

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Anže Molan

Spletna aplikacija za analizo poslovnih procesov
podjetja

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentorica: doc. dr. Mira Trebar

Ljubljana, 2012



Št. naloge: 00250/2012

Datum: 05.04.2012

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

Kandidat: **ANŽE MOLAN**

Naslov: **SPLETNA APLIKACIJA ZA ANALIZO POSLOVNIH PROCESOV
PODJETJA**
**WEB APPLICATION FOR ANALYSIS OF COMPANY BUSINESS
PROCESSES**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj preuči spletne tehnologije, orodja in specifikacije standardov EPCglobal za razvoj aplikacije, ki omogoča pregled poslovanja podjetja na osnovi zbranih podatkov o sledljivosti v preskrbovalni verigi. Podatki o proizvodni, ki se nahajajo v podatkovni bazi EPCIS, naj bodo uporabljeni za pregled poslovnega procesa v posameznih fazah in morebitnih neskladnostih med naročili in poslanimi pošiljkami. Izdelaj naj spletno aplikacijo, ki omogoča pregled poslovanja v ribogojnici in nadzor v procesih pakiranja, transporta, shranjevanja in dostave z dvema različnima pristopoma za pridobivanja podatkov. Preuči naj rezultate testiranja in poskuša ugotoviti prednosti in slabosti predlaganih rešitev.

Mentor:

doc. dr. Mira Trebar

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Anže Molan,

z vpisno številko 63050176,

sem avtor diplomskega dela z naslovom:

Spletna aplikacija za analizo poslovnih procesov podjetja.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire 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 3.7.2012

Podpis avtorja: _____

Zahvala

Zahvaljujem se mentorici doc. dr. Miri Trebar za vodenje, podporo in pomoč pri izdelavi diplomskega dela. Za dodatne nasvete pri analizi podatkov se zahvaljujem tudi Andreju Grahju, sodelavcu v Laboratoriju za računalniško arhitekturo.

Hvaležen sem svojim staršem za brezkompromisno podporo med študijem.

Kazalo

1	Uvod.....	1
2	Sledljivost v preskrbovalni verigi	3
2.1	Kaj je sledljivost in zakaj je potrebna	3
2.2	Postopek sledljivosti.....	4
2.3	Vrste sledljivosti	6
2.3.1	Notranja sledljivost	6
2.3.2	Zunanja sledljivost	7
2.3.3	Mogoči načini sledljivosti.....	7
2.4	Tehnologije označevanja	8
2.4.1	Črtna koda.....	9
2.4.2	Tehnologija RFID	10
2.5	RFID v prehrabni industriji.....	13
3	Pregled tehnologij in orodij za razvoj spletne aplikacije	15
3.1	Tehnologije za dostop do repozitorija EPCIS	15
3.2	Orodja in spletne tehnologije za razvoj aplikacije	18
3.3	Uporabljene metodologije in pogosto uporabljene funkcije	22
4	Sistem sledljivosti za razvoj spletne aplikacije	25
5	Aplikacija za pregled poslovanja podjetja	29
5.1	Analiza zahtev in podatkov	29
5.2	Načrtovanje aplikacije	30
5.3	Izvedba aplikacije.....	32
5.3.1	Neposredno dostopanje do podatkov	32
5.3.2	Dostop do podatkov z uporabo Fosstraka.....	32
5.4	Moduli aplikacije.....	33
5.4.1	Modul Nastavitve.....	33
5.4.2	Modul Stranke.....	34
5.4.3	Modul Prodaja.....	37
5.4.4	Modul Ribogojnica	39

5.4.5	Modul Sledenje.....	41
5.5	Testiranje aplikacije in analiza rezultatov	43
6	Sklepne ugotovitve	47

Seznam uporabljenih kratic

AJAX (ang. Asynchronus JavaScript and XML) – tehnologija za asinhrono zahteve

CSS (ang. Cascading Style Sheets) – slogovna predloga za spletne strani

DAL (ang. Data Access Layer) – program za preprostejše dostopanje do podatkovne baze

DQPB (neposredno poizvedovanje podatkovne baze)

EPC (ang. Electronic Product Code) – elektronska koda izdelka

EPCIS (ang. Electronic Product Code Information Services) – EPC-informacijske storitve

FQPB (poizvedovanje podatkovne baze Fosstrak)

GIAI (ang. Global Individual Asset Identifier) – globalni identifikacijski ključ

GLN (ang. Global Location Numbers) – globalni identifikator lokacij

GRAI (ang. Global Returnable Asset Identifier) – globalni identifikator povratne embalaže

GTIN (ang. Global Trade Item Number) – globalni identifikator izdelka

MVC (Model – View – Controller)

PHP (ang. Hypertext Preprocessor) – programski jezik za spletne rešitve

QR Code (ang. Quick Response Code) – hitro odzivna koda

SGTIN (ang. Serialized Global Trade Item Number) – globalna serijska številka izdelka

SOAP (ang. Simple Object Access Protocol) – protokol za izmenjavo strukturiranih podatkov na spletu

SSCC (ang. Serial Shipping Container Code) – serijska številka zaboja

UPC (ang. Universal Product Code) – univerzalna številka izdelka

XHTML (ang. eXtensible HyperText Markup Language) – označevalni jezik za izdelavo spletnih strani

XML (ang. Extensible Markup Language) – razširljiv označevalni jezik

Povzetek

V zadnjem času uporablja splet čedalje več uporabnikov, zato se čedalje bolj uveljavlja razvoj spletnih aplikacij. Uporabniki želijo imeti dostop do obdelave svojih podatkov z vsakega računalnika, tudi prenosnega, povsod, kjer je mogoča povezava z internetom. Eden glavnih razlogov za to je, da lahko svoje delo opravljajo kadarkoli in kjerkoli, v pisarni ali na sestankih imajo dostop do podatkov. Ti so v aplikaciji vedno na voljo za prebiranje in urejanje na poljubni lokaciji. Zaradi omenjenih zahtev smo se odločili izdelati spletno aplikacijo za pregled poslovanja podjetja, ki ima na voljo vse podatke o svojih procesih na lokalnih računalnikih oziroma na strežnikih.

Za diplomsko nalogo smo izdelali spletno aplikacijo, ki omogoča uporabniku vpogled v poslovanje podjetja, ki vključuje poslovne procese v preskrbovalni verigi na podlagi zbranih podatkov v sistemu sledljivosti. Predstavljeni so standardi GS1, predvsem smo se podrobneje posvetili globalnemu standardu EPC in z njim povezanimi tehnologijami in orodji. V ozadju imamo podatkovno bazo EPCIS, v katero se beležijo dogodki pri sledljivosti. Aplikacija je bila izdelana s programskim jezikom PHP in podatkovno bazo MySQL. Postopek izdelave in izbire tehnologij je podrobno opisan v nalogi. Predstavljeni rezultati spletne aplikacije omogočajo pregled in analizo podatkov, ki so bili zbrani v pilotni izpeljavi evropskega projekta RFID from Farm to Fork.

Ključne besede:

Spletna aplikacija, sledljivost, poslovni procesi, preskrbovalna veriga, EPCIS

Abstract

Recent years the Web is gaining more attention in the field of applications. Users want to access and process their data from any computer with an Internet connection. One major reason for this is working and accessing data anytime and anywhere. Data is always available for reading and editing at any location. Because of these requirements, we decided to implement a Web application for review of business operations in a company that has all the data stored on local computer or servers.

The aim of this thesis was to implement a Web application that allows users access to company data, including business processes on data collected in the track and trace system. In thesis we acquainted with GS1 standards, especially the more detailed focus on EPCglobal standards and related technologies and tools. In the background we have EPCIS database where all events are being stored. The application was developed using the programming language PHP with MySQL database. Process of designing and the choice of technologies is described in detail. Presented results of a web application provides a review and analysis of data collected in the pilot implementation of the European project »RFID from Farm to Fork«.

Key words:

Web application, traceability, bussiness processes, supply chain, EPCIS

1 Uvod

Globalizacija in odprtost trgovanja sta omogočili prenos različnih dobrin iz oddaljenih dežel. Med transportom se izdelek prenaša po številnih posrednikih, v tem času se lahko tudi pokvari ali pa z njim ne ravnajo po pravilih, kar še posebno velja za živila. Zaradi takšnih težav je nastala potreba po strožjem nadzoru in sledenju živil. Že v preteklosti so uporabljali različne načine beleženja porekla, sestavin in poti, ki ga je izdelek opravil. Od klasičnega beleženja v zvezek se od razvoja računalnikov podjetja odločajo za nadgradnjo sistema, ki je podprt računalniško. Najbolj osnovni sistemi so namizne aplikacije, ki jih podpirajo različne identifikacijske tehnologije. Najbolj vsakdanja je črna koda, vendar pa potreba po preprostejšem beleženju izdelkov pospešuje razvoj v smeri mobilnih in spletnih aplikacij.

Veliko podjetij stremi k prehodu z namiznih v spletne aplikacije. Prednosti sta lažja dostopnost in varnost podatkov. Poudarek je na ažurnosti podatkov in praktičnosti uporabe. Diplomaska naloga prikazuje razvoj aplikacije, ki omogoča pregled podatkov, zajetih v različnih fazah prehranske verige.

V sklopu pilotnega projekta Laboratorija za računalniško arhitekturo na Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko, z naslovom RFID from Farm to Fork smo izdelali aplikacijo, ki omogoča pregled podatkov, zajetih v posameznih fazah prehranske verige. Aplikacija deluje na dva različna načina pridobivanja podatkov: neposredno poizvedovanje v EPCIS-podatkovno bazo in poizvedovanje z vmesnikom Fosstrak Query.

Predstavljeni so pojem sledljivosti in najpogosteje uporabljene metode sledenja. Prikazane so tudi mogoče tehnologije, ki se uporabljajo za identifikacijo artiklov ali izdelkov. Opisana sta načrtovanje in razvoj aplikacije, prikazane pa tudi potrebne spletne tehnologije in orodja, ki nam omogočajo dostop do EPCIS-podatkov. Moj glavni prispevek pri diplomski nalogi je bil razvoj spletne aplikacije, ki omogoča kombiniran dostop do podatkov in razumevanje ter uporabljanje tehnologij za sledenje.

2 Sledljivost v preskrbovalni verigi

2.1 Kaj je sledljivost in zakaj je potrebna

Sledljivost je zmožnost izsleditve in prepoznave poljubnih predmetov, izdelkov. Njen namen je ugotoviti pretekle ali hipne lokacije ter poznati njihovo zgodovino. V nadaljevanju je opis sledljivosti povzet po dokumentih, ki obravnavajo omenjeno problematiko v preskrbovalni verigi [11,15].

V zakonodaji Evropske unije, ki velja tudi v Republiki Sloveniji, je sledljivost obravnavana kot možnost sledenja živilom, krmi, živalim in izdelkom ali snovem, ki bodo uporabljene za hrano. Če imamo vpeljano sledljivost, se lahko spopadamo z nekaterimi nevarnostmi, ki lahko nastanejo v zvezi s hrano ali krmo, in tako evropskim državljanom omogočimo uživanje varnih in zdravih živil. Kadar pristojni organi ugotovijo oporečnost, morajo imeti možnost vpogleda v izvor živila, da se lahko takšno živilo čim prej izloči in prepreči stik s kupcem. S sledljivostjo lahko natančno ugotovimo tudi mesto in s tem v zvezi tudi vzrok pokvarjenosti živila ter o tem obvestimo javnost.

Sledljivost se je že pokazala kot ena od uspešnih metod zajemanja podatkov v prehrabni verigi, ko je bilo na primer treba hitro odkriti in izločiti nevarna živila, ki so bila onesnažena z bovino spongiformno encefalopatijo (BSE) oziroma boleznijo norih krav [11].

Sledljivost je precej več kot le nadzor onesnaženih izdelkov in njihov odpoklic. Sistemi sledljivosti lahko med drugim omogočajo tudi preverjanje prisotnosti ali odsotnosti določenih značilnosti, pomembnih za kupce. Sledljivost omogoča nadzor nad organskim in ekološkim načinom pridelave, uporabo alergenov v kozmetični industriji ali dietičnih izdelkov in izvor hrane in še marsikaj drugega. V številnih državah po svetu imajo zakonska določila o njeni uporabi, predvsem na področju pokvarljivih živil [15].

S sledljivostjo tako obvladamo:

- varnost izdelkov,
- primernost izdelkov za uporabo,
- trajnost,
- etični odnos do kupca.

Cilj sledljivosti je prepoznavanje serije, skupine, lota izdelkov ali surovin med proizvodnjo in celotno distribucijo. Je najbolj razširjena v zvezi z neustreznimi ali

pokvarjenimi izdelki, kadar jih je treba kar najhitreje odstraniti s prodajnih polic.

Razlogi za vzpostavitev sistema sledljivosti so:

- omejitev odgovornosti,
- čim manjša količina izdelkov za umik,
- čim manjša škoda, ki nastane pri odpoklicu izdelkov,
- čim manjša obdobja blokade blaga ali podjetja,
- optimizacija proizvodnih postopkov,
- prvi pogoj za posebne marketinške trditve (biološka pridelava, zaščita živali),
- prvi pogoj za posebne standarde kakovosti.

Subjekti pri sledljivosti so:

- prevoznik/ponudnik logističnih storitev,
- predelovalec/proizvajalec,
- prodaja na drobno ali upravljavec storitev,
- skladišče,
- oblast.

Sledljivi predmeti so:

- pošiljka (ena ali več logističnih enot, tovor na ladji, tovarnjaku),
- logistična enota (en ali več izdelkov, paleta, zabojnik),
- izdelek za prodajo na debelo (karton, vrečke),
- izdelke za prodajo na drobno.

2.2 Postopek sledljivosti

V svetu delujejo za določanje postopkov v sistemih sledljivosti organizacije, ki skrbijo za uveljavljanje standardov in se pojavljajo pod skupnim imenom GS1: »GS1 je nevtralna in

nepridobitna organizacija, katere cilj je načrtovati in uveljavljati globalne standarde, tehnologije in rešitve za izboljšanje učinkovitosti preskrbovalnih in oskrbnih verig z dodajanjem uporabnih informacij katerikoli izmenjavi dobrin in storitev« [15]. Organizacija je nastala z združitvijo EAN International in Uniform Code Council (UCC) in deluje na podlagi več kot 30 let izkušenj v več kot 145 državah sveta. Standard GS1 ima pet faz v postopku sledljivosti.

Načrt in organizacija

V prvi fazi sledljivosti je treba določiti načine izpeljave sledljivosti ter izbor, deljenje in shranjevanje podatkov o njej. Določiti je treba tudi način upravljanja povezav med vnosi, notranjimi postopki in rezultati. Načrt in organizacijo sledljivosti pripravijo partnerji pri sledljivosti.

Uskladitev matičnih podatkov

Trgovski partnerji morajo biti globalno in unikatno identificirani. GS1 uporablja identifikator GLN. V naslednjem koraku je treba unikatno identificirati vse fizične lokacije in sredstva, pri katerih moramo poznati odpošiljatelja in prejemnika. Pri tem uporabimo identifikator GIAI, za povratna sredstva pa GRAI. Lastniki blagovnih znamk so zavezani edinstveno identificirati vse izdelke, ki sodelujejo pri procesu sledljivosti. Uskladitev matičnih podatkov je končana, kadar si vsi partnerji izmenjajo in uskladijo podatke.

Zapis podatkov o sledljivosti

V tej fazi je treba definirati načine za določanje, uporabljanje in zajemanje identifikacije s sledljivih predmetov in kako jih zbirati, deliti ter shranjevati med fizičnim prenosom. Sledljivi predmet dobi enolično identifikacijo najpozneje ob ustvarjenju. Identifikator je GTIN ali GTIN + številka serije ali serijska številka (SGTIN) pri prodajnih enotah in SSCC pri logističnih enotah.

Vsi sledljivi predmeti morajo biti označeni z globalno edinstveno identifikacijo, ki mora biti na sledljivem predmetu, sredstvu sledljivega predmeta ali na priloženem dokumentu. Identifikacija mora biti priložena predmetu, tudi če je pakiran v večje embalaže, in ostati na njem do zaužitja, prodaje ali uničenja. Prejemnik sledljivega predmeta podatke zajema in jih posreduje ali shrani. Zapis podatkov se konča pri dostavi predmeta končnemu prodajalcu, ki ima neposreden stik s kupcem, ali pri izgubi ali uničenju predmeta.

Zahtevek za sledenje

Zahtevek za sledenje lahko sproži vsak partner v verigi. Partner lahko izda zahtevek zaradi prošnje oblasti ali kupca in začne postopek za umik izdelka. Glavno vlogo ima tukaj subjekt,

ki je podal zahtevek za sledenje. Pri zahtevku za sledljivost je treba priložiti najmanj eno informacijo za pomoč pri iskanju želenih rezultatov:

- identifikacija izdelka,
- identifikacija partnerjev,
- identifikacija lokacije,
- datumski okvir,
- identifikacija dogodka ali postopka.

Subjekt, ki je izdal zahtevek za sledenje, pridobi informacijo o izdelku ali sporočilo, da informacije ni mogoče pridobiti. S tem je zahtevek za sledenje končan.

Uporaba informacij

Rezultat te faze je uporaba informacij, na podlagi katerih je mogoče ukrepati skladno z zakonodajo in poslovnimi zahtevami.

2.3 Vrste sledljivosti

Poznamo predvsem dva načina sledljivosti. Prvi, imenovan notranji, se uporablja za sledenje različnih vrst izdelkov in polizdelkov v nekem podjetju. Drugi, tako imenovan zunanji, pa za sledenje izdelkov na poti med poslovnimi partnerji, vse do porabnika oziroma prodajalca [11].

2.3.1 Notranja sledljivost

Pri različnih dejavnostih proizvodnje je sledljivost organizirana različno. Skupine izdelkov in izdelki se delijo in različno označujejo. Različni tokovi različnih izdelkov potujejo drugače. Te premike sledimo z ukrepi notranje sledljivosti, ki so navadno prostovoljni. Ukrepi so namenjeni zagotavljanju dovolj visoke kakovosti, preglednosti, notranji logistiki in upravljanju skladišča.

Notranji postopek je skupek več povezanih faz v podjetju za potrebe predelave in priprave izdelka. Takšen postopek z vidika sledljivosti ne potrebuje sodelovanja z drugimi zunanjimi partnerji. Vsako podjetje ali partner je odgovoren za shranjevanje podatkov, ki vsebujejo vse faze predelave.

Da imamo lahko neki postopek za notranjega, mora vsebovati najmanj eno od naslednjih faz: premik, predelava, hramba in uničenje.

2.3.2 Zunanja sledljivost

Zunanja sledljivost je uporabna pri predaji sledljivega izdelka med dvema partnerjema. Pri tem lahko beležimo dva načina: sledljivost nazaj in sledljivost naprej. Prvi nam pove, kateri dobavitelj nam je dobavil izdelek, drugi pa, katero podjetje, ki je oddalo izdelek, ve, kateremu partnerju ga je predalo.

Vsak izdelek mora imeti identifikacijsko oznako, ki mu jo dodeli vir v fazi nastajanja. Oznaka mora biti edinstvena. Navadno je to globalna trgovinska identifikacijska številka (GTIN, ang. Global Trade Item Number) ali zaporedna koda zabojnika (SSCC, ang. Serial Shipping Container Code). Ta oznaka se potem beleži med celotno preskrbovalno verigo.

Sledljivost ne pomeni, da mora vsak partner v verigi imeti vse podatke, saj je dovolj natančno, da imata sosednja partnerja zabeleženo skupno točko. Takšen način omogoča sledljivost izdelka naprej in nazaj.

Za zagotovitev najbolj pregledne sledljivosti morajo vsi udeleženci v preskrbovalni verigi doseči notranjo in zunanjo sledljivost. Tako lahko izdelku sledimo med prevozom med partnerji in predelavo. Rešitve za sistem sledljivosti se razlikujejo po vrsti podjetja, velikosti podjetja in vrsti izdelka.

2.3.3 Mogoči načini sledljivosti

Sledljivost ene stopnje pomeni, da imamo le podatke za eno stopnjo naprej in nazaj. Takšen način nam omogoča pregled po stopnjah. Če se veriga razteza čez več podjetij, je potreben večji napor za vpogled v vse stopnje dobavne verige. Takšen sistem zahteva uporabo enoličnega identifikatorja, na primer črtne kode, in informacijskega sistema za hranjenje podatkov.

Z nadgradnjo informacijskega sistema je mogoče doseči sledljivost različnih stopenj (Slika 1). Takšen način omogoča vpogled v vse stopnje dobavne verige. V skupni ali

porazdeljeni podatkovni bazi hranimo podatke o vseh stopnjah verige. Vsi vključeni partnerji uporabljajo enak informacijski sistem oziroma enak standard sledljivosti. S tem ukrepom si zagotovimo zbirko podatkov, ki vsebuje celotno pot izdelka. Na tem sistemu je mogoče preprostejše in hitrejšo poizvedovanje [11].



Slika 1: Sledljivost različnih stopenj [11].

2.4 Tehnologije označevanja

Za sledenje poljubne enote (izdelek, zaboj ...) v preskrbovalni verigi je treba določiti njen enolični identifikator. To je zapis, ki nam omogoča unikatno prepoznavnost vsakega izdelka in predstavlja povezavo fizičnega sveta z informacijskim sistemom. Identifikator se prebere pri vsakem prehodu ali spremembi opazovanega izdelka, kar nam omogoča, da informacijski sistem zabeleži potrebne podatke.

Za identifikatorje poznamo več standardov. Najbolj uveljavljen je standard GTIN (Global Trade Item Number), ki ga je oblikovala organizacija GS1 [15]. GTIN vsebuje tudi EAN-številko, ki je unikatna evropska številka izdelka, zdaj mednarodno. Vsebuje tudi standard UPC (Universal Product Code), ki se uporablja za črtne kode. Če GTIN razširimo v SGTIN (Serial Global Trade Item Number), ga lahko uporabljamo v standardu EPC (Electronic Product Code). GTIN se uporablja za označevanje vrste izdelka, SGTIN v EPC pa za označevanje posameznega izdelka.

Za označevanje sledljivih enot so v preteklosti najpogosteje uporabljali linearno črtno kodo. V zadnjem času se čedalje bolj uveljavljajo dvodimenzionalne oblike črtnih kod, kot je koda QR (Quick Response). Z razvojem tehnologije RFID se čedalje bolj uveljavljajo značke ali nalepke RFID, ki so resni tekmeč sistemom označevanja [4].

2.4.1 Črtna koda

Črtna koda (Slika 2) je optična predstavitev podatka, ki določa kodo izdelka. Predstavljajo jo črte različnih debelin in razmikov med njimi. Črtni kode so prvič uporabili za prepoznavanje vagonov, ampak niso bile uspešne. To je veljalo, dokler jih niso začeli uporabljati za avtomatski zajem podatkov o izdelkih (cena) in za beleženje zalog. Zato so postale nepogrešljive in so se uveljavile za označevanje zelo različnih enot, od izdelkov, palet in drugega. Njena prednost pred RFID sta zagotovo cena in zmožnost hitrega razširjanja, ne zagotavlja pa enoličnega označevanja posameznega izdelka.



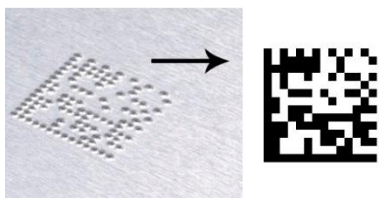
Slika 2: Primer črtni kode UPC A.

Črtna koda je pogosto natisnjena na nalepko, ki se namesti na produkt, ali pa je vtisnjena že na embalažo. Z optičnim čitalnikom kodo preberemo in zapisani podatek pošljemo računalniku. Implementacija čitalnika je zelo preprosta, saj deluje po enakem načelu kot tipkovnica, ki avtomatsko prenese prebrani niz iz čitalnika v računalnik. Pravzaprav je branje kode enakovredno vnosu podatka z uporabo tipkovnice.

Matrična koda

Matrična koda, imenovana tudi 2D-črtna koda, je sestavljena iz črnih in belih kvadratkov, ki predstavljajo optično berljiv zapis. Matrične kode lahko vsebujejo veliko več podatkov kot navadne linearne črtni kode. Koda Data Matrix (Slika 3) lahko shrani do 2335 alfa-numeričnih znakov. Prednost matričnih kod je tudi v popravljanju napak. Tudi če je vzorec delno poškodovan, ga je še vedno mogoče prebrati. Odpornost proti napakam pa je odvisna od uporabljenega standarda.

Matrične kode vsebujejo tudi tako imenovani iskalni vzorec, s katerim čitalnik najde in izloči zapis iz okolice. Vzorec je viden na kodi, kot velika črka L, ki obdaja kodo z leve in spodnje strani.



Slika 3: Primer označevanja izdelka v kodi Data Matrix.

V zadnjem času je postala zelo popularna koda QR (ang. Quick Response), (Slika 4). Najprej so jo uporabljali v avtomobilski industriji, zdaj jo lahko opazimo na številnih plaktih, knjigah in številnih drugih mestih. Uporablja se predvsem zaradi svoje hitre berljivosti in velike zmogljivosti, v primerjavi s črtnimi kodami UPC (ang. Universal Product Code). Razširjenost pametnih telefonov je QR-kodo približala uporabnikom za vsakdanjo uporabo. QR-koda ima več funkcij. Lahko komunicira z aplikacijami na telefonu, prikaže lokacijo na zemljevidu, odpre določeno spletno stran ali pokliče telefonsko številko.



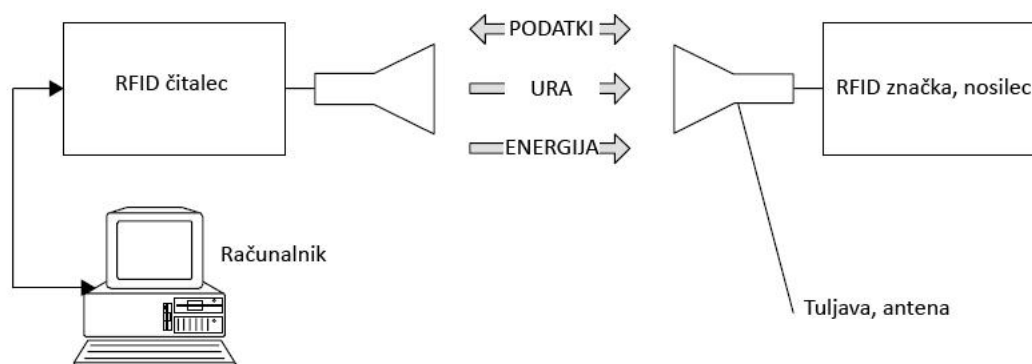
Slika 4: Primer kode QR, ki na vašem telefonu odpre spletni naslov.

2.4.2 Tehnologija RFID

Radiofrekvenčna identifikacija (RFID) pomeni uporabljanje radiofrekvenčnih elektromagnetnih polj za brezžični in brezkontaktni prenos podatkov, ki so zapisani na znački RFID za potrebe avtomatske identifikacije izdelka [1,2]. Komponente sistema RFID (Slika 5) so:

- značka RFID – nosilec informacije (enolični identifikator),
- čitalnik RFID – skrbi za prenos podatkov med značko RFID in računalnikom,

- računalnik – zagotavlja zbiranje in shranjevanje podatkov.

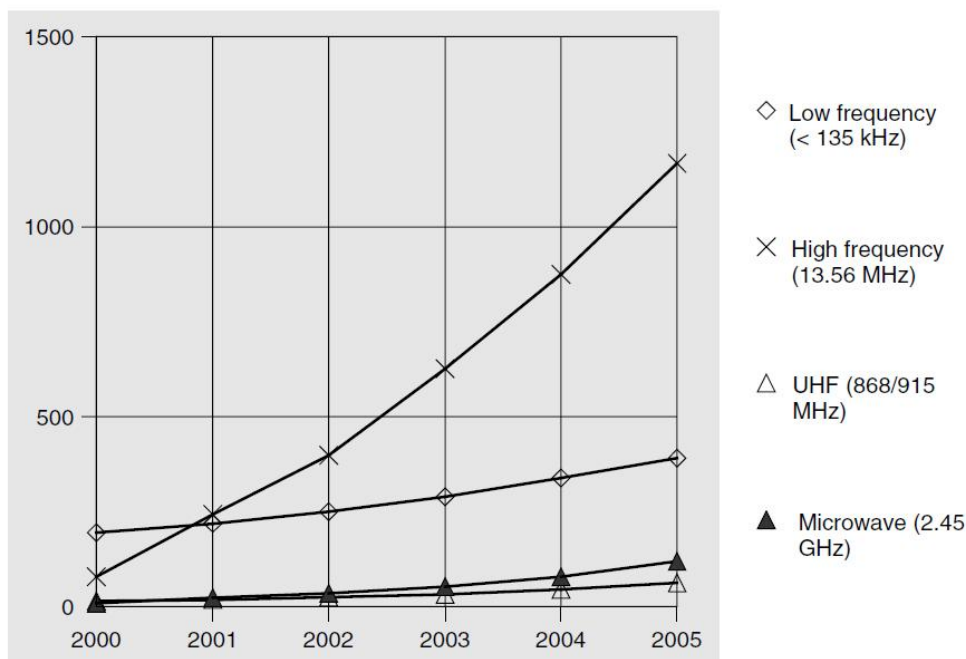


Slika 5: Sistem RFID.

Začetki razvoja tehnologije RFID segajo v obdobje druge svetovne vojne. Uporabili so jo za identifikacijo sovražnih in prijateljskih letal. Prvi predhodnik današnjega modernega RFID je razvil Mario Cardullo leta 1973. Njegova naprava je bila pasivna, to pomeni, da je energijo za napajanje prejela iz čitalnika po inducirani napetosti v anteni oziroma tuljavi značke.

Zdaj poznamo aktivne (ang. active) in pasivne (ang. passive) značke RFID. Aktivne imajo lasten vir napajanja – baterijo, zato ne potrebujejo energije čitalnika za prenos podatkov. Njihove prednosti so v večji zanesljivosti in daljšem doletu. Prav tako lahko vsebujejo tudi druge komponente, na primer senzorje in spominska vezja. Pasivne značke so preprostejše in cenejše za izdelavo. Obstajajo tudi polaktivne (ang. semiactive) ali polpasivne (ang. semipassive) značke, ki imajo tudi lastno napajanje, ampak le za delovanje aktivnih komponent, kot je zapis senzorskih podatkov v spomin. Za oddajanje signala uporabljajo inducirano napetost, tako kot pasivne značke [3].

Značke in čitalniki RFID delujejo v različnih frekvenčnih območjih. Vsako ima svoje prednosti in slabosti [3]. Čedalje bolj pa uporabljajo takšne, ki delujejo v visokofrekvenčnem območju (Slika 6).



Slika 6: Porazdeljenost uporabe različnih frekvenc [3].

Frekvenčna območja, ki so na voljo, so:

- **LF (ang. Low Frequency):** Značke nizkega frekvenčnega območja, od 120 do 150 kHz, se navadno uporabljajo za identifikacijo živali, omejevanje dostopa in razne ključavnice. Doseg takšnih naprav je razmeroma kratek, le do polovice metra. Hitrost branja je manjša kot pri napravah z višjimi frekvencami.
- **HF (ang. High Frequency):** Značke delujejo v območju 13,56 MHz, z dometom do enega metra, in se uporabljajo za omejevanja dostopov, za sledenje različnih izdelkov, na primer v knjižnicah in farmaciji. Pametne kartice delujejo na tem frekvenčnem območju. Cena takšnih naprav je nižja od nižjefrekvenčnih, čeprav imajo večjo hitrost branja.
- **UHF (ang. Ultra High Frequency):** Značke, ki delujejo v območju od 868 do 915 MHz, imajo območje dosega do približno treh metrov ali več, odvisno od antene, ki jo uporablja čitalnik. Hitrost branja je enaka ali večja od HF-značk. Takšne naprave se uporabljajo v elektronskem cestninjenju, v prehranjevalni verigi ter distribucijski in logistični industriji. UHF je temelj standarda EPC.

Optični RFID, znan tudi s kratico OPID, temelji na optičnih čitalnikih. Njegova glavna prednost pred tradicionalnim RFID je nizka cena. Optični RFID deluje na elektromagnetnem spektru, med frekvencama 333–750 THz. Identifikator se prebere na podlagi odboja. Optični

RFID omogoča večjo zaščito podatkov, saj je za branje podatkov potrebna vidna povezava med čitalnikom in značko [12].

2.5 RFID v prehrambni industriji

Napredek na področju RFID omogoča zelo veliko možnosti uporabljanja tehnologije na številnih področjih, še posebno v različnih panogah živilske industrije. Njena uporaba se uveljavlja tudi na področju logistike in pri označevanju prehrambnih izdelkov [5].

Pridelava, predelava in proizvodnja hrane se je močno spremenila z uporabo sodobnih tehnologij in orodij, ki omogočajo boljši in temeljitejši pregled od diagnostike, analize in učinkovitejše načrtovanje poslovnega procesa.

Sistemi RFID se pojavljajo kot zelo učinkovita oprema pri identifikaciji, sledenju in spremljanju živali na kmetijah. Od prvih patentov iz leta 1981 se je uporabljanje elektronskega označevanja živine povečevalo in je že tako rekoč vsakdanja praksa v večini industrijske pridelave hrane. Lahko bi rekli, da obstaja sistem sledljivosti na začetku preskrbovalne verige. V kombinaciji z informacijskim sistemom sistem označevanja živali RFID omogoča podroben pregled zdravljenja in rasti živali, premike, nakupe in prodajo.

Čitalnik lahko uporabljamo v kombinaciji s prenosnim ali lokalnim računalnikom, s katerim beležimo in hranimo podatke. Podatki so lahko tako s prilagojenim informacijskim sistemom dostopni proizvajalcu, preostalim partnerjem in tudi veterinarski službi.

Vsak dan je proizvedenih, prepeljanih ali shranjenih na milijone ton temperaturno občutljive hrane. Za vse te produkte je nadzor nad temperaturo zelo pomemben. Takšne aplikacije uporabljajo RFID-značke z vgrajenim temperaturnim senzorjem. RFID-sistemi lahko tudi pospešijo postopke sprejetja in izdaje v logističnih procesih. Beleženje dogodkov je avtomatizirano in hitrejše. S tem lahko v verigi ugotovljamo nepooblaščen dostop in zagotovimo preglednost prevoza.

Prednosti RFID-tehnologije so vidne predvsem ob veliko označenih enotah, ki jih je treba pregledati pri sprejetju ali izdaji, pri velikih količinah podatkov in pri zahtevah po prilagodljivih bralnih razdaljah.

Velike količine podatkov, ki bi jih morali prebrati in zabeležiti z uporabo konvencionalnih orodij, kjer še vedno najdemo papirnate dokumente, bi se lahko izgubile oziroma bi bil njihov pretok prevelik za obdelavo, ali pa se vnašajo napake pri prepisovanju podatkov v računalnik. Z uporabo RFID lahko za to poskrbi računalniški sistem [5].

Bralna razdalja je lahko ena od prednosti ali pa ena od težav, predvsem v povezavi z nekaterimi uporabljenimi materiali pakiranja (kovine, steklo) in značilnostmi predmetov

(vlaga). Določene tehnologije RFID značk omogočajo prebiranje podatkov z večjih razdalj, pri čemer čitalniki črtne kode odpovedo, ker mora biti vidna povezava med značko in čitalnikom.

Glavne slabosti uporabe RFID so v ceni, varnosti podatkov in reciklaži. Cena RFID značk je še vedno visoka, približno 20–30 centov za manjše količine (nekaj deset tisoč značk RFID), pri čemer je lahko cena precej nižja, tudi okrog 5 centov (nabavljena količina značk RFID je okrog 1.000.000). To je zelo veliko za izdelek, ki stane 50 centov, na primer sadje in zelenjava. Primer je črna koda, ki stane manj kot cent in je v takšnih primerih primernejša.

Pri hrambi in izmenjavi informacij s partnerji je treba biti pazljiv in vnaprej določiti njihova pravila.

RFID-značke so lahko nameščene na embalažo, redkeje v izdelek, pri tem pa pogosto niso primerno odvržene ali reciklirane. Se pa čedalje pogosteje uporabljajo v povratni embalaži (palete, plastični zaboji, kontejnerji). S čedalje pogostejšo uporabo RFID-tehnologije postaja možnost vpliva na okolje čedalje bolj verjetna. Prav tako lahko vpliva na reciklažo papirja, stekla, plastike in kovin.

Pri uporabljanju sodobnih tehnologij je treba v podjetju skrbeti za ustrezno usposobljeno osebje, ki je pripravljeno sprejemati izzive pri uporabljanju najnovejših tehnologij. Veliko podjetij nima primernih možnosti ali sredstev, zato je preskok na uporabo tehnologije RFID težji. Vpeljava uporabe RFID v informacijski sistem potrebuje podporo inženirjev računalništva in tudi strokovnjakov s področja agrikulture. Predstavljena rešitev pa je ena od možnosti, da je, kadar podjetje še vedno uporablja papirne dokumente v svojih procesih, izbira nove tehnologije preprostejša ali učinkovitejša rešitev od kakšne druge.

3 Pregled tehnologij in orodij za razvoj spletne aplikacije

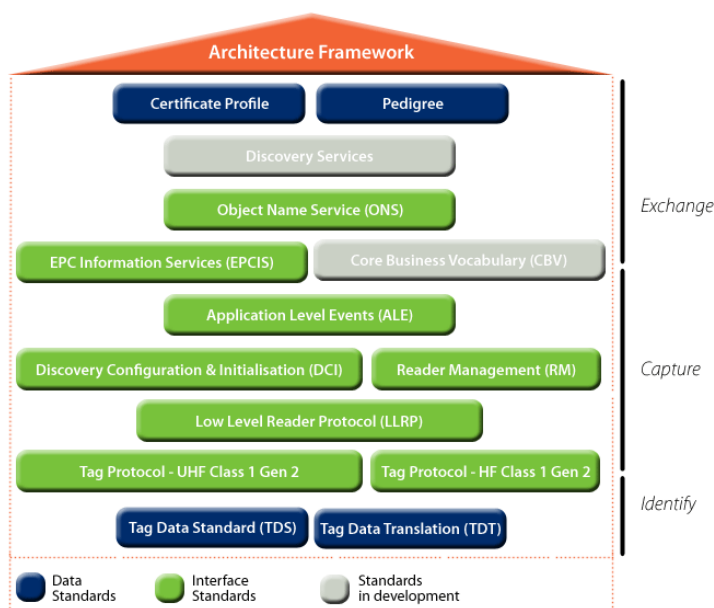
Razvoj spletne aplikacije za pregled poslovanja podjetja na podlagi podatkov, ki so bili zbrani v sistemu sledljivosti, vključuje uporabo standardov GS1 na področju omrežja EPCglobal, programska orodja in strežnik Glassfish za dostop do repozitorija EPCIS (EPC Information Services). Pri izdelavi aplikacije smo se odločili za odprtokodno rešitev, v kombinaciji strežnika MySQL in programskega jezika PHP.

3.1 Tehnologije za dostop do repozitorija EPCIS

Omrežje EPCglobal je uporabljeno za vizualizacijo informacij o stanju in premikih izdelkov in sredstev v procesu poslovanja podjetja v preskrbovalni verigi med poslovnimi partnerji. Zaradi potrebe po izmenjavi podatkov je treba upoštevati standarde, ki zagotavljajo učinkovito zbiranje in izmenjavo podatkov [15].

Standardi GS1 EPCglobal

Vse pomembne standarde RFID (Slika 7) nadzorujejo različne organizacije, med katerimi sta najvplivnejši ISO (ang. International Organization for Standardization) in EPCglobal.



Slika 7: Standard EPCglobal.

EPCglobal je vodilna pri razvoju industrijskih standardov za sistem označevanja z uporabo EPC (ang. Electronic Product Code) v povezavi s tehnologijo RFID za avtomatično identifikacijo in izboljšanje upravljanja preskrbne verige. Standardi določajo, kaj in kako se zapisuje v značko RFID, kakšna je komunikacija s čitalniki in kako poteka izmenjava podatkov med različnimi poslovnimi partnerji.

Koda EPC je standardizirana enolična oznaka in se uporablja v sistemih RFID v povezavi z EPCIS za izmenjavo podatkov. EPC-koda je lahko zapisana v binarni obliki za branje in zapisovanje v značke, prav tako jo lahko zapišemo tudi v berljivi obliki Internet Uniform Resource Identifier (URI), ki se uporablja za izmenjavo podatkov in v podatkovni bazi. V EPC-formatu imamo s standardi GS1 EPCglobal opredeljene vse GS1-identifikatorje, katerih standardizacija je opisana v GS1 EPC Tag Data Standardu.

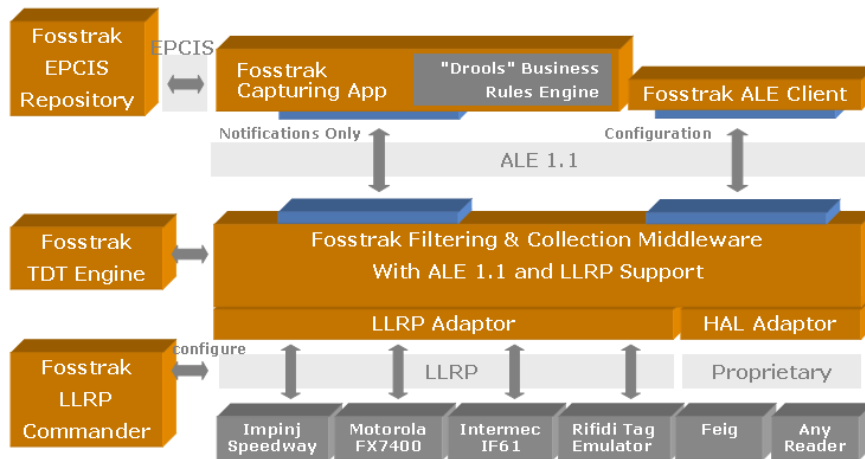
EPC Information Services (EPCIS)

EPCIS ima poseben pristop za izmenjavo podatkov. Razdeljen je na podatkovne standarde, ki definirajo predstavitev, obliko zapisa in definicije skupnih podatkov, ter na standarde vmesnikov. Ti zagotavljajo vnaprej definiran niz operacij za zajem (ang. capture) in shranjevanje podatkov v repozitorij EPCIS ter nato za pridobivanje podatkov (ang. query) iz repozitorija EPCIS. Deluje s katerimi koli podatki in opredeljuje uporabo zapisovanja podatkov v obliki dogodkov, ki označujejo podane štiri parametre:

- kaj (EPC-številka, LOT, datum poteka ...),
- kje (lokacija),
- kdaj (čas dogodka, čas zapisa),
- zakaj (pošiljanje, sprejetje ...).

Fosstrak

Fosstrak je odprtokodno orodje, ki deluje na RFID-programski platformi [7]. Namenjeno je razvijalcem aplikacij za shranjevanje in dostopanje do podatkov o sledljivosti, ki so v repozitoriju Fosstrak EPCIS. Na sliki 8 vidimo pregled vmesnika ALE Middleware, ki je sestavljen iz filtrirnega in zbirnega strežnika, samostojnih odjemalcev in spletnih odjemalcev. ALE za komunikacijo z RFID-čitalniki uporablja protokol LRP (ang. Low Level Reader Protocol).

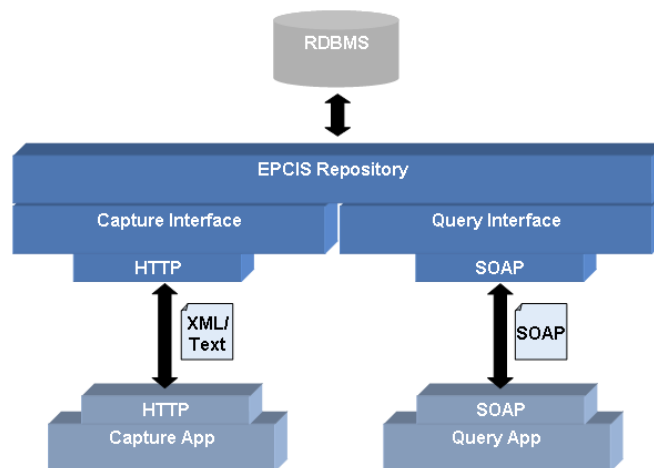


Slika 8: Arhitektura Fosstrak.

Modul Fosstrak EPCIS (Slika 9) je popolna implementacija EPCIS-standarda. Sestavljen je iz treh modulov:

- EPCIS repository,
- interaktivna aplikacija EPCIS Capture,
- interaktivna aplikacija EPCIS Query.

Vsebuje tudi tako imenovan Webadapter za povezovanje prek spletnih protokolov. V našem primeru smo uporabili REST-protokol za poizvedovanje po zabeleženih dogodkih.



Slika 9: Modul Fosstrak EPCIS.

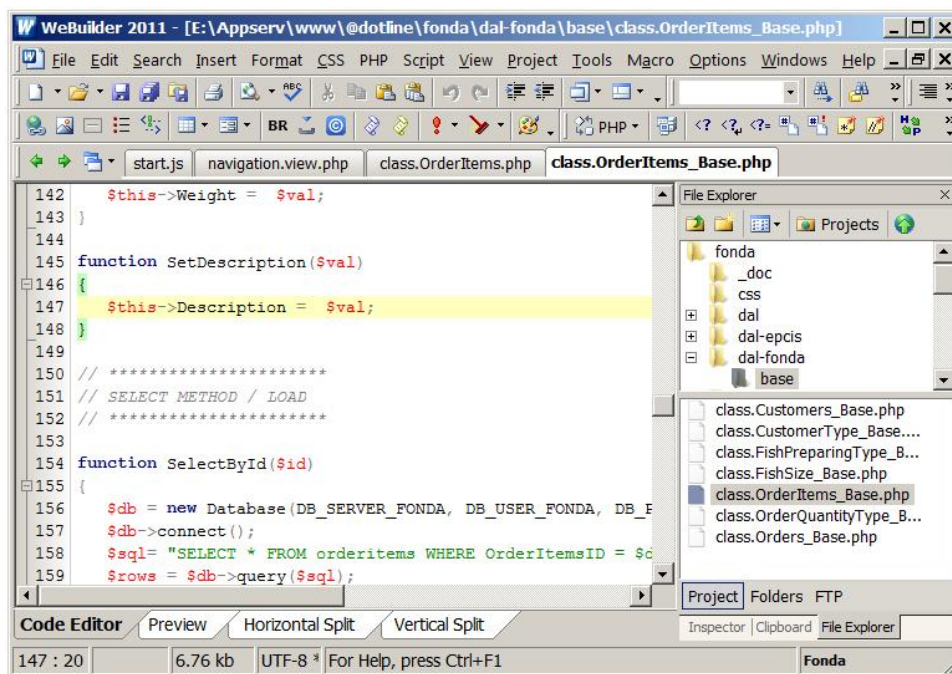
Strežnik Oracle GlassFish

Aplikacijski strežnik GlassFish nam omogoča delovanje repozitorija EPCIS in EPCIS Query klienta [8]. GlassFish je brezplačen odprtokodni sistem, ki nam omogoča gostovanje javanskih aplikacij. To omogoča razvijalcem razvoj močnih aplikacij, ki so prenosljive in skalabilne.

3.2 Orodja in spletne tehnologije za razvoj aplikacije

WeBuilder 2011

WeBuilder je izdelek podjetja Blumentals Software. To je orodje (Slika 10) za urejanje poljubne kode, povezane s spletom. Omogoča urejanje HTML, CSS, JavaScript, PHP, ASP in kode Ruby. Vsebuje veliko integriranih orodij, ki omogočajo lažje in hitrejše delo. Omogoča realno časovno preverjanje sintakse jezika in iskanje oziroma odstranjevanje napak (ang. debug). S tem urejevalnikom je bila napisana koda, ki je potrebna za delovanje aplikacije.



Slika 10: Okolje WeBuilder.

Strežnik MySQL

V odprtokodni sferi je strežnik MySQL eden najpopularnejših sistemov za upravljanje podatkovnih baz. Njegove prednosti so preprostost in odprtokodna implementacija, kar pomeni, da je brezplačen [10]. Podatkovno bazo smo potrebovali za hranjenje informacij o

naročilih strank in naročenih izdelkih. Tudi sistem EPCIS deluje na strežniku MySQL. Razen za interne nastavitve aplikacije smo v tej nalogi uporabljali MySQL le za poizvedovanje po lokalnih bazah in poizvedovanje po EPCIS-bazi pri neposrednem dostopu.

Primer poizvedbe po izdelkih v naročilu:

```
SELECT * FROM orderitems WHERE OrderItemsID = $db->escape($id)
```

Programski jezik PHP

PHP ali PHP Hypertext Preprocessor je zelo razširjen odprtokodni programski jezik. Uporablja se za razvoj strežniških aplikacij in dinamičnih spletnih strani [13]. PHP je podoben jezikoma C in Perl in omogoča razvoj zahtevnejših aplikacij brez dolgotrajnega učenja. Teče na spletnem strežniku, kjer kot vhod dobi izvorno kodo in generira spletno stran kot izhod. PHP lahko uporabimo tudi v HTML-datotekah. Če spremenimo končnico v .php, se najprej izvede php-koda, nato se pošlje brskalniku kot spletna stran.

PHP-skripte med seboj komunicirajo na podlagi dveh metod: POST in GET. Podatke med sejo pa lahko shranjujemo v sejno spremenljivko \$_SESSION.

Metoda POST se uporablja za pošiljanje podatkov iz obrazcev ali formularjev. Navadni uporabnik ne vidi, kateri in kakšni podatki so bili poslani. Z metodo POST lahko pošljamo tudi datoteke in daljše alfa numerične podatke. Do podatkov, ki so bili poslani z metodo POST, dostopamo po php-spremenljivki \$_POST, ki predstavlja tabelo vseh vnosov.

Metoda GET se navadno uporablja za razne nastavitve prikaza, kot je na primer številka strani ali parameter aplikacije, ki spremeni način delovanja. Poslani podatki so vidni v naslovni vrstici spletne strani. Primer, ki nam pove, da zahtevamo prikaz modula skladiščenje za naročilo ID 350:

```
fonda/index.php?mod=skladiscenje&id=350
```

Sejna spremenljivka \$_SESSION je super globalna spremenljivka, ki vsebuje tabelo podatkov, ki smo jo zapisali vanjo. Ti podatki se hranijo na strežniku. Pogosto se uporablja za hranjenje informacij o prijavi in uporabniku. Vanjo lahko shranimo tudi informacije, ki jih potrebujemo večkrat. V svoji aplikaciji v sejno spremenljivko shranjujemo vnose obrazcev in tako hranimo zgodovino zadnjih vsebin. To nam omogoča prehajanje med zavijki aplikacije brez izgube podatkov, ki so bili prikazani. Sejna spremenljivka se uniči, kadar poteče piškotek ali kadar zapremo brskalnik.

Označevalni jezik XHTML

XHTML so razvili za potrebe razširitve HTML 4.01 in delovanje z več različnimi oblikami podatkov. Označevalni jezik se uporablja za izdelavo spletnih strani. HTML-dokument sestavljajo oznake – gradniki, ki so lahko gnezdeni. Vsaka oznaka ima svojo specifikacijo in pravila. Z oznakami povemo strukturo strani, ki jo brskalnik pretvori v grafično obliko [16].

Stilski jezik CSS

CSS, drugače tudi kaskadne stilske predloge, so predstavljene v obliki preprostega slogovnega jezika (Slika 11), ki brskalniku pove, kako naj prikaže stiliziran element oziroma oznako. S stili določamo barve, obrobe, pozicije, velikosti in vrsto drugih atributov, ki pripomorejo k sodobnemu videzu spletnih strani. Bistvo predlog CSS je ločitev strukture spletne strani od njene predstavitve. S tem zmanjšamo količino in ponavljanje kode, saj lahko več strani ali elementov strani uporablja iste stile [6].

```
23 #searchResults
24 {
25     padding:5px;
26     display:block;
27     overflow:hidden;
28 }
```

Slika 11: Primer stila CSS za element z ID searchResults.

Programski jezik JavaScript in ogrodje jQuery

JavaScript je prototipni programski jezik, ki nam omogoča ustvarjanje interaktivnih spletnih strani. Program se lahko vključi tudi v HTML-kodo, kjer posluša za uporabnikove akcije, ki aktivirajo funkcije, ki so bile določene (Slika 12).

jQuery je knjižnica za JavaScript, ki poenostavi programiranje interaktivnih aplikacij in omogoča premostitev različne interpretacije JavaScripta različnih brskalnikov [9]. Uporablja pogon Sizzle za izbiranje elementov iz DOM (Document Object Model), ki predstavlja objekte HTML, XHTML in XML. Omogoča spreminjanje DOM s CSS-izbirniki. Vsebuje veliko funkcij z raznimi vizualnimi efekti in animacijami. Omogoča uporabo in razvoj vključkov (ang. Plugins), ki še dodatno razširijo možnosti interaktivnosti. jQuery se uporablja za razvoj asinhronih aplikacij (AJAX). Mi smo uporabili AJAX za hitro iskanje strank in naročil.

```

8     $('#search_input').bind('keyup', function() {
9         var searchQuery = $(this).val();
10        var tipStranke = $("#tipstranke").val();
11        if(searchQuery.length >= 0)
12            {
13            $.post("modules/stranke/model/_ajax.searchStranke.model.php",{
14                tip: tipStranke, search: searchQuery},
15                function(data) {
16                    $('#searchResults').html(data);
17                });
18            $(".searchResults").slideDown("fast");
19        }
20    });

```

Slika 12: Koda jQuery za hitro iskanje.

AJAX

Asinhroni JavaScript z XML se uporablja za asinhrono pošiljanje in dobivanje podatkov. Med zahtevo se stran ne spremeni (ponovno naloži). Kljub imenu XML ni potreben, včasih se uporablja JSON (JavaScript Object Notation). Lahko pa zahtevamo tudi navadno besedilo, ki ga po končani zahtevi vključimo v DOM. V enem primeru se zahteva AJAX opravi (Slika 12) pri hitrem iskanju. Tukaj smo uporabili POST-metodo, s katero pošljemo iskano besedo PHP-skripti, ki nam nato odgovori z zadetki. Te dodamo v element, ki ga s kratko animacijo padanja prikažemo.

Označevalni jezik XML

Razširljiv označevalni jezik XML se uporablja za predstavitev podatkov. Struktura podatkov ni definirana prav zaradi potrebe po razširljivosti. Pomembna je doslednost uporabljanja pravil XML. Pogosto se uporablja za prenos podatkov med različnimi sistemi [17]. Prav za to smo ga uporabili tudi mi (Slika 13). Sporočilo SOAP (Simple Object Access Protocol) je XML-dokument, ki ga pošljemo strežniku (EPCIS Query Client).

Sporočilo SOAP vsebuje:

- Envelope: ovojnico sporočila, ki pove, da gre za SOAP-sporočilo,
- Header: glavo sporočila,
- Body: zahtevane podatke ali parametre.

```

<?xml version="1.0" encoding="UTF-8"?>
- <soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  - <soap:Body>
    - <epcisq:Poll xmlns:epcisq="urn:epcglobal:epcis-query:xsd:1">
      <queryName>SimpleEventQuery</queryName>
      - <params>
        - <param>
          <name>eventType</name>
          - <value>
            <string>ObjectEvent</string>
          </value>
        </param>
        - <param>
          <name>MATCH_epc</name>
          - <value>
            <string>
              http://wmrfid.org/farm2fork/383004776/product/0037/133
            </string>
          </value>
        </param>
      </params>
    </epcisq:Poll>
  </soap:Body>
</soap:Envelope>

```

Slika 13: Primer zahtevka za seznam ObjectEventov ki se ujemajo s podanim EPC.

3.3 Uporabljene metodologije in pogosto uporabljene funkcije

Model-view-controller (MVC)

MVC je arhitektura, s katero dosežemo ločeno obdelavo podatkov za potrebe prikaza informacij. »Model« je sestavljen iz poslovne logike in pridobivanja podatkov, »controller« skrbi za zahteve uporabnika in jih pretvori v ukaze modela ali viewa. View skrbi za prikaz obdelanih podatkov. Lahko je različnih oblik, od tekstovnega do grafičnega. Ker je prikaz ločen od poslovne logike in podatkov, nam to omogoča hitrejša grafične spremembe uporabniškega vmesnika in uporabo predlog.

Data access layer (DAL)

DAL je program, ki poenostavi dostop do podatkovne baze. Vsaka tabela v podatkovni bazi je v DAL predstavljena kot objekt, ki ima lastne spremenljivke namesto polj v bazi. To nam omogoča višjo raven abstrakcije. Namesto da uporabljamo ukaze, kot so *insert*, *select*, *delete* ali *update*, lahko napišemo svoje metode, ki to poenostavijo. V DAL lahko vključimo poslovno logiko, ki nam predpripravi podatke iz podatkovne baze. Tako smo dosegli, da smo v »modelu« že dvignjeni nad raven podatkovne baze in le kličemo metode, ki poskrbijo za dostop do baze in poizvedbe.

Funkcija epcisQuery

S to funkcijo (Slika 14) pošiljamo zahteve na repozitorij EPCIS. Zahteva vsebuje XML-sporočilo s parametri proizvodnje. Kot odgovor dobimo XML-datoteko, ki vsebuje zahtevane dogodke. XML-odgovor je treba pretvoriti v php-tabelo za lažjo obdelavo podatkov.

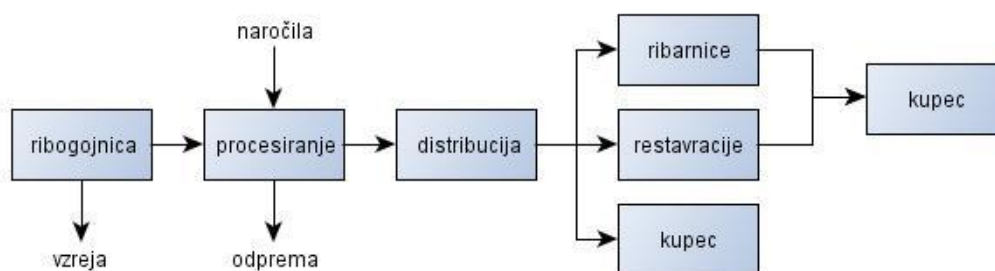
```
360 function epcisQuery($xml_data)
361 {
362     $ch = curl_init(EPCISURL);
363     $headers = array(
364         "Content-type: text/xml;charset=utf-8",
365         "Accept: text/xml",
366         "Cache-Control: no-cache",
367         "Pragma: no-cache",
368         "SOAPAction: \"run\"",
369         "Content-length: ".strlen($xml_data),
370     );
371
372     curl_setopt($ch, CURLOPT_MUTE, 1);
373     curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
374     curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
375     curl_setopt($ch, CURLOPT_POST, 1);
376     curl_setopt($ch, CURLOPT_PORT, EPCISPORT);
377     curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
378     curl_setopt($ch, CURLOPT_POSTFIELDS, $xml_data);
379     curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
380     $output = curl_exec($ch);
381     curl_close($ch);
382     $result = xmlstr_to_array($output);
383     $podatki = $result['soap:Body']['ns3:QueryResults']['resultsBody']['EventList'];
384     return $podatki;
385 }
```

Slika 14: Funkcija epcisQuery.

Knjižnica libcurl nam omogoča komunikacijo PHP z različnimi strežniki z različnimi protokoli. Podpira tudi metode POST, PUT in FTP za pošiljanje datotek.

4 Sistem sledljivosti za razvoj spletne aplikacije

Sistem sledljivosti v preskrbovalni verigi vključuje zbiranje podatkov z uporabo tehnologije RFID vse od ribogojnice do mesta prodaje ali končnega kupca. Za razvoj aplikacije je bila uporabljena pilotna izvedba sledljivosti rib (Fonda piranski brancin), ki je bila implementirana v sklopu evropskega projekta RFID from Farm to Fork [14]. Slika 15 prikazuje mesta za zajem in uporabo podatkov ter posamezne člene v verigi, ki so vključeni v sledljivost. Podjetje ima v svojem procesu vključeno celotno preskrbovalno verigo, od ribogojnice do končnega kupca. Podatki so se zapisovali in shranjevali v papirni obliki ter nato prepisovali v računalnik za različne analize in pregled poslovanja. Uporaba tehnologije RFID je avtomatizirana rešitev, ki zagotavlja zbiranje podatkov v repozitoriju EPCIS in dveh lokalnih bazah.



Slika 15: Preskrbovalna veriga rib.

Sledljivost na področju sveže hrane ali na konkretnem primeru podjetja Fonda.si, d. o. o., je izredno pomembna. Ribe so hitro pokvarljivo živilo, zato so informacije o ravnanju z njimi zelo pomembne za podjetje in tudi kupca. V podjetju že od vsega začetka uporabljajo označevanje posameznega brancina s papirnato značko, ki predstavlja blagovno znamko (Slika 16).



Slika 16: Fonda piranski brancin.

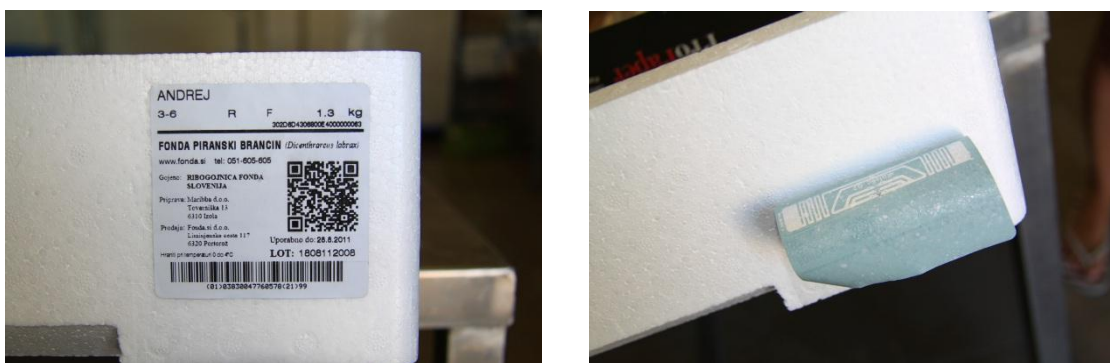
Kot pri vsaki vzreji se pojavlja vprašanje, zakaj je smiselno slediti z uporabo tehnologije RFID ali slediti posamezni ribi ali večjim enotam, to so zaboji. V podjetju so se skupaj z izvajalci projekta dogovorili za sledenje zabojev pripravljenih rib, ki so opremljeni z ustreznim identifikatorjem (Slika 17). Zaradi občutljive narave svežih rib se v zaboj doda tudi aktivna RFID-značka, ki beleži temperaturo okolja in ribe. Podatki o temperaturah so porabljeni v drugi aplikaciji za prikaz na spletni strani v povezavi s predstavitvijo produkta končnemu kupcu in niso vključeni v pregled poslovanja podjetja.

V sistemu sledljivosti so definirane štiri faze preskrbovalne verige:

- ribogojnica,
- procesiranje,
- distribucija,
- ribarnice/restavracije/kupec.

V ribogojnici se zbirajo podatki o sprejemu mladice v kletke, o hranjenju rib, pregledi rib, zdravljenje in smrtnost v kletkah ter beleženje ulova in transporta rib do sortirnice. Vsi zbrani podatki se shranjujejo v bazo EPCIS v obliki dogodkov, povezanih s posamezno kletko.

Procesiranje vključuje aplikacijo naročil, ki je podlaga za določanje količine rib za ulov in beleženje dejanskega ulova (teža in število ulovljenih rib) ter zagotavlja podatke za pripravo in pakiranje rib za odpremo po strankah. Podatki so natisnjeni na nalepki RFID, unikatna EPC-koda pa je v RFID-znački, ki je sestavni del nalepke. Podatki o naročilih strank se shranijo v lokalno bazo naročil, podatki o količinah rib v zabojih in odpremi pa v bazo EPCIS. Slika 17 prikazuje nalepko RFID za dostavo zasebni stranki.



Slika 17: RFID-nalepka na zaboju za dostavo zasebni stranki.

V distribucijo so vključeni prevoz zabojev rib v hladilnico, shranjevanje rib do naslednjega dne, ko se opravi odprema rib z dostavo v trgovine, ribarnice, restavracije in

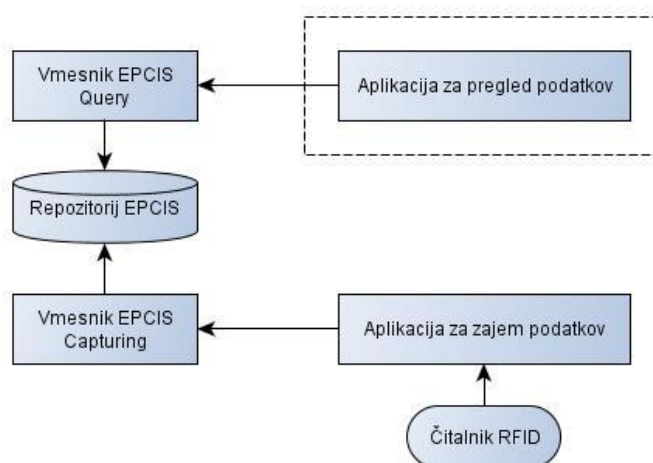
končnim kupcem. Podatki so namenjeni sledenju zabojev ob sprejemu v ali izdaji iz hladilnice in omogočajo podjetju pregled pripravljenih naročil za dostavo strankam.

Ribarnice/restavracije/kupec – ob dostavi zabojev se zabeležijo podatki o sprejemu rib v zadnji fazi preskrbovalne verige. S tem je končan postopek sledenja, ki je na voljo podjetju za pregled in analizo v spletni aplikaciji.

Za zajem podatkov se lahko uporabljajo aplikacije z grafičnimi vmesniki za vpis podatkov in aplikacije za avtomatski zajem podatkov na fiksnih in mobilnih RFID-napravah. Podatki se zbirajo v obliki dogodkov, ki so bili predstavljeni v opisih repozitorija EPCIS in vključujejo:

- Transaction event – uporabljen za identifikacijo zabojev povezanih z naročilom strank,
- Quantity event – uporabljen za podatke o številu rib in teži zabojev,
- Object event – za zajem podatkov o stanju kletk v ribogojnici in zbiranje podatkov o premiku zabojev v preskrbovalni verigi,
- Aggregation event – omogoča združevanje več manjših enot v novo sestavljeno enoto (združevanje zabojev na paleti).

Vsak dogodek vključuje informacijo o sledljivem predmetu (KAJ), času (KDAJ) in lokaciji (KJE) ter poslovnem koraku (ZAKAJ), ki označuje stanje ali akcijo, povezano s sledljivim predmetom. Če želimo zagotoviti še analizo in pregled podatkov, ki so shranjeni v repozitoriju EPCIS, je treba izdelati aplikacijo, ki uporablja vmesnik EPCIS Query za pregled podatkov v sistemu sledljivosti (Slika 18).



Slika 18: Povezava aplikacije z repozitorijem EPCIS.

5 Aplikacija za pregled poslovanja podjetja

Pri načrtovanju aplikacije za pregled poslovanja podjetja smo poskušali slediti sodobnim in priporočenim smernicam za razvoj spletne programske opreme. Pred začetkom izdelave aplikacije smo temeljito analizirali zahteve podjetja in definirali zelene rezultate. Postopek izdelave lahko razdelimo na štiri dele:

- analiza zahtev in podatkov,
- načrtovanje aplikacije,
- izvedba aplikacije,
- testiranje in predstavitev rezultatov.

5.1 Analiza zahtev in podatkov

Izvedba spletne aplikacije je bila povezana z zahtevami podjetja, ki ima vpeljan testni sistem sledljivosti izdelkov v preskrbovalni verigi. Podatki o posameznih procesih za določanje premikov ali stanja proizvoda so v repozitoriju EPCIS, ki je eden od standardov GS1 in se uporablja v prehranski industriji. Standard določa shemo baze, ki pa ne ponuja vseh funkcionalnosti za analizo poslovanja podjetja. Podatkov o strankah, naročilih in notranjih procesih ni v EPCIS, ker so to interni podatki podjetja. Zato smo morali uporabiti še dve lokalni bazi, ki se uporabljata v sistemu sledljivosti v projektu RFID from farm to Fork, to sta: baza naročil in baza ribogojnice.

Poglavitna težava pri analizi zahtev je bilo poizvedovanje po podatkih, ki so zapisani v repozitoriju EPCIS. Obstajata dve možnosti, kjer je prva predstavljena z neposrednim dostopom, kot drugo možnost pa smo izbrali dostop prek vmesnika query, ki je na voljo v RFID-programski platformi Fosstrak. Vendar pa smo po analiziranju zahtev ugotovili, da vseh podatkov tudi ne bo mogoče pridobiti z uporabo vmesnika, ampak je potrebna tudi uporaba neposrednega dostopa. Aplikacija bo delovala v dveh načinih:

- DQPB (za direktni/neposredni dostop do podatkovne baze je uporabljen DAL),
- FQPB (za dostop do podatkovne baze je uporabljen Fosstrak).

Pri DQPB bo za vse podatkovne baze uporabljen samo neposredni dostop. Pri načinu dostopa FQDB bo za repozitorija EPCIS uporabljen Fosstrak, poleg tega pa tudi neposredni dostop do lokalnih baz, saj drugače ni mogoče do podatkov o naročilih in strankah, ki so hranjeni v bazi naročil, ter podatkov o procesih gojenja rib, ki so v podatkovni bazi ribogojnice.

Pri izdelavi spletne aplikacije sta zelo pomembni varnost in možnost dostopa tudi zunaj delovnega okolja zaposlenih. Odločili smo se za prijavno okno, ki z uporabniškim imenom in geslom dovoljuje dostop do modulov, ki ponujajo različne funkcionalnosti. Pri tem je najpomembnejši in obvezni administrativni račun, ki lahko neomejeno dodaja nove uporabnike sistema in jim določa pravice – dostope. Vsakemu zaposlenemu se lahko določi, do katerih modulov ima dostop.

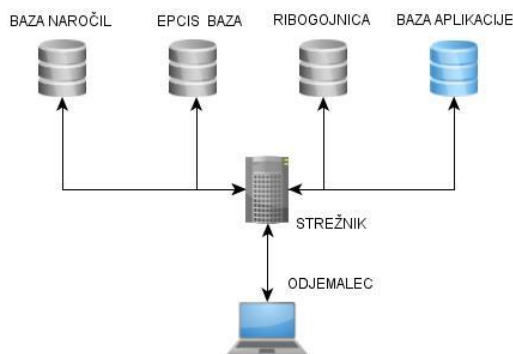
Na podlagi analize podatkov, ki so shranjeni v treh podatkovnih bazah, so bile definirane zahteve, ki so v aplikaciji realizirane kot posamezni moduli:

- pregled naročil in prodaje po strankah,
- pregled prodaje v podjetju,
- pregled procesov in ulova rib v ribogojnici,
- sledljivost naročila (zabojev rib) v preskrbovalni verigi.

5.2 Načrtovanje aplikacije

Po analizi in ugotovitvi, da se za EPCIS-sistem uporablja baza MySQL, je bila na vrsti odločitev o tem, da bomo za realizacijo poslovne logike uporabili programski jezik PHP. Za izdelavo in pogon takšne aplikacije potrebujemo strežnik s podporama PHP in MySQL. Spletna aplikacija se izdelava v označevalnem jeziku HTML, ob katerem smo uporabili stilni jezik CSS. Za pomoč pri interaktivnosti aplikacije smo uporabili ogrodje jQuery.

Pri izdelavi aplikacije smo potrebovali še podatkovno bazo aplikacije, ki omogoča shranjevanje uporabnikov sistema in njihovih pravic. Za predstavitev aplikacije kot poslovne rešitve ta baza nima večjega pomena. Aplikacija mora delovati z lokalnimi in tudi oddaljenimi bazami. V procesu razvoja smo uporabljali tri različne baze: lokalno bazo naročil, kopijo EPCIS-baze in bazo ribogojnice (Slika 19). Bazo aplikacije smo pri opisu funkcionalnosti izpustili, saj ni tako pomembna za pregled poslovanja podjetja.



Slika 19: Koncept delovanja aplikacije.

Za boljšo skalabilnost in nadgradljivost aplikacije sem se odločil za dvojni MVC-pristop. Prvi »controller« skrbi za prijavo in odjavo uporabnika, pa tudi za prikaz zahtevanega modula. Druga plast MVC pa skrbi za prikaz vsebin, validacijo podatkov in komunikacijo z DAL in vmesnikom Fosstrak v vsakem modulu. Takšna rešitev je preprostejša zaradi možnosti upoštevanja že vnaprej nepredvidenih zahtev in kompleksnejših rešitev problema aplikacije. S tem načinom rešujemo specifične zahteve na ravni vsakega modula posebej, ker ne veljajo za vso aplikacijo. Tako se lahko izognemo poznejšim zapletom pri dodajanju novih zmožnosti.

Sestavo datotek in skript sem razdelil v mape po načelu modularnosti in preglednosti:

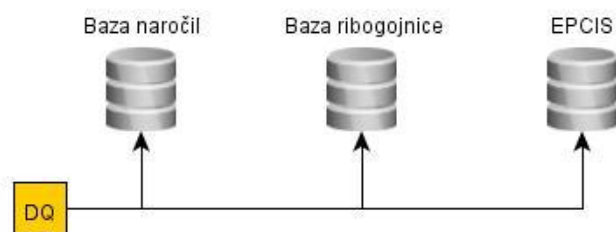
- css (mapa stilnih predlog aplikacije),
- dal (DAL-razredi za dostop do baze podatkov o uporabnikih sistema),
- dal-epcis (DAL-razredi za dostop do EPCIS-baze),
- dal-farmdb (DAL-razredi za dostop do baze ribogojnice),
- dal-fonda (DAL-razredi za dostop do baze naročil),
- js (mapa datotek javascript),
- lib (nastavitvene datoteke, pogosto uporabljene knjižnice, controller prve ravni),
- modules:
 - IME MODULA (za vsak modul)
 - controller (controller trenutnega modula),
 - css (stilne predloge trenutnega modula),
 - js (javascript datoteke trenutnega modula),
 - model (poslovna logika modula),
 - view (prikaz rezultatov poizvedb),
- view (prikaz ogrodja aplikacije in modulov).

5.3 Izvedba aplikacije

Ker smo določili, da bo aplikacija delovala na dva načina, je bilo treba namestiti stikalo, s katerim lahko preklapljamemo med različnima pristopoma poizvedovanja podatkov. To stikalo je vedno vidno v glavi aplikacije in klik nanj povzroči spremembo v sejni spremenljivki `$_SESSION['fonda']`. Vse naslednje zahteve so izpeljane na izbran način.

5.3.1 Neposredno dostopanje do podatkov

Neposredno dostopanje (DQPB – Direktnen query podatkovne baze) deluje po programu DAL, ki nam preslika tabele iz podatkovne baze MySQL v PHP-razrede. Za delovanje aplikacije so potrebni trije različni DAL-programi, ki so povezani vsak s svojo podatkovno bazo (Slika 20).



Slika 20: Neposreden query podatkovne baze.

Za neposredno dostopanje potrebujemo spremenljivko razreda tiste tabele, do katere želimo dostopati.

Primer kode za neposredno dostopanje (pridobivanje pravic prijavljenega uporabnika):

```
$level = new UserLevel();  
$level->SelectById($users[0]->GetLevel());
```

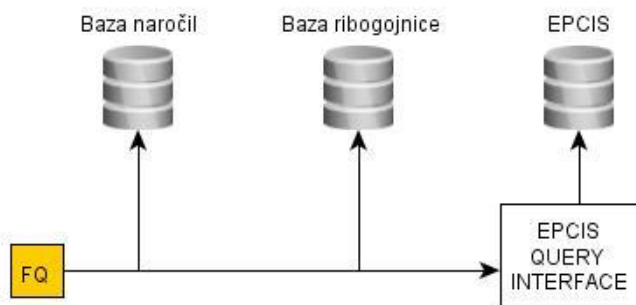
Primer klica metode za pridobivanje seznama ID-naročil glede na vrsto stranke in datumski izbor:

```
$orderIds = Orders::DobiNarocila($tipstranke, $od, $do);
```

5.3.2 Dostop do podatkov z uporabo Fosstraka

Za dostop do repozitorija EPCIS po FDBQ (Slika 21) potrebujemo aplikacijski strežnik GlassFish, na katerem je nameščen Fosstrak EPCIS repository. Repozitorij EPCIS podpira tudi RESTful-arhitekturo, s katero lahko dostopamo do podatkov ne glede na operacijski sistem in programski jezik. Komunikacija poteka dvosmerno. Najprej odjemalec pripravi

SOAP-zahtevo, ki jo oblikuje v XML-datoteko in pošlje strežniku. Strežnik odgovori z XML-datoteko, ki vsebuje zahtevane podatke.



Slika 21: Fosstrak query podatkovne baze.

Primer klica funkcije `epcisQuery`, ki nam vrne PHP-tabelo s seznamom zabojev, kot argument pa sprejme XML-datoteko:

```
$xml_data = getXmlProdaja($od,$do);  
$podatki = epcisQuery($xml_data);
```

5.4 Moduli aplikacije

Aplikacija je sestavljena iz stikala za izbiranje načina dostopa do podatkov in petih modulov za analizo in pregled poslovanja. Stikalo omogoča preklap med načini delovanja (DQPB-dostop je označen z DAL, FQPB-dostop pa z EPCIS). Moduli so dostopni kot gumbi v glavi aplikacije (Slika 22) in lahko pripravijo stran za tiskanje ali izvoz.



Slika 22: Glava aplikacije.

Vsak modul vsebuje navigacijsko ploščo, na kateri so izbirna in vnosna polja, ter prikazano ploščo, na kateri se prikažejo zahtevani podatki in izračuni v obliki urejenih tabel.

5.4.1 Modul Nastavitve

Nastavitve so namenjene le uporabnikom, ki imajo administracijske pravice (Slika 23). Na navigacijski plošči je mogoč hiter dostop do uporabnikov in njihovih pravic, ki so vezane na pregled posameznih modulov. Na prikazni plošči lahko vidimo podrobnosti o vsakem uporabniku. S klikom na uporabniško ime se nam odpre okno, v katerem vidimo te podatke. Prav tako lahko tukaj te podatke spremenimo in dopolnimo.

Nastavitve vsebujejo tudi dnevnik dostopov, v katerega se beležijo vsi dostopi do aplikacije – uspešni in tudi neuspešni. Vidna je tudi vsaka sprememba pravic uporabnikov.

ID	Uporabnik	Ime	Dostop	Datum	E-mail	Zadnja aktivnost	Uredi	Briši
1	admin	Anže Molan	* super Admin *	1.1.1970 1:00		23.5.2012 18:43		
10	Anže	Anže Molan TEST	Zaposleni	24.11.2011 13:51		24.11.2011 16:32		
11	fonda	Test Fonda	Administrator	12.11.2011 22:23		9.4.2012 12:00		

Slika 23: Modul Nastavitve.

5.4.2 Modul Stranke

Vsako podjetje, ki se ukvarja s prodajo, ima svojo bazo strank, za katero mora vestno skrbeti. Pregled strank podjetju omogoča optimizacijo svojih delovnih procesov, saj ima večina rednih strank pogosto enake potrebe po dobavljenosti rib. Med drugim jim omogoča tudi izvedbo projekcije in načrtovanje prihodnje proizvodnje.

Modul omogoča pregled strank, njihovih dosedanjih naročil in dostav v izbranih časovnih obdobjih (Slika 24). Pomemben del tega dela aplikacije omogoča vpogled v naročene izdelke in dejansko zapakirane in dostavljene. Velikokrat se lahko zgodi, da naročila ne morejo izpolniti v takšnem obsegu, kot je bilo zahtevano. Takrat se zapakirajo ribe drugih kategorij v dogovoru s stranko.

Na levi strani aplikacije se nam pokaže izbirno polje strank. Izberemo lahko tip stranke, ki je lahko: fizična, podjetje ali borza. Ko v polje za ime stranke začnemo tipkati ime stranke, ki nas zanima, se nam sproti prikazujejo mogoči zadetki, tako da nam ni treba poznati pravega in celotnega naziva stranke. Takšen način prikaza nam omogoča tehnologija AJAX in jQuery. Aplikacija zazna, da je uporabnik nehal tipkati, in pošlje zahtevo za prikaz strank, ki ustrezajo do zdaj vtipkanemu nazivu stranke in izbranemu tipu stranke. Rezultati se pokažejo kot izbirni seznam, na katerega lahko kliknemo in tako izberemo stranko.

Drugi del izbirnega polja je sestavljen iz datumskega izbora. Izbiramo lahko med več možnostmi: en dan, obdobje med dvema izbranimi dnevnoma, ali vsa naročila. Ko izberemo vsa merila iskalnika po strankah, kliknemo gumb Prikaži in na glavni plošči se nam pokažejo podatki o naročilih in zapakiranih dostavah. Vsaka postavka naročila se prikaže v svoji vrstici z naročenimi izdelki. Zraven vidimo tudi dejansko zapakirano količino, če je naročilo že bilo sprejeto v obdelavo. Na skrajni desni strani obrazca vidimo tudi podatek o tem, ali so bili vsi zaboji tega naročila dostavljeni. S klikom na indikator se nam odpre modul Sledenje.

Pod prvo tabelo imamo še dve dodatni tabeli, ki prikazujeta vsoto naročil in vsoto zapakiranih rib s težami glede na velikost rib. Vidimo lahko, koliko zabojev rib je stranka naročila in kakšne količine so bile dostavljene v izbranem časovnem obdobju.

Pregled stranke (Ribarnica Fonda, Datum: 20.10.2011 - 20.11.2011)									
Naročeno					Zapakirano				
Datum	Št. naročila	Količina	Priprava	Velikost	(velikost)	Količina rib / Teža	Opombe	Dostavljeno	
20.10.2011	360	2 Z	N	3-4	(3-4)	14 / 4.2Kg		✓	
20.10.2011	360	1 Z	N	4-6	(6-8)	24 / 15.9Kg		✓	
20.10.2011	360	1 Z	N	6-8	Skupaj: 38 / 20.1Kg			✓	
03.11.2011	432	4 Z	N	3-6	(3-4)	11 / 3.1Kg		✓	
					(3-6)	11 / 4.6Kg			
					(4-6)	11 / 4.7Kg			
					(6-8)	8 / 5.3Kg			
					Skupaj: 41 / 17.7Kg				
Vsota naročil					Vsota zapakiranih rib				
Datum	Velikost	Št. zabojev	Št. rib	Kg	Datum	Velikost	Št. rib	Kg	
20.10.2011	3-4	2			20.10.2011	3-4	14	4.2	
20.10.2011	4-6	1			20.10.2011	6-8	24	15.9	
20.10.2011	6-8	1			03.11.2011	6-8	8	5.3	
03.11.2011	3-6	4			03.11.2011	4-6	11	4.7	
					03.11.2011	3-4	11	3.1	
					03.11.2011	3-6	11	4.6	
					Skupaj:		79 Rib	37.8 Kg	

Slika 24: Modul Stranke.

Neposredno dostopanje do podatkov (DQPb)

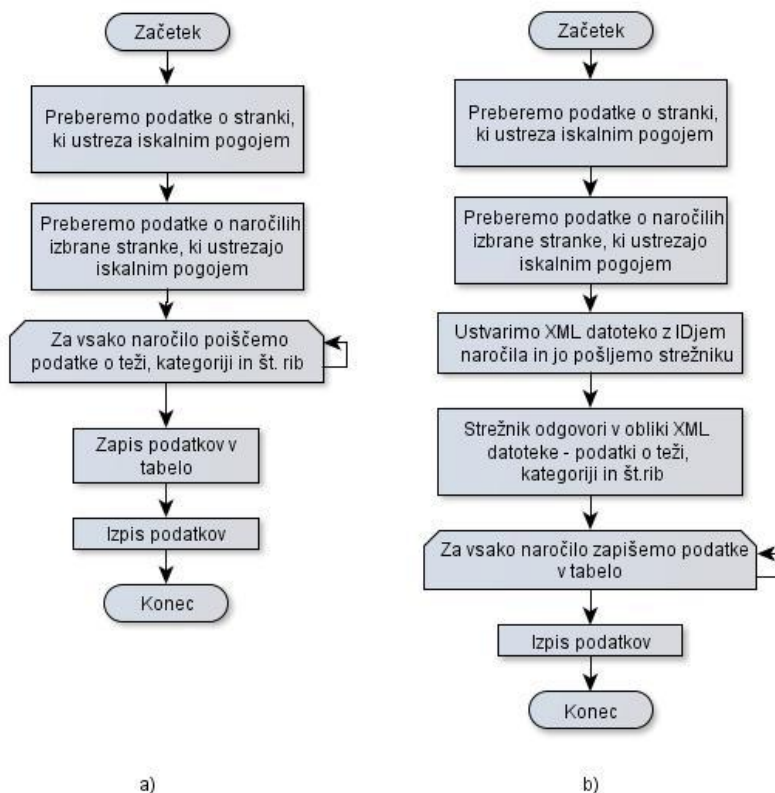
Opis neposrednega dostopanja do podatkov (Slika 25a):

1. Iz lokalne baze naročil izberemo stranko (ime stranke, tip stranke, časovno obdobje proizvodnje).
2. Iz lokalne baze naročil pridobimo vsa naročila stranke v izbranem časovnem obdobju.
3. Za vsako naročilo v EPCIS-bazi poiščemo BizTrans_id, po katerem lahko dostopamo do podatkov v Quantity Eventih.
4. Vsak Quantity Event, ki nam pove količino zapakiranih rib in težo za posamezno velikost, shranimo v PHP-tabelo.
5. Prikažemo izpis.
6. Za vsako prikazano naročilo še preverimo, ali je bila dostava izpeljana (modul Sledenje).

Dostopanje do podatkov po Fosstraku (FQPB)

Dostopanje poteka prek podatkov, ki jih dobimo v obliki XML (Slika 25b):

1. Iz lokalne baze naročil izberemo stranko (ime stranke, tip stranke, časovno obdobje poizvedbe).
2. Iz lokalne baze naročil pridobimo vsa naročila stranke v izbranem časovnem obdobju.
3. Za vsak ID-naročila ustvarimo SOAP-zahtevek v obliki XM in ga pošljemo strežniku (EPCIS-baza).
4. Za vsak odgovor strežnika preverimo, ali ustreza časovnemu okviru, ki smo ga določili.
5. Podatke o naročilu in količinah ter dostavi shranimo v PHP-tabelo.
6. Prikažemo izpis.
7. Za vsako prikazano naročilo še preverimo, ali je bila dostava izpeljana (modul Sledenje).



Slika 25: Zajem podatkov o naročilih in dostavi: a) neposredni dostop, b) Fosstrak.

Pri obeh načinih dostopanja na koncu še seštejemo podatke iz lokalne baze naročil in podatke o številu in teži rib iz EPCIS-baze. Tako dobimo skupen seštevek naročenih in

dostavljenih izdelkov glede na kategorijo rib. Na koncu še uredimo izpis seštevka in ga prikažemo.

5.4.3 Modul Prodaja

Modul Prodaja nam omogoča pregled prodaje rib po kategorijah glede na tip stranke. Na levem izbirnem in vnosnem polju lahko izberemo tip stranke, za katerega želimo izvedeti podatke o prodaji, izbiramo lahko med fizičnimi strankami, podjetji, borzami ali vsemi. Določimo tudi časovni okvir, za katerega želimo imeti prikaz podatkov. S klikom na gumb Prikaži pošljemo zahtevana merila strežniku, ki glede na izbrani način delovanja aplikacije prikaže zahtevane podatke (Slika 26).

Prikažejo se nam podatki, ki ustrezajo iskalnim pogojem, to so podatki o skupni teži in skupnem številu rib. V stolpcu pa so razvrščeni podatki glede na kategorijo zapakiranih rib. Zraven vsakega stolpca imamo tudi podatek o deležu. Na vrhu prikazane plošče imamo gumb Prikaži/skrij stranke, s klikom nanj pa lahko vidimo tudi, katere stranke so bile vključene v izračun prodaje.

Tip	Teža (kg)	%	Št. rib	%
2-3	3	5.93	10	9.71
3-4	7.7	15.22	24	23.30
4-6	17.4	34.39	35	33.98
6-8	22.5	44.47	34	33.01
Skupaj:	50.6	100	103	100

Slika 26: Modul Prodaja.

Neposredno dostopanje do podatkov (DQPБ)

Opis neposrednega dostopanja do podatkov (Slika 27a):

1. Glede na vnesena merila preberemo seznam strank iz lokalne baze naročil.
2. Za vsako stranko izberemo seznam naročil, ki ustrezajo datumu.
3. Z vsakim ID-naročila po BizTransu_id dostopamo do Quantity Eventa.
4. Za vsak dogodek pridobimo podatke o kategoriji rib, številu rib in težah.
5. Zapišemo količino in težo glede na tip rib v PHP-tabelo.
6. Oblikujemo in izpišemo podatke.

Na koncu se ustvari še seznam strank, ki je bil zajet v izračunu. Seznam strank je v izpisu skrit, tako da s klikom na gumb za prikaz strank ni treba spet pregledati vseh naročil in izločati podatke o strankah.

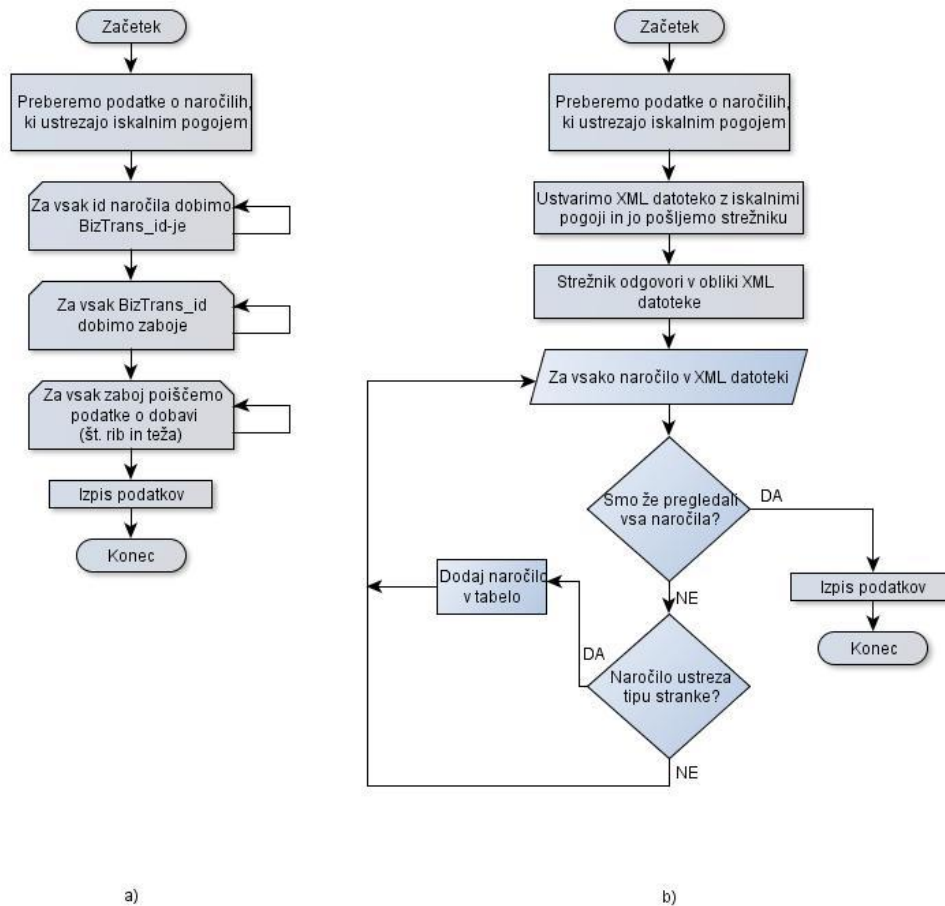
Dostopanje do podatkov po Fosstraku (FQPB)

Ker po EPCIS-vmesniku ne moremo ugotoviti tipa stranke, smo se odločili najprej poslati zahtevek za naročila, ki ustrezajo datumu, in nato po lokalni bazi preveriti, katera naročila nas zanimajo (Slika 27b).

Opis delovanja:

1. Glede na vnesena merila preberemo seznam strank iz lokalne baze naročil.
2. Za vsako stranko izberemo seznam naročil, ki ustrezajo datumu.
3. Ustvarimo SOAP-zahtevek v obliki XML, ki vsebuje zahtevani datumski okvir naročil.
4. Strežnik odgovori s seznamom naročil.
5. Za vsako naročilo preverimo, ali ustreza pogoju »tip stranke«:
 1. Iz lokalne baze preberemo podatke o naročilu.
 2. Glede na ID-stranke iz naročila izberemo podatke o stranki.
 3. Če je tip stranke enak zahtevanemu podatku o naročilu, zapišemo v PHP-tabelo.
6. Če smo že pregledali vsa naročila, oblikujemo podatke in jih izpišemo, drugače se vrnemo na obdelavo naročil.

Med pregledom naročil se beležijo tudi podatki o strankah, ki so nastopale v izračunu.



Slika 27: Diagram za dostop do podatkov o prodaji: a) neposredni dostop, b) Fosstrak.

5.4.4 Modul Ribogojnica

Modul ribogojnica je namenjen pregledu podatkov o vzreji rib vse od prihoda mladice do ulova. Nanašajo se na generacije vstavljenih rib v kletke, pregled hranjenja rib, stanja kletk in zamenjave mrež ter drugo. Na levi strani aplikacije imamo dva obrazca, prvega za izbor generacije, drugega za pregled posamezne kletke.

Če želimo pregled generacije (Slika 28) v izbirnem meniju, kliknemo na želeno generacijo in pritisnemo gumb Prikaži. Na desni strani se nam odpre tabelarni pregled podatkov izbrane generacije. V tabeli so podatki o vnosu generacije, kletke, v katere je bila razdeljena, in deleže. Klik na ime kletke nam pokaže pregled izbrane kletke. Tabela vsebuje tudi skupne podatke o številu ulovljenih rib, najdenih mrtvih ribah in količini hrane. Skupni podatki se nanašajo na generacijo ne glede na to, v kateri kletki so bile ribe.

Ribogojnica		Ribogojnica - pregled generacije (2008/1)											
Generacija:	2008/1	Vnos - datum	Število rib	13.05.2008 - Selitev (št. rib)	27.02.2009 - Selitev (št. rib)	03.03.2009 - Selitev (št. rib)	27.03.2009 - Selitev (št. rib)	31.03.2009 - Selitev (št. rib)	17.03.2009 - Selitev (št. rib)	24.03.2008 - Selitev (št. rib)	Št. mrtvih rib	Hrana (kg)	
<input type="button" value="Prikaži"/>		2008/1	13.05.2008	181710	A1 (17624) A2 (17643) A3 (26074) A4 (26104) A5 (26278) A6 (26369) A7 (17495) A8 (24203)	A1 ->G10 (17569)	A2 ->G1 (17574)	A8 ->G11 (24119) A4 ->G22 (26019) A3 ->G5 (26000)	A5 ->G23 (26180)	A6 ->G39 (26295)	A7 ->G40 (17421)	6350	182202
<input type="button" value="Prikaži"/>		Kletka:	G1										

Slika 28: Pregled generacije.

Kadar izpolnimo obrazec Kletka ali kliknemo na kletko v pregledu generacije, se nam prikažejo podatki izbrane kletke (Slika 29). V tabeli pomeni vsaka vrstica generacijo, ki je bivala v njej. Razberemo lahko datum vnosa, število vstavljenih rib, število mrtvih rib v tej kletki, ulov, količino porabljene hrane, stanje rib, praznjenje mrež, preglede mrež in njihove morebitne zamenjave.

Ribogojnica		Ribogojnica - pregled kletke (G1)									
Generacija:	2008/1	Generacija (datum vnosa)	Kletka izpraznjena	Št. vstavljenih rib	Št. mrtvih rib	Št. ulovljenih rib	Stanje rib	Hrana (kg)	Biometrija (kg)	Pregledi mrež	Menjave mrež
<input type="button" value="Prikaži"/>		2009/1 (05.06.2009)	-	25271	0	0	25271	0		28.09.2010 20.05.2011	
<input type="button" value="Prikaži"/>		2008/1 (03.03.2009)	-	17574	629	0	16945	15768		28.09.2010 20.05.2011	

Slika 29: Pregled kletke z oznako G1.

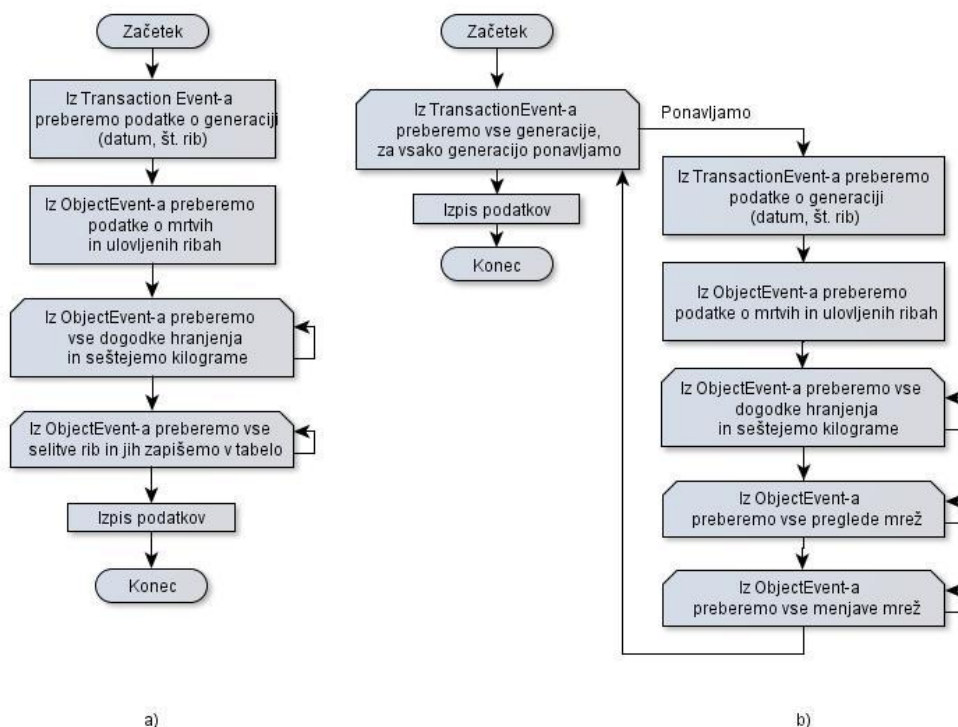
Dostopanje do podatkov o generaciji (FQPB, Slika 30a)

1. Ustvarimo PHP-tabelo, v katero bomo shranjevali podatke o generaciji.
2. Iz TransactionEventa, ki predstavlja prihod generacije, preberemo datum in število rib.
3. Iz ObjectEventa preberemo podatke o mrtvih in ulovljenih ribah.
4. Iz ObjectEventa preberemo vse dogodke hranjenja in seštejemo kilograme porabljene hrane.
5. Iz ObjectEventa preberemo vse selitve generacije.
6. Izpišemo podatke.

Dostopanje do podatkov o kletkah (FQPB, Slika 30b)

1. Ustvarimo PHP-tabelo, v katero bomo shranjevali podatke o kletki.
2. Iz TransactionEventa preberemo vse generacije, ki so bile v tej kletki. Korake od 3 do 7 ponovimo za vsako generacijo.
3. Za generacijo shranimo podatke o datumu vložitve, številu rib in nazivu generacije.
4. Iz ObjectEventa preberemo podatke o mrtvih in ulovljenih ribah.

5. Iz ObjectEvent-a preberemo vse dogodke hranjenja in seštejemo kilograme porabljene hrane.
6. Iz ObjectEvent-a preberemo vse preglede mrež.
7. Iz ObjectEvent-a preberemo vse zamenjave mrež.
8. Izpišemo podatke.



Slika 30: Diagram pridobivanja podatkov za a) generacijo in b) kletko.

5.4.5 Modul Sledenje

Modul se uporablja za sledenje zabojev (EPC) za obdelana naročila. Lahko se zgodi, da v verigi določen paket ne pride do stranke v zahtevanem času oziroma se izgubi. S tem modulom je omogočen pregled sledljivosti pošiljke. Pregled, ali je bilo naročilo v celoti dostavljeno, je implementiran že v modulu «Stranke», pa vendar lahko s tem modulom vidimo čas in datum, ko je zaboj prešel posamezno stopnjo v verigi. Prav tako vidimo zabeležene dogodke dostave zaboja v ribarnico ali k stranki (Slika 31).

Če želimo ročno pregledati določena naročila, lahko to storimo z obrazcem, ki je na levi plošči modula. Imamo iskalnik strank, ki uporablja tehnologijo AJAX, tako da lahko s sprotnim pisanjem izbiramo mogoče ponujene zadetke. Ob kliku na ime stranke se nam pokaže seznam naročil te stranke, skupaj z datumom naročila. S klikom na naročilo, ki nas

zanima, se nam na levi plošči pokaže tabela časa in lokacij. Skrajno levi stolpec nam pove identifikator zaboja EPC, s katerim lahko enolično sledimo zaboj v verigi.

Modul lahko sprejme ID-naročila tudi kot argument v naslovni vrstici:

`index.php?mod=skladiscenje&id=267`



EPC	Izdaja iz sortirnice	Sprejem v hladilnici	Izdaja iz hladilnice	Sprejem pri stranki
797	2011-10-12 14:42:29			
798	2011-10-12 14:42:29			

Slika 31: Modul Sledenje.

Iz zgornje slike lahko razberemo, da je bilo naročilo z ID 267 v lokalni bazi naročil izdano iz sortirnice v dveh zabojih: EPC 797 in EPC 798 na dan 13. 10. 2011, vendar ni prešlo sprejetja v hladilnici in preostalih korakov. Verjetno je bilo naročilo obdelano zunaj standardnega poteka. To lahko tudi pomeni, da zaboj ni bil dostavljen.

Neposredno dostopanje do podatkov (DQPB)

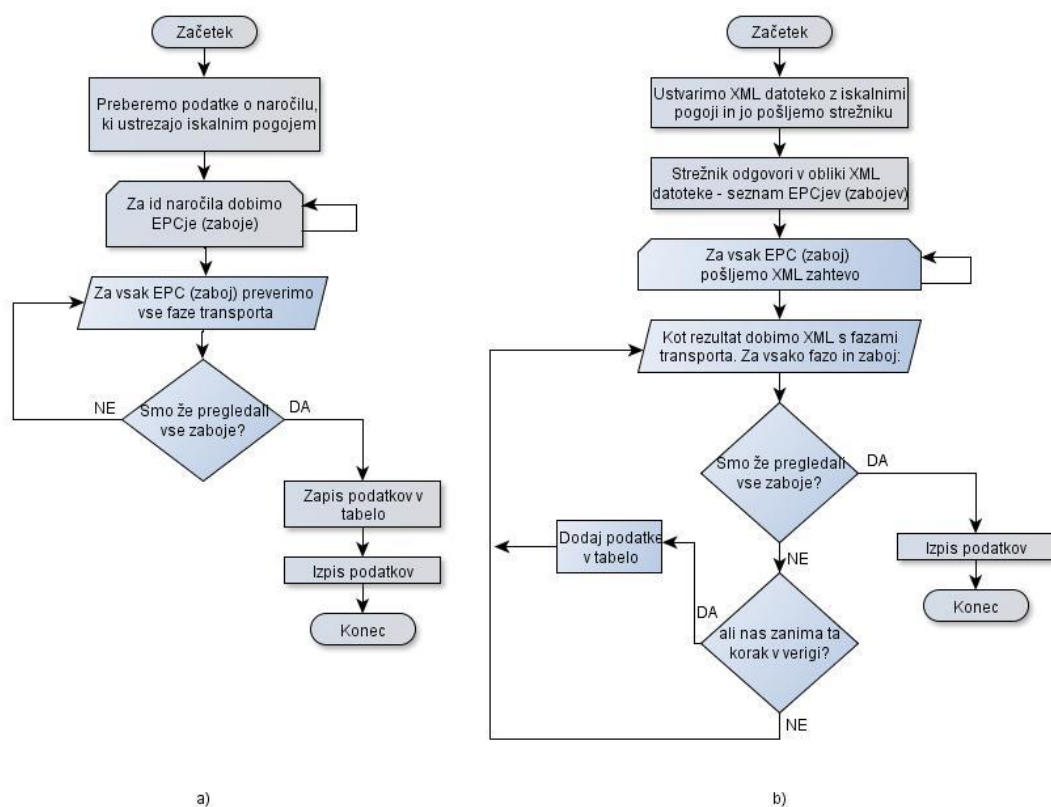
Opis poteka pridobivanja podatkov (Slika 32a):

1. Glede na izbran ID-naročila po TransactionEventu in številke naročila pridemo do EPC, ki predstavljajo zaboj.
2. Za vsak EPC izberemo ObjectEvente, ki vsebujejo EPC.
3. Za vsak ObjectEvent preverimo vse korake v verigi.
4. Za vsako korak shranimo podatke v PHP-tabelo.
5. Kadar preiščemo vse ObjectEvente in vse kombinacije dogodkov, podatke izpišemo.

Dostopanje do podatkov po Fosstraku (FQPB)

Opis dostopanja do podatkov (Slika 32b):

1. Ustvarimo SOAP-zahtevo na podlagi iskalnih pogojev v obliki XML, s katero pridobimo seznam EPC-naročila.
2. Za vsak EPC ustvarimo novo SOAP-zahtevo, ki vrne faze transporta.
3. Za vsako fazo zapišemo podatke v PHP-tabelo.
4. Uredimo in izpišemo podatke.



Slika 32: Diagram pridobivanja podatkov o sledljivosti zabojev: a) neposredni dostop, b) Fosstrak.

5.5 Testiranje aplikacije in analiza rezultatov

Testiranje aplikacije je bilo na vrsti po končani fazi izdelave. Treba je bilo preveriti pravilnost prikaza in izračuna podatkov. Zato smo uporabili oba implementirana dostopa do podatkov. Podatki se v EPCIS zapisujejo z uporabo čitalnikov na lokacijah in prenosnih čitalnikov. Preverjanje pravilnosti delovanja smo začeli s primerjanjem izpisov obeh načinov dostopanja do podatkovne baze. Dokončna potrditev pravilnosti je bila na vrsti po primerjanju izpisov aplikacije z izvozi XML-datotek iz EPCIS-baze.

Primer pregledovanja pravilnosti podatkov pri izpisu naročila stranke:

1. XML-TransactionEventa, ki predstavlja obdelano naročilo (Slika 33).
2. XML-QuantityEventa, ki predstavlja en zaboj (Slika 34).
3. Pregled izpisanih podatkov aplikacije (Slika 35).

```

<TransactionEvent>
  <eventTime>2011-10-20T10:53:53+02:00</eventTime>
  <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
  <bizTransactionList>
    <bizTransaction type="urn:epcglobal:cbv:fmcg:btt:po">http://wmrfid.org/farm2fork/383004776/document/00001/360</bizTransaction>
  </bizTransactionList>
  <epcList>
    <epc>http://wmrfid.org/farm2fork/383004776/product/0031/1042</epc>
    <epc>http://wmrfid.org/farm2fork/383004776/product/0033/1043</epc>
    <epc>http://wmrfid.org/farm2fork/383004776/product/0033/1044</epc>
    <epc>http://wmrfid.org/farm2fork/383004776/product/0033/1045</epc>
  </epcList>
  <action>ADD</action>
  <bizStep>http://wmrfid.org/farm2fork/383004776/bizstep/packing</bizStep>
  <disposition>http://wmrfid.org/farm2fork/383004776/disposition/in_progress</disposition>
  <readPoint>
    <id>http://wmrfid.org/farm2fork/383004776/location/02/10</id>
  </readPoint>
  <bizLocation>
    <id>http://wmrfid.org/farm2fork/383004776/location/02/01</id>
  </bizLocation>
</TransactionEvent>

```

Slika 33: XML TransactionEventa, ki predstavlja obdelano naročilo.

```

<QuantityEvent>
  <eventTime>2011-10-20T10:52:08+02:00</eventTime>
  <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
  <epcClass>http://wmrfid.org/farm2fork/383004776/product/0031</epcClass>
  <quantity>14</quantity>
  <bizStep>http://wmrfid.org/farm2fork/383004776/bizstep/packing</bizStep>
  <disposition>http://wmrfid.org/farm2fork/383004776/disposition/in_progress</disposition>
  <readPoint>
    <id>http://wmrfid.org/farm2fork/383004776/location/02/10</id>
  </readPoint>
  <bizLocation>
    <id>http://wmrfid.org/farm2fork/383004776/location/02/01</id>
  </bizLocation>
  <bizTransactionList>
    <bizTransaction type="urn:epcglobal:cbv:fmcg:btt:po">http://wmrfid.org/farm2fork/383004776/document/00001/360</bizTransaction>
  </bizTransactionList>
  <Fonda:weight xmlns:Fonda="Fonda/epcis/extensions/">4.2</Fonda:weight>
  <Fonda:lot xmlns:Fonda="Fonda/epcis/extensions/">20101119091409</Fonda:lot>
</QuantityEvent>

```

Slika 34: XML QuantityEventa, ki predstavlja zapakiran zaboj.

Pregled stranke (Ribarnica Fonda, Datum: 20.10.2011)						
		Naročeno			Zapakirano	
Datum	Št. naročila	Količina	Priprava	Velikost	(velikost) Količina rib / Teža	
20.10.2011	360	2 Z	N	3-4	(3-4) 14 / 4.2Kg	
20.10.2011	360	1 Z	N	4-6	(6-8) 24 / 15.9Kg	
20.10.2011	360	1 Z	N	6-8	Skupaj: 38 / 20.1Kg	

Slika 35: Izpis aplikacije.

Iz slike 33 je razvidno, da je bilo naročilo razdelano v štiri zaboje (EPC). Vsak zaboj je opisan v svojem QuantityEventu, ki označuje izdajo zaboja iz sortirnice. Na sliki 34 je primer xml zapisa enega zaboja in na sliki 35 vidimo, da je bilo v tem zaboju zapakiranih 14 rib velikosti 3–4, s skupno težo 4,2 kg. Po pregledu preostalih zabojev lahko ugotovimo, da se količinski seštevki ujemajo z izpisom aplikacije.

Primer pregledovanja podatkov pri ribogojnici:

Modul Ribogojnica je narejen samo z FQPB načinom, ker se razlikuje od ostalih po tem, da tu preverjamo stanja v kletkah ter generacijah in nima neposredne povezave s sledljivostjo zabojev. Izpisani podatki iz aplikacije so se preverjali z XML datotekami, ki so bile posredovane Fosstrak vmesniku (Capture).

Za preverjanje izpisov, ki predstavljajo generacije rib (Slika 28), smo naredili primerjavo z naslednjimi datotekami:

1. XML-TransactionEventa, ki predstavlja prihod generacije (bizStep: arriving_fish).
2. XML-ObjectEventa, ki predstavlja štetje mrtvih rib (bizStep: counting_dead_fish).
3. XML-ObjectEventa, ki predstavlja hranjenje rib (bizStep: feeding).
4. XML-ObjectEventa, ki predstavlja prvo vstavljanje rib (bizStep: inserting_fish).
5. XML-ObjectEventa, ki predstavlja selitev rib (bizStep: transferring).

Za prikaz kletke (Slika 29) smo naredili primerjavo z naslednjimi datotekami:

1. XML-ObjectEventa, ki predstavlja prvo vstavljanje rib (bizStep: inserting_fish).
2. XML-ObjectEventa, ki predstavlja selitev rib (bizStep: transferring).
3. XML-ObjectEventa, ki predstavlja štetje mrtvih rib (bizStep: counting_dead_fish).
4. XML-ObjectEventa, ki predstavlja ulov rib (bizStep: harvesting).
5. XML-ObjectEventa, ki predstavlja hranjenje rib (bizStep: feeding).
6. XML-ObjectEventa, ki predstavlja pregled mrež (bizStep: changing_fish_net).
7. XML-ObjectEventa, ki predstavlja menjavo mrež (bizStep: inspecting_fish_net).
8. XML-ObjectEventa, ki predstavlja merjenje biometrije (bizStep: biometry).

Primerjava obeh načinov dostopa do repozitorija EPCIS:

Med izdelavo aplikacije se je neposredni dostop izkazal za zahtevnejšega, med tem ko vmesnik Fosstrak za dostop in delovanje potrebuje strežnik GlassFish in Fosstrak. Ugotovili smo, da se odvisno od števila poizvedb izkaže, da ima vsak način dostopa svoje prednosti in slabosti (Tabela 1).

Tabela 1: Povprečni dostopni časi v sek.

Modul / dostop	Stranke	Prodaja	Ribogojnica - generacija	Ribogojnica - kletka	Sledenje
DQPB	0,0566	2,9038	2,1723	0,5161	1,5141
FQPB	0,0409	6,5708	-	-	0,1871

Direktni dostop potrebuje veliko število poizvedb, kar bi lahko rešili z uporabo view-ov v EPCIS podatkovni bazi. Prodaja je pri FQPB obsežna, ker za vsak dogodek v izbranem časovnem obdobju preverjamo ali tip stranke ustreza iskalnim pogojem.

Končni cilj izvedbe aplikacije je uporaba dostopa FQPB do repozitorija EPCIS, katerega želimo ohraniti in tako omogočiti možnost povezave več neodvisnih partnerjev v verigo sledljivosti po GS1 standardu. Po pregledu rezultatov, lahko rečemo da je Fosstrakov vmesnik bolj vsestranski, lažji za uporabo in bolj praktičen pri razširitvi sistema.

6 Sklepne ugotovitve

Diplomsko delo zajema pregled tehnologij in orodij za načrtovanje in razvoj spletne aplikacije, ki je zasnovana na podlagi podatkov v sistemu sledljivosti izdelkov v preskrbovalni verigi. Aplikacija je bila izdelana zaradi potrebe po pregledu poslovnih podatkov, ki jih lahko zajemamo iz repozitorija EPCIS. Pri razvoju sem se srečal s težavami, ki so bile povezane z neposrednim dostopom v lokalno bazo naročil in preslikavo identifikatorjev v EPCIS-bazo. Neposredni dostop zahteva več časa za učenje in razumevanje arhitekture EPCIS-baze, dostop EPCIS pa omogoča elegantnejše in za razvoj hitrejša rešitve. Neposredni dostop prednjači pred dostopom EPCIS z večjim naborom možnosti. Z dostopom EPCIS namreč ne moremo iskati po delni številki LOT, kar lahko naredimo le z neposrednim dostopom.

Za nadaljevanje razvoja aplikacije bi lahko implementirali MySQL view-e nad EPCIS tabelami, saj bi s tem zmanjšali število poizvedb in pohitrili pridobivanje podatkov. Veliko možnosti je tudi v izboljšavi grafičnega vmesnika, ki bi ga lahko naredili še preprostejšega in preglednejšega. Potrebni bi bili še dodatni testi varnosti prenosa podatkov in morebitne varnostne nadgradnje.

Literatura

- [1] M. Bolić, D. Simplot-Ryl, I. Stojmenović, »RFID Systems, Research trends and challenges«, Wiley, 2010.
- [2] H. Chabanne, P. Urien, J. Susini, »RFID and the Internet of Things«, Wiley, 2011.
- [3] K. Finkenzeller, »RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification«, Munich: Wiley, 2003.
- [4] J. M. Myerson, »RFID in the Supply Chain, A guide to Selection and Implementation«, New York: Auerbach Publication, 2007.
- [5] L. Ruiz-Garcia, L. Lunadei, »The role of RFID in agriculture: Applications, limitations and challenges«, Computers and Electronics in Agriculture, Št. 79, str. 42–50, 2011.
- [6] (2012) Cascading Style Sheets. Dostopno na:
<http://www.w3schools.com/css/>
- [7] (2012) Fosstrak. Dostopno na:
<https://code.google.com/p/fosstrak/>
- [8] (2012) GlassFish. Dostopno na:
<http://glassfish.java.net/>
- [9] (2012) jQuery. Dostopno na:
<http://www.w3schools.com/jquery/>
- [10] (2012) MySQL. Dostopno na:
<http://www.mysql.com/>
- [11] (2012) Navodila za sledljivost živil v trgovini ter njihov umik. Dostopno na:
<http://www.mg.gov.si/>
- [12] (2012) Optical RFID. Dostopno na:
http://en.wikipedia.org/wiki/Optical_RFID
- [13] (2012) PHP. Dostopno na:
<http://php.net/manual/en/index.php>
- [14] (2012) Projekt RFID from Farm to Fork. Dostopno na:
<http://www.rfid-f2f.eu/>
- [15] (2012) Standard sledljivosti GS1. Dostopno na:
<http://www.gs1si.org>
- [16] (2012) XHTML. Dostopno na:
<http://www.w3.org/TR/xhtml1/>
- [17] (2012) XML. Dostopno na:
<http://www.w3.org/standards/xml/>