antene 14dbm, kar je znatno nižje od primera, v katerem je antena pod tekočim trakom (Slika 20). Oddajna moč mora biti čim manjša, saj lahko drugače pride do napake, ko čitalnik RFID prebere podatke iz več RFID značk. Ta napaka se na primer pojavi, ko ima operater za sabo voziček z že sortirano prtljago.

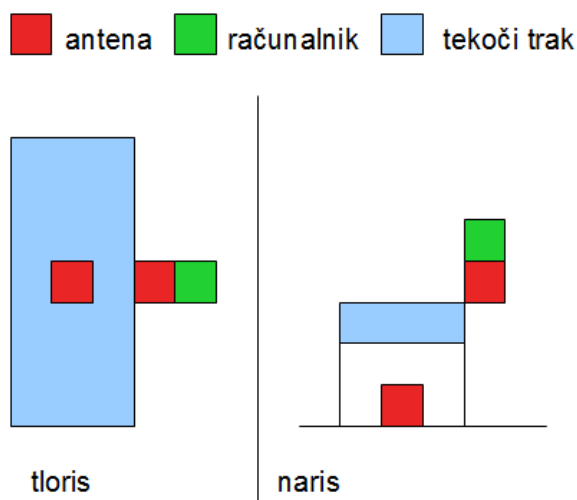
Preizkušanje smo nadaljevali z uporabo obeh anten. S tem smo želeli povečati zanesljivost branja podatkov iz RFID značk. Kot prvo smo preizkusili postavitev dveh anten pod tekočim trakom (Slika 22).



Slika 22: Konfiguracija z dvema antenama, postavljenima pod tekoči trak

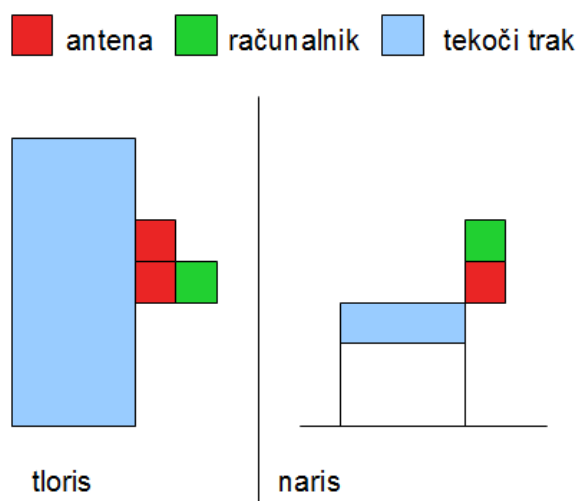
Žal morajo biti oddajne moči anten še vedno v zgornji tretjini. Branje je zaradi večje pokritosti uspešnejše, vendar pa se poveča občutljivost na nezadosten razmak med prtljagami in večkrat pride do napake, ker čitalnik RFID prebere več RFID značk hkrati. Nadaljevali smo s kombinacijo obeh prvi preizkusov (Slika 23). Slednja se je izkazala za najuspešnejšo.

Obema antenama smo lahko zmanjšali oddajno moč, saj sta veliko bolje pokrili območje prtljage in posledično RFID značke.



Slika 23: Konfiguracija z dvema antenama, postavljenima pod in ob tekoči trak

Za konec smo preizkusili še postavitev dveh anten ob računalnik (Slika 24). Težave te postavitve so enake kot pri postavitvi dveh anten pod tekoči trak. Ob tem nastane dodatna težava, da preberemo več kot samo RFID značko, v primeru, ko se voziček z že sortirano prtljago nahaja neposredno za uporabnikom.



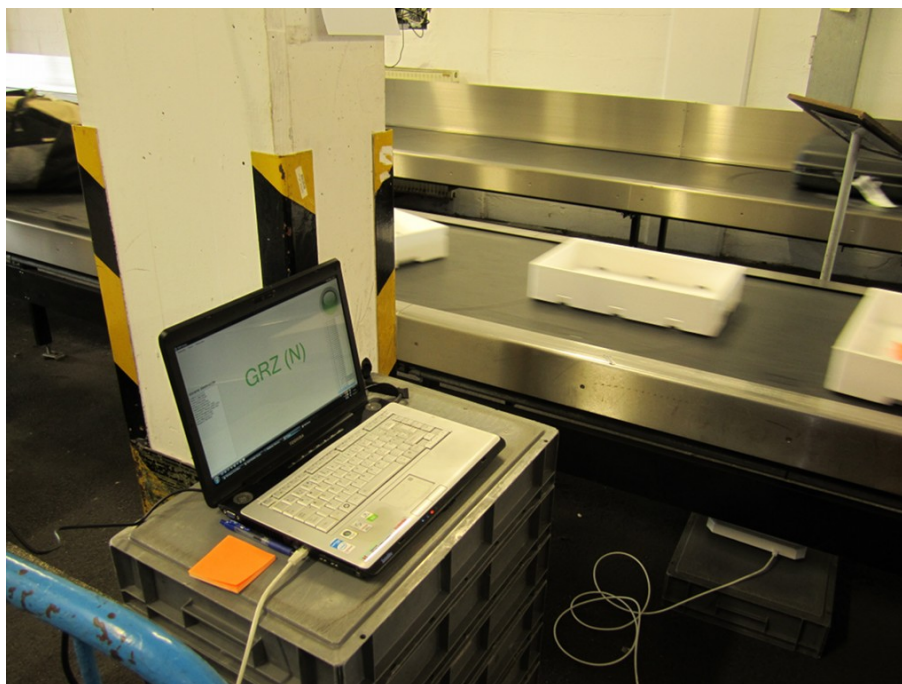
Slika 24: Konfiguracija z dvema antenama, postavljenima ob računalnik

Pravilna postavitev anten je zelo pomembna, še posebej če potrebujemo hitro in zanesljivo branje. Za doseg tega morajo antene s svojim snopom pokrivati samo prostor, ki ga zavzema prtljaga. Njihova oddajna moč mora biti čim večja, vendar ob tem ne smejo brati podatke z RFID značk, ki se ne nahajajo neposredno pod njimi.

4.2. Testiranje na letališču

Po preizkušanju v laboratoriju smo se odpravili na letališče Jožeta Pučnika Ljubljana v sortirnico B, kjer smo sistem preizkusili še v realnem okolju. Za simulacijo prtljage smo uporabili stiroporne banice velikosti 60 krat 40 krat 10 centimetrov (DxŠxV). Na vsako banico smo prilepili RFID značko in jo s pomočjo aplikacije za dodajanje prtljage dodali v bazo. Dodeljeno destinacijo smo za lažje preizkušanje napisali na listek, ki smo ga položili vanjo.

Preizkušanje smo začeli s postavitvijo ene antene pod tekoči trak (Slika 25). Sledilo je polaganje banic na tekoči trak z medsebojno razdaljo pol metra. Preizkus smo začeli z oddajno močjo, ki smo jo določili v laboratoriju, vendar se je ta ob prvem prehodu prtljage izkazala za premajhno. Da smo lahko uspešno prepoznavali prtljago, smo morali oddajno moč povečati na njeno največjo vrednost 31,50dbm. Kljub temu aplikaciji večkrat ni uspelo prepoznati prtljage, zato smo se odločili, da preizkušanje nadaljujemo z dvoantenskimi konfiguracijami.



Slika 25: Preizkušanje z eno anteno v sortirnici B

Kot prvo dvoantensko konfiguracijo smo preizkusili konfiguracijo z dvema antenama pod tekočim trakom. Njuni oddajni moči smo takoj nastavili na največjo vrednost, saj ob manjši sistem ni deloval. Ob 50 centimetrskem razmaku med prtljagami je bilo branje uspešno, vendar je ob zmanjšanju razmaka, uspešnost branja hitro padala. Do manjše razdalje prihaja zaradi ovinkov, ki sestavljajo sortirni rondo. Prtljaga se tam lahko zaleti ob zid ali celo obstane.

Vedno bolj opazen problem je postajal čas, ki ga aplikacija potrebuje za celoten postopek prepoznavanja prtljage. Tekoči trak ima hitrost pol metra na sekundo, kar pomeni, da se prepoznana prtljaga ob prikazu rezultatov nahaja najmanj pol metra stran od računalnika.

Rešitev težav je prestavljanje anten ali pospešitev bralnega časa. Odločili smo se, da začnemo preizkušati najobetavnejšo postavitev iz laboratorija, postavitev antene pod računalnik in pod tekoči trak (Tabela 3).

Tabela 3: Začetne nastavitve moči anten ob dvoantenski konfiguraciji

Nastavitev anten	
Pod računalnikom	Pod tekočim trakom
20 dbm	27 dbm

Slednje so se izkazale za premočne (Tabela 3). Zaradi ozkosti sortirnega rondoja je prišlo do primera, ko sta bili banici vzporedni in je aplikacija prebrala več kot eno RFID značko (Slika 26). Nadaljevali smo s preizkušanjem različnih oddajnih moči in določili optimalne (Tabela 4).



Slika 26: Premočne nastavitve moči anten in več prebranih RFID značk hkrati

Tabela 4: Optimalne nastavitve moči anten v sortirnici B

Nastavitev anten	
Pod računalnikom	Pod tekočim trakom
16 dbm	25,5 dbm

V tej konfiguraciji (Tabela 4) smo prav tako poskušali skrajšati čas branja RFID značk. Vsako nastavitev časa smo preizkusili s preходом najmanj desetih RFID značk (Tabela 5).

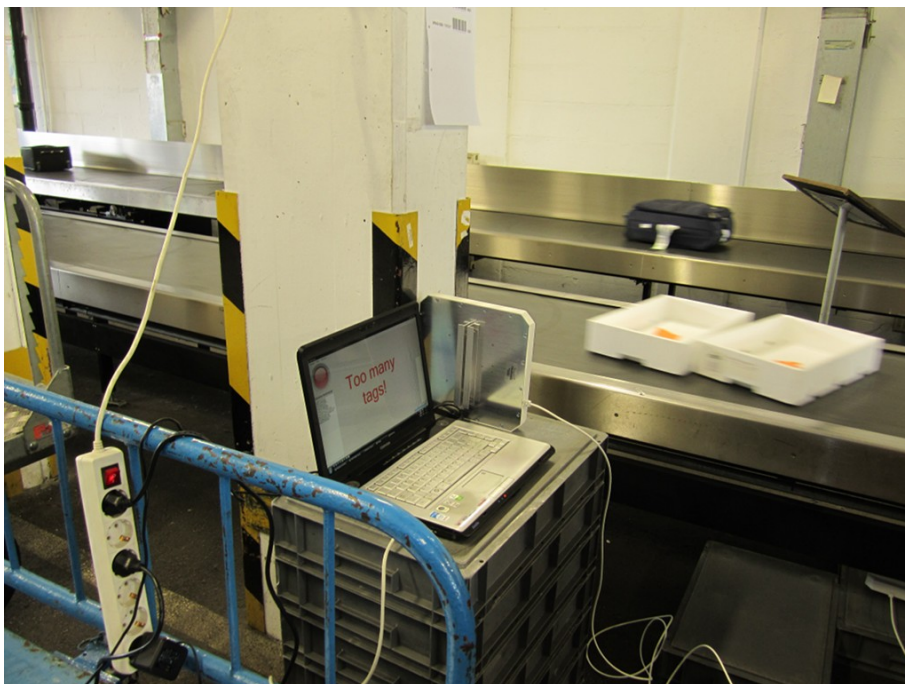
Tabela 5: Uspešnost branja RFID značk glede na čas v sortirnici B

Čas branja	Uspešnost
0,05 s	10%
0,2 s	50%
0,5 s	80 %
0,8	90%

Izkazalo se je, da je čas ene sekunde, ki smo ga določili v laboratoriju, najustreznejši tudi za realno okolje sortirnice B. Problem hitrosti branja v primeru sortirnice B ne moremo označiti kot težavnega. Ker gre za ročno sortiranje, kjer hitrost traku ni velika in problem, da prtljaga prepotuje razdaljo pol metra, preden jo aplikacija prepozna, lahko rešimo s postavitvijo anten pol metra višje od sortirne postaje.

Za večji problem se izkaže pravilna nastavitev anten. Najboljša bi bila postavitve v obliki polkroga, ki bi visel nad tekočim trakom. Po celem loku bi bile postavljene antene, ki bi kar najbolj pokrile področje prehoda prtljage. Takšno konfiguracija bi lahko brala hitreje in zanesljiveje, ter bi jo lahko uporabili pri nadgradnji sistema, ki bi uporabljal robotske izmetne roke. Vendar tudi polkrožna postavitve anten ne bi mogla uspešno prepoznati prtljage, ki imajo premajhno medsebojno razdaljo (Slika 27).

Ta primer zahteva ali ukrep operaterja ali postavitve dodatnega samodejnega povratnega tokokroga, ki bi tako prtljago ponovno ločil na najnižjo medsebojno razdaljo.



Slika 27: Neuspešno branje RFID značk, zaradi premajhne medsebojne razdalje

5. Sklepne ugotovitve

V sklopu diplomske naloge smo razvili testni sistem za sortiranje prtljage. Glavni namen sistema je lažjanje dela operaterja tako, da temu ni potrebno ročno branje identifikacije prtljage. Slednje opravlja sistem s pomočjo RFID tehnologije in tako razbremeni operaterja. Ob tem smo opravili veliko testiranj tako v laboratoriju kot v sortirnici B na letališču Jožeta Pučnika Ljubljana. Izdelali smo tudi aplikacijo, ki komunicira z čitalnikom RFID in bere podatke z RFID značk, ki označujejo prtljago. S temi podatki nato v bazi poišče destinacijo prtljage in jo izpiše na zaslon. Operater prebere destinacijo in prtljago sortira. Aplikacija je zaradi svoje modularne zasnove enostavno nadgradljiva za uporabo v sistemih, ki za sortiranje uporabljajo robotske roke.

V laboratoriju smo določili najkrajši čas branja, ki ustreza zahtevam za sortiranje prtljage. Ta čas je ena sekunda in smo ga s testiranjem v sortirnici B tudi potrdili (Tabela 5). Prav tako smo v laboratoriju preizkusili različne postavitev anten z namenom, da bi izbrali najugodnejšega. Slednje se je izkazalo za težavno, saj smo tekoči trak simulirali s pomočjo osebe, ki se je premikala s hitrostjo pol metra na sekundo. Tudi sam laboratorij je bil za tovrstno testiranje premajhen. Zato smo večino testiranja s postavitvami anten opravili v sortirnici B. S pomočjo meritev smo določili, da je najobetavnejša postavitev z dvema antenama. Ena antena se nahaja pod tekočim trakom, druga pa ob tekočem traku (Slika 22). Ob tem smo določili tudi moči, ki najbolj ustrezajo za sortirno postajo. Moč antene pod tekočim trakom mora biti 25,5 dbm, moč antene ob traku pa 16 dbm (Tabela 4). Razlog, da je moč antene, ki je ob tekočem traku, toliko manjša, je bližina vzporednega tekočega traku. Pri večjih močeh te antene prihaja do branja prtljage, ki se nahaja na vzporednem – povratnem tekočem traku. Med testiranjem v sortirnici B se je pokazala pomembnost pravilnega odmika položaja anten od sortirne postaje. Celoten čas branja podatkov z RFID značke in povezovanje s podatki v bazi traja več kot eno sekundo in v tem času prtljaga prepotuje več kot pol metra. Torej se prebrana prtljaga ne nahaja več pod sortirno postajo. Slednje lahko zmede operaterja. V primeru krmiljenja roke, pa bi se robotska roka sprožila veliko prepozno. Po končanem preizkušanju smo določili izboljšave, ki bi ta sistem približale uporabnosti v realnem okolju. Te so:

- Zmanjšanje bralnega časa

Da bi sistem lahko krmil robotske roke, bi mu nujno morali zmanjšati čas, ki ga potrebuje za branje podatkov iz RFID značk. Za doseg tega je potrebna uporaba več anten in krajšanje bralnega časa aplikacije. Ob tem bi morali zagotoviti, da se RFID značka, v času branja čitalnika RFID, vedno nahaja v snopu anten. Ena boljših postavitev anten bi bila postavitev v polkrogu nad tekočim trakom. Za dodatno zmanjšanje motenj in napak, bi celoten lok obdali s faradejevo kletko, proženje bralnega cikla čitalnika RFID pa bi upravljal stikala, ki bi se vklopila ob prihodu prtljage pod antene. Takšna postavitev bi omogočila zmanjšanje časa, ki ga aplikacija potrebuje za uspešno branje in prepoznavanje prtljage in bi bila uporabna za krmiljenje robotskih rok.

- Samodejno popravljanje razmaka med prtljagami

Problem najmanjšega razmaka med prtljagami in posledično preveč prebranih RFID značk bi reševali s pomočjo izločanja te prtljage na povratni tekoči trak. Trak bi nato sam poskrbel za ponovno vzpostavitev najmanjšega razmaka med prtljagami in jih vrnil na sortirni rondo.

- Uporaba drugačnih RFID značk

Ob možnosti zapisovanja podatkov na RFID značke bi lahko celoten sistem preoblikovali tako, da ta ne bi potreboval glavnega računalnika, ki s prebranimi podatki z RFID značke po bazi išče njen končni cilj. Namesto tega bi celotno pot prtljage do ustrezne destinacije napisali na RFID značko, ki ima dodan uporabniški pomnilnik. Pred vsako robotsko roko bi čitalnik RFID to pot prebral in se na podlagi nje odločil, ali jo bo sprožil.

Ob razvoju sortinega sistema sem dodobra spoznal RFID tehnologijo, njene prednosti in slabosti. Prav tako sem spoznal vse težave, ki jih prinašajo realna okolja, ki jih je težko simulirati v laboratoriju. S tem sem pridobil boljši vpogled v to, kako potekajo razvoji večjih industrijskih sistemov.

Dodatek A

Seznam slik

Slika 1: Manjši sortirni sistem.....	4
Slika 2: Različna čitalnika RFID.....	6
Slika 3: Različne RFID značke.....	6
Slika 4: Shema sortirnice B na letališču Jožeta Pučnika Ljubljana.....	8
Slika 5: Sortirna postaja.....	9
Slika 6: Čitalnik RFID in anteni.....	10
Slika 7: RFID značka.....	10
Slika 8: Uporabniški vmesnik za vnos nastavitvev čitalnika RFID.....	12
Slika 9: Tabela Bags.....	16
Slika 10: Tabela Airports.....	16
Slika 11: Tabela BagTransferType.....	17
Slika 12: Podatkovni model.....	17
Slika 13: Shematska slika uporabniškega vmesnika.....	19
Slika 14: Izdelan uporabniški vmesnik aplikacije za sortiranje prtljage.....	20
Slika 15: Uporabniški vmesnik za izbiranje destinacij prtljage.....	20
Slika 16: Program za dodajanje in urejanje prtljage.....	22
Slika 17: Izgled aplikacije ob zagonu in začetku dela z njo.....	23
Slika 18: Izgled aplikacije ob prebranju nastavljenih in nenastavljenih destinacij.....	23
Slika 19: Izgled aplikacije ob sporočanju napak.....	24
Slika 20: Postavitev ene antene, postavljene pod tekoči trak.....	24
Slika 21: Konfiguracija z eno anteno, postavljeno ob tekoči trak.....	25
Slika 22: Konfiguracija z dvema antenama, postavljenima pod tekoči trak.....	25
Slika 23: Konfiguracija z dvema antenama, postavljenima pod in ob tekoči trak.....	26
Slika 24: Konfiguracija z dvema antenama, postavljenima ob računalnik.....	26
Slika 25: Preizkušanje z eno anteno v sortirnici B.....	27
Slika 26: Premočne nastavitve moči anten in več prebranih RFID značk hkrati.....	28
Slika 27: Neuspešno branje RFID značk, zaradi premajhne medsebojne razdalje.....	30

Dodatek B

Seznam tabel

Tabela 1: Prednosti in slabosti aktivnih in pasivnih RFID sistemov.....	7
Tabela 2: Primerjava prednosti in slabosti RFID frekvenčnih območij.....	7
Tabela 3: Začetne nastavitve moči anten ob dvoantenski konfiguraciji.....	28
Tabela 4: Optimalne nastavitve moči anten v sortirnici B.....	29
Tabela 5: Uspešnost branja RFID značk glede na čas v sortirnici B.....	29

6. Seznam uporabljenih virov

- [1] (2010) Radio-frequency identification. Dostopno na:
http://en.wikipedia.org/wiki/Radio-frequency_identification
- [2] (2010) How RFID Works. Dostopno na:
<http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid.htm>
- [3] (2010) How Baggage Handling Works. Dostopno na:
<http://science.howstuffworks.com/transport/flight/modern/baggage-handling.htm>
- [4] (2005) McCarran Airport RFID System Takes Off. Dostopno na:
<http://www.rfidjournal.com/article/view/1949>
- [5] (2007) Milan's Malpensa Airport Prepares for RFID Baggage Handling. Dostopno na:
<http://www.rfidjournal.com/article/view/3534>
- [6] (2009) Hong Kong Airport Says It Now Uses Only RFID Baggage Tags. Dostopno na:
<http://www.rfidjournal.com/article/view/4885>
- [7] (2010) Aalborg Airport Debuts Baggage-Handling System With High-Memory RFID Tags. Dostopno na:
<http://www.rfidjournal.com/article/view/7642>
- [8] (2008) Copenhagen Airport Pilots RFID Tags for Passengers. Dostopno na:
<http://www.rfidjournal.com/article/view/4104>
- [9] V. D. Hunt, A. Puglia, M. Puglia, RFID – A Guide to Radio Frequency Identification, New Jersey, Wiley-Interscience, 2007
- [10] IATA, Airport development reference manual 9th edition effective 2004, IATA, 2004
- [11] (2010) XML Schema Definition Tool (Xsd.exe). Dostopno na:
[http://msdn.microsoft.com/en-us/library/x6c1kb0s\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/x6c1kb0s(v=vs.80).aspx)
- [12] (2010) Database. Dostopno na:
<http://www.microsoft.com/express/Database/>
- [13] (2010) ConfigurationManager Class. Dostopno na:
<http://msdn.microsoft.com/en-us/library/system.configuration.configurationmanager.aspx>
- [14] (2010) System.Xml.Serialization Namespace. Dostopno na:
<http://msdn.microsoft.com/en-us/library/system.xml.serialization.aspx>
- [15] (2010) SQL Stored procedure, Dostopno na:
<http://databases.about.com/od/sqlserver/a/storedprocedure.htm>
- [16] (2010) Walkthrough: Arranging Controls on Windows Forms Using a TableLayoutPanel. Dostopno na:
[http://msdn.microsoft.com/en-us/library/w4yc3e8c\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/w4yc3e8c(v=VS.85).aspx)
- [17] (2010) Speedway Revolution Reader. Dostopno na:
<http://www.impinj.com/products/reader-speedway-rev.aspx>
- [18] (2010) Faster RFID Application Development with the Octane SDK. Dostopno na:
<http://learn.impinj.com/search/label/SDK>
- [19] (2010) LLRP Toolkot. Dostopno na:
<http://www.llrp.org/>