

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Pavuna

**Mobilna aplikacija za predstavitev  
sledljivosti in hladne verige v  
prehranski verigi**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mira Trebar

Ljubljana 2013



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*





Št. naloge: 00440 / 2013  
Datum: 12.4.2013

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

Kandidat: **GREGOR PAVUNA**

Naslov: **MOBILNA APLIKACIJA ZA PREDSTAVITEV SLEDLJIVOSTI IN  
HLADNE VERIGE V PREHRANSKI VERIGI  
MOBILE APPLICATION FOR PRESENTATION OF TRACEABILITY  
AND COLD CHAIN DATA IN FOOD SUPPLY CHAIN**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Sodobne informacijsko komunikacijske tehnologije omogočajo uporabniku pregled podatkov o sledljivosti produkta v celotni prehranski verigi. V primeru, da se v logističnih procesih, kot sta transport in skladiščenje, izvaja tudi nadzor temperatur je mogoče vključiti tudi pregled meritev. V diplomskem delu naj kandidat analizira sistem sledljivosti z označevanjem izdelkov, ki ga predpisuje standard GS1 za uporabo tehnologije RFID in senzorske podatke na strežniku ter izdela mobilno aplikacijo za pametne naprave (telefon, tablični računalnik). Prikazana naj bo celotna zgodovina ravnanja z živilom, njegov izvor, podatki o roku uporabnosti, kakor tudi o izmerjenih temperaturah v povezavi s predpisanimi pogoji shranjevanja. Rezultati delovanja aplikacije naj bodo prikazani za sledljivost rib, ki je bila implementirana v pilotni izvedbi projekta RFID-F2F (»RFID – from Farm to Fork«).

Mentor:

doc. dr. Mira Trebar

*Mira Trebar*



Dekan:

prof. dr. Nikolaj Zimic

*Nikolaj Zimic*



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Gregor Pavuna, z vpisno številko **63080342**, sem avtor diplomskega dela z naslovom:

### **Mobilna aplikacija za predstavitev sledljivosti in hladne verige v prehranski verigi**

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 16. decembra 2013

Podpis avtorja:



*Zahvaljujem se vsem, ki so mi kakorkoli pomagali pri izdelavi Diplom-  
skega dela. Še posebej se zahvaljujem mentorici doc. dr. Miri Trebar, za  
organizacijo projekta, nasvete in pomoč pri pisanju.*

*Zahvaljujem se tudi družini, ki me je vsa leta mojega študija podpirala.*



# Seznam uporabljenih kratic in simbolov

**EPC** – **E**lectronic **P**roduct **C**ode (standardizirani globalni identifikator)

**EPCIS** – **E**lectronic **P**roduct **C**ode **I**nformation **S**ervices (standard, ki opredeljuje obliko in izmenjavo podatkov)

**F2F** – **F**rom **F**arm **T**o **F**ork (produkt od izvora do mize ali od vilic do mize)

**GMO** – **G**enetically **M**odified **O**rganisms (genetsko spremenjeni organizmi)

**HTML** – **H**yper**T**ext **M**arkup **L**anguage (označevalni jezik za opisovanje spletnih strani)

**HTTP** – **H**yper**T**ext **T**ransfer **P**rotocol (protokol za prenos informacij na spletu)

**JSON** – **J**ava**S**cript **O**bject **N**otation (struktura za prenos podatkov med strežnikom in aplikacijo)

**MySQL** – **M**y **S**tructured **Q**uery **L**anguage (odprtokodni programski jezik za upravljanje poizvedb nad podatkovno bazo)

**NFC** – **N**ear **F**ield **C**ommunication (visokofrekvenčna komunikacijska tehnologija kratkega dosega)

**QR Code** – **Q**uick **R**esponse **C**ode (matrična oz. dvodimenzionalna(2D) črtna koda)

**RFID** – **R**adio-**F**requency **I**Dentification (radiofrekvenčna identifikacija)

**SQL** – **S**tructured **Q**uery **L**anguage (programski jezik za upravljanje poizvedb nad podatkovno bazo)

**SQLite** – **S**tructured **Q**uery **L**anguage (odprtokodni programski jezik za

upravljanje z majhnimi podatkovnimi bazami)

**XML** – e**X**tensible **M**arkup **L**anguage (struktura za prenos podatkov med strežnikom in aplikacijo)

# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Hladna veriga in sledljivost v prehranski verigi</b>	<b>5</b>
2.1	Opis problema . . . . .	6
2.2	Pilotna rešitev sistema sledljivosti . . . . .	7
2.3	Obstoječe rešitve . . . . .	8
2.3.1	Global tracking systems . . . . .	8
2.3.2	Integra2 . . . . .	9
2.3.3	HarvestMark . . . . .	10
<b>3</b>	<b>Uporabljene tehnologije in programska oprema</b>	<b>13</b>
3.1	Mobilne tehnologije . . . . .	13
3.1.1	Operacijski sistem Android . . . . .	14
3.1.2	AChartEngine . . . . .	14
3.1.3	Aplikacija ZXing barcode scanner . . . . .	15
3.1.4	Google Maps API v2 . . . . .	15
3.1.5	QR Code . . . . .	15
3.1.6	NFC in pametne naprave . . . . .	16
3.2	Spletne tehnologije . . . . .	16
3.2.1	RESTful spletna storitev . . . . .	16

3.2.2	JSON in XML . . . . .	17
3.2.3	HTML in CSS . . . . .	18
3.3	Programski jeziki in orodja . . . . .	18
3.3.1	Eclipse . . . . .	18
3.3.2	Oracle GlassFish . . . . .	19
3.3.3	Podatkovne baze . . . . .	19
3.3.4	Programski jezik Java . . . . .	20
<b>4</b>	<b>Mobilna aplikacija TColdChain</b>	<b>21</b>
4.1	Sistem za izvedbo hladne verige . . . . .	21
4.2	Načrtovanje aplikacije . . . . .	24
4.3	Strežnik . . . . .	27
4.3.1	Podatkovne baze . . . . .	27
4.3.2	RESTful spletna storitev . . . . .	30
4.4	Delovanje Android aplikacije . . . . .	32
<b>5</b>	<b>Opis in testiranje aplikacije</b>	<b>35</b>
5.1	Grafični vmesnik . . . . .	35
5.1.1	Vstopna stran in funkcionalnosti aplikacije . . . . .	36
5.1.2	Sledljivost . . . . .	39
5.2	Testiranje . . . . .	43
<b>6</b>	<b>Sklepne ugotovitve</b>	<b>47</b>

# Slike

2.1	Prikaz informacij o iskanem produktu na spletni strani podjetja HarvestMark. . . . .	11
2.2	Slika mobilne aplikacije za operacijski sistem Android. . . . .	11
4.1	Arhitektura sistema za izvedbo hladne verige z uporabo tehnologije NFC. . . . .	22
4.2	Skica načrtov pri načrtovanju grafičnega vmesnika za: (a) prvo stran; (b) stran opisa produktov; (c) stran prikaza procesov. . . . .	26
4.3	Relacijska shema tabel iz podatkovnih baz. . . . .	28
4.4	Dve SQL poizvedbi. Prva pridobi podatke o sledljivosti za določen produkt. Druga pa dobi podatke o lokaciji, kje se je kakšen proces zgodil. . . . .	30
4.5	Primer XMLja epcis za produkt z oznako 0042. . . . .	31
4.6	Primer XMLja Sensor za produkt z oznako 0042. . . . .	32
4.7	Prikaz sočasnega izvajanja pridobivanja podatkov iz senzorskega oz. drugega XMLja. . . . .	34
5.1	Vstopna stran aplikacije TColdChain. . . . .	38
5.2	Izgled strani kartice, informacij in zgodovine. . . . .	38
5.3	Primeri dinamičnega izrisa podatkov o produktu: (a) sledljivost in hladna veriga; (b) hladna veriga; (c) sledljivost. . . . .	40
5.4	Primer časovnega grafa iz aplikacije. . . . .	41
5.5	Faze sledljivosti. . . . .	42

5.6	Primer dobavne verige – lokacije. . . . .	43
5.7	Rezultat testiranja na napravi z velikostjo in gostoto zaslona small-ldpi: (a) opis produkta; (b) časovni graf. . . . .	45
5.8	Rezultat testiranja na napravi z velikostjo in gostoto zaslona large-tvdpi: (a) opis produkta; (b) časovni graf. . . . .	46

# Povzetek

Sledljivost živil v prehranski verigi postaja za podjetja in za potrošnike zelo pomembno, a še vedno pomanjkljivo obdelano področje z vidika informacijsko komunikacijskih tehnologij. Na voljo imamo množico naprav v internetnem ali mobilnem omrežju, s katerimi lahko povsod pridemo do informacij, če so nam na voljo. Pregledali smo nekatere obstoječe rešitve, ki nam omogočajo pridobivanje podatkov o hrani, ki jo kupujemo. Podrobno je bila preučena tudi pilotna rešitev sistema sledljivosti za ribe RFID-F2F, ki je zagotavljala realne podatke o sledljivosti rib v Sloveniji. Cilj diplomske naloge je bil razvoj mobilne aplikacije za sistem Android, katera povezuje podatke sledljivosti in hladne verige iz treh podatkovnih baz na strežniku. Mobilna aplikacija uporabniku omogoča pregled informacij o produktu, njegov izvor, vse korake v poslovnem procesu in še druge informacije. Zasnovana je bila tako, da uporabnik skenira QR Code ali prisloni mobilno napravo k NFC pametni kartici ali nalepki na pakiranem živilu. Ta nato pošlje zahtevek preko HTTP metode GET na strežnik, kjer se nahaja RESTful spletna storitev, katera pridobi vse potrebne podatke in jih pošlje nazaj v obliki XML na mobilno aplikacijo. Ta nato vse informacije prikaže uporabniku na pametni napravi in mu na ta način pomaga pri odločitvi o nakupu izbranega produkta.

**Ključne besede:** Android, RESTful spletna storitev, Hladna veriga, Sledljivost, RFID-F2F, NFC, QR koda



# Abstract

Food traceability is becoming more and more important for both business and consumers, but is still inadequate in terms of treated scope of information and communication technologies. We have a plurality of devices connected on the internet or via mobile network, which can get us the needed informations that are available to us. We have reviewed some of the solutions, which are enabling us to access informations about the food we buy. The RFID-F2F pilot traceability system solution for fish was examined to its details, which grants us real data on fish traceability in Slovenia. The aim of the thesis was to develop a mobile application for Android, which links traceability and cold chain data from three different databases on the server. The mobile application enables the user to review information about the product, its origin, all steps in the business process and other information. It was designed as a user friendly application, since the user can simply scan a QR code or place a device near a NFC smart card or label on packaged food. This sends a request via HTTP GET method on the server where the REST web service is located. This service then obtains all the necessary information and sends them back to the mobile application as an XML structure. After the data is received on a mobile device, it is presented to the user. Then it is up to the user to decide, whether he is going to purchase the product or not.

**Keywords:** Android, REST web service, Cold chain, Traceability, RFID-F2F, NFC, QR Code



# Poglavje 1

## Uvod

V svetovnem okolju je vse več podjetij v preskrbovalni verigi, ki morajo ali pa želijo zagotoviti zbiranje in izmenjavo podatkov o sledljivosti v elektronski obliki. Tu se pojavi sodelovanje večjega števila podjetij, partnerjev, kateri delujejo kot: proizvajalec, prevoznik, prodajalec, ipd. Zaradi potreb po izmenjavi informacij je pri sledenju v papirnati obliki (preko naročilnic) zelo zamudno ali skoraj nemogoče ugotoviti, kaj se je dogajalo v primeru pokvarjenega produkta, kje se je pokvaril in kdo je odgovoren za nastalo situacijo. Potrošniki pa tudi ne morejo pogledati in se osebno prepričati o tem ali je produkt, ki ga želijo kupiti res takšen, kot piše na embalaži. Najnovejše tehnologije omogočajo napredne rešitve za elektronsko sledljivost produktov in izvedbo hladne verige. Avtomatska sledljivost v preskrbovalni verigi se je v številnih primerih izkazala kot najzanesljivejši način zagotavljanja varne oskrbe z dobrinami. Omogoča nam nadzor nad vsakim procesom in zagotavlja informacije o ravnanju s produkti v primeru zahtevanega nadziranja temperatur v obdobju skladiščenja ali prevoza vse od začetne faze spremljanja produkta pa do končnega uporabnika-potrošnika v realnem času. Le na ta način lahko imamo zagotovljene pogoje, da se na prodajnih policah pojavi sveža in kvalitetna hrana. Če pride do kakršnih koli problemov, pa lahko takoj ugotovimo kje in kdaj je prišlo do napačnega ravnanja s proizvodi.

Da bi zagotovili zanesljivo elektronsko sledljivost in hladno verigo je po-

trebno izbrati primerno strojno in programsko opremo ter komunikacijsko omrežje. V strojni opremi se vse bolj uveljavlja tehnologija radiofrekvenčne identifikacije, kot so RFID čitalci, zapisovalniki temperatur, antene za oddajanje, GPS tehnologija za shranjevanje lokacije, ipd. Pri programski opremi pa je potrebno zagotoviti različne namizne ali mobilne aplikacije s katerimi bodo uporabniki (delavci) zapisovali ali urejali ustrezne podatke ter jih shranjevali v podatkovne baze na spletnih strežnikih. Na ta način bodo lahko uporabniki imeli na voljo tako spletne, kot tudi mobilne aplikacije, katere bodo prikazovale zahtevane informacije.

V zvezi z omenjenimi izzivi na področju mobilnih in spletnih tehnologij smo se odločili za implementacijo samostojnega sistema hladne verige, ki pa ga je možno vključiti v obstoječo pilotno rešitev projekta RFID-F2F (RFID-from Farm to Fork). Predlagana izvedba sledljivosti je bila zasnovana na tehnologiji RFID. Naša nadgradnja pa vsebuje tehnologijo NFC (angl. Near Field Communication), ki jo je možno uporabiti na pametnih napravah (telefoni, tablični računalniki). Arhitektura sistema v tem projektu vključuje tri aplikacije: (i) upravljanje kontekstnih podatkov o podjetju in produktu; (ii) nadzor in meritve temperatur s senzorskimi pametnimi značkami; (iii) predstavitev informacij o sledljivosti in hladni verigi produkta.

V diplomskem delu je predstavljena rešitev za tretji del arhitekture, to je prikaz informacij za izvedbo sledljivosti iz projekta RFID-F2F v povezavi z izvedbo hladne verige in uporabo tehnologije NFC. Razvita je bila mobilna aplikacija, katera povezuje podatke o sledljivosti in hladni verigi iz treh podatkovnih baz na skupnem strežniku. Za njeno delovanje je bilo potrebno razviti tudi RESTful spletno storitev, katera pošilja zahtevane podatke iz strežnika na mobilno aplikacijo. V fazi testiranja so bili vključeni realni podatki podjetja Fonda.si, kateri so bili na voljo iz pilotne rešitve sledljivosti rib vse od ribogojnice do končnega kupca, oz. do našega testnega mesta v Laboratoriju za računalniško arhitekturo na FRI.

V prvem delu diplomske naloge smo opisali področje našega dela in nekatere že obstoječe rešitve sledljivosti in hladne verige. Na podlagi teh informa-

cij smo se odločili, katere funkcionalnosti naj ima naša aplikacija in na kakšen način predstavimo informacije na pametnih napravah. V nadaljevanju smo se posvetili načrtovanju in izdelavi mobilne aplikacije za povezavo in prikaz informacij v sistemu za izvedbo hladne verige z uporabo tehnologije NFC. Opisali smo tehnologije, s pomočjo katerih smo razvili aplikacijo, njeno izdelavo, kakor tudi spletne aplikacije (RESTful spletna storitev na GlassFish strežniku). In na koncu prikazali in opisali teste, katere smo opravili na več napravah.



## Poglavje 2

# Hladna veriga in sledljivost v prehranski verigi

Sledljivost produktov v prehranski verigi postaja ena od pomembnih metod za zagotavljanje varnejše oskrbe s hrano med proizvajalci in potrošniki. Predstavitev informacij je vse bolj pomembna, ker se vseskozi srečujemo s pojavi, kot so mešanje govejega mesa in konjskega mesa, živalske bolezni v mesu in vprašanja v zvezi z gensko spremenjenimi organizmi (GMO – Genetically modified organisms). Tudi zakonodaja po vsem svetu postavlja zahteve po vzpostavljanju sistemov, ki omogočajo nadzor nad procesi, ki so pomemben člen vsake prehranske verige. Na ta način je omogočeno potrošnikom, da ugotovijo ali je hrana res takšna, kot piše na embalaži in kot jo želijo kupiti [39]. Hladna veriga se ukvarja z učinkovitostjo nadzora temperatur v prehranski verigi. Omogoča nam pregled senzorskih meritev samega produkta in okolja vse od izvora pa do trgovine oz. končnega kupca in s tem ohranja varnost proizvodov in zmanjšanje izgube. Ohranja tudi svežost in kvaliteto v prehranski verigi [6].

## 2.1 Opis problema

Številna podjetja se vse bolj zavedajo, da si lahko zagotovijo svoje konkurenčne prednosti na svetovnem trgu samo z informacijskimi sistemi, ki jim omogočajo takojšnjo povezavo s kupci. Pri tem je pomembno upoštevati dejstvo kako pomembna je tehnologija, ki omogoča hiter pregled nad produkti, poslovnimi koraki in temperaturam v realnem času. Posamezna podjetja imajo svoje sisteme sledljivosti v elektronski obliki ali pa še zmeraj uporabljajo papirnato dokumentacijo, ki postane pri velikih količinah produktov nepregledna. Če se hrana pri prevozu ali skladiščenju pokvari, se pogosto zgodi, da jo je potrebno zavreči. Lahko se zgodi, da šele potrošnik ob nakupu ugotovi, da je hrani potekel rok trajanja ali da je slabe kvalitete in nastane problem ugotavljanja vzrokov za nastalo situacijo. Iskanje krivca ali vzrokov je pogosto zamuden proces. Če nimamo podatkov o temperaturah in vlažnosti pri posameznih zabojih hitro pokvarljivih živil. Zaboju oz. vsaki škatli od začetka, pa do trgovine oz. privatnega kupca, nikakor ne moremo vedeti, kje so bile temperature previsoke oz. prenizke in se je produkt pokvaril. V tem primeru ima tudi trgovec škodo, saj hrane oz. produkta ne more vrniti, ker ne ve, niti ne more dokazati ali se je hrana pokvarila v trgovini ali že prej.

Vse več potrošnikov si želi, da bi imeli na voljo zdravo in ekološko pridelano hrano. Zaradi tega prihaja do povpraševanja po informacijah o hrani, o njenem izvoru, kako je bila pridelana (škropljenje, gensko spremenjena hrana). Problem je, da potrošniki ne morejo videti, kaj se je z produktom oz. hrano dogajalo od vsega začetka pa do trenutka, ko jo želijo kupiti.

## 2.2 Pilotna rešitev sistema sledljivosti

V Sloveniji je bil razvit pilotni sistem sledljivosti z analizo hladne verige na Fakulteti za računalništvo in informatiko v Ljubljani za podjetje Fonda.si, katera se ukvarja z gojenjem rib. V rešitvi je predlagana implementacija RFID (Radiofrekvenčne identifikacije), ki vključuje namizne, in mobilne aplikacije za vnašanje in obdelavo podatkov na štirih lokacijah podjetja: (i) ribogojnica, (ii) obrat za predelavo in pripravo naročil, (iii) hladilnica in (iv) ribarnica. Vsi sistemi morajo biti povezani v internetno omrežje in omogočajo shranjevanje podatkov v podatkovni bazi EPCIS (angl. Electronic Product Code Information Services) na strežniku. V ribogojnici so uporabljeni ročni čitalci RFID, kateri zbirajo podatke o ribah, hrani in temperaturi pošiljke do predelovalnega obrata. V predelovalnici se ribe sortirajo glede na naročilo stranke. Tukaj se v škatle vstavi tudi RFID zapisovalnike, kateri merijo temperaturo med prevozom do hladilnice in v času skladiščenja. Ko pride škatla v hladilnico, RFID čitalec (angl. RFID reader) na glavnih vratih prebere kodo EPC in s tem ugotovi ali je škatla prišla v hladilnico in to zabeleži. Naslednji dan se ribe odpelje iz hladilnice privatnim strankam, restavracijam in trgovinam. Ko produkt prispe do končne točke, se vzamejo RFID zapisovalniki iz zabojev z ribami, zapisane temperature in podatki o lokacijah se preberejo z mobilnim RFID čitalcem, da se zapišejo v podatkovno bazo na strežniku. Vsi podatki, ki so shranjeni v EPCISu in v podatkovni bazi z izmerjenimi temperaturami, omogočajo predstavitev na spletni strani. Vidimo lahko podatke o hranilni vrednosti izbrane ribe, procese, kateri so bili izvedeni nad produktom, prevoženo pot na zemljevidu in temperaturni graf [30].

Pri izvedbi sledljivosti je bil za identifikacijo posameznih zabojev rib uporabljen sistem GS1 EPC/RFID. To je standard, ki povezuje tehnologijo RFID in sistem označevanja EPC (angl. Electronic Product Code). Sistem označevanja EPC se uporablja z namenom za neposredno enolično avtomatsko identifikacijo in za spremljanje posameznega produkta v preskrbovalni verigi [18]. V projektu RFID-F2F so bili uporabljeni naslednji identifikatorji:

- SGTIN – identifikator za produkte ("urn:epc:id:sgtin:383004776.0042.1006268"),
- GRAI – identifikator za sredstva ("urn:epc:id:grai:383004776.004.1"),
- SGLN – identifikator za lokacije ("urn:epc:id:sgln:383004776.020.01").

## 2.3 Obstoječe rešitve

V zadnjem obdobju se pojavljajo različne mobilne in spletne aplikacije, ki omogočajo pridobivanje informacij o izvoru in ravnanju z živili v prehranski verigi. Številni proizvajalci, predvsem manjša podjetja želijo na ta način prepričati potrošnike o zdravi, kvalitetni hrani. Obstajajo tudi rešitve, ki se navezujejo na programske rešitve za sisteme sledljivosti v povezavi z nadzorom hladne verige, ki jih ponujajo posamezna podjetja, ampak so običajno plačljive. Podrobnejših analiz za takšne rešitve žal ni bilo mogoče opraviti. Obstajajo pa tudi posamezne mobilne aplikacije za pametne telefone, ki omogočajo povezave na spletne strani proizvajalcev hrane. Poleg njih pa se pojavljajo tudi druge rešitve, kjer so proizvodi označeni s kodo QR in omogočajo pridobivanje pravih podatkov o sledljivosti. V nadaljevanju si oglejmo tri aplikacije, ki ponujajo različne pristope in rešitve za informiranje uporabnikov.

### 2.3.1 Global tracking systems

Ponujajo celovit paket strojne in programske opreme, storitev in vzdrževanja. Pri izgradnji njihove rešitve ponujajo rešitve, ki so po meri za vsakega kupca. Na voljo je v celoti prilagojen portal za hladno verigo, ki se lahko nahaja na strežniku podjetja naročnika ali strežniku ezTRACK, ki je bil narejen na strani njihovega partnerja GS1 Hong Kong. GTS (angl. Global tracking systems) portal za hladno verigo ponuja globalno sledljivost in spremljanje vseh proizvodov, da bi dosegli end-to-end vidnost. To pa pomaga podjetju doseči večjo donosnost naložbe. Storitve, ki jih podjetje GTS ponuja:

- Zagotavlja RFID (angl. Radio-Frequency Identificaton) oznake za sledljivost paletam in kosom dobrin, antene, mobilne naprave, ročne ali stacionarne bralnike in naprave z tehnologijo GPS,
- shranjuje se več informacij naenkrat vključno s točko poti, lokacijo, čas, temperature in informacijo naročilnice,
- zraven temperature shranjuje še vlažnost in na željo uporabnika se ti dve informaciji ločita,
- zavarovalne prilagoditve in funkcionalnost preverjanja [12].

Ponujajo tudi tehnologijo CLASS-3 BAP RFID, kateri ponuja številne zmogljivosti. Tehnologija je brezžična, tako da namestitev in implementacija te tehnologije ne vpliva na obstoječe poslovne procese. Glede na to, da je tehnologija brezžična, RFID oznake lahko direktno postavimo zraven produkta in jih preberemo ne da bi bilo potrebno odpirat embalažo. Prednosti Class-3 BAP RFID so:

- Razširjeno območje branja vse do približno 300 metrov. V primerjavi z starejšo tehnologijo C1G2, ko ima doseg branja le približno 10 metrov je bolj robusten in se bolj izkaže v zahtevnih okoljih obdanih z kovinami, tekočinami znotraj palet in zabojnikov,
- nizka poraba energije,
- velik razširjen pomnilnik,
- varnost podatkov,
- sposobnost dodati/vključiti senzorje za temperaturo, vlažnost, ipd. [13].

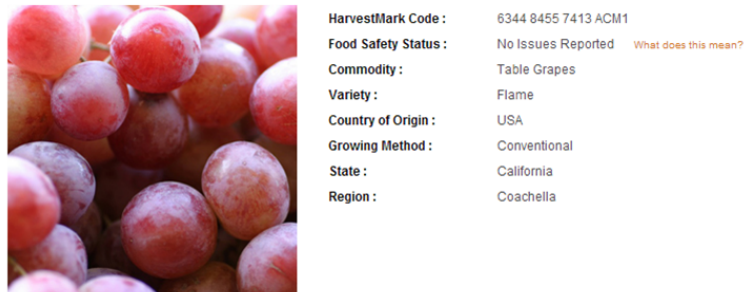
### 2.3.2 Integra2

Sledljivost in hladna veriga sta zelo pomembni v farmacevtski industriji. Integra predstavlja rešitev v transportu, ki uporabnikom (zaposlenim) omogoča

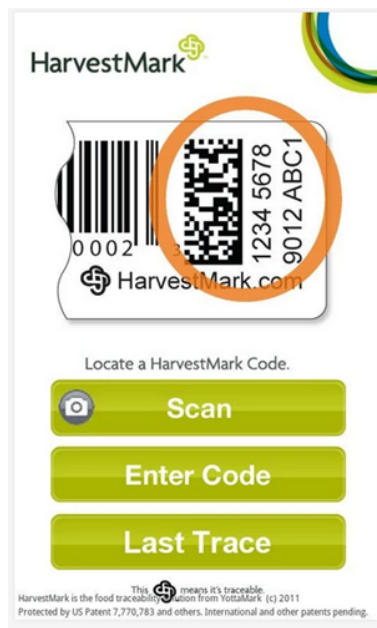
pregled nad lokacijami in temperaturam zdravil preko spletne strani in preko mobilne aplikacije. Mobilna aplikacija omogoča pregled poti po kateri je pošiljka potovala in sledljivost procesom. Na spletni strani lahko uporabniki sledijo tudi svoji pošiljki tako, da vpišejo referenčno številko produkta in poštno številko [21].

### 2.3.3 HarvestMark

Podana rešitev vključuje sledljivost živilom vse od njihovega izvora pa do končnega kupca. Na spletni strani omogočajo sledljivost produktu po njihovi HarvestMark kodi, ki je dolga 16 znakov. Na sliki 2.1 lahko vidimo kakšne podatke ponujajo za določen produkt. Rešitev je bolj namenjena za pridelke kmetovalcev, ki so pridelani na polju [19]. Vsebujejo tudi mobilno aplikacijo za operacijska sistema Android in iOS. Uporabnikom nudijo pregled lokacij pridelovanja in podatke o tem, da je ta hrana tudi zdrava. Aplikacija omogoča zgodovino pregledanih produktov, tako da si lahko produkte ponovno pregledujemo tudi pozneje. Za ogled produkta na mobilni aplikaciji lahko skeniramo QR kodo na zelenjavi, sadju in perutnini ali pa vpišemo kodo, ki je dolga 16 znakov. Na sliki 2.2 si lahko pogledamo izgled mobilne aplikacije za operacijski sistem Android [27].



Slika 2.1: Prikaz informacij o iskanem produktu na spletni strani podjetja HarvestMark.



Slika 2.2: Slika mobilne aplikacije za operacijski sistem Android.



## Poglavje 3

# Uporabljene tehnologije in programska oprema

Za razvoj aplikacije smo uporabljali mobilne in spletne tehnologije, programska orodja, programski in opisni jezik. Za shranjevanje podatkov smo uporabili podatkovno bazo MySQL in SQLite. Aplikacijo, ki se izvaja na operacijskem sistemu Android smo razvili v programskem jeziku Java, v programskem razvojnem okolju Eclipse in z dodatkom Android orodja za razvijalce (ADT - Android Developer Tools). Vsebine smo urejali in oblikovali z uporabo opisnega jezika HTML in preprostega slogovnega jezika CSS. Za RESTful spletno storitev (aplikacija na strežniku) smo uporabili programski jezik Java z JAX-RS. Podatke iz strežnika GlassFish pridobivamo s tehnologijo XML. Za skeniranje QR kode smo uporabili odprtokodno Android aplikacijo ZXing barcode scanner. Graf izrisujemo s pomočjo tehnologije AChartEngine. Lokacije se v zemljevidu rišejo z Android Google Maps API v2. Podatke iz strežnikov Google Maps prejemo z vključitvijo tehnologije JSON.

### 3.1 Mobilne tehnologije

Med mobilne tehnologije sodijo standardi, ki se uporabljajo v mobilnem svetu. Zraven sodijo tudi operacijski sistemi, ki s pomočjo dodatnih aplikacij

omogočajo, kar se da največji izkoristek naprave. Uporabili smo operacijski sistem Android, AChartEngine knjižnico za izrisovanje grafov, aplikacijo za skeniranje QR kod, Google Maps za risanje poti po zemljevidu in tehnologije NFC na mobilnih napravah za pridobivanje vhodnih podatkov.

### 3.1.1 Operacijski sistem Android

Android je mobilni operacijski sistem, ki temelji na Linux jedru, ki ga je razvil Google. Poganja ga JVM (angl. Java Virtual Mashine) in je najbolj razširjena neplačljiva mobilna platforma. Je operacijski sistem, ki deluje na večih jedrih in podpira delovanje več nitnih aplikacij. Vsebuje Google play trgovino v kateri lahko najdemo zastonske in plačljive aplikacije. Je zelo prilagodljiv, saj si lahko prilagodimo gumbe in domačo stran po svoji volji. Omogoča vključevanje gradnikov (angl. widgets), kot so spletni radio, vreme, zapiski (angl. notes), ipd. Google je razvil Android tako, da delujejo vse Googlove storitve z največjo možno učinkovitostjo (Google Maps, Gmail, ipd.). Za delovanje aplikacij operacijski sistem Android potrebuje Google Play Storitve, ki sedaj z novjšimi različicami pridejo že med namestitvijo operacijskega sistema. Aplikacije se nadgrajujejo preko Google Play trgovine s pomočjo Google Play storitve. Aplikacije, ki so plačljive se kupijo s pomočjo Google Checkout [22]. To je storitev, ki procesira plačevanje s kreditno kartico. Razvoj aplikacij je za razliko od drugih mobilnih operacijskih sistemov zelo enostaven in vsa orodja, ki jih ponuja Google so brezplačna [14].

### 3.1.2 AChartEngine

Je knjižnica grafikonov za operacijski sistem Android. Podpira Android OS 1.6 in novejši. Vsebuje naslednje vrste grafov: Črtni graf, območni graf, raztreseni graf, časovni graf, palični graf, tortni graf, mehurčkast graf, kolobarni graf, kubični črtni graf in kombinirani graf [1]. Z njeno uporabo je možno v aplikacijah dodajati grafične predstavitve različnih senzorskih in drugih

podatkov.

### 3.1.3 Aplikacija ZXing barcode scanner

ZXing (izgovori se 'zebra crossing') je odprtokodna aplikacija za več mobilnih operacijskih sistemov. Aplikacija s pomočjo kamere prebere in dešifrira Hitro odzivno kodo (angl. Quick Response Code). Uporabi se lahko za kodiranje in dekodiranje črtnih kod. Trenutno podpira naslednje formate: UPC-A in UPC-E, EAN-8 in EAN-13, Code 39, Code 93, Code 128, ITF, Codabar, RSS-14 (vse različice), RSS Expanded (skoraj vse različice), QR Code, Data Matrix, Aztec in PDF 417 [44]. Aplikacijo smo integrirali v našo aplikacijo in jo uporabljamo za branje QR kod produktov.

### 3.1.4 Google Maps API v2

Z Google Maps Android API v2 lahko dodajamo zemljevide v aplikacije, ki vsebujejo podatke iz Google Maps strežnikov, s katerimi API samodejno ureja komunikacijo. S pomočjo APIja lahko pridobimo podatke o lokacijah na poti, ki poteka od ene točke do druge točke. V njih lahko rišemo označevalce po zemljevidu. To so ikone, ki so zasidrane na določenih položajih na zemljevidu [15]. Za delovanje je bilo potrebno narediti posebej račun pri Google Consoli in vključiti storitev Google Maps Android API v2.

### 3.1.5 QR Code

QR kratica pomeni Hitri odziv (angl. Quick response). Prihajajo iz Japonske, kje so zelo razširjene. Lahko se jih prebere s kamero mobilne naprave. QR code na mobilnih napravah lahko shranjuje navaden tekst, spletni naslov, vnaprej pripravljen klic, vnaprej napisan sms, kontaktno kartico z že napisanimi podatki o neki osebi, ipd. QR koda je dvodimenzionalna in ima tri oporne kvadratke (v spodnjem levem kotu, v zgornjem levem kotu in v zgornjem desnem kotu) [43].

### 3.1.6 NFC in pametne naprave

Near Field Communication (NFC) predstavlja brezkontaktno komunikacijo med napravami. Podatki se prenašajo na kratke razdalje, največ do 10 cm. NFC je tehnologija z nizom standardov za prenosne naprave. Omogoča nam, da vzpostavimo 'peer-to-peer' (p2p) radijsko komunikacijo med dvema napravama ali med napravo in kartico ali značko NFC (angl. Tag), katera nam omogoča pretok informacij shranjenih v spominu. V NFC karticah se lahko shranjujejo navadno besedilo (angl. Plain text), spletni naslovi (URL - Uniform Resource Locator) in drugo [42]. Običajno so ti podatki uporabljeni za identifikacijo produktov ali oseb, ali pa za povezavo na spletni naslov, ki omogoča prikaz informacij v brskalniku na pametnih NFC napravah.

Pametne naprave so vsak dan bolj priljubljene pri vseh generacijah. S pomočjo spletnih trgovin (npr. Google play trgovina ali Applov AppStore) so pametne naprave postale še pametnejše in so že skoraj boljše in bolj natančne kot kakšna orodja, ki jih uporabljamo pri hišnih ali katerih drugih opravilih (npr. s pomočjo kamere lahko izmerimo dolžino dveh točk, uporabimo napravo kot vodno tehtnico ali kot kompas). Naprave, ki vsebujejo tehnologijo NFC pa lahko uporabljajo storitev Google Wallet in takojšnje plačevanje z tehnologijo NFC.

## 3.2 Spletne tehnologije

Zbirko tehnologij, standardov, programskih in opisnih jezikov ter orodij za razvoj spletnih strani in storitev imenujemo Spletne tehnologije. Pri izdelavi spletnega dela je bila uporabljena RESTful spletna storitev, JSON in XML. V Android aplikaciji pa so bili uporabljeni HTML in CSS za urejanje besedila.

### 3.2.1 RESTful spletna storitev

Je način prenosa podatkov med strežnikom in odjemalcem. REST (angl. Representational State Transfer) [32, 33, 35] je tehnologija, ki ni odvisna od

programskega jezika, temveč temelji na HTTP (angl. HyperText Transfer Protocol) metodah, vendar ni omejena samo na njih. Vhodne podatke dobi z uporabo naslova URL, izhodne podatke pa vrne v obliki JSON ali XML (po potrebi vrne podatke tudi v ostalih oblikah kot so: XHTML (angl. Extensible HyperText Markup Language), navaden tekst, ipd.). Uporaba HTTP metod po načelih RESTful:

- GET – se uporablja za pridobivanje podatkov ali opravi poizvedbo o viru,
- POST – se uporablja za ustvarjanje novega vira,
- PUT – se uporablja za posodobitev obstoječih virov ali podatkov in
- DELETE – se uporablja za odstranitev vira ali podatkov.

V spletni aplikaciji smo uporabili samo GET metodo. Z njo smo pridobivali podatke iz podatkovnih baz na strežniku.

### 3.2.2 JSON in XML

JavaScript Object Notation (JSON) je sintaksa za shranjevanje in izmenjavo tekstovne informacije. Je enostaven format za izmenjavo besedilnih podatkov, je samo opisni in razumljiv. Je neodvisen od programskih jezikov, ampak uporablja JavaScript sintakso za opisovanje objektov [40]. Preko JSON smo pridobivali podatke o poti med dvema točkama na zemljevidu iz Googlovih strežnikov.

Extensible Markup Language (XML) je bil narejen za strukturiranje, shranjevanje in prenos informacij. Objekte in informacije ima shranjene v oznakah (angl. Tag). Oznake v XML niso nikjer definirane, ampak si jih sami lahko definiramo, ko izdelujemo XML za naše potrebe oz. informacije [23]. Razlika med JSON in XML je, da JSON vsebuje manj znakov pri strukturiranju istih podatkov in je lažje berljiv za stroj, XML pa je lažje berljiv za človeka. XML smo uporabili za prenos podatkov dobljenih iz RESTful spletne storitve.

### 3.2.3 HTML in CSS

HyperText Markup Language (HTML) je označevalni jezik za opisovanje spletnih strani. HTML dokumenti vsebujejo HTML oznake (angl. Tags) in navadno besedilo (angl. Plain text) [20]. Za urejanje golega besedila v HTML dokumentu uporabimo preprosti slogovni jezik Cascading Style Sheets (CSS). S katerim oblikujemo velikost pisave, barvo pisave, barvo ozadja, ipd. CSS je bil narejen z namenom, da olajša spletnim programerjem oblikovanje HTML dokumentov [7].

HTML in CSS smo uporabili v razvoju mobilne aplikacije pri grafičnem vmesniku, saj nam tisti elementi niso ponujali obojestranskega poravnanja in ga je bilo potrebno omogočiti s pomočjo HTML in CSS.

## 3.3 Programski jeziki in orodja

Za programiranje Android aplikacije smo uporabili programsko okolje Eclipse in programski jezik Java. Pri shranjevanju podatkov smo si pomagali s podatkovnima bazama MySQL za shranjevanje podatkov o sledljivosti in hladni verigi in SQLite za shranjevanje zgodovine lokalno na mobilni napravi. Za spletni strežnik smo uporabili Oracle GlassFish strežnik.

### 3.3.1 Eclipse

Je programsko razvojno okolje za pisanje namiznih, mobilnih in spletnih aplikacij v različnih programskih jezikih. Narejen je v programskem jeziku Java, zato deluje na vseh operacijskih sistemih. Da omogočimo pisanje v drugih programskih in opisnih jezikih je potrebno namestiti določene dodatke [26]. Za pisanje programov za OS Android smo morali namestiti dodatek ADT (ADT - Android Developer Tools), ki zagotavlja nabor orodij, ki so povezana z Eclipse IDE (IDE - Integrated development environment). To nam ponuja dostop do številnih funkcij s katerimi si pomagamo pri razvijanju poljubne Android aplikacije. Zagotavlja grafični uporabniški vmesnik (angl.

Graphic User Interface) za dostop do številnih orodij v kompletu za razvijalca programske opreme (SDK - Software Development Kit) [4].

### 3.3.2 Oracle GlassFish

Je brezplačna programska oprema razvita v podjetju Oracle Corporation. To je nadgradnja odprtokodnega projekta GlassFish prvega aplikacijskega strežnika za platformo Java EE, ki ga je začel Sun Microsystems. Deluje na vseh operacijskih sistemih [11, 28].

### 3.3.3 Podatkovne baze

MySQL je najbolj popularna odprtokodna podatkovna baza. Uporablja se za shranjevanje večjih količin kategoričnih podatkov po tabelah. Tabela je zbirka povezanih podatkov in je sestavljena iz atributov (stolpcev) in vrstic. Vsak atribut je neka lastnost podatka, ki ga imamo shranjenega po vrsticah v tabeli. Podatke pridobimo s pomočjo poizvedb, ki se obnašajo kot neka vprašanja oz. zahteve. V ta namen sta uporabljena ukaz SELECT in pogoj, ki ga navedemo. Pridobljene podatke lahko razporedimo, lahko jih omejimo glede na to koliko jih želimo uporabiti in še drugo. Če želimo naknadno vstavljati podatke v tabele, si pomagamo z ukazom INSERT INTO. S tem ukazom vstavimo v tabelo nov vnos oz. naredimo novo vrstico. Za spremembo že vpisanega določenega vnosa uporabimo ukaz UPDATE ime\_tabele, za izbris pa uporabimo ukaz DELETE FROM ime\_tabele. Če želimo izbrisati celo tabelo napišemo ukaz DROP TABLE [36].

SQLite je okrnjena različica MySQL, kar pomeni, da je narejena za shranjevanje manjših količin podatkov in da ima manj ukazov. V operacijskem sistemu Android pridobivamo podatke v aplikaciji iz podatkovne baze SQLite s pomočjo Cursorja [37]. Cursor je ene vrste gonilnik, ki skrbi za vrnjene podrazrede razreda Cursor, ki se zgodijo pri poizvedbi podatkovne baze [2].

### 3.3.4 Programski jezik Java

Programski jezik Java se je razvil iz jezika Oak, katerega so razvili v začetku devetdesetih pri podjetju Sun Microsystems kot prvega, ki je bil neodvisen od platforme oz. operacijskega sistema. Nekaj lastnosti Jave:

- Neodvisnost od platforme – Java prevajalniki ne proizvajajo izvirnega objekta kode za določeno platformo, temveč interpretirajo v "Java byte-code" navodila (to so navodila, ki jih razume JVM) za Javin navidezni stroj (JVM – Java Virtual Mashine),
- objektno orientirana – kar pomeni, da je vse v Javi objektno,
- sintaksa je podobna C++ in C#,
- vsebuje funkcijo za zbiranje smeti (angl. Garbage collection), katera sama počisti spomin (RAM).

Java je visokonivojski programski jezik in sodi v tretjo generacijo programskih jezikov. Obstaja 5 razredov [24, 38]. Javo smo uporabljali kot osnovni programski jezik pri izdelavi mobilne aplikacije in pri izdelavi RESTful spletne storitve.

## Poglavje 4

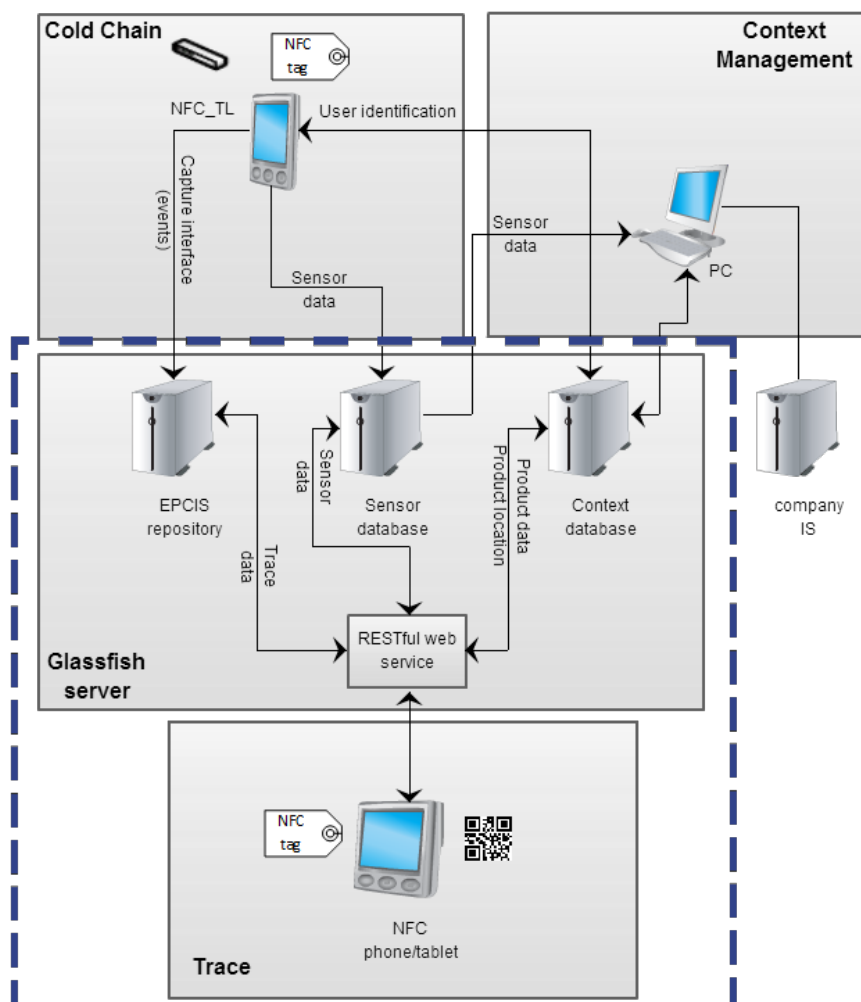
# Mobilna aplikacija TColdChain

TColdChain je aplikacija za mobilne naprave z operacijskim sistemom Android. Vključena je v sistem, ki podjetju omogoča izvedbo hladne verige s tehnologijo NFC. Za popolno delovanje aplikacije potrebujemo mobilni ali internetni dostop do strežnika. Z aplikacijo lahko skeniramo QR kodo ali prislonimo mobilno napravo k znački NFC. Omogoča nam pregled podatkov o sledljivosti v celotni prehranski verigi in pregled ravnanja s hitro pokvarljivimi živili, če so bile v logističnih procesih izvedene meritve zahtevanih temperatur.

### 4.1 Sistem za izvedbo hladne verige

Arhitektura sistema za izvedbo hladne verige je predstavljena na sliki 4.1. Sestavljajo ga tri aplikacije: (i) Context Management; (ii) Cold Chain; in (iii) Trace. Podatkovne baze, ki vsebujejo podatke o ravnanju z živili v preskrbovalni verigi (angl. EPCIS repository), temperature (angl. Sensor database) in informacije o podjetju (angl. Context database) so shranjene na strežniku GlassFish.

Aplikacija 'Context Management' je zasnovana za vnos, urejanje in shranjevanje podatkov o produktu, podjetju in lokacijah. Je spletna aplikacija, ki omogoča upravljanje identifikatorjev in podatkov o hladni verigi. Služi



Slika 4.1: Arhitektura sistema za izvedbo hladne verige z uporabo tehnologije NFC.

kot vmesnik za vnos podatkov, spreminjanje podatkov v podatkovni bazi in omogoča pregled nad trenutnim dogajanjem [8]. Spletna stran 'Context Management' uporablja kontekstno podatkovno bazo. Podatki, ki se v njej shranjujejo, so imena poslovnih korakov, dispozicij, podatki o produktih, podrobnosti lokacij in podatki o podjetjih, ki so neposredno vključeni v proizvod oz. procesiranje produktov.

'Cold Chain' – je mobilna aplikacija za operacijski sistem Android.

Omogoča upravljanje in nadzor temperaturnih meritev z uporabo pametnih kartic s temperaturnimi senzorji (NFC-TL) ter njihovo shranjevanje v senzorski podatovni bazi. Za uporabo aplikacije se moramo identificirati preko kartice NFC. To naredimo tako, da prislonimo kartico k napravi (pametni telefon), katera preveri našo avtentičnost v bazi uporabnikov. Shranjuje podatke o temperaturah, ki so shranjene na pametnih karticah, začetno lokacijo meritve, končno lokacijo meritve, čas začetka meritve in čas konca meritve ter kje je ta meritev potekala (ali je to bilo na paleti, tovornjaku, vlaku, ipd.) v senzorsko podatkovno bazo. Podatke o vmesnih lokacijah, procesih, ki so bili izvedeni nad produktom pa shrani v podatkovno bazo EPCIS. Senzorska baza vsebuje podatke o temperaturah, zemljepisne dolžine in širine (začetek meritve in konec meritve), interval med meritvami v sekundah, zgornjo dovoljeno mejo, spodnjo dovoljeno mejo, začetek meritve in konec meritve.

'Trace' je rešitev, ki omogoča predstavitev podatkov izvajalcem preskrbovalne verige, kakor tudi potrošnikom in je predstavljena v diplomski nalogi. Aplikacija, ki deluje na operacijskem sistemu Android združi podatke iz treh predstavljenih podatkovnih baz na enem mestu in nam prikaže vse pomembne in zanimive podatke o skeniranem produktu in podjetju, kot so:

- kratek opis produkta s podatki o hranilni vrednosti,
- datum poteka ali rok uporabnosti,
- prikaz temperatur na grafu, katerim je bil produkt izpostavljen med pripravo, prevozom in shranjevanjem,
- izvor in pot produkta na zemljevidu in
- časovni prikaz procesov v preskrbovalni verigi.

Iz podatkovne baze EPCIS so uporabljeni podatki o sledenju produkta, procesih, dispozicijah in hladni verigi v preskrbovalni verigi. Podatki, ki jih pridobi iz senzorske baze, so temperaturne meritve, katere prikaže na grafu. EPCISove podatke uporabi za ključ v kontekstni bazi in pridobi podrobne podatke o lokacijah, podjetju in produktu.

## 4.2 Načrtovanje aplikacije

Aplikacija potrebuje dostop do podatkov na strežniku. V primeru, da ni možna povezava v mobilno ali internetno omrežje, je realizirana funkcija, ki shrani EPC (angl. Electronic Product Code) podatek o produktu v lokalni napravi. Ko ima uporabnik dostop do spleta, si lahko ogleda skeniran produkt.

Razvoj aplikacije je zahteval izvedbo v dveh delih. Prvi del je strežniški del, kateri teče na GlassFish strežniku in ima podatke shranjene v relacijski podatkovni bazi MySQL. Drugi del je Android aplikacija, katera prikazuje podatke v uporabniku razumljivi obliki. Strežnik in aplikacija komunicirata preko RESTful spletne storitve.

Načrtovanje rešitve je potekalo v štirih fazah. V prvi fazi smo preučevali označevanje in strukturo podatkov in zapisov v datotekah XML, kateri so bili definirani in uporabljeni za izvedbo pilotne rešitve sistema sledljivosti v projektu RFID from Farm to Fork [30]. Preverili smo katere informacije so nam na voljo v podatkovni bazi EPCIS, senzorski bazi in kontekstni bazi. Iz njih smo izbrali uporabne tabele in sestavili relacijsko shemo, ki je predstavljena na sliki 4.3.

V nadaljevanju smo se odločili, da naredimo RESTful spletno storitev v programskem jeziku Java (aplikacija za strežnik Glassfish), katera nam bo vračala dva različna zapisa v XMLju. Takšno rešitev smo izbrali zaradi varnosti saj ni dobro, če aplikacija dostopa neposredno do strežnika in podatkovnih baz. Uporabljeni podatki so dinamični. To pomeni, da lahko za posamezne faze podatki o sledenju ali o hladni verigi manjkajo. V ta namen smo sestavili več SQL poizvedb, ki se zgodijo samo ob določenih primerih in kombinirajo podatke iz treh podatkovnih baz.

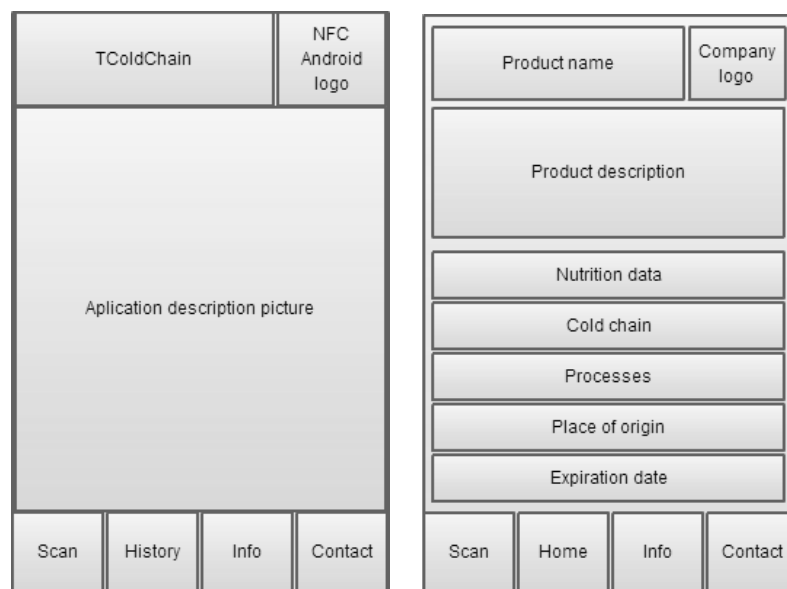
Tretji del je vključeval komunikacijo mobilne aplikacije z RESTful spletno storitvijo in način shranjevanja pridobljenih podatkov za nadaljno uporabo v aplikaciji. Odločili smo se, da podatke, ki jih bomo dobili kot XML shranimo objektno po razredih, saj je potem uporaba precej lažja.

Zadnja faza je predstavljala izdelavo grafičnega vmesnika za pametne na-

prave. Razdeljen je na več nivojev oz. na več strani in podstrani. Prvi nivo je domača stran oz. uvodna stran aplikacije (slika 4.2(a)). Na vrhu je predvidena slika z logotipom aplikacije, na sredini bo slika, katere naloga bo prikazati funkcionalnost aplikacije in spodaj meni (Sken, Zgodovina, Info, Kartica). Drugi nivo omogoča uporabo vseh funkcionalnosti aplikacije. To so: (i) aplikacija ZXing barcode scanner, (ii) prikaz zgodovine, (iii) opis produkta, (iv) informacije in (v) kartica. V tretji nivo spadajo prikaz temperatur (hladna veriga - graf), sledljivost (prikaz procesov – slika 4.2(c)) in lokacije (Google maps). Strani, ki ne vsebujejo grafa bodo zaklenjene v portretni orientaciji. To pomeni, da se aplikacija ne bo vrtela skupaj z napravo. Stran, ki prikazuje graf bo zaklenjena v "landscape" oz. ležeči orientaciji (napravo obrnemo za  $90^\circ$ ). Za ležečo orientacijo smo se odločili zaradi tega, ker imajo naprave v portretni orientaciji ožji zaslon, mi pa uporabljamo časovni graf, kateri potrebuje nekaj več prostora po x osi. Aplikacija vsebuje devet različnih strani. Stran na kateri se prikaže opis produkta bo prilagodljiva podatkom, ki jih bomo dobili iz strežnika. Skica strani na sliki 4.2(b) prikazuje opis produkta in je razdeljena v več področij, ki so izvedena z gumbi. Omogočajo odpiranje naslednjega nivoja. Če bodo manjkali podatki za:

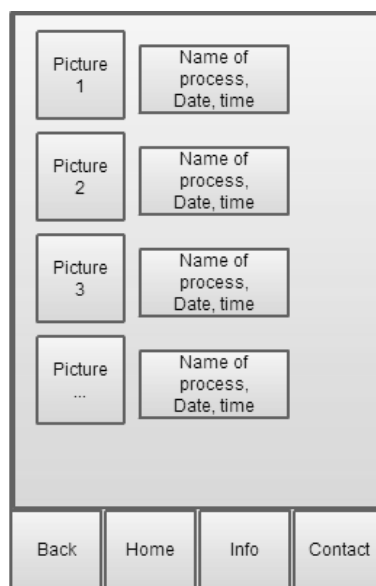
- hladno verigo – se gumb "Cold chain" ne bo prikazal in graf ne bo na voljo,
- procese – se gumb "Processes", "Place of origin" in "Expiration date" ne bodo prikazali in bodo na voljo samo gumb za podatke o hranilni vrednosti in hladni verigi.

V spodnjem delu aplikacije se bo nahajal meni, ki bo prisoten v vseh straneh razen pri prikazu hladne verige (graf). Meni bo vseboval gumbe kot so skeniraj, kateri odpre aplikacijo barcode scanner s katerim skeniramo QR kodo, prikaži zgodovino, prikaži informacije o aplikaciji, pošlji elektronsko pošto (kartica), gumb domov, katerega naloga nas je vrniti na uvodno stran, gumb nazaj, kateri nas popelje 1 nivo nazaj oz. na prejšnjo stran (slika 4.2(c)).



(a)

(b)



(c)

Slika 4.2: Skica načrtov pri načrtovanju grafičnega vmesnika za:  
(a) prvo stran; (b) stran opisa produktov; (c) stran prikaza procesov.

## 4.3 Strežnik

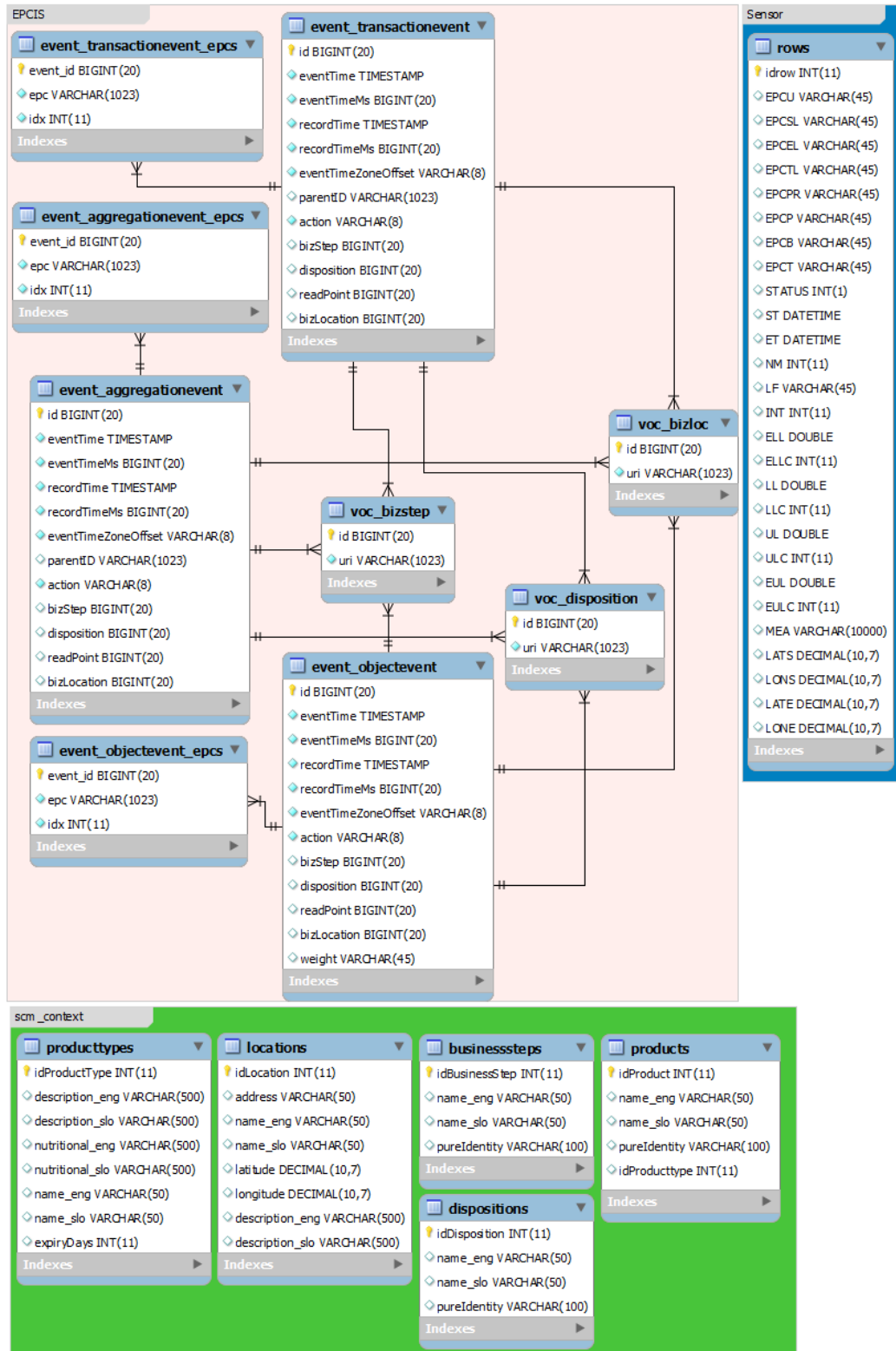
Podatkovne baze in RESTful spletna storitev delujejo na strežniku GlassFish. Mobilna aplikacija bo s klicem storitve posredovala informacijo o skeniranem produktu, izvedla potrebne poizvedbe v podatkovnih bazah in vrnila rezultate v strukturi XML.

### 4.3.1 Podatkovne baze

Imamo tri različne podatkovne baze, kjer se nahajajo podatki o sledljivosti v prehranski verigi (farma, priprava, transport, skladiščenje, trgovina ali kar končni kupec), podatki o hladni verigi, podatki o lokaciji produkta in informacije o podjetju. Slika 4.3 prikazuje relacijsko shemo tabel, ki se uporabljajo v mobilni aplikaciji.

Iz podatkovne baze EPCIS pridobimo vse potrebne podatke v treh korakih:

- **Korak 1:** Iz tabel "event\_aggregationevent" in "event\_aggregation-event\_epcs" pridobimo podatke, kdaj se je dogodek začel (eventTime, ko je action ADD), kdaj končal (eventTime, ko je action DELETE) in s katerim sredstvom je bila temperatura merjena (parentID oz. epcTL). V tabeli "event\_aggregationevent" se nahajajo zapisi o dogodkih sledenja, kje in kdaj so se začeli in končali. V "event\_aggregationevent\_epcs" pa se nahajajo oznake produkta s serijsko številko (v nadaljevanju EPC – Electronic Product Code), katere potem uporabimo kot ključ za pridobitev podatkov o sledenju produkta.
- **Korak 2:** Pridobitev podatkov o hladni verigi, če le te obstajajo. Atributi katere potrebujemo za hladno verigo so:
  - "eventTime" – čas dogodka,
  - "voc.bizstep.uri" – podatek je del spletnega naslova in
  - "voc.disposition.uri" – podatek shranjen kot del spletnega naslova.



Slika 4.3: Relacijska shema tabel iz podatkovnih baz.

Te podatke dobimo iz tabel:

- "epcis.event\_objectevent" – podatki o sledenju produkta v posameznih fazah,
- "epcis.event\_objectevent\_epcs" – vsebuje vse EPCje produktov, kateri imajo shranjene podatke o sledenju,
- "epcis.voc\_bizstep" – shranjeni vsi poslovni koraki,
- "epcis.voc\_disposition" – shranjene dispozicije, to so vsa stanja produkta v povezavi z EPC.

Dele spletnega naslova, ki jih dobimo iz "voc\_bizstep.uri" in "voc\_disposition.uri" uporabimo kot IDje v kontekstni bazi za pridobitev imena poslovnih korakov in dispozicij (bolj natančno iz tabele "scm\_context.dispositions" in "scm\_context").

- **Korak 3:** Za pridobivanje podatkov o sledljivosti sta na sliki 4.4 prikazani dve SQL poizvedbi. Tukaj uporabimo attribute iz hladne verige (Korak 2) in dodamo nov atribut "voc\_bizloc.uri" (shranjen kot del spletne povezave), ki je dodan v poizvedbo. Za pridobitev ustreznih podatkov je potrebno narediti unijo med dvema SQL poizvedbama. Prva je ista kot pri hladni verigi, druga pa vsebuje attribute iz tabel:

- "event\_transactionevent" – podatki o transakcijah oz. o procesih,
- "event\_transactionevent\_epcs" – vsebuje EPCje produktov, kateri so bili prisotni pri večih procesih,
- "voc\_bizloc" – vsebuje IDje lokacij.

ID lokacije iz atributa "voc\_bizloc.uri" uporabimo za pridobitev podatkov o lokaciji dogodka iz kontekstne baze, ki se nahajajo v tabeli "scm\_context.locations".

Podatke o temperaturah, ki so shranjene v senzorski bazi, pridobimo na podlagi zapisov o tem, kdaj se je dogodek začel, kdaj končal in s katerim

```

(SELECT voc_disposition.uri as disp, voc_bizstep.uri as bizs, eventTime, voc_bizloc.uri as loc
FROM epcis.event_objectevent, epcis.event_objectevent_epcs, epcis.voc_bizstep, epcis.voc_disposition, epcis.voc_bizloc
WHERE epcis.event_objectevent.id = epcis.event_objectevent_epcs.event_id
      AND epcis.event_objectevent.bizStep = epcis.voc_bizstep.id
      AND epcis.event_objectevent.disposition = epcis.voc_disposition.id
      AND epcis.event_objectevent.bizLocation = epcis.voc_bizloc.id
      AND epcis.event_objectevent_epcs.epc = 'http://wmfId.org/farm2fork/383004776/product/0042/1006268')
UNION (SELECT voc_disposition.uri as disp, voc_bizstep.uri as bizs, eventTime, voc_bizloc.uri as loc
FROM epcis.event_transactionevent, epcis.event_transactionevent_epcs,
      epcis.voc_bizstep, epcis.voc_disposition, epcis.voc_bizloc
WHERE
      epcis.event_transactionevent.id = epcis.event_transactionevent_epcs.event_id
      AND epcis.event_transactionevent.bizStep = epcis.voc_bizstep.id
      AND epcis.event_transactionevent.disposition = epcis.voc_disposition.id
      AND epcis.event_transactionevent.bizLocation = epcis.voc_bizloc.id
      AND epcis.event_transactionevent_epcs.epc = 'http://wmfId.org/farm2fork/383004776/product/0042/1006268')
ORDER BY eventTime ASC;

SELECT voc_disposition.uri as disp, voc_bizstep.uri as bizs, eventTime
FROM epcis.event_objectevent, epcis.event_objectevent_epcs, epcis.voc_bizstep, epcis.voc_disposition
WHERE epcis.event_objectevent.id = epcis.event_objectevent_epcs.event_id
      AND epcis.event_objectevent.bizStep = epcis.voc_bizstep.id
      AND epcis.event_objectevent.disposition = epcis.voc_disposition.id
      AND bizstep = 16
      AND eventTime LIKE '2013-02-28%'
LIMIT 1;

```

Slika 4.4: Dve SQL poizvedbi. Prva pridobi podatke o sledljivosti za določen produkt. Druga pa dobi podatke o lokaciji, kje se je kakšen proces zgodil.

sredstvom je bil merjen. Podati moramo še serijsko številko, ID produkta in ID proizvajalca. Poleg temperatur dobimo še podatke o tem, kje se je začelo merjenje in kje končalo. Te podatke uporabimo v primeru, da sledljivost o nekem produktu ni zabeležena v EPCIS repozitoriju.

### 4.3.2 RESTful spletna storitev

Ta storitev služi kot vmesnik za komunikacijo med aplikacijo in podatkovnimi bazami. Napisana je bila v Javi z JAX-RS. JAX-RS je kratica za Java API za RESTful spletno storitev [25]. RESTful spletna storitev (tracecc) se zažene, ko aplikacija pošlje HTTP zahtevo na strežnik, naslov mora vsebovati: protokol HTTP, naslov strežnika, ime storitve (tracecc) in ime razreda (trace) zglada tako: "http://naslov\_streznika/tracecc/trace/". Vsebuje dve metodi, katere se odzivajo na GET zahteve. Pred deklaracijo metode vsebujejo JAX-RS značke @GET, @Path("path") in @Produces(application/xml).

Prva metoda se odzove na @GET zahtevek, če spletni naslov ustreza naslovu v @Path("manID:[0-9]\*/productprodID:[0-9]\*/boxID:[0-9]\*/language:[0-9]"). Atributi, ki se jih pošlje preko naslova so številka proizvajalca (manID), številka produkta (prodID), serijska številka (boxID) in številka jezika (language=2 je za Slovenski jezik, 1 ali karkoli drugega je za Angleški

```

<epcis name="Seabass - cleaned 400-600g">
  <Expiration displayUntil="2013-03-10" useBy="2013-03-10"/>
  <Coldchain maxLimit="10.0" minLimit="0.0" start="2013-02-28 13:10:23.0" bizstep="Packed"
  disposition="Ready for transport" sensorUri="sensors/Temperature/383004776/0042/1006268/2013-02-
  28+13:10:23/2013-02-28+14:06:10/.0?uri=383004776/asset/004/1"/>
  <Coldchain maxLimit="4.0" minLimit="0.0" start="2013-02-28 14:06:10.0" bizstep="Received at coldstore"
  disposition="Coldstore" sensorUri="sensors/Temperature/383004776/0042/1006268/2013-02-
  28+14:06:10/2013-03-01+05:56:06/.0?uri=383004776/asset/004/1"/>
  <Coldchain maxLimit="4.0" minLimit="0.0" start="2013-03-01 05:56:06.0" bizstep="Shipped from
  coldstore" disposition="Transport" sensorUri="sensors/Temperature/383004776/0042/1006268/2013-03-
  01+05:56:06/2013-03-01+09:55:44/.0?uri=383004776/asset/004/1"/>
  <Process at="2013-02-28 11:36:29.0" bizstep="Received from sea" disposition="Stored for processing"/>
  <Process at="2013-02-28 13:01:08.0" bizstep="Packed" disposition="Ready for transport"/>
  <Process at="2013-02-28 13:22:34.0" bizstep="Shipped from sorting room" disposition="Transport"/>
  <Process at="2013-02-28 14:06:10.0" bizstep="Received at coldstore" disposition="Coldstore"/>
  <Process at="2013-03-01 05:56:06.0" bizstep="Shipped from coldstore" disposition="Transport"/>
  <Process at="2013-03-01 09:56:42.0" bizstep="Received by customer" disposition="Sold"/>
  <PointOfInterest address="Seča" name="Fish farm" latitude="45.4876120" longitude="13.5789010"
  description="Fish farm in Piran bay"/>
  <PointOfInterest address="Seča" name="Sorting room" latitude="45.4997010" longitude="13.5886010"
  description="Sorting room in Seča"/>
  <PointOfInterest address="Lucija" name="Cold store" latitude="45.5396630" longitude="13.6622710"
  description="Cold store in Lucija"/>
  <PointOfInterest address="Ljubljana" name="Ljubljana-LRA" latitude="46.0446520" longitude="14.4890870"
  description="UL, FRI (LRA) - Private customer"/>
</epcis>

```

Slika 4.5: Primer XMLja epcis za produkt z oznako 0042.

jezik). Zapis z oglatimi oklepaji in številkami v njem se imenuje Regular expressions (regex). To je poseben niz, ki opisuje iskalni vzorec [31]. Ta nam pomaga pri filtriranju pravih vzorcev naslova. Metoda preko SQL poizvedb (naštete v poglavju 4.3.1) poišče potrebne podatke in jih vrne v XML obliki. Obstajajo tri različne variante, ki so odvisne od zapisov za posamezen produkt v bazi EPCIS:

- Obstajajo vsi podatki za sledljivost in hladno verigo v podatkovni bazi EPCIS. Izgled zapisov v XML je prikazan na sliki 4.5.
- Obstajajo samo podatki o sledljivosti produkta.
- Obstajajo podatki za celotno hladno verigo.
- Nepopolni podatki o procesih in/ali meritvah.

Klic druge metode se izvede v primeru, da spletni naslov ustreza naslovu v @Path(sensors/Temperature/manID/prodID/boxID/startTime/endTime/-). Vse attribute pretvorimo v spremenljivke s funkcijo @PathParam(). Pomen atributov: manID - številka proizvajalca, prodID - številka produkta, startTime - začetni čas, endTime - končni čas. Zadnji del naslova dobimo s

```

<Sensor end="2013-02-28 14:06:10" readings="10" start="2013-02-28
13:10:23" lats="45.4997010" lons="13.5886010" late="46.0446520"
lone="14.4890870" lf="Dense">
  <Temperature at="2013-02-28 13:10:23.0" temp="19.89"/>
  <Temperature at="2013-02-28 13:16:23" temp="13.42"/>
  <Temperature at="2013-02-28 13:22:23" temp="11.81"/>
  <Temperature at="2013-02-28 13:28:23" temp="7.5"/>
  <Temperature at="2013-02-28 13:34:23" temp="4.98"/>
  <Temperature at="2013-02-28 13:40:23" temp="3.73"/>
  <Temperature at="2013-02-28 13:46:23" temp="3.19"/>
  <Temperature at="2013-02-28 13:52:23" temp="2.83"/>
  <Temperature at="2013-02-28 13:58:23" temp="2.65"/>
  <Temperature at="2013-02-28 14:04:23" temp="2.11"/>
</Sensor>

```

Slika 4.6: Primer XMLja Sensor za produkt z oznako 0042.

pomočjo @QueryParam("uri") in vsebuje manID - številka proizvajalca, assetID - številka sredstva in log\_ID - številka zapisovalnika. V senzorski podatkovni bazi imamo shranjene temperature v eni spremenljivki, podani pa so še interval meritev, začetni čas in atribut za tip meritev (angl. Dense, Out of limits and Limit crossing). Tako, da z SQL poizvedbo dobimo vse temperature, za katere izračunamo čas shranjevanja meritve na podlagi začetnega časa in intervala meritev (številka temperature v zaporedju \* interval + začetni čas). Izgled drugega oz. senzorskega XMLja je prikazan na sliki 4.6.

Namestitev spletne storitve na spletni strežnik poteka preko grafičnega vmesnika imenovanega "asadmin" (lahko tudi preko terminala). Za namestitev spletne storitve potrebujemo .war datoteko, ki je arhivska datoteka (kot .jar za Javo) za Glassfish strežnik. War datoteko kreiramo tako, da izvirno kodo iz datoteke .java prevedemo v .class datoteko. Ko imamo prevedeno .class datoteko jo skupaj s potrebnimi knjižnicami (.jar) stisnemo v .war arhiv in nato namestimo na spletnemu strežniku, kot aplikacijo.

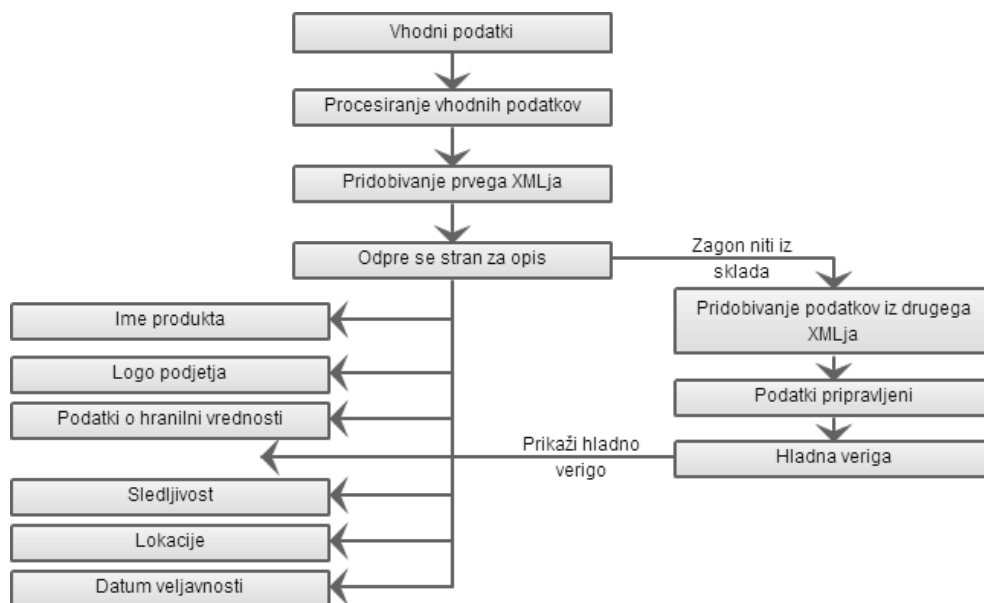
## 4.4 Delovanje Android aplikacije

Android aplikacija, ki smo jo poimenovali TColdChain, je več nitna aplikacija in deluje na operacijskem sistemu Android. Podpira vmesnik Android API 8 in novejši [9]. Aplikacijo namestimo na napravo tako, da si posnamemo namestitveno datoteko TColdChain.apk na napravo in jo namestimo.

TColdChain lahko sprejme vhodne informacije preko QR kode ali preko NFC oznak oz. kartic, če ima naprava vgrajeno dodatno funkcionalnost, ki podpira uporabo tehnologije NFC. Vhodna informacija je podana kot spletni naslov <http://wmrfid.org/farm2fork/trace.html?uri=tag/300D6D4306800A80000F5ABC>, ki je bil definiran v projektu RFID-F2F [30] za pregled informacij z uporabo spletnega brskalnika. Na osnovi delujoče pilotne izvedbe sistema sledljivosti rib je bilo uporabljeno pridobivanje podatkov o sledljivosti tudi za razvoj in testiranje naše aplikacije. Za njeno delovanje je pomemben zapis, ki se začne od "tag/" naprej (300D6D4306800A80000F5ABC) in predstavlja unikatno oznako produkta v preskrbovalni verigi. To je koda EPC. QR Code ali NFC oznaka lahko vsebujejo celoten spletni naslov ali pa samo EPC, ki je dolg 24 znakov v šestnajstiškem zapisu in lahko vsebuje številke od 0 do 9 in črke od A do F. Pridobivanje oznak je izvedeno na dva načina:

- Aplikacija zazna NFC oznake, če imajo za mime vrsto navadno besedilo (angl. `mimetype text/plain`) ali shemo HTTP. Mime vrsta (angl. `mimetype`) je oznaka za določeno vrsto podatkov, tako da programska oprema ve, kako naprej s podatki [41]. Shema HTTP pa deluje kot filter, če kartica vsebuje v oznaki (angl. `Tag`) HTTP in podan strežnik, ki je v našem primeru "wmrfid.org", potem odpre našo aplikacijo.
- QR kodo preberemo s pomočjo programa ZXing barcode scanner. Ta integracija potrebuje dodatni program "barcode scanner", kateri nam pomaga prebrati QR kodo. Če programa nimamo nameščenega na napravi in hočemo skenirati QR kodo (v naši aplikaciji z gumbom Scan) dobimo obvestilo, da je potrebno ta program namestiti. Pojavi se pojavno okno in nas vpraša, če ga želimo namestiti.

Koda EPC ustreza standardu GS1, ki za označevanje produktov uporablja oznako SGTIN (angl. `Serialized Global Trade Item Number`). Vsebuje glavo, ki je velika 8 bitov, filter 3 bite, particija 3 bite, številka proizvajalca 30 bitov, številka produkta 14 bitov in serijska številka produkta, ki je velika 41 bitov [17]. Ko aplikacija dobi vhodno informacijo EPC, jo iz šestnajstiškega



Slika 4.7: Prikaz sočasnega izvajanja pridobivanja podatkov iz senzorskega oz. drugega XMLja.

številkega sistema pretvorimo v binarni številski sistem. Nato vse 4-bitne zapise združimo v niz in ga nato razbijemo po delih, da dobimo številko proizvajalca (GS1 oznaka, ki jo ima podjetje, ki je vključeno v prehransko verigo), številko produkta (označuje tip produkta) in serijsko številko produkta v desetiškem številskem sistemu. Preden sestavimo spletni naslov za željeni produkt potrebujemo še informacijo o jeziku, katerega uporabnik uporablja na napravi. Spletni naslov, ki je potreben za pridobivanje podatkov je sestavljen v naslednji obliki: "http://naslovStreznika/tracecc/trace/številka-proizvajalca/številka-produkta/serijska-številka/jezik". Pridobljen epcis oz. prvi XML razdelimo po razredih in podrazdredih. Podrazredi, kateri potrebujejo še dodatne podatke iz senzorskega oz. drugega XMLja se dodajo v sklad kot niti. Ko si uporabnik ogleduje opis produkta, se niti iz sklada začnejo zaganjati in sočasno prenašajo podatke iz strežnika ne da bi uporabnik kaj čakal. Sočasno delovanje je prikazano na sliki 4.7.

# Poglavje 5

## Opis in testiranje aplikacije

Grafični del aplikacije je razdeljen na več nivojev in dejavnosti. Ker so podatki dinamični se v vsaki dejavnosti posebej izvede logika za prikaz elementov na zaslon. Testiranje pa je potekalo na večih različnih pametnih telefonih, tablicah in na virtualnih napravah.

### 5.1 Grafični vmesnik

Sestavljen je iz devetih dejavnosti, nekatere so med seboj povezane in lahko uporabnik dostopa do njih preko menija, ki ga ima pri skoraj vsaki dejavnosti v spodnjem delu zaslona. Grafični vmesnik je razdeljen na funkcionalnosti aplikacije in dejavnosti sledljivosti. Aplikacija ima naslednje funkcionalnosti:

- Vstopno stran, ki se nam prikaže ob vstopu v aplikacijo.
- Prikaz zgodovine, katera nam omogoči vpogled v že pregledane produkte ali pa samo skenirane in še ne ogledane.
- Skeniranje s pomočjo programa barcode scanner, kateri aplikaciji kot rezultat pošlje vrednost QR kode.
- Kartica, katera odpre privzeti program (če je izbran, drugače izbiramo med večimi programi) za pošiljanje elektronske pošte. Preko katerega pošljemo vprašanje administratorju.

- Informacije, ki služijo kot info točka. Je lahko dostopna stran, ker se nahaja pri vseh straneh z menijem.

Dejavnosti za prikaz sledljivosti vključujejo:

- Opis produkta – to je stran na kateri je opisan produkt, na kateri so podatki o hranilni vrednosti, datum veljavnosti in gumbi za hladno verigo, sledljivost in lokacije.
- Hladna veriga – tu lahko odčitavamo temperature na časovnem grafu. Lahko vidimo, kje so bile zgornje in spodnje meje presežene in v katerem procesu je bilo to storjeno.
- Sledljivost – opisno in slikovno nam prikaže, kakšni procesi so bili izvedeni na tem produktu.
- Lokacije – z uporabo Google map se nam izriše pot in označke na katerih vidimo pot, katero je prepotoval produkt.
- Datum, ki označuje uporabnost produkta.

### 5.1.1 Vstopna stran in funkcionalnosti aplikacije

Vstopna stran je oblikovana tako, da že ob prvem pogledu lahko vidimo, da aplikacija oz celoten sistem nadzora temperatur vsebuje tehnologijo NFC. Je enostavna za uporabo, na sredini vsebuje slike procesov in funkcionalnosti v preskrbovalni verigi rib, ki prikazujejo delovanje aplikacije. Vstopno stran si lahko ogledamo na sliki 5.1. Spodaj vsebuje meni z gumbi, kateri vodijo na druge funkcionalnosti in dejavnosti, kot so barcode scanner, informacije, zgodovina in kartica. Na Sliki 5.2 lahko vidimo izgled strani kartica, strani za informacije in primer zgodovine.

”Barcode scanner” lahko odpremo na dva načina. Prvi je iz vstopne strani, drugi pa iz Opisa produktov. Barcode scanner uporablja kamero s katero slika/skenira QR kodo. Ko slišimo pisk v programu to pomeni, da je uspešno prebral kodo in jo vrnil naši aplikaciji kot vhodno informacijo. Za

integracijo tega programa v našo aplikacijo smo se odločili iz dveh razlogov. Program Barcode scanner se nadgrajuje ločeno od naše aplikacije. Je enostaven za uporabo in za integracijo. Program je licenciran pod licenco Apache License, Version 2.0 [5].

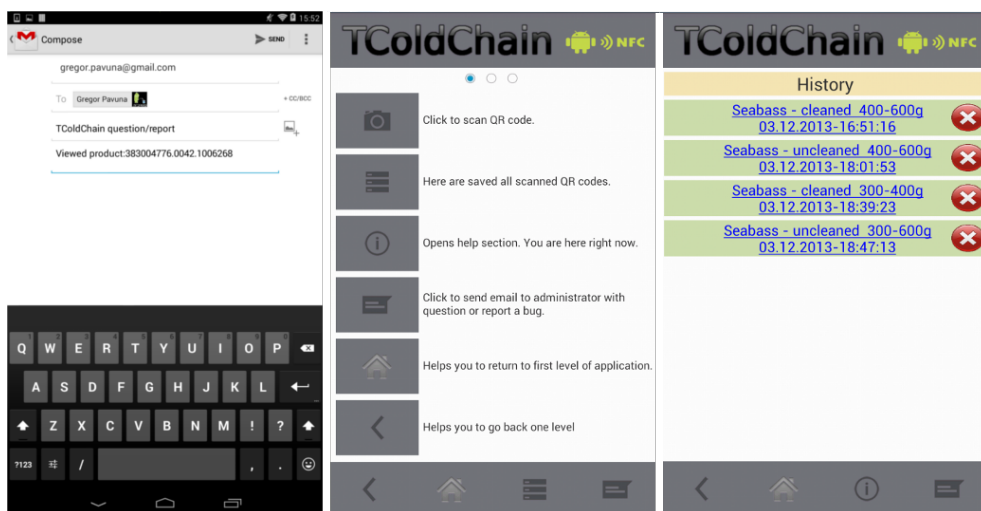
Funkcionalnost "Informacija" se obnaša kot info točka in je dostopna iz skoraj vseh strani. Pisno in slikovno nam razloži, kateri gumb in napis kaj pomenita. Informacije se nahajajo na večih straneh. Te strani so označene na izbirnih tipkah (angl. radio buttons), na katere ni mogoče klikniti, ampak kar s prstom povlečemo v levo za naslednjo stran in v desno za prejšnjo stran.

"Kartica" nam omogoča kontaktiranje administratorja, v primeru, če imamo kakšno vprašanje ali če želimo poročati o kakšnem hrošču. Nahaja se na vseh straneh, katere vsebujejo menije. Ko kliknemo na kartico se nam odpre izbirnik aplikacij za pošiljanje sporočil oz. elektronske pošte. Če imamo že nastavljen privzeti program za pošiljanje elektronske pošte pa preskoči izbirnik in se odpre privzeti program. Elektronska pošta že vsebuje naslovnika, predmet in nekaj besedila v jedru (številka skeniranega produkta). Če se aplikacija sesuje, pošlje obvestilo na Android NotificationManager, ki nas obvešča o dogodkih med delovanjem. NotificationManager skrbi za obvestila, ki se prikažejo uporabniku [3]. S pomočjo tega povemo kdaj se zgodi nepredvidena operacija ali napaka. Ko pritisnemo na obvestilo, se nam odpre program za urejanje in pošiljanje elektronske pošte z že napisanim naslovnikom, predmetom in napako v jedru besedila. Ročno pošiljanje elektronske pošte je narejeno zato, ker pošiljamo iz svojega elektronskega naslova.

"Zgodovina" nam omogoča vpogled v že skenirane produkte. Shranjuje se na napravo v lokalno podatkovno bazo SQLite. Ker naprava potrebuje dostop do spleta, če si želimo ogledati podrobnosti produkta, se izkaže ta funkcija zelo uporabna za primere, ko nimamo mobilne ali internetne povezave. Skeniran produkt se nam shrani v zgodovino in zraven dopiše še datum pregleda. Ko imamo možnost povezave na splet, si lahko ogledamo podrobnosti produkta. Skenirane produkte lahko ročno odstranimo z gumbom, ki je zraven produkta v zgodovini.



Slika 5.1: Vstopna stran aplikacije TColdChain.



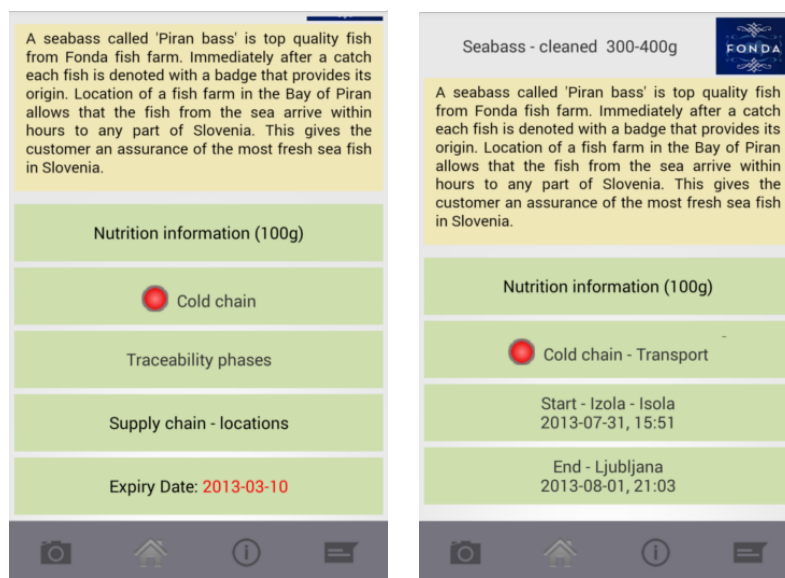
Slika 5.2: Izgled strani kartice, informacij in zgodovine.

### 5.1.2 Sledljivost

Opis produkta je stran, katera se prikaže, ko prislonimo NFC kartico ali skeniramo QR kodo. Stran se dinamično prilagaja dobljenim podatkom. Vsebuje naslednja opisna polja in gumbe:

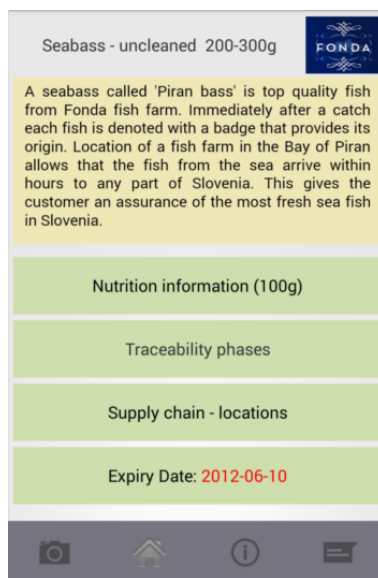
- ime produkta – to je opisno polje v katerem se prikaže ime produkta,
- logotip podjetja – prikaže sliko podjetja, ki je odgovorno za skeniran produkt,
- opis produkta – opisno polje v katerem je opisan produkt,
- podatki o hranilni vrednosti – s pritiskom na gumb se animirano prikažejo podatki o hranilni vrednosti. S ponovnim pritiskom na gumb začnejo podatki izginjati,
- hladna veriga – gumb z lučko, katera je lahko obarvana rdeče ali zeleno. Če je zelena pomeni, da so bile izmerjene temperature znotraj zgornje in spodnje meje. Če je rdeča pa pomeni, da so temperature v nekem času presegle zgornjo in spodnjo mejo,
- faze sledljivosti in dobavne verige (lokacije) – gumba sta povezana, če ni sledenja ni podatkov o lokacijah,
- datum veljavnosti – opisno polje, katero nam napiše datum do katerega je produkt uporaben. Če je datum potekel potem napiše datum z rdečo barvo, v nasprotnem primeru pa z modro.

Aplikacija lahko dobi podatke za: (i) sledljivost in hladna veriga – prikažejo se vsi gumbi (slika 5.3(a)); (ii) hladna veriga – prikažeta se gumba s podatki o hranilnih vrednostih in gumb za hladno verigo (slika 5.3(b)); (iii) sledljivost – prikažejo se vsi gumbi razen gumba za hladno verigo (slika 5.3(c)). Obstaja še ena izjema in sicer če podatkov o sledljivosti in hladni verigi nimamo v podatkovni bazi EPCIS, gremo podatke iskat v senzorsko podatkovno bazo.



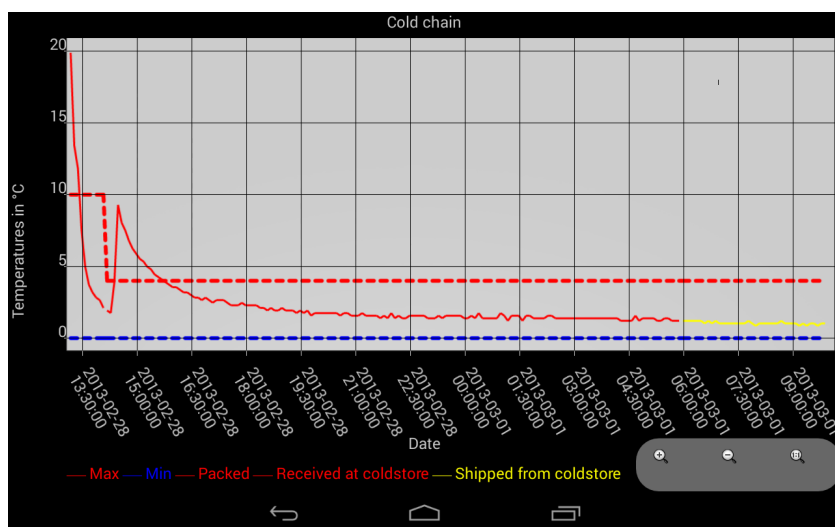
(a)

(b)



(c)

Slika 5.3: Primeri dinamičnega izrisa podatkov o produktu: (a) sledljivost in hladna veriga; (b) hladna veriga; (c) sledljivost.



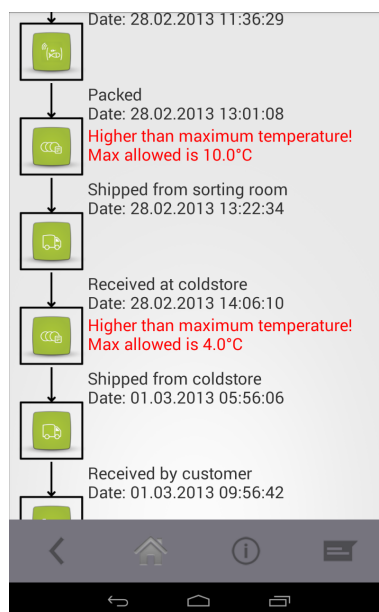
Slika 5.4: Primer časovnega grafa iz aplikacije.

Če se podatki o skeniranem produktu nahajajo tam, izrišemo gumb hladna veriga in dve opisni polji s podatkom, kje se je meritev začela in kje končala.

S pritiskom na gumb za hladno verigo (angl. Cold chain) se nam prikaže graf, kateri je generiran s pomočjo knjižnice AChartEngine in vsebuje podatke iz senzorskega XMLja. Vrsta grafa je časovni graf, ki po x-osi kaže čas, po y-osi pa temperature. Omogoča nam, da s premikom prstov povečamo graf, pomanjšamo graf in ga lahko premikamo v vse strani. Pri manjših napravah z manjšo gostoto zaslona in velikostjo običajno ni mogoče povečati ali pomanjšati grafa. Na sliki 5.4 lahko vidimo primer časovnega grafa iz aplikacije za testne podatke v bazah na strežniku.

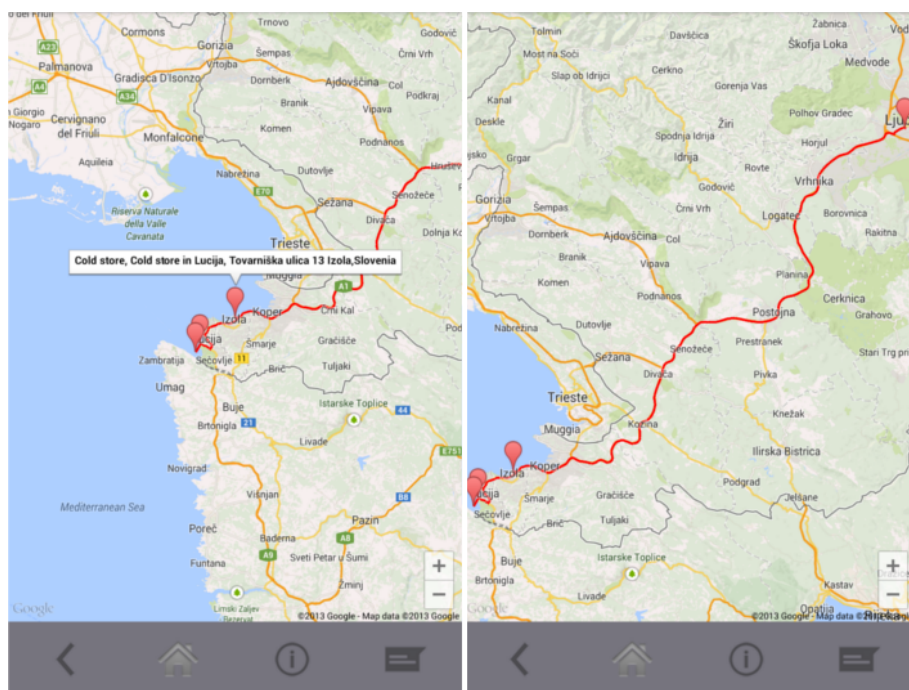
Faze sledenja (angl. Traceability phases) nam s slikami in opisom prikazujejo, kaj se je s produktom dogajalo v vsakem procesu verige. Če je šlo v kateri fazi kaj narobe in je temperatura izven dovoljenih mej, nam to aplikacija napiše z rdečo barvo. V primeru, da je vse v redu s temperaturami pa izpiše samo ime procesa, datum in uro (slika 5.5).

Lokacije v preskrbovalni verigi (angl. Supply Chain – locations) nam prikažejo pot, ki jo je produkt prepotoval od začetka (običajno proizvajalci,



Slika 5.5: Faze sledljivosti.

kjer se izvede prvi korak sledljivosti) pa do trenutka, ko ga skeniramo. Pot lahko izrišemo le če imamo podatke o sledenju in podatke na katerih lokacijah se je produkt nahajal. Za izris mape sveta in poti uporabljamo Google Maps Android API v2. Za implementacijo le teh v našo aplikacijo je bilo potrebno narediti v Google Cloud Console račun in aktivirati Google Maps Android API v2 storitev. Za uporabo Google Maps je potrebno imeti nameščen najnovejši Google Play Services, ki skrbi za posodobitev aplikacij Google v OS Android [16]. Ker v podatkih iz lokacij ne dobimo naslova, ampak zemljepisno širino in zemljepisno dolžino je potrebno s pomočjo Geocoderja pridobiti naslov lokacije. Geocoder je razred za ravnanje geokordiniranja in povratnega geokordiniranja. Geokordiniranje je proces pridobivanja iz naslova ulice ali drugega opisa lokacije zemljepisno širino in zemljepisno dolžino. Povratna geokordinacija pa je, da s pomočjo zemljepisne širine in zemljepisne dolžine pridobimo naslov lokacije [10]. Ko imamo naslov lokacije, lahko narišemo na mapo označevalnike in jim damo pomen v sistemu sledljivosti. To pomeni, da napišemo kateri poslovni korak se tam izvaja. Za pridobitev poti upora-



Slika 5.6: Primer dobavne verige – lokacije.

bimo začetne zemljepisne koordinate in končne zemljepisne koordinate. Te koordinate nato vstavimo v spletni naslov in ga pošljemo preko HTTP zahtevka na Google maps spletno storitev, ki se nahaja na tem spletnem naslovu: ”maps.googleapis.com/maps/api/directions/json?origin=začetne-zemljepisne-geografske-koordinate&destination=končne-zemljepisne-geografske-koordinate”. Storitve nam vrne JSON z zbirko lomljenih črt (angl. Poly-lines), katere je potrebno najprej dekodirati in pridobiti ven poligone. Ko imamo poligone pa se izriše pot produkta. Na Sliki 5.6 je prikazana pot produkta, to je rib iz Piranskega zaliva do Ljubljane.

## 5.2 Testiranje

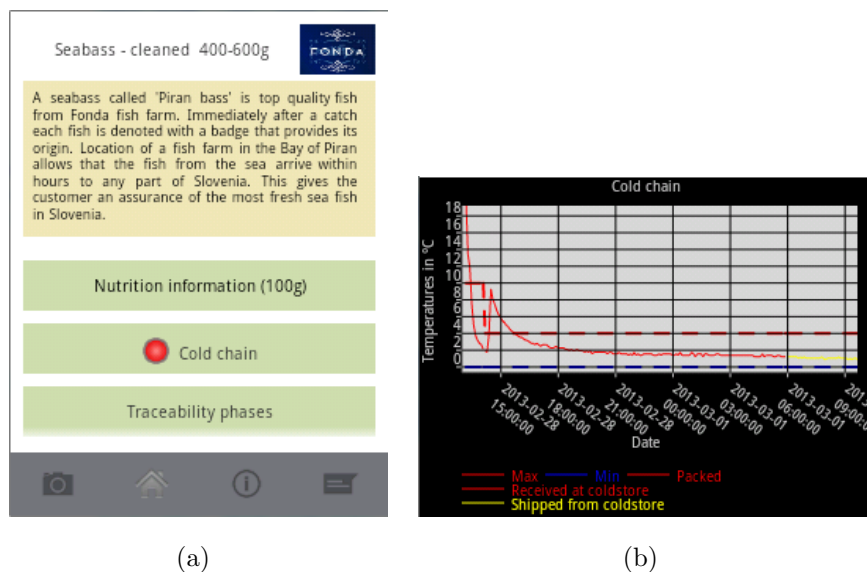
Za operacijski sistem Android obstaja velik nabor naprav. Naprave so razdeljene po velikosti in gostoti zaslonov. Ker aplikacija gradi grafični vmesnik

programsko na podlagi pridobljenih podatkov je bilo potrebno nastavljanje nekatere spremenljivke oz. konstante. Testiranje in nastavljanje smo razdelili na tri dele:

- Prvi del je vseboval testiranje in nastavljanje velikosti pisave za gumbe, menije, naslove, elemente v grafu in za opis produkta (kateri je sestavljen iz HTMLja zaradi obojestranskega poravnjanja besedila),
- drugi del je nastavitev konstante za različne gostote in velikosti pri izrisu grafa in informacije,
- tretji del pa je nastavljanje pravih prostorov elementov (angl. Margin, padding).

Velikost zaslona se meri po diagonali zaslona naprave. Obstajajo štiri skupine: mala (angl. small), normalna (angl. normal), velika (angl. large) in zelo velika (angl. xlarge - extra large). Gostota zaslona vsebuje pet skupin: nizka (ldpi - low dots per inch), srednja (mdpi - medium dots per inch), visoka (hdpi - high dots per inch), ekstra visoka (xhdpi - extra high dots per inch) in primerna za televizije (tvdpi - tv dot per inch). Količina pik znotraj območja zaslona se imenuje dpi (angl. dots per inch). Kar pomeni, da ima nizka (ldpi) manj pik znotraj območja zaslona kot srednja (mdpi) [29].

Testiranje je potekalo na napravah Samsung galaxy 5, kateri ima najmanjšo možno velikost in gostoto zaslona (angl. small-ldpi), tablica Asus Nexus 7 (angl. large-tvdpi), tablica Google Nexus 10 (angl. xlarge-xhdpi), tablica Samsung Galaxy tab 2 (angl. large-mdpi) in preko virtualnih naprav. V testiranje preko virtualne naprave so bile vključene naslednje velikosti-gostote zaslonov: small-ldpi, normal-mdpi, large-mdpi, xlarge-mdpi, normal-hdpi in normal-xhdpi. V primeru, da aplikacija ne podpira določene velikosti in gostote zaslona generira elektronsko pošto in pošlje obvestilo uporabniku. Elektronska pošta vsebuje podatke o velikosti in gostoti zaslona. S pomočjo tega potem naredimo popravke za aplikacijo. Opazili smo, da razlike zaslonov najbolj pridejo do izraza pri izrisu grafa in opisu produktov. Graf je edini element, ki potrebuje veliko širine (zato je tudi postavljen v landscape

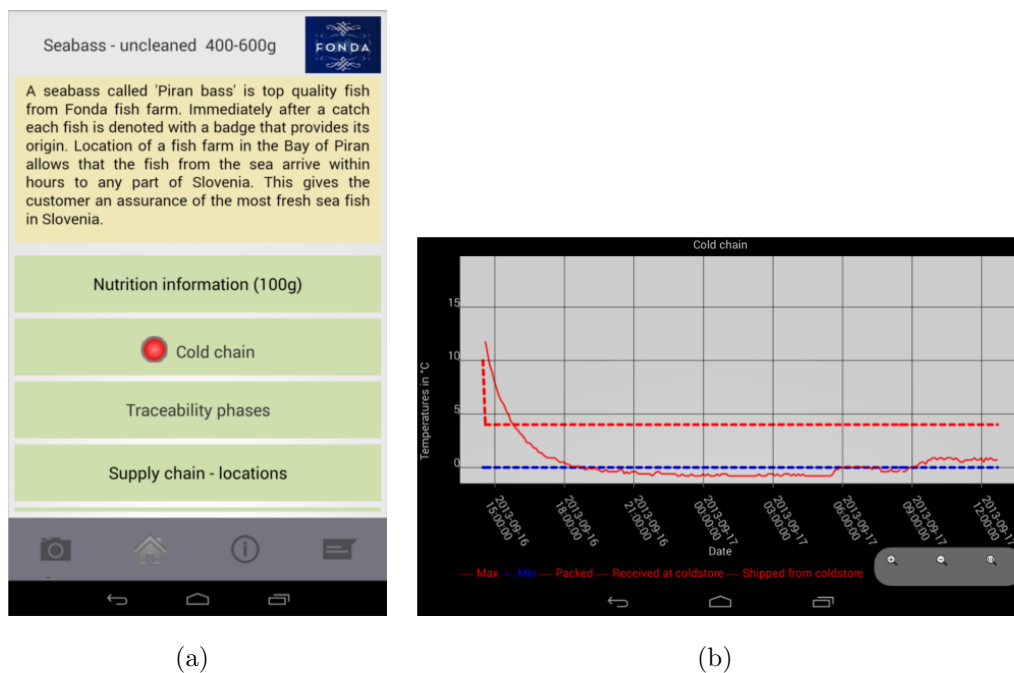


(a)

(b)

Slika 5.7: Rezultat testiranja na napravi z velikostjo in gostoto zaslona small-ldpi: (a) opis produkta; (b) časovni graf.

oz. ležečo pozicijo), opis produkta pa potrebuje višino. Na sliki 5.7(b) lahko vidimo, da je graf zelo majhen, ni prostora za orodno vrstico z gumbi (kateri nam omogočijo, da lahko povečamo, pomanjšamo ali postavimo graf na začetno stanje). Pri opisu produkta (slika 5.7(a)) pa je potrebno veliko pomikati zaslono gor in dol saj se prikaže zelo malo elementov na zaslonu. Pri zaslonu large-tvdpi pa je graf (slika 5.8(b)) lepo viden z vsemi elementi, lepo podan po širini ekrana. Pri opisu produkta pa so vidni skoraj vsi elementi (slika 5.8(a)).



Slika 5.8: Rezultat testiranja na napravi z velikostjo in gostoto zaslona large-tvdpi: (a) opis produkta; (b) časovni graf.

# Poglavje 6

## Sklepne ugotovitve

Cilje, kateri so bili zastavljeni pri načrtovanju projekta, so bili doseženi. Funkcionalnosti, ki so bile zastavljene pri samem načrtovanju so uspešno realizirane, delujejo in so bile testirane z realnimi podatki. Mobilna aplikacija omogoča branje QR kode, branje NFC kartic, shranjuje zgodovino prebranih produktov, omogoča informacije o elementih, ki se nahajajo v aplikaciji in vsebuje sam vmesnik za kontaktiranje administratorja v primeru vprašanja ali poročanja glede kakšnih hroščev. Najpomembnejši del pa je vsekakor prikaz informacij o sledljivosti in hladni verigi z uporabo pametnih telefonov in tabličnih računalnikov, ki podpirajo tehnologijo NFC. Uporabniku je predstavljen opis produkta s podatki o hranilnih vrednostih, rok uporabnosti, procesi, lokacije, kjer se je nahajal in pot po kateri je potoval. Podatki se na strežniku nahajajo v dveh jezikovnih različicah, v slovenščini in v angleščini, v mobilni aplikaciji pa se prikažejo v jeziku, ki je izbran na napravi.

Aplikacija ima veliko funkcionalnosti in nudi širok nabor informacij, a bi jo lahko še izboljšali z novimi funkcijami in podatki, ki so na voljo v podatkovnih bazah. Vključili bi lahko še podrobnejši prikaz podatkov o posameznih podjetjih, ki delujejo v posameznih poslovnih korakih. Zaenkrat se slike, ki so uporabljene v aplikaciji, nahajajo v spominu naprave, lahko bi dodali še slike o podjetjih in produktih in jih shranjevali na strežnikih. Ko bi bile potrebne za prikaz v aplikaciji, bi se s pomočjo HTTP zahteve pridobile iz

spletne storitve in shranile v lokalno podatkovno bazo. Za hitrejše delovanje in da uporabniku ne bi bilo potrebno čakati ob kliku na izbrani gumb, bi se to lahko izvajalo sočasno med tem, ko bi uporabnik že gledal informacije o produktu. Dodali bi lahko možnost za ročno nastavljanje jezika, podporo za prikaz senzorskih podatkov v različnih grafih ali pa izpis zahtevanih podatkov o produktu.

Aplikacija se uporablja v povezavi s pilotno rešitvijo RFID-F2F in omogoča prikaz sledljivosti za teste, ki jih občasno izvajajo. V nadaljevanju pa bi v povezavi z zadnjo rešitvijo nadzora temperatur z NFC zapisovalniki temperatur delovala kot celovita rešitev.

# Literatura

- [1] (2013) AchartEngine, charting library for Android. Dostopno na:  
<https://code.google.com/p/achartengine/>
  
- [2] (2013) android.database.sqlite. Dostopno na:  
<http://developer.android.com/reference/android/database/sqlite/package-summary.html>
  
- [3] (2013) Android notification manager. Dostopno na:  
<http://developer.android.com/reference/android/app/NotificationManager.html>
  
- [4] (2013) Android orodja za razvijalce. Dostopno na:  
<http://developer.android.com/tools/help/adt.html>
  
- [5] (2013) Apache Licence, Version 2.0. Dostopno na:  
<http://www.apache.org/licenses/LICENSE-2.0.html>
  
- [6] (2013) Cold Chain Management. Dostopno na:  
<http://ccm.ytally.com/index.php?id=7>
  
- [7] (2013) CSS Introduction. Dostopno na:  
[http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp)
  
- [8] (2013) Erjavec, Rok, 1990. Aplikacija za upravljanje identifikatorjev in podatkov v sistemu z NFC nadzorom temperatur. Dostopno na:  
<http://cobiss5.izum.si/scripts/cobiss?ukaz=DISP&id=1324202310196161&rec=2&sid=1>

- 
- [9] (2013) Explaining API levels. Dostopno na:  
<http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>
- [10] (2013) Explaining Geocoder. Dostopno na:  
<http://developer.android.com/reference/android/location/Geocoder.html>
- [11] (2013) GlassFish open-source. Dostopno na:  
<http://en.wikipedia.org/wiki/GlassFish>
- [12] (2013) Global tracking systems, Class-3 BAP RFID tehnology. Dostopno na:  
<http://www.gtsllc.co/class3rfid.html>
- [13] (2013) Global tracking systems, Cold Chain solutions. Dostopno na:  
<http://www.gtsllc.co/coldchainsolutions.html>
- [14] (2013) Google Checkout. Dostopno na:  
<http://google.about.com/od/c/g/checkoutdef.htm>
- [15] (2013) Google Maps Android API v2. Dostopno na:  
<https://developers.google.com/maps/documentation/android/start>
- [16] (2013) Google Play Services. Dostopno na:  
<https://developer.android.com/google/play-services/index.html>
- [17] (2013) GS1 EPC Tag Data Standard 1.7. Dostopno na:  
[http://www.gs1.org/sites/default/files/docs/tds/tds\\_1\\_7-Std.pdf](http://www.gs1.org/sites/default/files/docs/tds/tds_1_7-Std.pdf)
- [18] (2013) GS1 EPC/RFID. Dostopno na:  
<http://www.gs1si.org/1/standardi-in-resitve/gs1-epcrfid.aspx>

- [19] (2013) HarvestMark solutions. Dostopno na:  
<http://www.harvestmark.com/solutions.aspx>
- [20] (2013) HTML basics. Dostopno na:  
<http://www.w3schools.com/html/default.asp>
- [21] (2013) Integra Grupa Logista. Dostopno na:  
<http://www.integra2.es/en/news/Pages/integra2-goes-for-mobile-technology.aspx>
- [22] (2013) Introducing Android. Dostopno na:  
[http://google.about.com/od/socialtoolsfromgoogle/p/android\\_what\\_is.htm](http://google.about.com/od/socialtoolsfromgoogle/p/android_what_is.htm)
- [23] (2013) Introduction to XML. Dostopno na:  
<http://www.w3schools.com/xml/>
- [24] (2013) Java. Dostopno na:  
<http://www.webopedia.com/TERM/J/Java.html>
- [25] (2013) JAX-RS, RESTful Web Services in Java. Dostopno na:  
<http://www.devx.com/Java/Article/42873>
- [26] (2013) O programskem okolju Eclipse. Dostopno na:  
<http://lgm.fri.uni-lj.si/PA/ECLIPSE/index.html>
- [27] (2013) Ogljed aplikacije HarvestMark za operacijski sistem Android. Dostopno na:  
<https://play.google.com/store/apps/details?id=com.yotta.HarvestMark>
- [28] (2013) Oracle GlassFish Server. Dostopno na:  
<http://www.oracle.com/technetwork/middleware/glassfish/overview/index.html>

- [29] (2013) Overview of Screens Support. Dostopno na:  
[http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)
- [30] (2013) Projekt RFID-F2F. Dostopno na:  
<http://www.rfid-f2f.eu/news.asp?cat=2>
- [31] (2013) Regular expressions tutorial. Dostopno na:  
<http://www.regular-expressions.info/>
- [32] (2013) RESTful Web Services, Oracle Corporation. Dostopno na:  
<http://www.oracle.com/technetwork/articles/javase/index-137171.html>
- [33] (2013) RESTful Web Services: The Basics. Dostopno na:  
<http://www.ibm.com/developerworks/webservices/library/ws-restful/>
- [34] (2013) ScienceDirect, Advanced traceability system in aquaculture supply chain. Dostopno na:  
<http://www.sciencedirect.com/science/article/pii/S0260877413004597>
- [35] (2013) Simple RESTful web services with Glassfish. Dostopno na:  
<http://www.andygibson.net/blog/article/simple-restful-web-services-with-glassfish/>
- [36] (2013) SQL Tutorial. Dostopno na:  
<http://www.w3schools.com/sql/default.asp>
- [37] (2013) SQLiteDatabase. Dostopno na:  
<http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html>

- 
- [38] (2013) The Java Programming Language. Dostopno na:  
[http://groups.engin.umd.umich.edu/CIS/course.des/cis400/  
java/java.html](http://groups.engin.umd.umich.edu/CIS/course.des/cis400/java/java.html)
- [39] (2013) Traceability of food products. Dostopno na:  
[http://www.sciencedirect.com/science/article/pii/  
S0260877406006893](http://www.sciencedirect.com/science/article/pii/S0260877406006893)
- [40] (2013) What is JSON? Dostopno na:  
<http://www.w3schools.com/json/>
- [41] (2013) What is MIME type. Dostopno na:  
<http://www.aibn.com/help/Learn/mimetypes.html>
- [42] (2013) What is NFC? Dostopno na:  
[http://www.pcadvisor.co.uk/how-to/mobile-phone/3472879/  
what-is-nfc-how-nfc-works-what-it-does/](http://www.pcadvisor.co.uk/how-to/mobile-phone/3472879/what-is-nfc-how-nfc-works-what-it-does/)
- [43] (2013) What is QR Code? Dostopno na:  
<http://searchengineland.com/what-is-a-qr-code-and-why-do-you-need-one-27588>
- [44] (2013) Zxing team, barcode scanner. Dostopno na:  
<https://code.google.com/p/zxing/>