

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matic Odar

**Nadzor temperatur z uporabo  
tehnologije NFC – Android aplikacija  
v sistemu sledljivosti**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mira Trebar

Ljubljana, 2014



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.





Št. naloge: 00112 / 2013

Datum: 15.4.2013

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

Kandidat: **MATIC ODAR**

Naslov: **NADZOR TEMPERATUR Z UPORABO TEHNOLOGIJE NFC –  
ANDROID APLIKACIJA V SISTEMU SLEDLJIVOSTI  
USE OF NFC TECHNOLOGY IN TEMPERATURE MONITORING -  
ANDROID APPLICATION IN TRACEABILITY SYSTEM**

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje


Tematika naloge:

Tehnologija NFC (Near Field Communication) omogoča nadzor in pregled temperatur v logističnih procesih transporta in skladiščenja s pametnimi telefoni in tabličnimi računalniki. Kandidat naj v diplomskem delu preuči njene možnosti in v povezavi s senzorji zasnuje izvedbo mobilne aplikacije za operacijski sistem Android. Predlagana rešitev naj vključuje izvedbo hladne verige, ki zajema inicializacijo pametnih značk s temperaturnimi senzorji (SL13A), zajem podatkov, shranjevanje in pregled temperaturnih meritev. Testiranje naj bo predstavljeno v okviru sistema sledljivosti, ki je bil izdelan v pilotnem projektu »RFID from Farm to Fork« tako, da omogoča izvedbo celotnega procesa z uporabo pametnega telefona NFC v realnem okolju.

Mentor:

  
doc. dr. Mira Trebar

Dekan:

  
prof. dr. Nikolaj Zimic





## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matic Odar, z vpisno številko **63080123**, sem avtor diplomskega dela z naslovom:

### **Nadzor temperatur z uporabo tehnologije NFC – Android aplikacija v sistemu sledljivosti**

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 13. februarja 2014

Podpis avtorja:



*Zahvaljujem se vsem, ki so mi pomagali in me spodbujali pri pisanju  
diplomske naloge.*

*Posebna zahvala gre moji mentorici, doc. dr. Miri Trebar, za vodenje  
projekta, zavzetost, trud in pomoč pri izdelavi diplomskega dela.*

*Zahvaljujem se tudi mami in očetu za podporo in potrpežljivost skozi vsa  
leta študija.*

*Hvala!*



# Seznam uporabljenih kratic in simbolov

**ADT** – **A**ndroid **D**evelopment **T**ools (android orodja za razvijalce)

**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)

**ERP** – **E**nterprise **R**esource **P**lanning (celovita informacijska rešitev)

**GPS** – **G**lobal **P**ositioning **S**ystem (sistem globalnega pozicioniranja)

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

**ISO** – **I**nternational **O**rganization for **S**tandardization (mednarodna organizacija za standardizacijo)

**JDK** – **J**ava **D**evelopment **K**it (platforma za razvijanje javanskih aplikacij)

**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 (tehnologija komunikacije kratkega dosega)

**PHP** – **P**HP **H**ypertext **P**rocessor (skriptni programski jezik)

**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 Markup Language (struktura za prenos podatkov med strežnikom in aplikacijo)

**TDT** – **T**ag **D**ata **T**ranslation (knjižnica za prevajanje podatkov v oznake EPC po standardu GS1)

# Kazalo

Povzetek

Abstract

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Uvod</b>                                       | <b>1</b>  |
| <b>2</b> | <b>Sistem za nadzor temperatur</b>                | <b>3</b>  |
| 2.1      | Hladna veriga . . . . .                           | 4         |
| 2.2      | Zahteve načrtovanja . . . . .                     | 5         |
| 2.3      | Vključitev v projekt RFID-F2F . . . . .           | 7         |
| 2.4      | Obstoječe rešitve . . . . .                       | 9         |
| 2.4.1    | F2F-Fonda . . . . .                               | 9         |
| 2.4.2    | Smart-Trace . . . . .                             | 10        |
| 2.4.3    | Aktivni sistemi za nadzor hladne verige . . . . . | 11        |
| <b>3</b> | <b>Uporabljene tehnologije in oprema</b>          | <b>13</b> |
| 3.1      | Strojna oprema . . . . .                          | 14        |
| 3.1.1    | IDS-SL13A . . . . .                               | 14        |
| 3.1.2    | RFID značke . . . . .                             | 15        |
| 3.1.3    | IDS-R13MP . . . . .                               | 16        |
| 3.2      | Programska oprema . . . . .                       | 16        |
| 3.2.1    | Android . . . . .                                 | 16        |
| 3.2.2    | Eclipse in ADT . . . . .                          | 17        |
| 3.2.3    | SAL Read . . . . .                                | 17        |

|          |   |           |
|----------|---|-----------|
| 3.3      | Tehnologije . . . . .                               | 19        |
| 3.3.1    | RFID in NFC . . . . .                               | 19        |
| 3.3.2    | GS1 in EPC . . . . .                                | 20        |
| 3.3.3    | MySQL . . . . .                                     | 21        |
| 3.3.4    | GPS . . . . .                                       | 21        |
| 3.3.5    | PHP . . . . .                                       | 21        |
| <b>4</b> | <b>Mobilni aplikaciji TraSens</b>                   | <b>23</b> |
| 4.1      | TraSens . . . . .                                   | 24        |
| 4.1.1    | Načrtovanje . . . . .                               | 24        |
| 4.1.2    | Internet . . . . .                                  | 26        |
| 4.1.3    | Lokacija . . . . .                                  | 28        |
| 4.1.4    | Prijava . . . . .                                   | 28        |
| 4.1.5    | Grafični vmesnik . . . . .                          | 29        |
| 4.1.6    | Začetek meritev (START LOG) . . . . .               | 30        |
| 4.1.7    | Konec meritev (END LOG) . . . . .                   | 31        |
| 4.1.8    | Pregled meritev (READ LOG) . . . . .                | 31        |
| 4.1.9    | Sinhronizacija podatkovnih baz . . . . .            | 32        |
| 4.2      | TraSens Manager . . . . .                           | 35        |
| 4.2.1    | Grafični vmesnik . . . . .                          | 36        |
| 4.2.2    | Podatki o zapisovalniku (READ LOGGER) . . . . .     | 37        |
| 4.2.3    | Inicializacija zapisovalnika (INITIALIZE) . . . . . | 38        |
| 4.2.4    | Nastavitve zapisovalnika . . . . .                  | 38        |
| <b>5</b> | <b>Testiranje</b>                                   | <b>41</b> |
| 5.1      | Mobilne naprave . . . . .                           | 41        |
| 5.2      | Načini beleženja . . . . .                          | 42        |
| 5.3      | Aplikacija ScmContext . . . . .                     | 45        |
| <b>6</b> | <b>Zaključek</b>                                    | <b>47</b> |

# Kazalo slik in tabel

|     |   |    |
|-----|---|----|
| 2.1 | Primer uporabe. . . . .                               | 5  |
| 2.2 | Hladna veriga, proces delovanja. . . . .              | 6  |
| 2.3 | Identifikacijski ključ GS1 . . . . .                  | 8  |
| 2.4 | Razčlenitev formata GS1 na pomembne podatke . . . . . | 9  |
| 3.1 | Zapisevalnik SL13A. . . . .                           | 14 |
| 3.2 | Čitalec R13MP. . . . .                                | 16 |
| 3.3 | Grafični vmesnik SAL Read. . . . .                    | 18 |
| 3.4 | Prikaz temperatur v programu SAL Read. . . . .        | 19 |
| 4.1 | TraSens, graf prehodov. . . . .                       | 27 |
| 4.2 | TraSens . . . . .                                     | 29 |
| 4.3 | EPCIS XML dogodka ADD . . . . .                       | 33 |
| 4.4 | TraSens Manager, graf prehodov. . . . .               | 35 |
| 4.5 | TraSens Manager . . . . .                             | 36 |
| 4.6 | Podatki zapisevalnika. . . . .                        | 37 |
| 5.1 | Dense: TraSens . . . . .                              | 42 |
| 5.2 | Out Of Limits: TraSens . . . . .                      | 43 |
| 5.3 | Limits Crossing: TraSens . . . . .                    | 44 |
| 5.4 | Graf temperatur v aplikaciji TraSens. . . . .         | 45 |
| 5.5 | Vnos v spletni aplikaciji ScmContext. . . . .         | 46 |
| 5.6 | Graf temperatur v aplikaciji ScmContext. . . . .      | 46 |



# Povzetek

Hladna veriga je izrednega pomena, tako v farmaciji in živilski industriji, kot tudi v drugih dejavnostih. Ob dobri informiranosti lahko ločimo kakovostne in zdrave dobrine od pokvarjenih, kar je lahko v nekaterih primerih življenjsko pomembno.

V diplomski nalogi je predstavljen sistem za zagotovitev informacij o hladni verigi, od zasnove do prototipa. Rešitev vključuje dve Android aplikaciji, ki sta namenjeni delu z radiofrekvenčnimi zapisovalniki v tehnologiji NFC (Near Field Communication) in omogočata nastavljanje parametrov, zajem podatkov iz zapisovalnikov in shranjevanje teh podatkov na oddaljenem strežniku.

Opisana je problematika načrtovanja in programiranja projekta v smeri, ki omogoča vključitev v večji sistem sledljivosti, kot tudi samostojnega delovanja za nadzor temperatur. Predstavljene so zahteve, katerim mora rešitev ustrezati, ki jim sledi opis tehnologij in strojne ter programske opreme, ki smo jo uporabili. V nadaljevanju sta opisani mobilni aplikaciji, na koncu pa testiranje celotnega sistema.

**Ključne besede:** Android, Hladna veriga, Sledljivost, RFID, NFC, EPC, Zapisovalnik temperatur, Živilska industrija



# Abstract

The cold chain is of paramount importance, both in the pharmaceutical and food industries, as well as in other activities. Having good quality information we can separate healthy from faulty goods, which, in some cases, may be of life importance.

The thesis describes a solution of a system to provide cold chain information, from concept to prototype. The solution includes two Android applications that are intended for use with data loggers in NFC (Near Field Communication) technology and enable setup of parameters, data recovery from loggers and storage of data on a remote server.

Described are the problems of planning and programming of a project in a direction that allows integration into a larger traceability system as well as stand-alone operation used in temperature monitoring. Presented are the requirements which our solution must meet, followed by a description of the hardware and software that was used. The following describes the mobile applications, and finally testing of the entire system.

**Keywords:** Android, Cold chain, Traceability, RFID, NFC, EPC, Temperature logger, Food industry



# Poglavje 1

## Uvod

V današnji, moderni družbi, sta hlajenje in zamrzovanje živil najbolj primerni metodi konzerviranja hitro pokvarljivih živil. Proizvodnja zamrznjenih in hlajenih živil je ena izmed najhitreje rastočih panog v živilski industriji, med drugim tudi zaradi časovne stiske posameznika v vsakdanjem življenju. V dobi globalizacije nekatera živila potujejo preko celega planeta, preden pridejo do mize potrošnika. Pri tem je pomembno, da so pravilno hranjena in da so na voljo podatki o njihovem ravnanju.

Hladna veriga pomeni neprekinjeno vzdrževanje predpisane, dovolj nizke temperature živila, da je zagotovljena varnost in čim boljša kakovost živila v celotni živilski verigi; od proizvodnje, prevoza, shranjevanja in razdeljevanja do porabe pri končnem potrošniku.

V primeru spremenjenih pogojev v hladni verigi je potrebno te motnje definirati in izbrati ustrezne ukrepe o uporabnosti produkta. Za zagotavljanje dobre hladne verige je potrebno imeti dober pregled nad temperaturami produkta na celotni poti od dobavitelja do kupca. Za to je potrebna primerna programska in strojna oprema. Vedno bolj se uveljavlja tehnologija radiofrekvenčne identifikacije, kot so pametne značke, RFID (radio frequency identification) čitalci, zapisovalniki temperatur, GPS (global positioning system) itd. 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. Za uporabno vrednost v industriji morajo biti te aplikacije hitre in preproste za uporabo.

Z definiranimi zahtevami na področju mobilnih in spletnih tehnologij smo se odločili za implementacijo samostojnega sistema hladne verige, ki pa ga je možno vključiti v obstoječ projekt RFID-F2F (RFID-from Farm to Fork). Predlagana izvedba sledljivosti je bila zasnovana na tehnologiji RFID. Naša rešitev pa vsebuje tehnologijo NFC (near field communication), ki omogoča uporabo pametnih naprav (telefoni, tablični računalniki). Arhitektura sistema v tem projektu vključuje tri aplikacije:

- I** upravljanje z opisnimi podatki,
- II** zajem in nadzor meritev temperatur s senzorskimi zapisovalniki,
- III** predstavitev informacij o sledljivosti in hladni verigi produkta.

V diplomskem delu je predstavljena rešitev za drugi del arhitekture, to je zajem in nadzor meritev temperatur z zapisovalniki RFID, ki delujejo v povezavi z NFC. Razvili smo dve aplikaciji za operacijski sistem Android, ki omogočata celovit sistem za nadzor hladne verige. Prva aplikacija skrbi za inicializacijo zapisovalnikov in testiranje RFID zapisov, druga pa omogoča beleženje in pregledovanje meritev temperatur in zapis teh meritev v podatkovno bazo na strežniku. Za vključitev v sistem sledljivost smo uporabili identifikatorje GS1, ki se uporabljajo pri projektu RFID-F2F. Za komunikacijo mobilne naprave z zapisovalnikom smo izkoristili vedno bolj uveljavljeno tehnologijo NFC, ki je vključena v vse večje število mobilnih naprav kot so pametni telefoni in tablični računalniki.

## Poglavje 2

# Sistem za nadzor temperatur

Za uspešno zasnovo in izvedbo rešitve moramo najprej dobro pregledati področje problematike in določiti najpomembnejše pojme, na katere se moramo osredotočiti pri našem delu. Nato moramo določiti opremo, postopke in tehnologije, s pomočjo katerih bomo dani problem kar najbolje rešili. Naš namen je bil razviti prototip za nadzor temperatur na terenu, ki bi omogočal:

- zajem in pregled temperatur produktov v hladni verigi,
- objavo teh meritev v ogled vsem, ki jih te zanimajo,
- čim bolj praktično uporabo zgoraj napisanih lastnosti,
- vključitev v sistem sledljivosti RFID-F2F (RFID-from Farm to Fork).

Prav tako je pomembno, da pri projektiranju, kot vedno, v kolikor je to mogoče, predvidimo težave, ki bi lahko nastale in te rešimo v začetni fazi razvoja. Eden od takih problemov je bil, kdaj se meritve zajemajo in shranjujejo, saj tehnologije ne omogočajo neprekinjenega dostopa do omrežja. Zato smo se zaradi preprostejše arhitekture odločili za sistem, ki bo deloval na mobilnih napravah in kjer se meritve ne shranjujejo samodejno, ampak le enkrat na koncu procesa, pregledamo pa jih lahko ob poljubnem času.

## 2.1 Hladna veriga

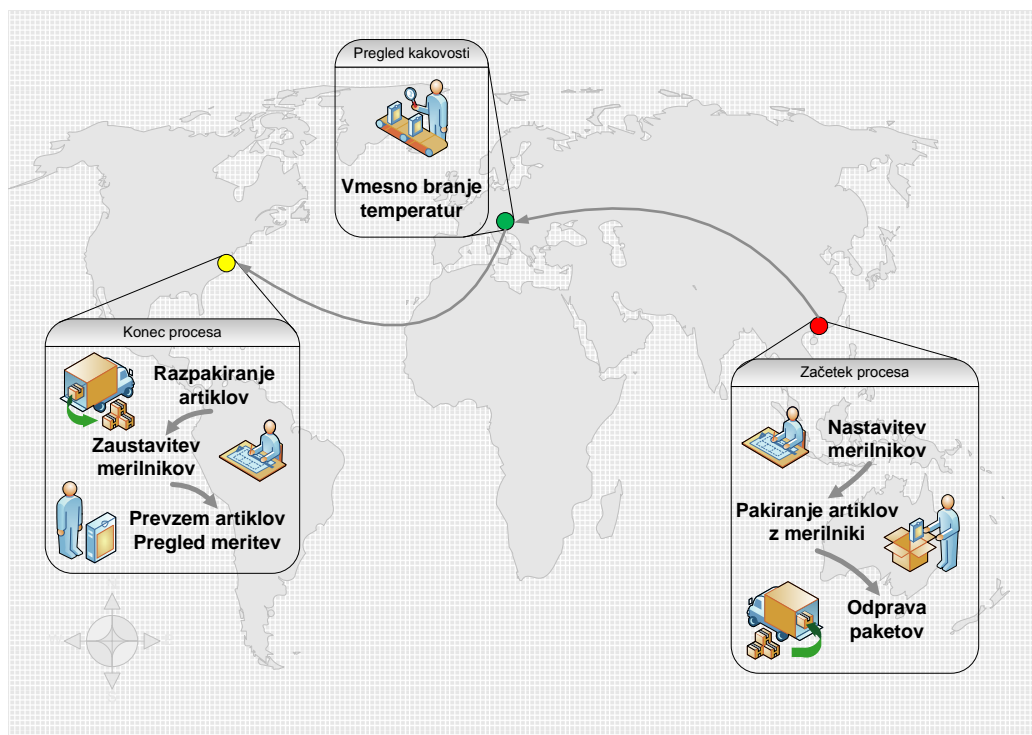
Hladna veriga je postopek, ki zahteva nadzorovanje temperature v preskrbovalni verigi [7]. Nepretrgana hladna veriga je niz skladiščne in distribucijske dejavnosti, ki ohranja določeno temperaturno območje. Take dejavnosti so proizvodnja, prevoz in skladiščenje produkta, kot je npr. shranjevanje v hladilnici ali hladilni omari v nakupovalnem centru. Uporablja se za zagotavljanje predpisanega roka trajanja izdelkov, kot so npr. kmetijski pridelki, morska hrana in druga hitro pokvarljiva hrana, pa tudi fotografski film, kemikalije in farmacevtska zdravila.

Hladna veriga pa ni uporabna le v preskrbovalni verigi. Primerna je tudi za domačo uporabo, ko potrebujemo ohranjanje priporočene temperature. Scenarij, ki tako potrebo dobro opiše, je pošiljanje paketov po pošti med dvema osebama. Rešitev za to je že veliko, npr. termalne škatle za prenos [13], vendar bi se prejemnik včasih rad prepričal o stabilnosti pogojev pri ravnanju prejetih dobrin. Zato je pametno, da naša rešitev podpira tudi take aplikacije.

V tej diplomski nalogi se bomo osredotočili na kakovost hladne verige med prevozom prehranskih dobrin. To pomeni, da se naš proces začne, ko so produkti pakirani v zaboje in razposlani po svetu, ter konča, ko so ti produkti vzeti iz zabojev. Za zagotavljanje neprekinjenosti potrebujemo napravo, ki nenehno meri temperaturo in jo hrani za kasnejši pregled. Za opravljanje te naloge smo se odločili za uporabo tehnologije RFID (Radiofrequency Identification), ki ponuja aktivne pametne značke s temperaturnimi senzorji. Te nam omogočajo natančno merjenje temperatur v intervalu, katerega sami določimo, pravtako pa te temperature shrani, da si jih lahko kasneje ogledamo. Taki zapisovalniki morajo biti fizično prisotni ob produktih, katerih temperaturo merimo.

Naš sistem mora podpirati uporabo pri številnih vrstah produktov, večih procesih preskrbovalne verige in različnih scenarijih. Eden takih scenarijev je opisan na sliki 2.1, ki je tudi dober primer, na katerem smo zgradili našo rešitev. Pri odpremi produktov iz proizvodnje so jim dodani temperaturni zapisovalniki. Do prihoda na cilj lahko meritve zapisovalnikov pregledamo in

po potrebi odstranimo pokvarjene produkte. Ob razpakiranju produktov se meritve temperatur preverijo še enkrat.



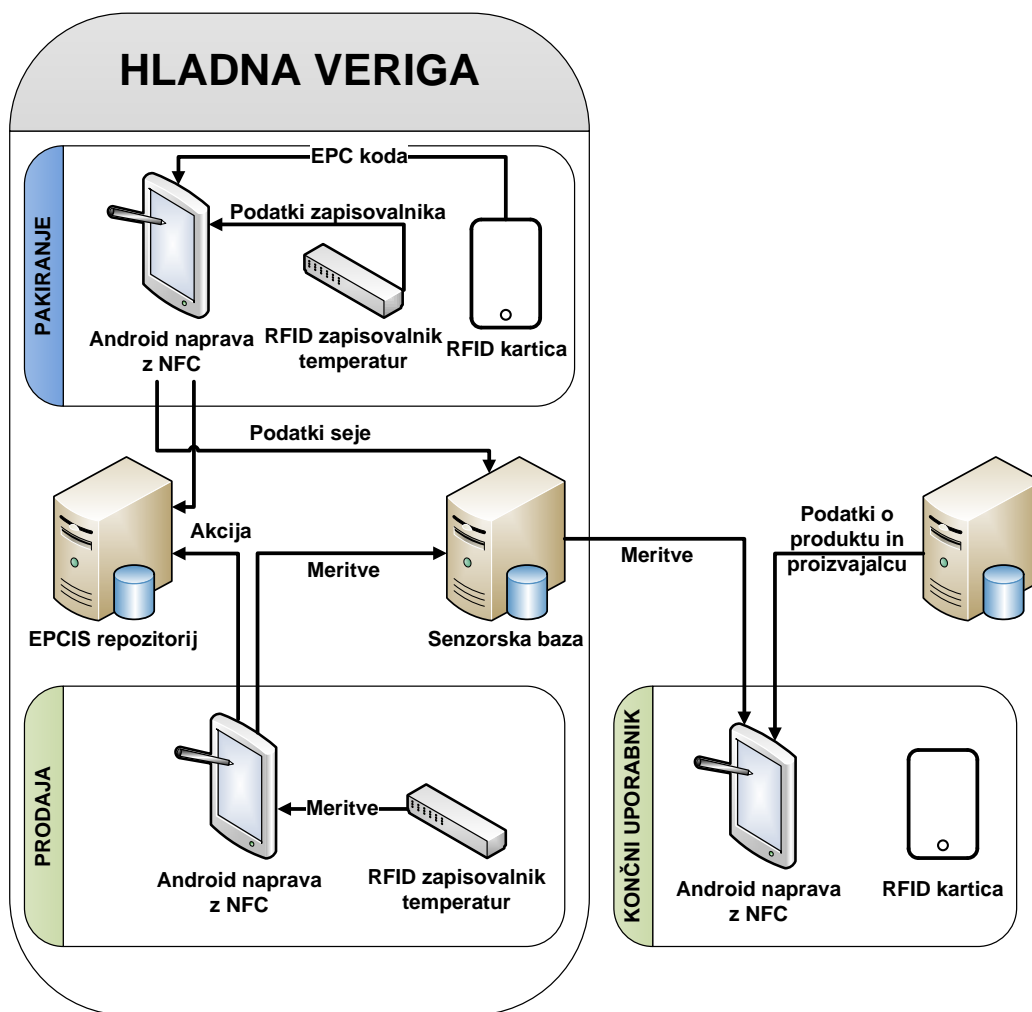
Slika 2.1: Primer uporabe.

## 2.2 Zahteve načrtovanja

Enostavnost in hitrost uporabe sta ključna parametra, ki ju moramo upoštevati v razvoju našega sistema, saj je pospeševanje procesa dela oziroma avtomatizacija ena glavnih zahtev razvoja v moderni industriji. Včasih je, kot kaže, ta celo bolj pomembna od zagotavljanja kakovosti produkta. Zato je zelo pomembno, da je naš sistem majhen, lahek in preprost za uporabo. V povezavi s prejšnjim poglavjem moramo zagotoviti tudi, da je ta dostopen večjemu številu uporabnikov. Paziti pa je treba tudi na ceno opreme, katera je potrebna za postavitve sistema.

Ob upoštevanju vseh zahtev, smo se odločili, da bodo uporabniki upra-

vljali sistem z mobilnimi telefoni na katerih teče operacijski sistem Android. To naredi sistem razmeroma poceni in dostopen vsem uporabnikom takih naprav. To pa ni edina prednost uporabe takih naprav. Večinoma so to namreč mobilni telefoni, ki imajo povezavo z internetom zagotovljeno s strani ponudnika mobilnih storitev, zato nam za način prenosa podatkov ni potrebno skrbeti.



Slika 2.2: Hladna veriga, proces delovanja.

Izbrati je bilo potrebno še medij, s katerim lahko merimo temperature in prenašamo podatke o produktih hladne verige. Da bi bil proces čim bolj

avtomatiziran, smo za to uporabili kartice RFID, ki omogočajo uporabo zapisovalnika temperatur - SL13A (angl. smart active label) [10]. Na njih lahko zapišemo poljubne podatke, ki zajemajo identifikacijsko oznako in temperaturne meritve. Za branje takih kartic pa potrebujemo mobilno napravo, ki omogoča tehnologijo NFC. S tem se krog Android naprav nekoliko zoža, vendar to tehnologijo uporablja vse več novih naprav, v prihodnosti pa po napovedih proizvajalcev lahko pričakujemo, da bo na voljo v vseh novih modelih. S preprostim pritiskom kartice ob tako napravo se iz nje prenesejo podatki, kar naredi proces zelo hiter. Velika prednost RFID kartic je, da so te zelo obstojne v različnih okoliščinah, kar jih naredi primerne tudi za uporabo v industriji.

Z uporabo vseh omenjenih tehnologij dobimo sistem, ki je predstavljen na sliki 2.2. Android naprava ob začetku in koncu hladne verige podatke iz RFID kartic in zapisovalnika pošlje na oddaljeni strežnik. Ti podatki so potem na voljo sistemu, ki je bil razvit v okviru dveh drugih diplomskih nalog [18][19].

## 2.3 Vključitev v projekt RFID-F2F

Ena od zahtev, ki smo jo definirali pri načrtovanju rešitve, je zmožnost vključitve našega sistema v pilotno rešitev sledljivosti v projektu RFID-F2F [5]. Ta za shranjevanje podatkov hladne verige in sledljivosti uporablja repozitorij EPCIS [1]. Podatke se v repozitorij nalaga z zajemom dogodkov EPCIS (angl. EPCIS event), kot so združitev (angl. Aggregation), transakcija (angl. Transaction), itd. Ti dogodki opisujejo akcijo, ki se nanaša na enega ali več produktov. V primeru agregacije dogodek opisuje akcijo, ki se nanaša na več opisanih sredstev in produktov, ki so fizično združeni skupaj. Dogodki se lahko nanašajo tudi na posamezno sredstvo, ali na več združenih sredstev, kjer ni opisan vsak posebej.

Uporabili bomo repozitorij EPCIS za upravljanje informacij o obstoječih procesih hladne verige na najvišjem sloju in bo deloval kot sistem ERP,

ki nam pove, ali obstajajo podrobnejši podatki o meritvah v nižjem sloju hierarhije. Na tak način lahko EPCIS uporabimo kot center infrastrukture večih manjših sistemov, ki lahko med sabo komunicirajo. Eden on takih manjših sistemov je tudi naša aplikacija za zajemanje temperatur.

Za potrebe našega projekta bomo uporabili dogodek združitve (angl. aggregation), saj vedno povežemo zapisovalnik temperatur z vsaj enim sredstvom. Ta sredstva so lahko produkt, zaboj, paleta ali metoda prevoza. S takim dogodkom naložimo v repozitorij EPCIS podatek, ki pove, ali je v poteku proces hladne verige oziroma je ta že končan. Za začetek procesa bomo uporabiki ukaz ADD, za konec pa ukaz DELETE. Ko sta v repozitoriju dogodka z obema ukazoma, vemo, da imamo v nižjem sloju podrobne podatke o meritvah hladne verige.

Še vedno pa moramo definirati identifikator, s katerim opisujemo sredstva. Sistem EPCIS za to uporablja kode EPC, tako da je za ohranjanje konsistenčnosti podatkov potrebno te šifre uporabiti tudi v naši rešitvi.

Za opis vseh objektov, ki obstajajo v testnem okolju nam zadostujejo ključi, ki so naštetni in opisani v tabeli 2.3[3].

| Označuje  | Ključ | Pomen                              | Uporaba          |
|-----------|-------|------------------------------------|------------------|
| Uporabnik | GSRN  | Global Service Relation Number     | prijava v sistem |
| Sredstvo  | GRAI  | Global Returnable Asset Identifier | enota logistike  |
| Produkt   | SGTIN | Global Trade Item Number           | oznaka izdelka   |
| Lokacija  | SGLN  | Global Location Number             | lokacija dogodka |

Tabela 2.3: Identifikacijski ključ GS1

Format GS1 v naši aplikaciji ni v celoti podprt, saj je ta izjemno zapleten, kar pomeni, da bi njegova implementacija pomenila veliko dodatnega dela. Osredotočili smo se na prevajanje primerov kod EPC, ki smo jih uporabili v testnem okolju. Formati identifikatorjev, ki so potrebni za delovanje našega sistema, so opisani v tabeli 2.4.

| Ključ | Tip      | Filter    | Podjetje   | Razred     | Serijska številka |
|-------|----------|-----------|------------|------------|-------------------|
| GSRN  | biti 0-7 | biti 8-10 | biti 14-43 | 0 bitov    | biti 44-71        |
| GRAI  | biti 0-7 | biti 8-10 | biti 14-43 | biti 44-56 | biti 58-95        |
| SGTIN | biti 0-7 | biti 8-10 | biti 14-43 | biti 44-56 | biti 58-95        |
| SGLN  | biti 0-7 | biti 8-10 | biti 14-43 | biti 44-54 | biti 55-95        |

Tabela 2.4: Razčlenitev formata GS1 na pomembne podatke

Nismo pa okrnili formata GS1 samo na štiri ključe, ki jih potrebujemo. Standard GS1 namreč podpira tudi različne dolžine razreda in serijske številke. To določajo biti polja Filter. Naša pretvorba teh bitov ne upošteva in razred in serijsko številko izračuna iz točno določenih dolžin, ki so podane v tabeli 2.4. Končni rezultat take pretvorbe iz heksadecimalnih znakov so naslednji primeri identifikatorjev:

- *urn:epc:id:sgln:383004776.030.1*
- *urn:epc:id:sgtin:383004776.0042.1006268*
- *urn:epc:id:gsrn:383004776.00000001*

Identifikatorji se nanašajo na pilotni projekt sledljivosti v podjetju Fonda.si (RFID-F2F), ki je opisan v poglavju 2.4.1. Predpona *urn:epc:id* nam pove, da gre za identifikator vrste EPC, ki se nanaša na ime spletnega vira.

## 2.4 Obstoječe rešitve

### 2.4.1 F2F-Fonda

Na UL, Fakulteti za Računalništvo in Informatiko, so v Laboratoriju za računalniško arhitekturo v okviru evropskega projekta "RFID from Farm to Fork" razvili sistem [5], katerega glavni cilj je bil zamenjati papirno dokumentacijo z elektronskimi podatki z uporabo tehnologije RFID in implementirati sistem sledljivosti. Projekt je bil izveden v sodelovanju z družinskim podjetjem

Fonda.si, katerih vizija je kupcem zagotoviti sveže in zdrave Piranske brancine najvišje kakovosti. Sistem omogoča sledenje rib od kletk za gojenje do potrošnika. Vsak zaboj rib ima pri prevozu svoj zapisovalnik temperatur RFID. Tako je kupcu zagotovljena informacija o svežini produkta. Identifikacija poteka preko pametnih značk in kod QR z identifikatorjem EPC, ki so dodeljene vsakemu zaboju rib. Potrošnik lahko tako s pametnim telefonom prebere kodo QR in preveri zanimive podatke o ribi, med drugimi poreklo, hranilne vrednosti in temperaturo med prevozom.

Sistem RFID-F2F za nadzor zapisovalnikov in identifikacijskih značk uporablja specializirane mobilne RFID čitalce. V naši rešitvi hladne verige lahko zapisovalnike in značke upravljamo z vsakim Android telefonom, ki ima vključeno tehnologijo NFC. Za označevanje so uporabljene enake kode EPC. Zapisovanje dogodkov v repozitorij EPCIS je tudi realizirano na enak način, tako da je na ta način omogočena vključitev v sistem RFID-F2F.

### 2.4.2 Smart-Trace

Sistem za izvedbo hladne verige Smart-Trace je zelo podoben naši rešitvi [6]. Omogoča pregledovanje podatkov preko spletne strani. Prednost njihove rešitve je način pošiljanja podatkov iz zapisovalnika, saj se ti na oddaljeni strežnik pošiljajo ob vsaki meritvi, zato je uporabnik obveščen o problemih skoraj takoj, ko do njih pride. Namenjen je le dobaviteljem in ne končnim, fizičnim uporabnikom produktov. Merilniki v tem sistemu so za enkratno uporabo in se jih ne da ponastavljati.

Sistem deluje po naslednjih korakih:

1. Čipi, v obliki pametnih kartic, so priloženi paletam s produkti v skladišču dobavitelja.
2. Delovanje čipa preveri delavec, nakar je ta s tovorom naložen v ohlajeno prikolico.
3. Merilnik vsakih 15 minut podatke o temperaturo pošlje brezžični točki, ki se nahaja v prikolici,

4. ta nato zbrane podatke skupaj s GPS podatki pošlje oddaljenemu strežniku, kjer so podatki obdelani.
5. Strežnik o morebitnih težavah dobavitelja obvesti preko sporočil SMS oziroma elektronske pošte. Ta si podatke lahko nato ogleda preko spleta.

### **2.4.3 Aktivni sistemi za nadzor hladne verige**

Podjetje Intellex iz ZDA ponuja celovite storitve, kot so nadzor hladne verige, sledenje sredstev in nadzor osebja s tehnologijo RFID. Za identifikacijo, sledenje in hladno verigo uporabljajo specializirane aktivne značke z dosegom do 100 metrov, kjer je napajanje z baterijo uporabljeno za shranjevanje meritev in tudi za komunikacijo s čitalcem. Za branje in uporabo značk ponujajo tudi fiksno nameščene in ročne čitalce, ki pa niso prav enostavni za uporabo. Za implementacijo manjšega sistema so njihove rešitve zelo drage.



## Poglavje 3

# Uporabljene tehnologije in oprema

Za predlagano izvedbo nadzora temperatur v hladni verigi smo uporabili zapisovalnik SL13A s podporo standardu ISO-15693, pri preprostosti rešitve pa pomagajo RFID značke, ki se nahajajo na produktih, lokacijah, prevoznih sredstvih, itd. Te nosijo identifikacijske kode v formatu GS1. V začetni fazi pisanja aplikacij smo uporabili tudi čitalnik R13MP.

Za uporabniški vmesnik smo uporabili telefon z operacijskim sistemom Android s podporo NFC, saj ta podpira protokol ISO-15693. Osnova aplikacij za Android je programski jezik Java, ki je tudi večinoma uporabljen za programiranje naših aplikacij. Aplikacije so bile razvite v okolju Eclipse z dodatkom ADT.

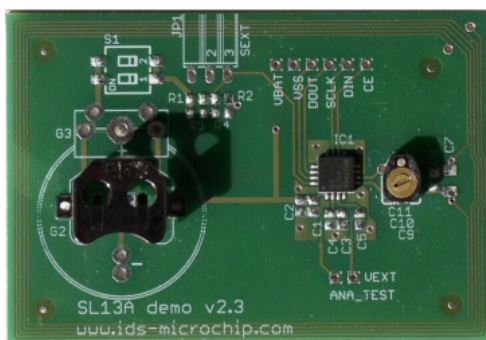
Ker so telefoni in prenos podatkov iz njih močno vezani na mobilno omrežje in njegovo stabilnost, je potrebno poskrbeti za ohranitev in preprečevanje izgub podatkov. To smo realizirali z lokalno podatkovno bazo SQLite. Ta se ob prisotnosti internetnega omrežja sinhronizira z oddaljeno bazo podatkov MySQL. Android žal ne podpira uporabe MySQL, zato smo to rešili z vmesno storitvijo v jeziku PHP, ki izvaja zahteve poslane iz telefona. S storitvijo se povežemo s protokolom HTTP. Vsak prenos podatkov se zapiše tudi kot dogodek EPCIS, z uporabo označevalnega jezika XML.

Kombinacija teh tehnologij omogoča programerju polno ustvarjalno moč in ne dopušča okrnjene uporabniške izkušnje zaradi manjkajočih funkcij.

## 3.1 Strojna oprema

### 3.1.1 IDS-SL13A

Zapisovalnik podatkov SL13A je semi-pasivna pametna značka izdelana v tehnologiji RFID z vgrajenim temperaturnim senzorjem [14], ki deluje po standardu ISO-15693. Za zapisovanje meritev v spomin je uporabljeno napajanje iz baterije (1.5V ali 3V), za branje podatkov pa se napaja iz polja radiofrekvenčnih valov (elektromagnetno valovanje iz RFID čitalnika). Zapisovalnik ima integriran temperaturni senzor z maksimalno natančnostjo  $0.5^{\circ}\text{C}$ , z delovnim območjem  $-40$  do  $110^{\circ}\text{C}$ , možen pa je tudi priklop zunanjega senzorja. Zmogljivost pomnilnika je maksimalno 765 meritev v zgoščenem načinu shranjevanja. Pomnilnik je razdeljen na tri dele: sistemski, uporabniški in meritveni del. Sistemski del je dostopen le s posebnimi ukazi in ima shranjene nastavitve za pravilno delovanje zapisovalnika. V uporabniški del lahko uporabnik zapiše poljubne podatke, prav tako lahko nastavi velikost tega dela. V meritveni del dostopa zapisovalnik, ko tja beleži izmerjene temperature in pa čitalec takrat, ko bere shranjene podatke. Zapisovalniku je potrebno za pravilno delovanje kalibrirati temperaturni senzor.



Slika 3.1: Zapisovalnik SL13A.

Na sliki 3.1 je predstavljen prototip s čipom SL13A. Njegove dimenzije so približno 48 x 70 mm.

Ker izdelana rešitev deluje po protokolu ISO-15693[10], to ni edini uporaben zapisovalnik, ampak se lahko uporabi vsak zapisovalnik, ki razume ukaze standarda ISO-15693. V uporabniškem delu pomnilnika so rezervirani 3 bloki po 32 bitov za shranjevanje kode EPC zapisovalnika.

### 3.1.2 RFID značke

Za zapis kod EPC so bile uporabljene ISO-15693 kompatibilne značke. Uporabljajo tehnologijo RFID in so v obliki nalepk in kartic. So pasivne, kar pomeni, da nimajo svojega izvora napajanja, ampak se pri branju ali pisanju napajajo iz čitalca, pa naj bo to mobilna naprava ali R13MP. V tem primeru se ustvari energija za komunikacijo in prenos podatkov na osnovi indukcije. Postopek je naslednji:

1. Podatki na kartici čakajo, da so prebrani,
2. antena značke prejme energijo iz antene RFID čitalnika,
3. s prejeto energijo pošlje radijske valove čitalcu,
4. čitalec prejme valove ter jih pretvori v pomenljive podatke.

Za vse vrste kod smo uporabili značke SL21CS20 proizvajalca NXP Semiconductors [15]. Za zapis kode o uporabniku smo uporabili značke v obliki bančne kartice, za vse ostale vrste pa so značke v obliki nalepke, katere lahko pritrdimo na produkt, zaboj, itd.

Značke imajo za podatke 28 namenskih blokov po 32 bitov. Za potrebe naše rešitve se uporabljajo le trije, saj so identifikacijske kode dolge 96 bitov. RFID značke imajo določeno prednost pred črtnimi in QR kodami, ker jih čitalec ne potrebuje v vidnem polju. Počasi jih tudi izpodrivajo v uporabi, ker bolje delujejo tudi v slabših delovnih razmerah, saj jih vreme in grobo ravnanje ne uniči tako hitro.

### 3.1.3 IDS-R13MP

Čitalec IDS-R13MP je integriran analogni vmesnik za sisteme RFID. Podpira protokole ISO-15693, ISO 14443 in HF EPC. Uporablja se za konfiguracijo in branje zapisovalnika SL13A. Z računalnikom ga povežemo s vmesnikom USB, preko katerega se tudi napaja. Za uporabo njegovih funkcij potrebujemo grafični vmesnik, kot je SAL Read, prav tako iz podjetja IDS.



Slika 3.2: Čitalec R13MP.

Na sliki 3.2 lahko vidimo čitalec R13MP. Uporabili smo ga za testiranje zapisovalnikov, zagotavljanje pravilnega delovanja Android aplikacij ter zapisovanje podatkov na značke.

## 3.2 Programska oprema

### 3.2.1 Android

Danes je najbolj razširjen mobilni operacijski sistem, ki predstavlja dobro osnovo za ustvarjanje aplikacij, ko govorimo o podpori in masivni distribuciji [21]. Zaradi omenjenih razlogov in relativno enostavnega programiranja smo se odločili za uporabo te platforme. Aplikacije za Android se programirajo v jeziku Java, za njihov izgled pa se uporablja opisni jezik XML. Uporaba funkcij operacijskega sistema je dobro dokumentirana s strani spletnega giganta Google.

### 3.2.2 Eclipse in ADT

Eclipse je razvojno okolje za programiranje v jeziku Java. Omogoča tudi uporabo različnih vtičnikov, kar daje programerju možnost uporabe drugih jezikov, npr. Perl, Python, Ruby, itd. Eclipse je večinsko uporabljeno razvojno okolje za Android aplikacije in tudi za programiranje naše rešitve. Programiranje za Android zahteva Googlov vtičnik ADT (Android Development Tools). To je dodaten program, kateri v Eclipse nudi nabor orodij, ki so integrirane v razvojno okolje. Daje nam dostop do številnih funkcij, ki pripomorejo hitremu razvoju Android aplikacij. Njegov grafični vmesnik nam pomaga pri izdelavi uporabniškega vmesnika, hitrega razvoja prototipov in projektiranja. ADT skrbi za prevajanje Java kode v format, razumljiv operacijskemu sistemu in je za programiranje Android aplikacij nujen.

### 3.2.3 SAL Read

SAL Read je program, ki nam omogoča uporabo vseh ukazov v protokolu ISO-15693 in kombinacijo teh za konfiguriranje zapisovalnika SL13A. Na sliki 3.3 vidimo funkcije in parametre s katerimi nastavimo delovanje zapisovalnika. Za testiranje smo uporabili naslednje funkcije:

**Initialize** S tem ukazom pobrišemo stare meritve in inicializiramo zapisovalnik. Uporabijo se parametri:

- Delay Time - Čas zakasnitve začetka meritev, ko uporabimo ukaz Start Logging.
- Number of Blocks For User Data - Število blokov v pomnilniku za podatke uporabnika. V našem primeru uporabljamo za zapis kode EPC 3 bloke.
- Usage - Ali je zapisovalnik za enkratno ali večkratno uporabo. Če nastavimo to, potem lahko do pomnilniškega dela za meritve dostopa samo zapisovalnik.

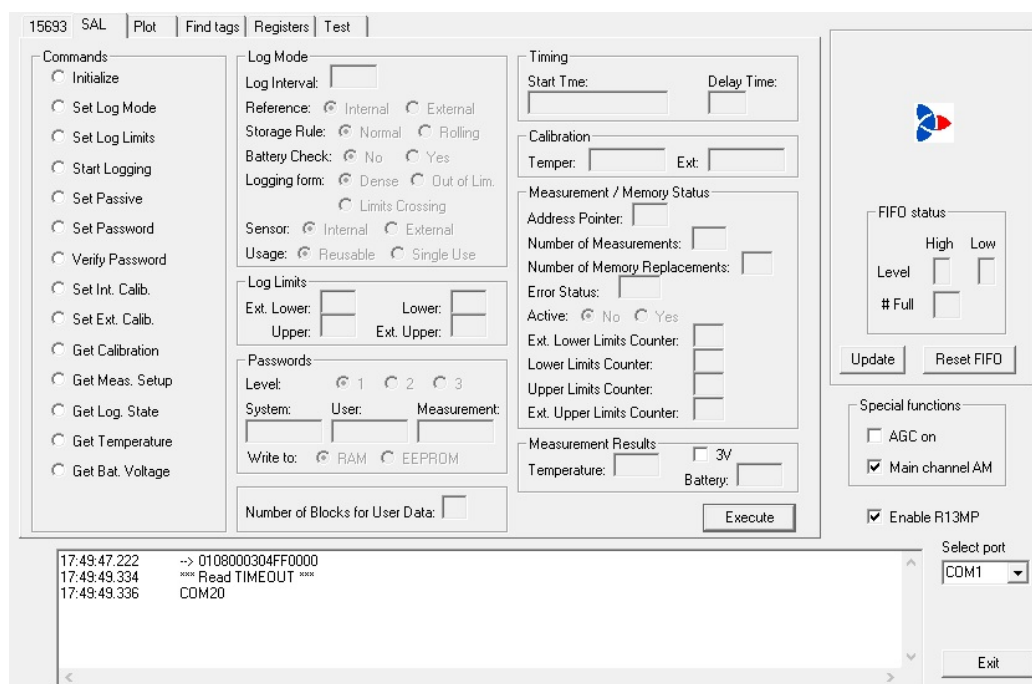
**Set Log Mode** Zapisovalniku s tem ukazom nastavimo način zapisovanja meritev. Pomembni parametri so:

- Log Interval - Čas med meritvami.
- Storage Rule - Ko enkrat presežemo velikost pomnilnika, ali prepisemo stare meritve.
- Logging Form - Parameter pove, kdaj zapisujemo temperature.
- Sensor - Ali se uporabi integrirani ali zunanji senzor.

**Set Log Limits** Meje temperatur, ki zapisovalniku povedo, kdaj naj shranjuje meritve.

**Start Logging** Ukaz, ki zapisovalniku pove, naj začne shranjevati meritve.

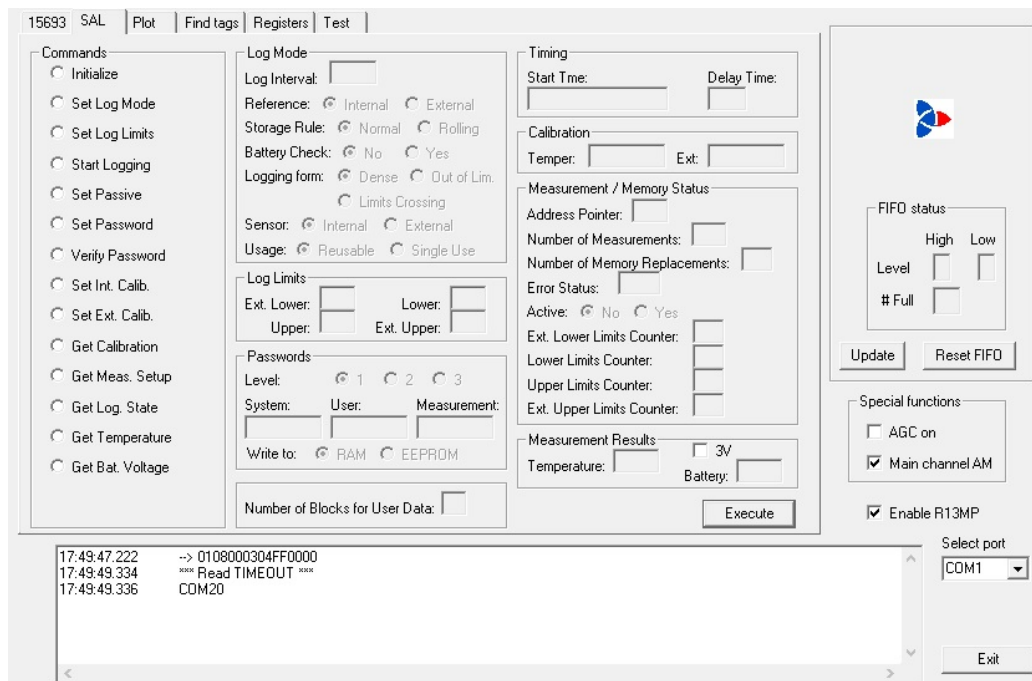
**Set Passive** Ukaz, ki zapisovalnik postavi v pasivno stanje, kjer ta ne shranjuje meritev.



Slika 3.3: Grafični vmesnik SAL Read.

Program omogoča tudi prikaz temperatur, izmerjenih na zapisovalniku, s katerim preverimo pravilno delovanje naše aplikacije. To lahko vidimo na sliki 3.4. Na levi strani so podatki o meritvah, kot je število meritev, začetek

shranjevanja, interval shranjevanja, itd.



Slika 3.4: Prikaz temperatur v programu SAL Read.

## 3.3 Tehnologije

### 3.3.1 RFID in NFC

Radiofrekvenčna identifikacija je tehnologija za prenos podatkov med čitalcem in elektronsko značko. Značka je sestavljena iz integriranega vezja, ki hrani in procesira podatke, ter izvaja modulacijo in demodulacijo signalov. Drugi del oddajnika je antena, ki sprejema in oddaja radijske signale. Signale RFID oddajnikov sprejema RFID čitalec, kar omogoča identifikacijo predmetov na katere je oddajnik pritrjen ali živih bitij, ki imajo vstavljen čip ('čipiranje domačih živali'). V primeru, da RFID znački pošljemo dovolj veliko energijo, lahko radijske valove pošljemo izjemno daleč, kar je zelo uporabno pri sledenju živali in vozil. Po drugi strani, ko želimo prenesti osebne in bančne podatke, takšen dolg dolet ni varen. Zato je bila razvita tehnologija Near

Field Communication (NFC).

NFC je visokofrekvenčna komunikacijska tehnologija kratkega dosega, ki omogoča izmenjavo podatkov na razdalji do 10 cm. NFC deluje pri frekvenci 13.56 MHz. Osnovana je na standardu ISO/IEC 14443 in združuje pametno kartico in čitalec v eno napravo. NFC naprava podpira tri načine delovanja [16]:

- Čitalec ali zapisovalec - V tem načinu lahko NFC naprava bere ali piše na pametne značke (ISO-15693 in ISO-14443).
- Emulator kartic - Naprava se v tem primeru obnaša kot pametna kartica, npr. bančna kartica (ISO-14443).
- Komunikacija med NFC napravami - Naprava lahko komunicira tudi z drugo NFC napravo (ISO-18092 in ISO-21481).

Zaradi svoje preprostosti se tehnologija vedno bolj uveljavlja v mobilnih napravah, saj potrebuje zelo malo vložka uporabnika. Omogoča veliko različnih aplikativnih funkcij, vendar je njena hitrost prenosa trenutno zelo omejena, zato je primerna za prenos krajših podatkov, kot so naprimer telefonske številke, tekstovni podatki in ukazi med napravami.

### 3.3.2 GS1 in EPC

Sistem GS1 je skupina medsebojno prepletenih standardov, ki temeljijo na globalni identifikaciji izdelkov, storitev in lokacij. Sistem GS1 je najširše uporabljeni standard v svetu, še posebej pa na področju preskrbovalnih verig. Z njim označujemo prodajne enote, storitve, lokacije, transportne enote, vračljivo embalažo, itd. Povezuje tehnologijo RFID in identifikator EPC [3].

EPC je elektronska koda izdelka, ki je zasnovana kot univerzalna identifikacijska oznaka. Njen zapis zagotavlja edinstveno identiteto vsakega fizičnega predmeta, kjerkoli na svetu, za vse večne čase. EPC je zaradi svoje dolžine in formata primerna za uporabo v pasivnih pametnih značkah, saj imajo te pogosto omejen pomnilniški prostor [12].

Pri označevanju je vedno zapisana kot 96 bitna koda, vendar jo lahko z dvostransko transformacijo spremenimo v človeku razumljivo obliko.

### 3.3.3 MySQL

MySQL je sistem za upravljanje s podatkovnimi bazami. To je odprtokodna implementacija relacijske podatkovne baze, ki za delo s podatki uporablja jezik SQL. MySQL deluje na principu odjemalec - strežnik, pri čemer lahko strežnik namestimo kot sistem, porazdeljen na več strežnikov. Obstaja veliko število odjemalcev, zbirk ukazov in programskih vmesnikov za dostop do podatkovne baze MySQL. Android ne podpira MySQL, zato moramo uporabiti drugo rešitev, kjer je ena od njih SQLite.

SQLite je sistem za upravljanje s podatkovnimi bazami vsebovan v majhni knjižnici in je zelo priljubljen za uporabo v majhnih, pomnilniško omejenih napravah, npr. mobilni telefoni, MP3 predvajalniki itd. Za razliko od ostalih sistemov za upravljanje podatkovnih baz SQLite ni ločen proces, do katerega dostopa aplikacija odjemalca, ampak je del le te.

V aplikaciji se SQLite uporablja za lokalno shranjevanje podatkov v primeru, da nimamo internetne povezave, in se na oddaljeni strežnik naložijo pozneje.

### 3.3.4 GPS

GPS (Global Positioning System) je satelitski navigacijski sistem, ki se uporablja za določanje točne lege in časa kjerkoli na Zemlji ali v zemeljski tirnici. Njegovi sateliti na potovanju okrog Zemlje uporabljajo srednjo krožno tirnico. Večina novejših mobilnih naprav ima vgrajen GPS čip.

V sistemu se GPS uporablja za primer, ko nimamo kode EPC za enolično določitev lokacije, zato je ta določena z latitudo in longitudo.

### 3.3.5 PHP

PHP je zelo razširjen odprtokodni skriptni programski jezik, ki se uporablja za strežniške storitve oziroma za razvoj dinamičnih spletnih vsebin. Podoben je strukturiranim programskim jezikom, najbolj jezikoma C in Perl, in najbolj

izkušanim programerjem dovoljuje razvijanje zapletenih rešitev problemov brez dolgega učenja.

V rešitvi se uporabljata dve PHP storitvi, ena za identifikacijo uporabnika ter druga za prenos podatkov. Deluje kot vmesni strežnik med mobilno napravo in podatkovno bazo, ki prejema ukaze in jih izvršuje nad oddaljeno podatkovno bazo in rezultate vrača uporabniku, ki jih je zahteval.

## Poglavje 4

# Mobilni aplikaciji TraSens

Za izvedbo nadzora temperatur sta bili razviti dve mobilni aplikaciji za operacijski sistem Android, TraSens in TraSens Manager. Zaradi enostavnejše arhitekture in uporabe smo se odločili za delitev funkcionalnosti na dve mobilni aplikaciji. Ime teh dveh aplikacij je izpeljanka dveh ključnih besed diplomske naloge, to sta Trace in Sensor, ki v angleščini pomenita sledljivost in zaznavanje.

Prva, TraSens, se uporablja za upravljanje s podatki zapisovalnika temperatur SL13A. Vključuje začetek in konec zajemanja podatkov in branje ter prikaz meritev temperatur. Dodatek k imenu druge aplikacije, Manager, označuje njeno uporabnost. TraSens Manager se uporablja za nastavljanje delovanja zapisovalnikov in zapisovanje kod EPC na zapisovalnike in pametne značke.

Obe aplikaciji sta narejeni z mislijo na čim bolj preprosto uporabo in povečanje hitrosti delovnega procesa. Grafična vmesnika sta zato izjemno enostavna in intuitivna za uporabo. Aplikaciji sta v celoti narejeni v angleškem jeziku, vendar nam način programiranja Android aplikacij omogoča hitro prevajanje, saj je ves viden tekst zapisan v eni datoteki XML.

## 4.1 TraSens

Aplikacija TraSens omogoča uporabniku tri funkcije, to so začetek beleženja, konec beleženja in pregled zabeleženih meritev. Sistem je narejen v treh slojih. V najnižjem sloju sta uporabnik in aplikacija TraSens. Na najvišjem sloju so podatkovne baze. Narejene so s programskim jezikom SQL, ki pa ga Android ne podpira, zato je bilo potrebno vključiti vmesni sloj (angl. middleware).

### 4.1.1 Načrtovanje

Načrtovanje aplikacije TraSens je potekalo v štirih delih. V prvem delu je bilo potrebno določiti podatke, ki bodo vključeni v ustrezno podatkovno strukturo, da bodo ti podatki smiselno povezani. Tukaj smo si zastavili štiri najpomembnejša vprašanja:

- KDO? - Meritve seveda potrebujejo podatek o tem kdo jih izvaja. Za to zadostuje koda EPC, ki enolično identificira uporabnika oziroma opremo (temperaturni zapisovalnik).
- KAJ? - Na vprašanje kaj odgovorimo s kodo EPC, ki pove na kaj se te meritve nanašajo. To je lahko izdelek ali pa logistična enota, kjer se ta izdelek nahaja.
- KJE? - Potrebujemo še podatke o lokaciji, to je lahko koda EPC ali koordinata v svetovnem koordinatnem sistemu.
- KDAJ? - Zadnji pomemben podatek je čas meritev, ki je določen z datumom in uro, ko zaženemo in ugasnemo zapisovalnik.

Zapisovalnik ima tudi funkcijo zamika začetka zapisovanja, zato je začetni čas določen kot vsota trenutnega časa in zamika. Najbolj pomemben podatek pa so seveda meritve, katerim so dodani vsi prej naštetih podatki. Vse skupaj tvori vsebinsko celoto za nadzor temperatur v hladni verigi.

Ko govorimo o optimalni strukturi podatkovne baze, imamo v mislih čim manj ponavljanja podatkov ter čim manjše število operacij za upravljanje z njimi. Ker je večino prej naštetih podatkov enoličnih, kot so kode EPC, lokacije, časi in meritve, je za zapis teh podatkov zadostna že ena tabela. V tabeli je en vnos za množico temperatur, ki so zgrajene v en niz po določenem pravilu, po katerem se niz tudi razčleni. Zadnji potreben podatek je bit, ki pove ali je merjenje že končano ali je še v teku. Tako dobimo optimalno zgradbo podatkovne baze, po kateri za iskanje zadošča enostaven stavek SQL SELECT, ki primerja po edinih enoličnih podatkih, ki jih imamo ob začetku in koncu merjenja. To sta koda EPC zapisovalnika in začetni čas merjenja.

Zasnovali bi lahko tudi drugačno rešitev z dvema tabelama, kjer bi bili v prvi vsi podatki v povezavi z merilnikom. Vse meritve pa bi bile v drugi tabeli, po pravilu en vnos za eno izmerjeno temperaturo, ki je povezana s prvo preko enoličnega identifikatorja. V tem primeru bi za vnos temperatur potrebovali  $N$  operacij, kjer je  $N$  število izmerjenih temperatur. To pa je zelo časovno potratno.

V drugem delu načrtovanja je bilo potrebno izdelati knjižnico za upravljanje z zapisovalnikom SL13A. Ker je bila na voljo priložena dokumentacija podjetja IDS in ker deluje po standardu ISO-15693, ki je dobro dokumentiran in podprt s strani Androida, komunikacija z zapisovalnikom ni predstavljala večjih težav. Problemi pri komunikaciji nastanejo takrat, ko naprava zazna in prebere zapisovalnik iz prevelike razdalje in takrat pogosto pride do napak.

Podatke iz zapisovalnika je potrebno nato shraniti v podatkovno bazo, ki smo jo definirali v prvem koraku. Za povpraševalni jezik za delo s podatkovnimi bazami smo se odločili za SQL, saj je ta enostaven, dobro podprt in prost za uporabo. Upravljanje s podatkovno bazo SQL poteka preko vmesnega sloja, to je strežnik s storitvama PHP, ki uporabniku posredujeta zahtevane podatke iz baze. Prva storitev se uporablja za identifikacijo, druga pa za vstavljanje podatkov v bazo.

Za vključitev v sistem sledljivosti RFID-F2F smo preučili zgradbo XML datotek, ki opisujejo dogodke vključitve v sistem. Potrebno je bilo tudi

programiranje gradnje identičnih XML datotek, ki jih ob dogodkih naložimo v repozitorij EPCIS.

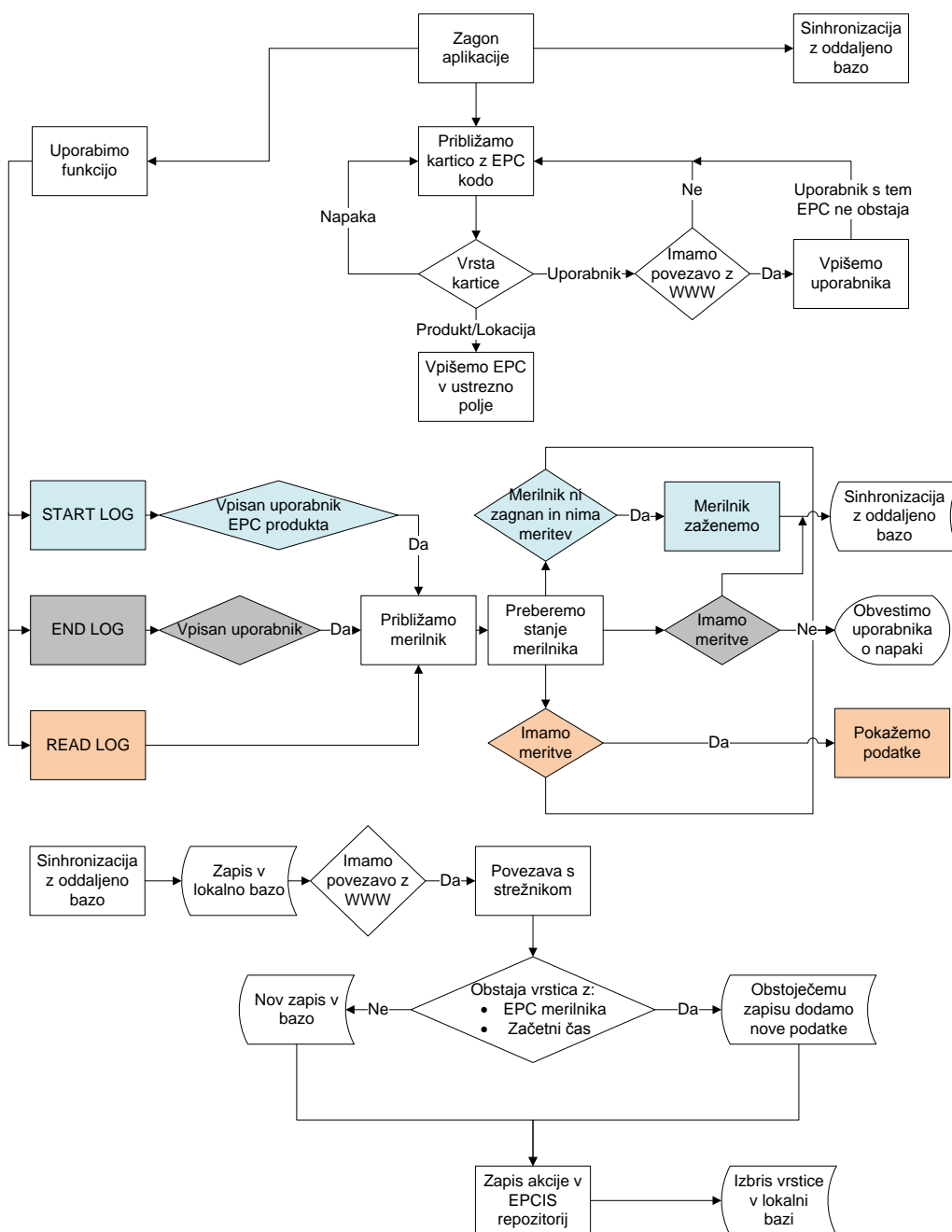
V zadnjem delu smo načrtovali grafični vmesnik, ki je preprost, hiter in mora prikazati vse pomembne podatke. Tako smo prišli do izgleda, ki je prijazen za oko, ni zapleten in ponuja celovit pregled nad pomembnimi podatki. Gumbi se ob pritisku obarvajo modro, ob branju zapisovalnika, ki včasih traja dlje časa, pa se prikaže zeleno pojavno okno, ki uporabniku pove, da poteka branje podatkov. Za prikaz temperatur smo uporabili knjižnico `AndroidPlot`, ki temperature prikaže kot točke na grafu.

Delovanje aplikacije je predstavljeno na sliki 4.1

### 4.1.2 Internet

Kljub vedno boljši pokritosti mobilnih omrežij, še vedno lahko pride do primerov, ko nimamo dostopa do interneta. Zato se ne moremo zanašati, da bomo lahko podatke vedno takoj naložili na oddaljeni strežnik. Da ne bi prišlo do izgube podatkov, potrebujemo lokalno shranjevanje le teh, dokler jih nimamo možnosti shraniti na strežnik. Tukaj pride v poštev knjižnica `SQLite`, ki je primerna za upravljanje manjših podatkovnih baz. Taka podatkovna baza je shranjena na mobilnem telefonu. V to bazo se v prvi fazi shranijo vsi podatki, ne glede na to ali imamo internet ali ne. Glede na izpolnjenost tega pogoja se ti podatki nato prenesejo v podatkovno bazo na oddaljenem strežniku. V tem koraku se ustvari tudi XML datoteka dogodka, ki se prenese v repozitorij EPCIS. Sinhronizacija baze je skicirana na sliki 4.1. Opis sinhronizacije podatkovnih baz se nahaja v poglavju 4.1.9.

S takim pristopom lahko v sistemu uporabljamo tudi mobilne naprave brez kartice SIM, kot so npr. tablični računalniki. Take naprave imajo dostop še bolj omejen, saj imajo povezavo z internetom le preko brezžičnih omrežij, če pa pazimo na varnost podatkov, se ta množica zoži na samo znana omrežja.



Slika 4.1: TraSens, graf prehodov.

### 4.1.3 Lokacija

Ob vsaki akciji, ki sproži shranjevanje v bazo, potrebujemo tudi lokacijo dogodka. V večini scenarijev našo lokacijo določa koda EPC, ki je zapisana na pametni znački. Lahko pa pride do primera, da te značke nimamo oziroma je ne preberemo. V tem primeru potrebujemo nek varnostni mehanizem, ki prepreči shranjevanje podatkov brez lokacije. To dosežemo z branjem lokacije telefona. Ob zagonu aplikacije se zažene tudi GPS senzor, ki poskuša čim bolj natančno pozicionirati, kje se nahaja mobilni telefon. Včasih zaradi zaprtosti prostora GPS senzor ne deluje, v tem primeru se prebere groba lokacija, ki je dobljena iz lokacije antene mobilnega omrežja, preko katere smo povezani. Lokacija se izboljša, če smo v bližini brezžičnih omrežij. GPS senzor oziroma grobo pozicioniranje nam vrneto zemljepisno dolžino in širino (angl. longitude, latitude), ki se v tej obliki tudi zapišeta v podatkovno bazo.

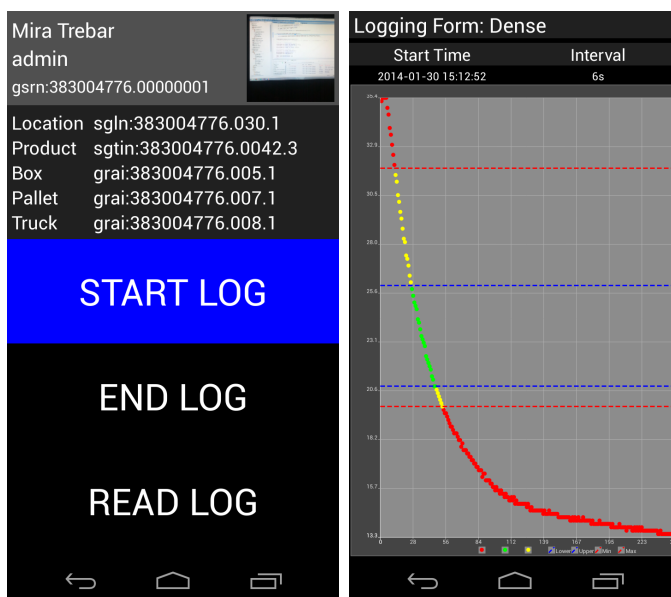
Če je GPS tehnologija na telefonu izklopljena, nas aplikacija ob zagonu na to opozori in brez vključitve ne moremo nadaljevati.

### 4.1.4 Prijava

Uporabnik je določen z enolično kodo EPC, ki je zapisana na pametni kartici. Na tak način ne potrebuje uporabniškega imena in gesla. V sistem se prijavi tako, da kartico približa telefonu. Ta nato kodo EPC pošlje storitvi PHP, ki v podatkovni bazi preveri, če uporabnik s to kodo obstaja. Storitev nato vrne ime in priimek uporabnika, ter njegovo vlogo v podjetju. Delovanje glede na vlogo ni spremenjeno, z manjšim dodatkom kodi aplikacije pa se lahko doda tudi ta funkcija. Za uporabo funkcij začetek (START LOG) in konec merjenja (END LOG) mora biti uporabnik prijavljen v sistem. Za prijavo je potreben dostop do interneta, vendar prijava poteka samo ob prvi uporabi aplikacije. Takrat se podatki o uporabniku zapišejo na telefon in tam ostanejo, dokler aplikacije ne odstranimo oziroma se vpišemo s kodo drugega uporabnika.

### 4.1.5 Grafični vmesnik

Začetni zaslon je razdeljen na tri dele: uporabniški podatki, kode EPC in funkcije, kot to prikazuje slika 4.2a. Uporabniški podatki zajemajo ime in priimek uporabnika, njegovo vlogo, identifikator EPC in sliko, ki jo lahko zajamemo s pritiskom na okvir. Ta slika se shrani na pomnilnik telefona in se nato prikazuje v okvirju za sliko. Kode EPC na začetnem zaslonu ločimo v dve skupini, to sta koda lokacije ('Location') in koda objekta. Koda objekta je lahko vezana na produkt ('Product'), zaboj ('Box'), paleta ('Pallet') ali prevozno sredstvo ('Truck'). Za uspešno delovanje sistema potrebujemo vsaj eno kodo objekta. Če te ni, nam aplikacija ne dovoli uporabe funkcije za začetek merjenja. Preostali prostor zasedajo tri velike tipke. Te tipke se ob pritisku obarvajo modro. Vmesnik je narejen tako, da uporabnik samo določi akcijo, ki se bo izvedla, ko približamo zapisovalnik. Na ta način se izognemo dodatnim zaslonom in pojavnim oknom, ki zmedejo uporabnika.



(a) Začetni zaslon.

(b) Graf meritev.

Slika 4.2: TraSens

Uporabniški vmesnik ima še četrti del, ki ni viden, in omogoča branje

kod EPC. Branje kode se zgodi vsakič ko približamo značko RFID. V tem trenutku se preberejo prvi trije bloki značke, ti podatki se prevedejo v identifikator EPC, ki se v primeru, da ni napake pri prevajanju, zapiše na ustrezno mesto v okviru za kode EPC.

#### 4.1.6 Začetek meritev (START LOG)

Proces hladne verige se začne s funkcijo START LOG. Za njeno uporabo moramo biti v sistem prijavljeni in prebran mora bit vsaj en EPC, ki je povezan s produktom. Sedaj lahko pritisnemo na tipko START LOG, približamo zapisovalnik in nato si dogodki sledijo v naslednjem zaporedju:

1. Zaposovalniku pošljemo ukaz, naj prebere stanje meritev, aktivnost zapisovanja in kodo EPC.
2. Če so na zapisovalniku še vedno meritve od prejšnega procesa hladne verige (zapisovalnik je še aktiven), če je napačna koda EPC ali če je prišlo do napake pri prenosu nas aplikacija o tem obvesti. Dokler tega ne popravimo, ne moremo nadaljevati s postopkom.
3. Zapisovalniku pošljemo ukaz za začetek beleženja.
4. Če pri prenosu ni prišlo do napake, lahko podatke shranimo v lokalno bazo in zaženemo sinhronizacijo baz.

V podatkovno bazo se zapiše začetni čas, ki je prebran iz zapisovalnika in ne tisti, ki je izmerjen v aplikaciji. To pa zato, ker je med izmerjenim časom in časom zagona zapisovalnika manjši zamik, kar bi povzročalo probleme pri operaciji END LOG, saj časa ob izvedbi začetka meritev ne bi imeli in bi bila sinhronizacija baz nemogoča. S tem je opravljen začetek beleženja.

### 4.1.7 Konec meritev (END LOG)

Za uporabo te funkcije moramo biti prijavljeni v sistem. Ko približamo zapisovalnik z izbrano tipko END LOG, pošljemo zapisovalniku zahtevo za njegovo stanje. Če nam pove, da je beleženje v teku, lahko nadaljujemo, sicer nas aplikacija obvesti o napaki. Nato preberemo njegov EPC, začetni čas ter časovni zamik zapisovanja. Sedaj lahko izračunamo enak čas, ki smo ga zapisali v bazo pri začetku beleženja. Nato iz zapisovalnika preberemo temperature. Če ni prišlo do napake pri branju, zapisovalnik postavimo v pasivno stanje. Dobljene podatke shranimo v lokalno podatkovno bazo in zaženemo sinhronizacijo baz. S tem se proces hladne verige konča, meritve pa se naložijo na strežnik.

### 4.1.8 Pregled meritev (READ LOG)

V procesu hladne verige lahko preverimo stanje meritev na SL13A zapisovalniku. Ta funkcija je narejena za spremljanje temperatur, čeprav nočemo ali ne želimo zaključiti procesa. Za njeno uporabo ni potrebno biti prijavljen v sistem, pravtako ni potreben noben podatek o produktu. Potrebujemo le dostop do zapisovalnika. Ta ukaz je zaradi hitrosti prenosa podatkov in njihove velikosti najbolj časovno zahteven, zato pri njem največkrat pride do napak. Ko pritisnemo na tipko READ LOG in približamo telefon zapisovalniku se temu pošlje ukaz za branje števca meritev. S tem dobimo število izmerjenih temperatur in iz tega lahko izračunamo, koliko blokov moramo prebrati iz zapisovalnika. Števec meritev je 10 bitna številka, ki pa se ob prelivu nastavi na 0. Narašča tudi po tem, ko je pomnilnik že zapolnjen z meritvami, zato moramo predvideti tudi ta primer. To naredimo tako, da iz intervala, zamika in začetka beleženja izračunamo približno število meritev do trenutka branja, ki sicer ni točno zaradi natančnosti ure zapisovalnika. Če je dobljena vrednost večja od 1023, vzamemo največje možno število meritev.

Branje se izvede v naslednjem koraku, seveda če zapisovalnik sploh ima podatke temperatur. V nasprotnem primeru obvestimo uporabnika o napaki.

Podatki, ki smo jih dobili iz zapisovalnika so 10 bitni zapisi števil, ki jih moramo še pretvoriti v stopinje celzija. Pretvorba je določena z enačbo:

$$T(^{\circ}C) = code * 0.169 - 92.7 - 5.4 \quad (4.1)$$

Ko imamo temperature v  $^{\circ}C$ , jih prikažemo v grafu temperatur v odvisnosti od zaporedne številke meritve.

Merilnik SL13A pozna več načinov shranjevanja temperatur. Verjetno najbolj prostorsko potraten je zgoščeni način (angl. dense), ki za vsak časovni interval shrani temperaturo v pomnilnik. V tem primeru so v vsakem bloku zapisane po tri temperature. Zaporedne številke si sledijo od 1 naprej. V večini primerov manj prostorsko potraten način shranjuje temperature samo takrat, ko so te izven dovoljenih vrednosti (angl. out of limits). V tem primeru ima vsak blok v sebi zapisano temperaturo in zaporedno številko meritve. Najmanj potraten način pa shranjuje le temperature, ko te prehajajo preko dovoljenih mej (angl. limits crossing). Tukaj je delitev blokov enaka kot pri prejšnem primeru.

Podatke o dovoljenih vrednostih temperatur ne prikažemo pri zgoščevalnem načinu, te le izrišemo na graf. To je prikazano na sliki 4.2b

#### 4.1.9 Sinhronizacija podatkovnih baz

Sinhronizacija lokalne in oddaljene podatkovne baze deluje kot nit, ki jo zaženemo vsakič, ko to zahteva funkcija za začetek ali konec beleženja oziroma ob zagonu aplikacije. Če internetna povezava ne obstaja, se postopek ustavi. V nasprotnem primeru se za vsak vnos v lokalni bazi izvedejo naslednji koraki:

1. Preverimo, če v oddaljeni bazi obstaja vnos z enako kodo EPC zapisovalnika in začetnim časom.
2. Če obstaja, ga posodobimo z novimi podatki.
3. Če ne obstaja, ustvarimo nov vnos s podatki iz lokalne baze.

4. Če ni napak, zgradimo XML datoteko z dogodkom EPCIS in jo naložimo v repozitorij. Primer za začetek beleženja je predstavljen na sliki 4.3. Ob koncu beleženja se spremeni samo polje *action* iz ADD v DELETE.

5. Če ni napak, vnos izbrišemo iz lokalne podatkovne baze.

```
<?xml version="1.0" encoding="UTF-8"?>
<epcis:EPCISDocument
xmlns:epcis="urn:epcglobal:epcis:xsd:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
creationDate="2013-12-30T15:13:02"
schemaVersion="1.0">
  <EPCISBody>
    <EventList>
      <AggregationEvent>
        <eventTime>2013-12-30T15:13:02+01:00</eventTime>
        <eventTimeZoneOffset>+01:00</eventTimeZoneOffset>
        <parentID>
          http://wmrfid.org/farm2fork/383004776/asset/004/3
        </parentID>
        <childEPCs>
          <epc>http://wmrfid.org/farm2fork/383004776/product/0042/3</epc>
          <epc>http://wmrfid.org/farm2fork/383004776/asset/005/1</epc>
          <epc>http://wmrfid.org/farm2fork/383004776/asset/007/1</epc>
          <epc>http://wmrfid.org/farm2fork/383004776/asset/008/1</epc>
        </childEPCs>
        <action>ADD</action>
        <bizStep>
          http://wmrfid.org/farm2fork/383004776/bizstep/shipping_processing
        </bizStep>
        <disposition>
          http://wmrfid.org/farm2fork/383004776/disposition/active
        </disposition>
        <readPoint>
          <id>http://wmrfid.org/farm2fork/383004776/location/030/1</id>
        </readPoint>
        <bizLocation>
          <id>http://wmrfid.org/farm2fork/383004776/location/030/1</id>
        </bizLocation>
        </AggregationEvent>
      </EventList>
    </EPCISBody>
  </epcis:EPCISDocument>
```

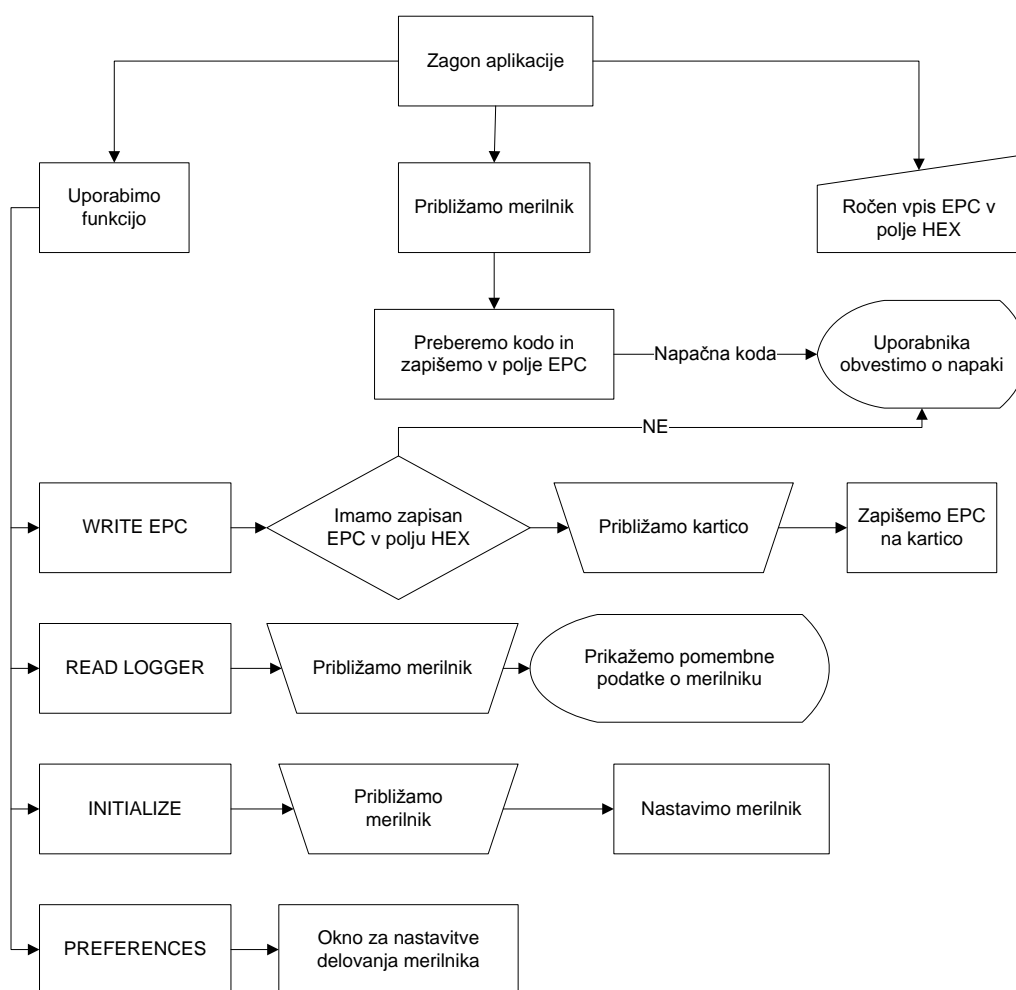
Slika 4.3: EPCIS XML dogodka ADD

Vsi koraki, razen zadnjih dveh potekajo preko PHP storitve, kateri pošljemo zgrajen SQL stavek, ta pa nam vrne 1 oziroma 0 v primeru napake. S takim vrstnim redom zagotovimo, da bodo podatki v oddaljeni bazi enaki, ne glede na to kdo je poslal nek podatek. Lahko se namreč zgodi, da beleženje ni začeto z istim telefonom kot je končano. Poskrbeti moramo le, da sta imela oba telefona po dogodkih vsaj enkrat povezavo z internetom. Datoteko

XML z dogodkom pošljemo repozitoriju EPCIS z metodo POST protokola HTTP neposredno iz naprave Android. Na repozitoriju EPCIS tečejo spletne storitve Fosstrak [17], ki obdelajo našo zahtevo. Če se je dogodek uspešno zapisal v repozitorij, nam Fosstrak vrne statusno kodo 200 (OK), sicer 400 (BAD REQUEST) ali 500 (INTERNAL SERVER ERROR). Če pride do napake pri SQL ali XML zahtevkih, se vnos iz lokalne baze ne izbriše in bo tam ostal do naslednje sinhronizacije baz.

## 4.2 TraSens Manager

Druga aplikacija rešitve je TraSens Manager. Z njo v končni fazi sistema popolnoma odpravimo potrebo po R13MP čitalcu. S to aplikacijo zapisujemo kodo EPC na pametne značke in zapisovalnik SL13A. Koda mora biti v heksadecimalnem formatu. Prav tako lahko z njo nastavljammo delovanje zapisovalnika in te nastavitve tudi preverjamo. Na sliki 4.4 je predstavljen postopek delovanja aplikacije TraSens Manager.

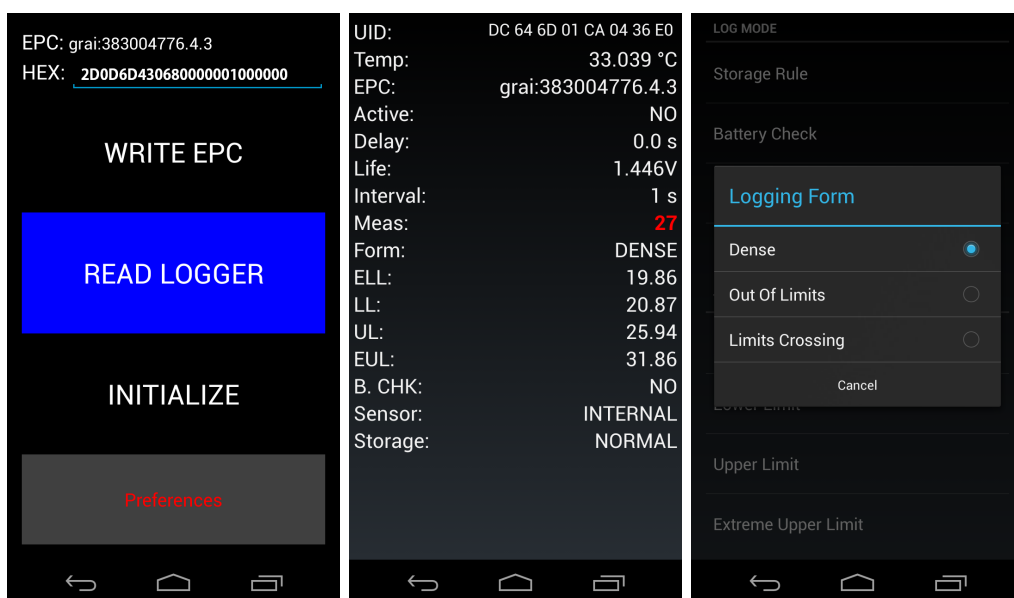


Slika 4.4: TraSens Manager, graf prehodov.

TraSens Manager ni namenjen končnemu potrošniku, niti ne prevozniku, ki je odgovoren za ohranjanje hladne verige. S to aplikacijo lahko ponastavljamo zapisovalnik in brišemo njegove podatke, kar bi v rokah napačnega uporabnika lahko privedlo do goljufanja in prirejanja meritev.

### 4.2.1 Grafični vmesnik

Grafični vmesnik pri TraSens Manager je podoben kot pri aplikaciji TraSens, kot to prikazuje slika 4.5a. Izbrane operacije se obarvajo modro, drugače se obnaša le tipka Preferences, ki odpre novo okno z nastavitvami. Polje HEX je namenjeno heksadecimalnemu zapisu kode EPC, ki se bo zapisala na pametno kartico RFID ali tudi na zapisovalnik. Ob samem vnosu lahko v to polje zapišemo vse znake, ne samo heksadecimalne, vendar nas bo aplikacija opozorila na napako pri izbiri funkcije za zapis kode EPC.



(a) Začetni zaslon.

(b) Podatki zapisovalnika.

(c) Nastavitve.

Slika 4.5: TraSens Manager

### 4.2.2 Podatki o zapisovalniku (READ LOGGER)

Z ukazom READ LOGGER preberemo vse podatke o zapisovalniku, ki jih lahko urejamo v nastavitvah. Na sliki 4.5b vidimo primer branja podatkov zapisovalnika. Prikazani so naslednji podatki:

| Podatek              | Ime           | Vrednost                                |
|----------------------|---------------|---|
| ID naprave           | UID           | heksadecimalna koda                     |
| Trenutna temperatura | Temp          | stopinje Celzija                        |
| Koda EPC             | EPC           | identifikator GS1                       |
| Aktivnost beleženja  | Active        | YES/NO                                  |
| Zamik beleženja      | Delay         | sekunde                                 |
| Napetost baterije    | Life          | volt                                    |
| Interval beleženja   | Interval      | sekunde                                 |
| Število meritev      | Meas          | 0 do 756                                |
| Način beleženja      | Form          | DENSE,OUT OF LIMITS,<br>LIMITS CROSSING |
| Meje temperatur      | ELL,EL,UL,EUL | stopinje Celzija                        |

Tabela 4.6: Podatki zapisovalnika.

Meje temperatur so lahko definirane kot:

- ELL - Skrajno nizka meja (angl. extreme lower limit)
- LL - Nizka meja (angl. lower limit)
- UL - Visoka meja (angl. upper limit)
- EUL - Skrajno visoka meja (angl. extreme upper limit)

Če je zapisovalnik še vedno aktiven, oziroma so na njem stare meritve, nas aplikacija o tem opozori tako, da te vrednosti obarva rdeče. Če je vsaj en podatek rdeče obarvan, zapisovalnik ni pripravljen za ponoven vstop v proces hladne verige. Rdeče se obarva tudi podatek o kodi EPC, če ta na zapisovalniku ni zapisana. Med podatki je tudi električna napetost baterije,

na katero pa mora uporabnik paziti sam, saj se ta pri nižjih vrednostih ne obarva rdeče. Napetost baterije od zapisovalnika dobimo kot 8 bitno številko, ki jo pretvorimo v napetost (V) po enačbi:

$$U(V) = code * 3.35mV + 860mV \quad (4.2)$$

### 4.2.3 Inicializacija zapisovalnika (INITIALIZE)

Z inicializacijo zapisovalnika tega ponastavimo in pripravimo na naslednji proces hladne verige. Ko imamo izbrano tipko INITIALIZE in približamo zapisovalnik, se ta postavi v pasivno stanje, stare meritve se pobrišejo in naložijo se nove nastavitve, ki smo jih prej nastavili v oknu Preferences. Rezervirajo se tudi trije bloki v pomnilniku za zapis kode EPC, ki jo lahko nanj zapišemo z uporabo ukaza WRITE EPC. Sedaj je zapisovalnik pripravljen za uporabo in čaka na ukaz START LOG aplikacije TraSens.

### 4.2.4 Nastavitve zapisovalnika

V oknu Preferences nastavljamo načine delovanja zapisovalnika SL13A. Nastavlja se z množico specializiranih ukazov, s katerimi dostopamo do sistema zapisovalnika. Operacijski sistem Android ponuja elegantno rešitev, kjer lahko to naredimo s enim klicem metode *transcieve*, ki ji nastavimo argumente kot so ukazna koda in parametri ukaza. Ti parametri so:

**Storage Rule** Način shranjevanja

- Normal - beleži vse meritve, dokler se pomnilnik zapisovalnika ne napolni. (Privzeto)
- Rolling - ko se pomnilnik napolni, prepisuje stare meritve.

**Battery Check** Podatku o temperaturi doda podatek o napetosti

- Da - Se ne uporablja pri načinu beleženja Dense.
- Ne - (Privzeto)

**Logging Form** Način beleženja

- Dense - beleži vse meritve. (Privzeto)
- Out of Limits - beleži samo tiste meritve, ki so izven določenih meja.
- Limits Crossing - beleži samo meritve, ki prečijo dovoljene meje.

**Sensor** Uporabljen termometer

- Internal - uporabi integriran termometer. (Privzeto)
- External - uporabi zunanji termometer.

**Limits** Dovoljene vrednosti temperatur na intervalu  $[-90,82]^{\circ}\text{C}$ 

- Extreme Lower Limit - skrajna spodnja meja
- Lower Limit - spodnja meja
- Upper Limit - zgornja meja
- Extreme Upper Limit - skrajna zgornja meja

**Interval** Interval beleženja temperatur na intervalu  $[0,32767]$  sekund. Tu je treba paziti, saj vpisana 0 pomeni interval 1 sekunda.

**Delay Time** Zakasnitveni čas začetka beleženja na intervalu  $[0,4095] * 8.53$  minut. (Privzeto 0 sekund)

Okno za nastavitve je prikazano na sliki 4.5c, ki kaže izbiro za način beleženja.



# Poglavje 5

## Testiranje

### 5.1 Mobilne naprave

Aplikaciji TraSens in TraSens Manager delujeta na vseh Android napravah, ki podpirajo tehnologijo NFC in GPS. Na napravah mora teči operacijski sistem Android verzije 13 (3.2) ali novejši.

Aplikacijo TraSens smo testirali na naslednjih napravah:

**Samsung Galaxy Nexus, Android 4.4.2** Aplikacija deluje normalno.

Velikost zaslona te naprave je bila uporabljena tudi pri načrtovanju grafičnega vmesnika v razvojnem okolju Eclipse, zato ni odstopanj od predvidevanega izgleda.

**Asus Nexus 7, Android 4.2** Aplikacija deluje normalno. Zaradi večjega zaslona ni bistvenih razlik od predvidevanega izgleda.

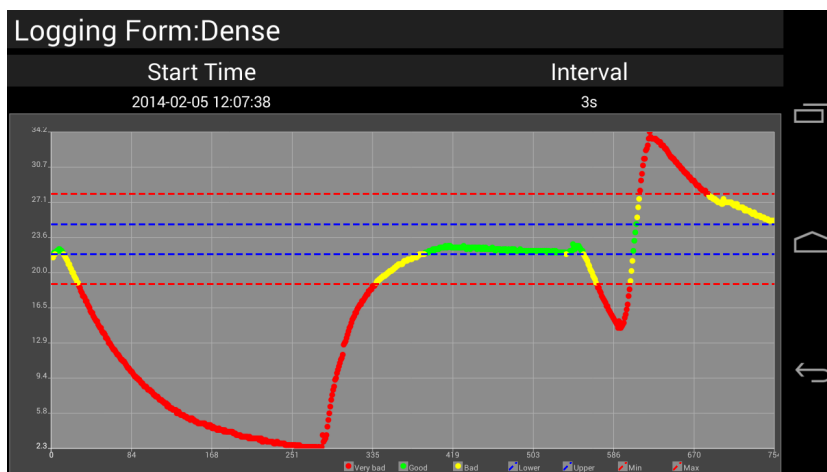
**Samsung Nexus S, Android 4.1.2** Aplikacija deluje normalno, ne deluje pa povezava z internetom, če smo v omrežje povezani s tehnologijo 3G. Napaka je v operacijskem sistemu Android 4.1.2, kjer javanski razred *HttpClient* ne deluje pravilno. Napako bi lahko odpravili z uporabno javanskega razreda *HttpURLConnection*. Zaradi te napake ne deluje prijava v aplikacijo in sinhronizacija podatkovnih baz. Za uporabo teh

se moramo povezati z WiFi omrežjem. Zaradi manjšega zaslona so se pojavile napake v grafičnem vmesniku, ki pa smo jih odpravili.

## 5.2 Načini beleženja

Pri načinih delovanja zapisovalnika se najbolj razlikujejo načini beleženja, zato smo se odločili, da te podrobno analiziramo in testiramo.

**Dense** Pri tem načinu beleženja je število zabeleženih meritev enako števcu meritev, ki se povečuje avtomatsko z intervalom. Zato imamo vedno znano število meritev, ki so zapisane v pomnilniku. Edini problem nastane, ker se števec meritev povečuje tudi po tem, ko se pomnilnik zapolne. Zaradi prostorske omejitve števec ni neskončen, vendar se pri vrednosti 1024 ponastavi na 0 in šteje naprej. Zato moramo število meritev omejiti s časovnim pragom, kot je to opisano v poglavju 4.1.8. Pri tem načinu nato ne prihaja do napak branja. Način ‘Dense’ najbolj prikaže, kaj se je dogajalo s temperaturo in je uporaben v večini scenarijev.



Slika 5.1: Dense: TraSens

**Out Of Limits** Pri načinu ‘Out Of Limits’ je število meritev drugačno od števca meritev, saj se meritve ne beležijo, ko je temperatura med zgornjo in spodnjo dovoljeno mejo. Zato moramo število zabeleženih meritev izračunati iz števcov, ki merijo, koliko meritev je bilo v vsakem intervalu med dovoljenimi mejami, razen v intervalu od spodnje do zgornje meje, na sliki 5.2 je to območje med modrima črtama. Tak izračun nam pove točno število zabeleženih meritev. Ti števci se obnašajo enako kot števec meritev, le da ti štejejo le do 256. To pomeni, da aplikacija v nekaterih primerih ne bo prebrala vseh meritev. Ti primeri pa so redki in se v testiranjih niso pojavili.

Način beleženja je uporaben, ko nas zanima, do katere vrednosti se je temperatura dvignila oz. spustila, ko ta ni bila v dovoljenih območjih. Pri dobri hladni verigi na zapisovalniku ne bo zabeleženih meritev.



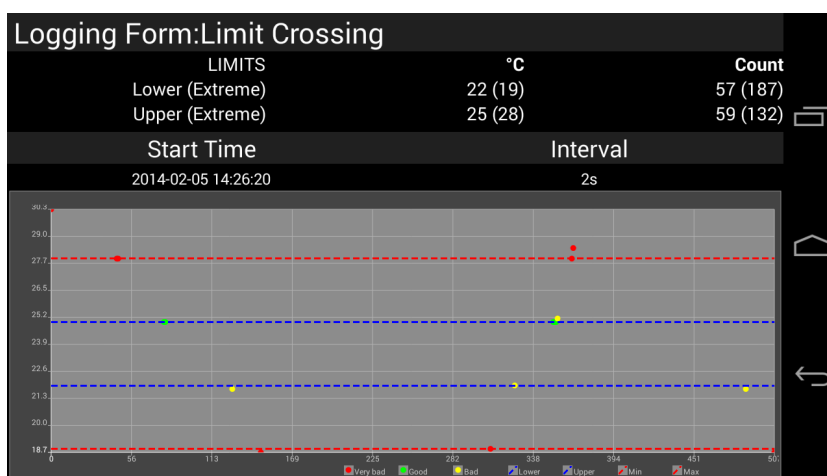
Slika 5.2: Out Of Limits: TraSens

**Limits Crossing** Pri načinu ‘Limits Crossing’ se meritve v pomnilnik beležijo le, ko temperatura prehaja preko dovoljenih mej. Na sliki 5.3 so to rdeče in modre črte. Iz zapisovalnika ne moremo prebrati podatka, ki nam pove, koliko meritev je zapisanih v pomnilniku. Zato lahko pravilno prikazemo meritve le, če pred začetkom beleženja pobrišemo vse

bloke v pomnilniku in ob branju preberemo vsakega, dokler ne naletimo na prazen blok. Vendar pa zapisovalnik SL13A ne podpira zapisovanja v več blokov naenkrat, zato moramo zapisovalniku poslati ukaz za vsak blok posebej, kar pa v realnosti traja približno 10 sekund. V tem času moramo zapisovalnik držati tesno ob telefonu, da ne pride do napak. Ponastavitev pomnilnika se v primeru načina ‘Limits Crossing’ naredi ob uporabi funkcije START LOG.

Ta način beleženja je uporaben v scenarijih, ko nas zanima le, ali je temperatura presegla dovoljene meje, ali pa takrat, ko imamo na zapisovalniku veliko lastnih podatkov in za meritve ostane precej manj pomnilniškega prostora.

Na sliki 5.3 je predstavljen graf meritev z načinom beleženja ‘Limits Crossing’. Na sliki opazimo, da manjka meritev čez modro črto pri zadnjem spuščanju temperature. Razlog za to je napaka v zapisovalniku.



Slika 5.3: Limits Crossing: TraSens

Večino problemov pri beleženju lahko rešimo s predvidevanjem, koliko časa bo hladna veriga trajala, in primerno temu nastavimo interval beleženja.

## 5.3 Aplikacija ScmContext

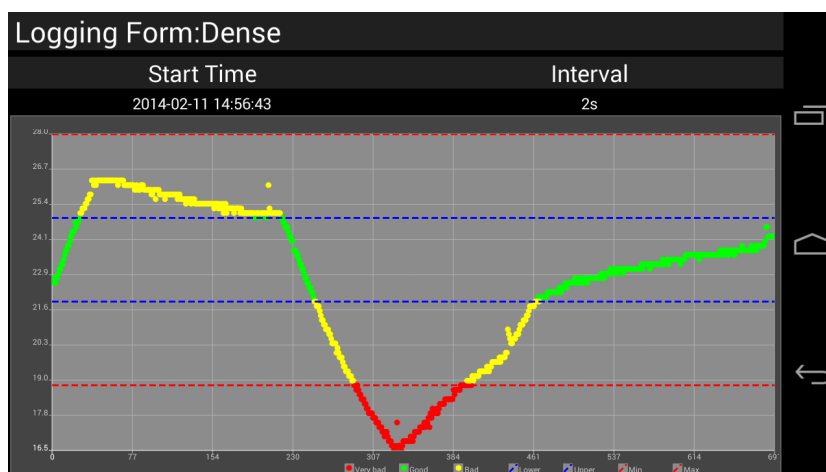
Aplikacija ScmContext nam omogoča pregled nad hladno verigo in podatki o vseh subjektih, povezanih z njo. Za testiranje aplikacije TraSens v povezavi s spletno aplikacijo ScmContext smo uporabili naslednje nastavitve zapisovalnika:

- Način beleženja - Dense
- Interval beleženja - 2 sekundi

Preden smo hladno verigo začeli, smo v aplikaciji TraSens prebrali naslednje kode EPC:

- Uporabnik - urn:epc:id:gsrc:383004776.00000002
- Produkt - urn:epc:id:sgtin:383004776.0042.257
- Lokacija - urn:epc:id:sgln:383004776.099.1 (to kodo EPC smo prebrali tudi ob koncu hladne verige)

Nato smo za čas 25 minut zapisovalnik postavili v okolje s spremenljivo temperaturo. Ob končanju hladne verige smo z mobilno napravo prebrali meritve iz zapisovalnika (slika 5.4).



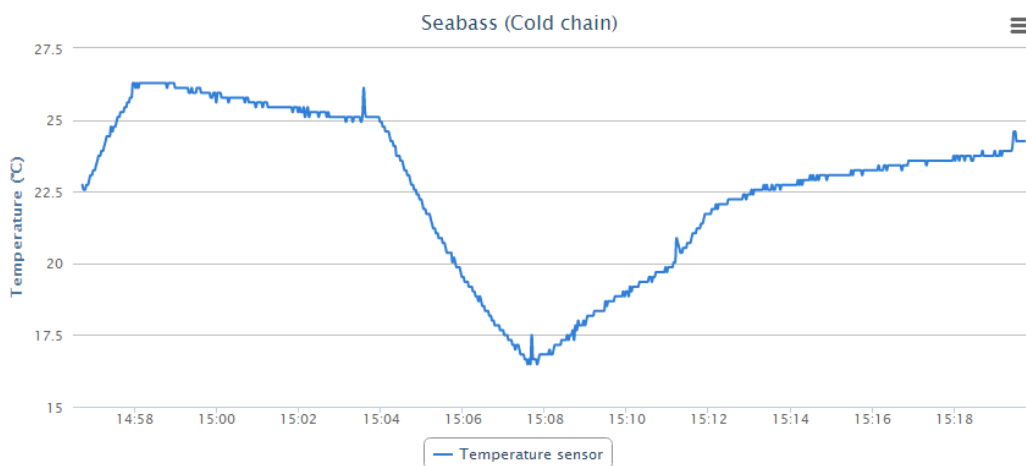
Slika 5.4: Graf temperatur v aplikaciji TraSens.

Rezultate testa lahko nato analiziramo tudi v spletni aplikaciji ScmContext [18], kjer so zabeleženi podatki o izvedeni hladni verigi v seznamu (slika 5.5). S klikom na gumb View pa je možno pogledati graf temperaturnih meritev (slika 5.6).

| Show 10 entries |        | Search:        |                        |              |                        |      |                      |
|-----------------|--------|----------------|------------------------|--------------|------------------------|------|----------------------|
| NFC-TL (004.y)  | Active | Start location | Start time             | End location | End time               | User | Graph                |
| 3               | 0      | 099.1          | 2014-02-11<br>14:56:43 | 099.1        | 2014-02-11<br>15:20:38 | 2    | <a href="#">View</a> |
| 3               | 0      |                | 2014-02-10<br>20:52:28 |              | 2014-02-10<br>20:54:45 | 2    | <a href="#">View</a> |
| 3               | 0      |                | 2014-02-10<br>14:24:10 |              | 2014-02-10<br>14:24:54 | 2    | <a href="#">View</a> |
| 3               | 0      |                | 2014-02-05<br>14:26:20 |              | 2014-02-05<br>14:45:09 | 2    | <a href="#">View</a> |
| 3               | 0      |                | 2014-02-05<br>13:35:05 |              | 2014-02-05<br>13:46:11 | 2    | <a href="#">View</a> |

First Previous 1 2 3 4 Next Last

Slika 5.5: Vnos v spletni aplikaciji ScmContext.



Slika 5.6: Graf temperatur v aplikaciji ScmContext.

# Poglavje 6

## Zaključek

V diplomski nalogi smo predstavili rešitev, ki omogoča nadzor temperatur v poljubnem procesu z uporabo mobilnih naprav in tehnologije NFC. Definirali smo ustrezne postopke, izbrati primerne tehnologije, razvili prototip in pridobiti čim več podatkov o težavah, ki bi se pri tem pojavile, kar bi nam bilo v pomoč kot osnova za razvoj končnega produkta. Podrobno smo opisali naprave, orodja in druge tehnologije, namenjene razvoju prototipa. Uporabili smo tudi razmeroma nove tehnologije, ki so še v postopku hitrega razvoja in dokazali njihovo uporabno vrednost.

Seveda razvoj ni potekal brez težav. V začetni fazi je bilo največ težav s problemom, ki ima sicer najmanjši pomen. Podatki na zapisovalniku so namreč shranjeni po pravilu tankega konca. To pomeni da si bloki v pomnilniku sledijo zaporedno, bajti v blokih pa so v obratnem vrstnem redu. Do problemov je prišlo tudi zaradi majhne robustnosti komunikacije z zapisovalnikom. To bi lahko rešili z optimizacijo komunikacije glede na razdaljo in povečanjem bralne razdalje pri usmerjenem toku podatkov. Seveda bi pri povečanju razdalje morali poskrbeti za boljšo varnost pri prenosu podatkov.

S prototipom smo dokazali, da bi lahko dobro deloval v okolju, za katerega je bil narejen, vendar bi bilo potrebno še dodatno delo, posebej na področju:

**Prevajanje identifikatorja GS1** Trenutno je prevajanje kod EPC nepopolno, saj podpira samo določene formate standarda GS1. Ta pre-

tvorba je namreč zelo zapletena, vendar ne nerešljiva. Obstaja namreč knjižnica TDT v okviru orodja Fosstrak, ki to prevajanje omogoča, vendar uporablja funkcije okolja JDK, katere na mobilnih napravah Android niso dostopne. Z vključitvijo programske kode iz knjižnice TDT bi dobili polno zmožnost integracije z obstoječim sistemom RFID-F2F.

**Komunikacija z zapisovalnikom** Velik problem prototipa je nezadostno upravljanje z napakami, ki v veliki meri nastanejo zaradi slabe robustnosti prenosa podatkov pri NFC tehnologiji. Do ostalih napak pride zaradi slabe kompatibilnosti z različnimi zapisovalniki. To bi popravili z boljšo definicijo in ustreznim ukrepanjem za vsako vrsto napake, do katere lahko privede. Rešitev sicer deluje z vsemi zapisovalniki, ki delujejo po standardu ISO-15693, vendar pri nekaterih z omejeno funkcionalnostjo. Potrebno bi bilo več testiranja na različnih zapisovalnikih.

**Funkcionalnost aplikacije** Za večjo preprostost bi bilo primernejše združiti aplikaciji TraSens in TraSens Manager, ki bi imela omejeno funkcionalnost glede na vlogo uporabnika, ki je prijavljen. Tako bi imeli eno aplikacijo za več različnih procesov v sistemu hladne verige.

**Zavarovanje podatkov zapisovalnika** Iz aplikacije TraSens, ki je namenjena za spremljanje meritev, sicer ni mogoče spreminjati podatkov na zapisovalniku, vendar lahko oseba, če ima dostop do aplikacije TraSens Manager oziroma čitalca, ki podpira standard ISO-15693, te podatke izbriše ali priredi. To bi lahko rešili z zaklepanjem funkcij zapisovalnika z geslom, kar je podprto s strani standarda ISO-15693.

### **Realno testiranje vključitve hladne verige v sistem sledljivosti**

V okviru diplomske naloge nam zaradi omejitev delovanja strežnika in aplikacije za pregled hladne verige ni uspelo testirati sistema sledljivosti. Za boljšo konsistenčnost bi bilo potrebno poenotiti zapise kod EPC na pametnih značkah za uporabo v aplikacijah TraSens, TraSens

Manager in TColdChain [19]. Za uporabo v aplikaciji TColdChain morajo biti kode EPC na pametni znački zapisane v formatu NDEF (NFC Data Exchange Format), za uporabo v aplikacijah TraSens in TraSens Manager pa so te zapisane v prvih treh blokih pomnilnika pametne značke.

Izdelani prototip je tudi dobra platforma za razvoj nadaljnjih samostojnih rešitev. Uporabili bi jo lahko za aplikacije, ki zahtevajo pregledovanje temperatur hladilnikov s pokvarljivimi živili v nakupovalnih centrih. Pokazali smo tudi izjemne možnosti tehnologije NFC v prihodnosti komunikacije na kratke razdalje.



# Literatura

- [1] Sistem EPCIS. Dostopno na:  
<http://en.wikipedia.org/wiki/EPCglobal>
  
- [2] FossTrak Tag Data Translation Engine. Dostopno na:  
<https://code.google.com/p/fosstrak/wiki/TdtMain>
  
- [3] GS1 identifikacijski ključi. Dostopno na:  
<http://www.gs1si.org/1/standardi-in-resitve/identifikacijske-stevilke/identifikacijski-kljuci.aspx>
  
- [4] Android Plot, knjižnica za risanje grafov na Androidu. Dostopno na:  
<http://androidplot.com/>
  
- [5] Projekt RFID-F2F. Dostopno na:  
<http://www.rfid-f2f.eu/details.asp?id=1583>
  
- [6] Smart-Trace, sistem za spremljanje hladne verige. Dostopno na:  
<http://www.smart-trace.com/>
  
- [7] Sofrigam, A better understanding of the cold chain, 2011.
  
- [8] NFC na sistemih Android. Dostopno na:  
<http://developer.android.com/guide/topics/connectivity/nfc/index.html>
  
- [9] NFC, na splošno. Dostopno na:  
<http://nfc-forum.org/what-is-nfc/about-the-technology/>

- 
- [10] Čip IDS-SL13A, 2008. Dostopno na:  
[http://www.ids.si/doc/sd/SL13A\\_SD.pdf](http://www.ids.si/doc/sd/SL13A_SD.pdf)
- [11] RFID. Dostopno na:  
<http://www.rfidjournal.com/site/faqs#Anchor-What-363>
- [12] Format EPC, 2011. Dostopno na:  
[http://www.gs1.org/gsmp/kc/epcglobal/tds/tds\\_1\\_6-RatifiedStd-20110922.pdf](http://www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_6-RatifiedStd-20110922.pdf)
- [13] Paketi za prenašanje cepiv. Dostopno na:  
<http://www.apex-international.org/cold-chain-equipment.html>
- [14] SL13A zapisovalnik. Dostopno na:  
<http://www.ams.com/eng/Products/NFC-HF-RFID/NFC-HF-Interface-and-Sensor-Tags/SL13A>
- [15] SL2ICS20 podatkovni list, 2005. Dostopno na:  
<http://www.datasheetarchive.com/dlmain/Datasheets-NXP/DSANXP01000589.pdf>
- [16] RFID in NFC. Dostopno na:  
<http://www.nfcnearfieldcommunication.org/radio-frequency.html>
- [17] Fosstrak EPCIS. Dostopno na:  
<https://code.google.com/p/fosstrak/wiki/EpcisMain>
- [18] Rok Erjavec, Aplikacija za upravljanje identifikatorjev in podatkov v sistemu z NFC nadzorom temperatur, 2013. Dostopno na:  
[http://eprints.fri.uni-lj.si/2219/1/Erjavec\\_R-1.pdf](http://eprints.fri.uni-lj.si/2219/1/Erjavec_R-1.pdf)
- [19] Gregor Pavuna, Mobilna aplikacija za predstavitev sledljivosti in hladne verige v prehranski verigi, 2013. Dostopno na:  
[http://eprints.fri.uni-lj.si/2316/1/Pavuna\\_G-1.pdf](http://eprints.fri.uni-lj.si/2316/1/Pavuna_G-1.pdf)

[20] EPC, 2007. Dostopno na:

[http://www.gs1.org/gsm/kc/epcglobal/epcis/epcis\\_1\\_0-presentation-20070619.pdf](http://www.gs1.org/gsm/kc/epcglobal/epcis/epcis_1_0-presentation-20070619.pdf)

[21] Android. Dostopno na:

<http://www.android.com/meet-android/>