

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andraž Hostnik

**Analiza in predstavitev podatkov iz  
hladne verige**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mira Trebar

Ljubljana, 2020

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

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

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Temperatura in vlaga sta pomembna parametra v nadzoru hladne verige hitro pokvarljivih živil. Z uporabo samostojnih merilnih naprav ali različnih senzorskih sistemov so pridobljeni podatki pogosto na voljo v obliki zapisov v datotekah. Kandidat naj v diplomskem delu izdelava spletno aplikacijo za vnos, obdelavo in prikaz izmerjenih temperatur in relativne vlage. Podatke naj analizira in uporabi za izračun preostale dobe uporabnosti ter rezultate predstavi v časovnem grafu. Pomembne ugotovitve naj predstavi tudi z enostavnim načinom obveščanja v tekstovni obliki. Rešitev naj omogoča dva tipa uporabnikov, za urejanje ali pregled. Izdelana naj bo z uporabo ogrodja Laravel, sodobnih spletnih tehnologij in podatkovno bazo MySQL. Implementacija in testiranje naj vključujeta predstavitev rezultatov z realnimi meritvami hladne verige.



*Zahvaljujem se mentorici doc. dr. Miri Trebar za vso pomoč in nasvete pri izdelavi diplomske naloge. Posebna zahvala gre tudi moji družini.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Pregled področja</b>	<b>3</b>
2.1	Hladna veriga . . . . .	3
2.2	Obstoječe rešitve . . . . .	4
2.3	Senzorski podatki . . . . .	5
<b>3</b>	<b>Uporabljene tehnologije</b>	<b>9</b>
3.1	Podatkovna baza . . . . .	9
3.2	Spletni strežnik . . . . .	10
3.3	Spletna aplikacija . . . . .	10
3.4	Programska oprema . . . . .	11
<b>4</b>	<b>Analiza, načrtovanje in arhitektura</b>	<b>13</b>
4.1	Analiza . . . . .	13
4.2	Načrtovanje in arhitektura . . . . .	17
<b>5</b>	<b>Razvoj</b>	<b>31</b>
5.1	Podatkovna baza . . . . .	31
5.2	Spletna aplikacija in storitev . . . . .	33
5.3	Prijava, odjava, registracija in potrditev uporabnika . . . . .	34

5.4	Uvoz in generiranje meritev . . . . .	36
5.5	Pregled, iskanje in filtriranje meritev . . . . .	40
5.6	Napoved preostale dobe uporabnosti izbranega živila . . . . .	43
5.7	Pregled in urejanje živil . . . . .	45
5.8	Pregled in urejanje lokacij . . . . .	46
5.9	Pregled senzorskih naprav . . . . .	47
5.10	Pregled in potrditev uporabnikov . . . . .	48
<b>6</b>	<b>Testiranje</b>	<b>49</b>
6.1	Testiranje spletne storitve . . . . .	49
6.2	Postavitev aplikacije na testno okolje . . . . .	50
<b>7</b>	<b>Sklepne ugotovitve</b>	<b>51</b>
	<b>Literatura</b>	<b>53</b>





# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>CSIRO</b>	Commonwealth Scientific and Industrial Research Organisation	Commonwealth Scientific and Industrial Research Organisation
<b>CSS</b>	Cascading Style Sheets	predloge, ki določajo izgled spletnih strani
<b>CSV</b>	Comma Separated Values	podatki so v datoteki ločeni z vejico
<b>ER diagram</b>	Entity Relationship Diagram	entitetno relacijski diagram
<b>FTP</b>	File Transfer Protocol	protokol za prenos datotek
<b>HTML</b>	Hyper Text Markup Language	označevalni jezik za izdelavo spletnih strani
<b>HTTP</b>	Hyper Text Transfer Protocol	komunikacijski spletni protokol
<b>FTP</b>	File Transfer Protocol	protokol za prenos datotek
<b>MVC</b>	Model-view-controller	model-pogled-krmilnik
<b>NFC</b>	Near Field Communication	tehnologija komunikacije kratkega dosega
<b>PDF</b>	Portable Document Format	format prenosnega dokumenta
<b>PHP</b>	Personal Home Page (PHP: Hypertext Preprocessor)	skriptni programski jezik
<b>REST</b>	Representational state transfer	predstavitveni prenos stanja
<b>URL</b>	Uniform Resource Locator	enolični krajevnik vira

# Povzetek

**Naslov:** Analiza in predstavitev podatkov iz hladne verige

**Avtor:** Andraž Hostnik

TODO naredi povzetek iz uvoda.

**Ključne besede:** hladna veriga, doba uporabnosti, spletna aplikacija, Laravel.



# Abstract

**Title:** Analysis and presentation of cold chain data

**Author:** Andraž Hostnik

TODO english abstract from uvod.

**Keywords:** cold chain, shelf life, web application, Laravel.



# Poglavje 1

## Uvod

Kakovostna in varna hrana predstavlja vse večji izziv tako na lokalnem, kot tudi globalnem trgu proizvodnje in distribucije. Potrošniki postajajo bolj osveščeni in pričakujejo vse več informacij o prehranski verigi živil, ki ne predstavlja tveganj za njihovo zdravje. V ta namen podatki iz hladne verige, ki so pridobljeni v vseh fazah, od začetka priprave proizvoda za prodajo in vse do prodajnega mesta, kjer so na voljo končnim uporabnikom. Pri zagotavljanju hladne verige živil gre lahko marsikaj narobe, zato sistemi za zajem podatkov in aplikacije za analizo in vizualizacijo tovrstnih podatkov odigrajo ključno vlogo pri ugotavljanju in odpravljanju težav ali napak.

Podatki, ki zagotavljajo meritve v celotni hladni verigi, običajno prihajajo iz različnih senzorskih sistemov in v različnih podatkovnih formatih. Na področju analize in prikaza podatkov iz hladne verige obstaja že kar nekaj senzorskih sistemov, ki so sestavni del interneta stvari v prehranskih verigah. Še vedno pa se pogosto zbirajo in shranjujejo v datotekah, kjer so zapisani tudi v različnih podatkovnih formatih in niso avtomatsko na voljo v spletnih aplikacijah. Njihovo analizo je nato mogoče izvesti z ročno obdelavo podatkov v ustreznih programskih orodjih, kar pa ne rešuje problema enostavnega in preglednega informiranja o morebitnih težavah, ki so pomembne za določanje preostale dobe uporabnosti hitro pokvarljivih živil. Zaradi izpostavljenosti višjim temperaturam je živilo še vedno primerno za uživanje, se je skrajšal

čas v katerem ga je potrebno porabiti.

V ta namen smo se odločili razviti enotno spletno aplikacijo, ki omogoča enostaven vnos in shranjevanje vhodnih podatkov, analizo in vizualizacijo celotnega postopka zajema senzorskih meritev. Na podlagi vnaprej definiranih zahtev o ravnanju z živili v hladni verigi in z analizo meritev in izbranih metod napovedovanja dobe uporabnosti bo možno napovedati preostalo dobo uporabnosti. Aplikacija nazorno prikaže tudi anomalije v podatkih in tako opozori na morebitne napake ali pomanjkljivosti v procesu hladne verige ali delovanja senzorskega sistema.

Aplikacija je razvita z sodobnimi spletnimi orodji in tehnologijami, prilagojena za vse vrste naprav in enostavna za uporabo. Za potrebe lažjega testiranja smo razvili tudi generator podatkov, ki omogoča dodatne analize simuliranih dogodkov v hladni verigi. Poleg tega vsebuje tudi dodajanje in urejanje posameznih parametrov ter dodeljevanje in urejanje pravic uporabnikov.

V diplomski nalogi najprej opišemo senzorske podatke s katerimi razpolagamo v aplikaciji. V drugem poglavju na kratko opišemo uporabljena orodja in tehnologije pri izdelavi spletne aplikacije. Sledi poglavje o predstavitvi funkcionalnih in nefunkcionalnih zahtev, podatkovnem modelu, načrtu arhitekture in opisu zaslonskih predlogov. V četrtem poglavju je podrobno predstavljen potek implementacije spletne rešitve z zaslonskimi slikami. Na koncu sledita še opis postopka testiranja in sklepne ugotovitve.



## Poglavje 2

### Pregled področja

V poglavju na kratko predstavimo hladno verigo, obstoječe spletne rešitve in opišemo senzorske podatke, ki so uporabljene v diplomski nalogi.

#### 2.1 Hladna veriga

Zahteve potrošnikov po raznoliki in kvalitetni hrani se iz leta v leto povečuje s tem pa se povečujejo tudi zahteve po prevozu in shranjevanju ohlajenih in zmrzjenih živil. Živila lahko potujejo preko celega planeta, preden pridejo do končnega uporabnika. Čez celoten logističen proces je potrebno zagotavljati nizke temperaturne pogoje, ki ohranjajo varnost in boljšo kakovost živila. Temu procesu rečemo hladna veriga. Hladna veriga torej pomeni vzdrževanje priporočene temperature, da zagotovimo rok trajanja živila [?].

Pri upravljanju tega procesa gre lahko marsikaj narobe, zlasti pri vzdrževanju priporočene temperature med logističnimi dogodki od proizvodnje, prevzema blaga, prevoza, skladiščenja, razdeljevanja, distribucije in dostave do končnega uporabnika, zato je potrebno celoten proces stalno nadzirati in optimizirati.

## 2.2 Obstoječe rešitve

Za zagotavljanje hladne verige odigrajo ključno vlogo sistemi za zajem podatkov in aplikacije za analizo in vizualizacijo tovrstnih podatkov.

Na trgu jih je kar nekaj, večina od njih pa ne rešuje problema o različnih vhodnih podatkovnih formatih.

Spodaj sta na kratko opisana dva primera.

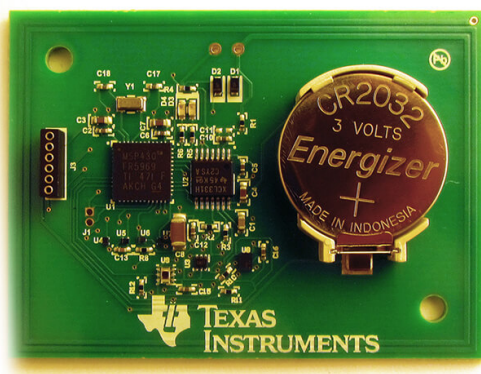
**Spletna aplikacija v okviru magistrskega dela Senzorski moduli NFC in varnost podatkov, Dean Črnigoj** : Omogoča analizo in pregled senzorskih podatkov tipa TIDA na način kjer se meritve pošiljajo na strežnik sproti ali na zahtevo. Sistem podpira še zaznavanje nepričakovanih dogodkov, obveščanje, izvoz podatkov in konfiguracijo senzorskega sistema NFC. Spletna aplikacija je bila razvita kot del magistrskega dela, ki prikaže uporabo celotnega sistema za potrebe hladne verige [21].

**Spletna aplikacija Coldchainview** : Optimizirana za delo z moduli Euroscan. Omogoča sprotno spremljanje in upravljanje izdelkov različnih panog znotraj hladne verige. Podprte so še naslednje funkcionalnosti: grafični prikaz podatkov, izvoz podatkov v format PDF in XLS, pošiljanje samodejnih alarmov preko e-pošte in SMS-ja, pregled in analiza posameznih senzorskih lokacij ter konfiguracija hladilnih enot. Spletna aplikacija je last podjetja ORBCOMM® in se nahaja na spletnem naslovu <http://www.coldchainview.com>. Njena uporaba je plačljiva, zato aplikacije nismo preizkusili, ampak smo se sklicevali na proizvajalcev opis produkta. [?]

## 2.3 Senzorski podatki

Na področju senzorskih meritev je različnih podatkovnih modelov ravno toliko kot imamo različnih senzorskih naprav. Spodaj smo opisali glavne lastnosti le dveh s katerim smo se srečali pri uporabi podatkov iz hladne verige.

Merilna naprava TIDA je senzorski modul NFC z zelo nizko porabo energije proizvajalca Texas Instruments. Deluje kot naprava za shranjevanje podatkov iz treh različnih senzorjev: temperature (T), svetlobe (E) in relativne vlage (RH) (slika 2.1) [21].



Slika 2.1: Merilna naprava TIDA

### Podatkovni model meritvene naprave TIDA :

- zapis datoteke v formatu CSV (.csv),
- vrednosti so ločene z znakom vejica (,),
- osnovni podatki o meritvi se začnejo z znakom lojtra (#),
- takoj za osnovnimi podatki so definirana imena vrednosti meritve,
- datum je zapisan v obliki leto–dan–mesec (yyyy-MM-dd),
- čas je zapisan v obliki ura:minuta:sekunda (HH:mm:ss),
- ostale vrednosti so zapisane v decimalnih številih.

Merilna naprava RHT10 z USB vmesnikom za enostavno namestitev in prenos podatkov. Beleži senzorske podatke o temperaturi (T) in relativni vlagi (RH). Ima možnost več intervalnih vrednosti - od 2 sekundi do 24 ur (slika 2.2) [20].



Slika 2.2: Merilna naprava RHT10

**Podatkovni model meritvene naprave RHT10 :**

- zapis datoteke v tekstovnem formatu (.txt),
- vrednosti so ločene z znakom za skok v naslednji odstavek (TAB),
- osnovni podatki so ločeni z dvema znakoma večje (>>),
- osnovni podatki in imena vrednosti meritve se ločujejo z vrstico, ki je sestavljena iz znakov minus (–),
- datum je zapisan v obliki dan–mesec–leto (dd-MM-yyyy),
- čas je zapisan v obliki ura:minuta:sekunda (HH:mm:ss),
- ostale vrednosti so zapisane v decimalnih številih.

Tretji podatkovni model smo si za potrebe simuliranja in testiranja meritev izmislili sami. Zaradi ročno ustvarjenih podatkov smo ta tip naprave poimenovali kar 'Generated'.

**Glavne lastnosti podatkovnega modela Generated :**

- podatke se generira direktno v spletni aplikaciji,
- določi se začetni datum in čas meritve,
- določi se časovni interval v sekundah,
- poljubno se dodaja število zapisov z isto temperaturo (°C), relativno vlago (%) in lokacijo.



## Poglavje 3

# Uporabljene tehnologije

Poglavje opisuje uporabljene tehnologije pri razvoju naše spletne aplikacije.

### 3.1 Podatkovna baza

Na področju informacijskih sistemov je osnovni vir informacij podatkovna baza. Podatkovna baza je zbirka shem, tabel, poizvedb, poročil, pogledov in drugih objektov. Upravljamo jo preko sistema za upravljanje baz podatkov. V grobem ločimo relacijske od nerelacijskih. Prve so najbolj razširjene in temeljijo na množici relacij, kjer je vsaka relacija tabela z vrsticami in stolpci. Nerelacijske opredeljuje visoka fleksibilnost, katero dosežejo tako, da nimajo vnaprej definirane sheme. Primeri sistemov za upravljanje relacijskih baz podatkov so MySQL, Oracle, PostgreSQL, nerelacijskih pa MongoDB, Redis, Elasticsearch [25].

V naši diplomski nalogi smo uporabili relacijsko podatkovno bazo z sistemom za upravljanje MySQL [16]. Do podatkov smo dostopali preko spletnega vmesnika `phpMyAdmin` [14].

## 3.2 Spletni strežnik

Spletni strežnik je računalniški sistem, ki obdeluje zahteve preko protokola HTTP. Njegove osnovne naloge so shranjevanje, obdelava in pošiljanje spletnih strani odjemalcem. Tipični primeri spletnih strežnikov so Apache, Nginx, Microsoft IIS. Vsak izmed njih podpira določene skriptne jezike kot so PHP, ASP, JavaScript [24].

V diplomski nalogi smo za razvoj spletne aplikacije izbrali skriptni jezik PHP, ki ima od verzije 5.4.0 vgrajen spletni strežnik za lokalni razvoj, katerega na zahtevo zaženemo preko ukazne vrstice s spodnjim ukazom:

```
php artisan serve
```

## 3.3 Spletna aplikacija

Skupek dinamičnih spletnih strani, ki omogočajo izvajanje akcij in opravil katerih spremembe se odražajo nad podatki, imenujemo spletna aplikacija. Glavne lastnosti le-te so neodvisnost od operacijskega sistema, dostopnost, centraliziran nazor in varnost. Zaradi omenjenih lastnosti je potreba po hitrem razvoju spletnih aplikacij vse večja. V ta namen razvijalci uporabljajo ogrodja, ki imajo temeljne gradnike že vnaprej definirane.

Ogrodje Laravel je v zadnjih letih popularno pri razvijalcih v programskem jeziku PHP. Njegov začetek beležimo junija 2011 z prvo verzijo, danes pa je aktualna že šesta, ki sledi zadnjim standardom na področju spletnega razvoja [8].

**Nekaj glavnih gradnikov omenjenega ogrodja :**

- arhitektura MVC,
- vse potrebno za delo z avtentikacijo uporabnika,
- konfigurirana obravnava napak in izjem,
- odpravljanje najpogostejših tehničnih ranljivosti,



- podpora za integracijo predpomnilniških sistemov,
- konfiguracija usmerjanja URL-jev,
- podpora za integracijo e-poštnih storitev.

Za hitrejši razvoj spletnih aplikacij v primerih, ko potrebujemo napredne grafične komponente, razvijalci dodatno izbirajo po predlogih HTML. Predloga HTML običajno vsebuje grafične komponente kot so gumbi, opozorilni boksi, tipografski elementi, spletni obrazci, grafi in tabele.

Dinamična komunikacija med ogrođjem in grafičnimi komponentami v spletni aplikaciji običajno poteka v obliki spletne storitve REST. Njen namen je definiranje principov, ki opisuje na kakšen način so viri definirani in naslovljeni, pri katerem se uporabljajo metode protokola HTTP. Te metode so lahko GET, POST, PUT ali DELETE, vsak vir podatkov pa ima svoj URL [19].

### 3.4 Programska oprema

Pri razvoju spletne aplikacije smo uporabili programsko orodje IntelliJ IDEA podjetja JetBrains. Produkt je na voljo v plačljivi in zastonjski verziji, namenjen Java razvijalcev, med drugim pa nudi podporo tudi za programski jezik PHP. Omogoča tudi integracijo s sistemi za nadzor verzij programske kode, v našem primeru s sistemom Git [18].

Git je sistem za upravljanje z izvirno kodo, s katerim lahko nadzorujemo spremembe, omogoča sodelovanje več razvijalcev na istem projektu, obenem pa nam kadarkoli omogoča obnovitev kode na prejšnjo - zgodnejše stanje [22].

Za beleženje in vodenje nalog v okviru diplomske naloge smo uporabili spletni servis GitHub. V osnovi omenjeni servis ponuja repozitorije Git v oblaku, omogoča pa tudi distribuirano upravljanje z izvirno kodo, kontrolo dostopa in storitve za kolaboracijski razvoj preko spletnega grafičnega vmesnika [2].



## Poglavje 4

# Analiza, načrtovanje in arhitektura

V poglavju najprej opredelimo funkcionalne in nefunkcionalne zahteve. V nadaljevanju podrobno opišemo podatkovni model, zaslonske predloge in arhitekturo aplikacije.

### 4.1 Analiza

Pred izvedbo smo definirali zahteve delovanja spletne aplikacije, ki smo jih razdelili v funkcionalne in nefunkcionalne.

#### 4.1.1 Funkcionalne zahteve

- Spletna aplikacija naj bo zavarovana in dostopna z uporabniškim imenom in geslom,
- vloge uporabnikov so razdeljene na administratorje (ang. Administrator), urednike (ang. Editor) in obiskovalce (ang. User),
- administrator naj ima dostop do vseh funkcionalnosti aplikacije vključno z funkcionalnostmi povezane z upravljanje z uporabniki (potrditev novega uporabnika in spreminjanje njegove vloge),

- urednik naj ima:
  - vse funkcionalnosti obiskovalca,
  - možnost uvoza dveh različnih tipov meritev,
  - možnost generiranja podatkov meritve,
  - za vsako meritev možnost brisanja, urejanja in izvoza v datoteko PDF,
  - možnost pregleda, dodajanje in urejanje živil,
  - možnost pregleda, dodajanja in urejanja lokacij,
  - možnost pregleda tipov senzorjev,
- obiskovalec naj ima:
  - celoten seznam, možnost iskanja in filtriranja vseh uvoženih ali generiranih meritev,
  - vidi podroben prikaz in analizo podatkov o meritvi,
  - na podlagi temperature sistem napove preostalo dobo uporabnosti živila.
- uredniki in obiskovalci se morajo pred prvo uporabo registrirati z naslednjimi podatki: ime in priimek, e-pošta in geslo,
- po registraciji mora urednika ali obiskovalca potrditi administrator,
- administrator prejme obvestilo o novem uporabniku na e-pošto,
- potrjeni uporabnik prejme obvestilo na registracijsko e-pošto,
- sistem naj prikaže čas trajanja meritve po lokacijah, izračuna največje in najmanjše vrednosti in obvesti o preseženih proporočenih vrednostih v posamezni meritvi,
- za potrebe simuliranja meritev in lažjega testiranja aplikacije razvijemo generator podatkov.

### 4.1.2 Modela za izračun preostale dobe uporabnosti

V diplomski nalogi smo za izračun preostale dobe uporabnosti živila za primerjavo uporabili dva različna modela. Na kratko sta opisana spodaj.

**Model CSIRO** (Commonwealth Scientific and Industrial Research) : Razvit v avstrijski organizaciji po kateri je dobil tudi ime, definira preposto matematično enačbo za razvoj bakterij v živilu [13].

$$k = (0,1T + 1)^2$$

$$SL = SL_{ref} - t \times k$$

Enačba za izračun preostale dobe uporabnosti po modelu CSIRO.

**Model SAL** : Uporaben predvsem za morsko hrano. Uporablja tabelo, ki ima za vsako temperaturo navedeno koliko časa je živilo uporabno, če je ves čas hranjena na taki temperaturi [6].

T(°C)	SL(d)	T(°C)	SL(d)	T(°C)	SL(d)	T(°C)	SL(d)
0	8	8	2,7	16	1,1	24	0,75
1	6,5	9	2,35	17	1	25	0,73
2	5	10	2	18	0,92	26	0,71
3	4	11	1,75	19	0,89	27	0,69
4	3,5	12	1,5	20	0,85	28	0,67
5	3,35	13	1,4	21	0,83	29	0,62
6	3,15	14	1,3	22	0,8	30	0,6
7	3	15	1,2	23	0,78		

Tabela 4.1: Vrednosti v modelu SAL, ki se uporablja za napovedovanje dobe uporabnosti.

### 4.1.3 Nefunkcionalne zahteve

# Varnost

Zaradi zagotavljanja ustrezne varnosti smo poskrbeli za izvedbo identifikacije, avtentikacije, avtorizacije in beleženja uporabnikov spletne aplikacije (ang. IAAA) [26].

Identifikacija je v našem primeru e-pošta uporabnika, z ustreznim geslom pa preverimo avtentikacijo oziroma prisotnost uporabnika. Zaradi lažje uporabnosti se nadaljnje preverjanje prisotnosti preverja na podlagi seje, ki ima aktivnost nastavljeno na 2 minute.

Na nivoju avtorizacije imamo poskrbljeno dva nivoja uporabnikov – urednik (ang. Editor) in administrator. Privzeto se ob registraciji dodeli vloga neodobrenega urednika, ki še nima dostopa do vsebin, dokler administrator-ska vloga ne potrdi omenjeno vlogo.

Za zadnji sklop varnosti beležimo še dnevnik prijav uporabnikov. Pri tem nam je v pomoč programsko ogrodje Laravel z knjižnico Monolog [12], ki z enostavnim ukazom shranimo vsako prijavo uporabnika v log datoteko na direktorij strežnika aplikacije.

## Obravnava napak

Ena iz med pomembnih nefunkcionalnih zahtev je tudi obravnava napak v sistemu. Obravnavamo strežniške (5xx) in odjemalčeve napake (4xx).

Na nivoju ogrođja imamo zagotovljeno avtomatsko beleženje napak v log datoteko. V sami predlogi pa imamo poskrbljeno tudi za izpis napake uporabniku.

Primer izvirne kode za izpis napak uporabniku:

```
1 <div class="alert alert-danger" role="alert">
2     @if ($errors->any())
3     <ul id="errors">
4         @foreach ($errors->all() as $error)
5             <li>{{ $error }}</li>
6         @endforeach
7     </ul>
```

```
8         @endif  
9     </div>
```

### Prilagodljiva grafična podoba

Grafični uporabniški vmesnik smo poizkušali izbrati preprost in pregleden, ter prilagodljiv na vse vrste ekranov. Pri tem smo dali poudarek na vizualizacijske komponente, kot so grafi in tabele. Izbrali smo odprtokodno predlogo HTML CoolAdmin [17], ki je vsebovala že v naprej definirane grafične komponente.

Grafične komponente so sestavljene iz kode HTML, datoteke CSS, slik in knjižnic Javascript. Primer knjižnice Javascript, ki skrbi za izris interaktivnih grafov v aplikaciji, je `highcharts.js` [5].

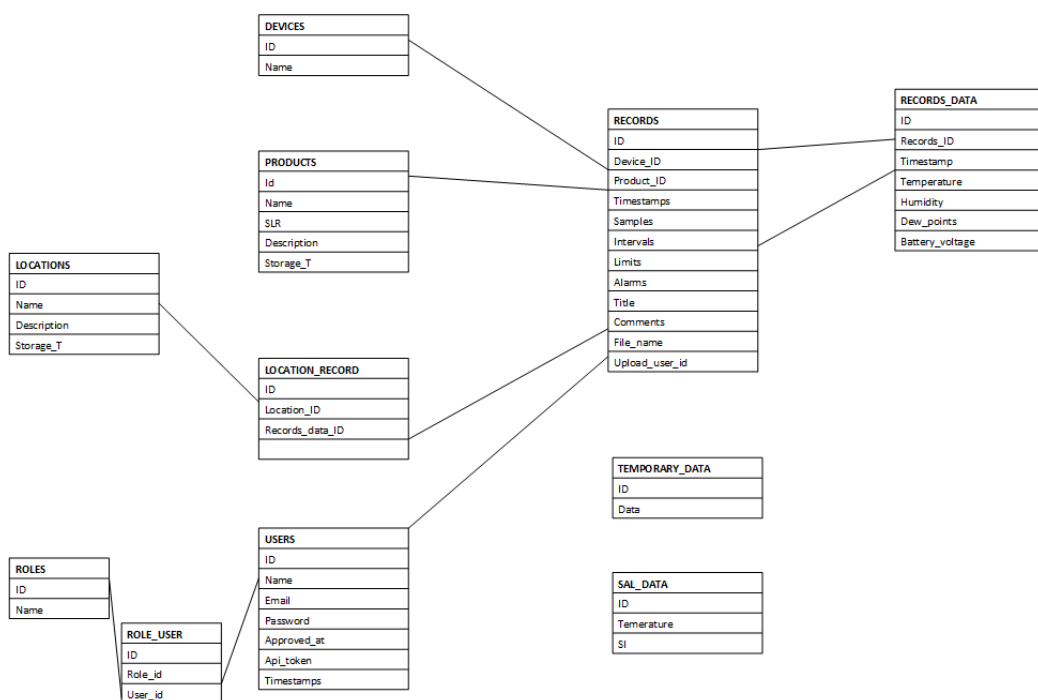
## 4.2 Načrtovanje in arhitektura

### 4.2.1 Podatkovni model

Načrtovanje podatkovnega modela nam je olajšalo kasnejše kreiranje objektov. S pomočjo ER diagrama in preslikave relacij v Migrations funkcionalnost ogrođa Laravel [10], smo ustvarili tabele v podatkovni bazi MySQL. Na kocu programske rešitve je bil podatkovni model sestavljen iz enajstih tabel. Te so prikazane na sliki 4.1.

**Tabela USERS** : Predstavlja podatke o uporabniku.

- ID - enolična oznaka uporabnika,
- Name - ime in priimek uporabnika,
- Approved\_at - status uporabnika, ki je lahko odobren ali neodobren,
- Email - email uporabnika
- Password - šifrirano geslo uporabnika,
- API\_token - žeton, ki je uporabljen pri avtentikaciji uporabnika.



Slika 4.1: ER diagram spletne aplikacije

**Tabela ROLES** : Vsebuje seznam vlog uporabljene v aplikaciji.

- ID – enolična oznaka vloge,
- Name – ime vloge (administrator, editor, viewer).

**Tabela ROLE\_USER** : Povezovalna tabela med uporabnikom in njegovimi vlogami.

- ID – enolična oznaka povezave,
- Role\_ID – enolična oznaka vloge,
- User\_ID – enolična oznaka uporabnika.

**Tabela DEVICES** : Vsebuje seznam naprav, ki so vsebovane v meritvi.

- UD – enolična oznaka naprave,
- Name – ime naprave.



**Tabela PRODUCTS** : Predstavlja seznam vseh živil, ki so lahko uporabljeni v meritvah.

- ID – enolična oznaka živila,
- Name – ime živila,
- SLR – predpisana doba uporabnosti živila,
- Description – opis živila,
- Storage\_T – območje priporočene temperature hranjenja živila.

**Tabela LOCATIONS** : Vsebuje seznam vseh lokacij, ki so lahko vsebovane v meritvi.

- ID – enolična oznaka lokacije,
- Name – ime lokacije,
- Description – opis lokacije,
- Storage\_T – območje delovne temperature lokacije.

**Tabela LOCATION\_RECORD** : Povezovalna tabela med lokacijami in meritvami. Ena meritev lahko vsebuje več različnih lokacij.

- ID – enolična oznaka povezave,
- Location\_ID – enolična oznaka lokacije,
- Records\_data\_ID – enolična oznaka posameznega zapisa meritve.

**Tabela RECORDS** : Vsebuje seznam vseh uvoženih ali generiranih meritev.

- ID – enolična oznaka meritve,
- Device\_ID – enolična oznaka naprave,
- Product\_ID – enolična oznaka živila,
- Timestamps – avtomatsko generirana časovna polja (created\_at, updated\_at), ki predstavlja čas uvoza in čas zadnje spremembe meritve,

- Samples – število zapisov v eni meritvi,
- Intervals – časovni interval med posameznimi zapisi meritve (enota milisekunda),
- Limits – polje, ki vsebuje izračunane podatke o robnih številkah meritve,
- Alarms – polje za opozorila ugotovljena iz meritve,
- Title – ime posamezne meritve,
- Comments – komentarji meritve,
- File\_name – ime uvožene datoteke,
- Upload\_user\_id – enolični identifikator uporabnika, ki je uvozil meritve.

**Tabela RECORDS\_DATA** : Vsebuje zapis posamezne meritve. Ena meritve ima več posameznih zapisov, med tem ko je lahko zapis vsebovan samo v eni meritvi.

- ID – enolična oznaka zapisa,
- Records\_ID – enolična oznaka meritve,
- Timestamp – točni čas zapisa,
- Temperature – izmerjena temperatura v cezijih,
- Humidity – izmerjena relativna vlaga v procentih,
- Dew\_points – izmerjeno rosišče,
- Battery\_voltage – napetost baterije merilne naprave.

**Tabela TEMPORARY\_DATA** : Tabela vsebuje trajno shranjene podatke, ki so uporabljene pri procesu shranjevanja meritve.

- ID – enolični identifikator trajno shranjenih podatkov,
- Data – podatki iz datoteke meritve.

**Tabela SAL\_DATA** : Tabela vsebuje podatke, ki so uporabljeni pri izračunu preostale dobe uporabnosti živila po modelu SAL.

- ID – enolični identifikator podatka,
- Temperature – temperatura hranjenja živila,
- SL – preostalo število dni uporabe živila.

### 4.2.2 Spletna aplikacija

Spletna aplikacija je namenjena uporabnikom, ki so odgovorni za kakovost in varnost hitro pokvarljivih živil. Omogoča uvoz, konfiguracijo, pregled, iskanje in filtriranje meritev v procesu hladne verige. V posamezni meritvi se analizira in vizualizira merljive podatke, izračuna največje in najmanjše vrednosti, prikaže skupni čas trajanja meritve po lokacijah, obvesti o preseženih priporočenih vrednostih. Znotraj meritve aplikacija napove tudi preostalo dobo živila po modelu CSIRO [13] in SAL [6].

V fazi načrtovanja smo naredili prototip aplikacije oziroma zaslonske predloge glavnih funkcionalnosti. Vsaka stran vsebuje levi menu, kjer so povezave do naslednjih strani.

Aplikacija kasneje vsebuje še prikazno sliko vloge in osnovne informacije uporabnika, logotip ter sekundarno navigacijo, ki ponazarja, kje se nahajamo na strani. Omenjene funkcionalnosti so bile izpuščene iz prototipa, saj niso igrale pomembno vlogo.

Na vstopni strani se prikaže obrazec za prijavo, kot prikazuje slika 4.2. Uporabnik mora ob prvem obisku navigirati na stran z registracijo s klikom na povezavo „Registration“.

Neregistrirani uporabnik vnese podatke o imenu, e-pošti naslov in ponovitvi gesla ter klikne na gumb „Register“.

V primeru, da je uporabnik že registriran, na prijavi strani vnesene e-poštni naslov in geslo, klikne na gumb „Login“ in ob primeru uspešne prijave ga preusmeri na seznam meritev.

The image shows two side-by-side wireframe panels. The left panel is titled 'Registration' and contains four input fields labeled 'Name', 'E-mail', 'Password', and 'Confirm Password'. Below these fields are two buttons: a grey 'Register' button and a blue 'Log In' link. The right panel is titled 'Login' and contains two input fields labeled 'E-mail' and 'Password'. Below these fields are two buttons: a grey 'Login' button and a blue 'Registration' link.

Slika 4.2: Zaslonska predloga registracije (levo) in prijave (desno).

V primeru uspešne registracije je uporabnik preusmerjen na opozorilo stran, kjer mu sporočimo, da je v postopku potrjevanja (slika 4.3). Ob izvedeni administratorski potrditvi uporabnik prejme obvestilo na registracijski e-poštni naslov.

The image shows a single wireframe panel titled 'Approval'. It features a grey box with the text 'Your account is waiting for our administrator approval.' Below this box is a blue 'Log In' link.

Slika 4.3: Opozorilna stran o administratorskem potrjevanju.

Pri uvozu datoteke v prvem koraku na gumb „Choose File“ izberemo željeno datoteko v ustreznem formatu. Dovoljena formata sta lahko zapisana v tekstovni datoteki ali datoteki CSV (vrednosti ločene z vejico). Takoj po izbiri sistem preveri datoteko in prikaže nadaljnje korake (slika 4.4).

Sledi konfiguracija vhodnih podatkov, kjer aplikacija prikaže imena vseh stolpcev s primerom dveh vrstic podatkov, vsebovanih v datoteki. Predstavljene stolpce lahko poljubno spreminjamo s stolpci določenih v aplikaciji.

Import data  
Records  
Settings

### Import record data

Choose File

Parse File Generate data

Complete record data:

date	time	temperature	battery	humidity
2018-10-26	14:58:00	25.00	12.2	52.0
2018-10-28	21:48:00	22.60	0.0	58.0

date time temperature battery humidity

Product: Fish Orado  
Location: Hladnica 32  
ID: 30003300...  
No. Samples: 142  
Start Date: 20 Feb 2016  
Start Time: 00:05:55  
Interval (s): 60

Title: \_\_\_\_\_  
Comment: \_\_\_\_\_

Import data Reset

Slika 4.4: Zgornja slika prikazuje spletno okno za uvoz meritev. Na izbiro imamo uvoz datoteke ali generiranje podatkov.

Vrednosti, ki jih lahko izbiramo iz spustnega seznama so: datum, čas, temperatura, relativna vlaga, napetost baterije, rosišče ali „–ignore–“. Z zadnjo vrednostjo povemo aplikaciji, da izbrani stolpec pri uvozu meritve izpusti.

V zadnjem koraku izpolnimo še preostale podatke o meritvi, kot so:

- izbira živila, ki je bila del meritve. Iz izpustnega seznama „Product“ lahko izbiramo živila, ki smo jih predhodno vnesli v aplikaciji. Predstavili smo: izbiro z Ribe – tuna, Ribe – brancin, Svinino, Govedo mleto, Govedo, Piščanec – file,
- izbira lokacije, kjer je bila opravljena meritev. Iz izpustnega seznama „Location“ lahko izbiramo lokacije, ki smo jih predhodno vnesli v aplikaciji,
- ime meritve v vnosnem polju „Title“,
- komentar meritve v vnosnem polju „Comment“.

Informativno se izpišejo tudi podatki o številu zapisov („No. Samples“), začetku in koncu meritve („Start Date“, „End Date“) in intervalni vrednosti med zapisi v sekundah („Interval(s)“).

S klikom na gumb „Reset“ se lahko ponovno nastavi uvožene podatke.

Čisto na koncu s klikom na „Import data“ shranimo uvožene podatke v bazo. Po uspešni shranitvi se nam prikaže obvestilo s povezavo do ogleda meritve.

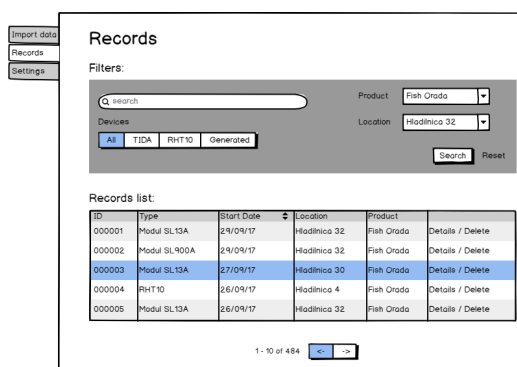
Slika 4.5: Zavihek „Generate data“ za ročno generiranje meritev.

Pri kliku na zavihek „Generate data“ se prikaže spletno okno iz slike 4.5. Omogoča ročno generiranje meritev, tako da na začetku vnesemo ime in komentar meritve, iz izpustnega seznama izberemo živilo („Product“) in nastavimo začetni čas („Start timestamp“) in interval med zapisi v sekundah („Interval“).

Na naslednjem razdelku vnesemo število in vrednosti posameznih zapisov. V polju „Records“ vnesemo število zapisov, v polju „Temperature“ temperaturo v °C, v polju „Humidity“ relativno vlago v % in iz izpustnega seznama „Location“ lokacijo.

S klikom na zeleno ikono „plus“ poljubno dodajamo nove intervalne vrednosti zapisov. S klikom na sivo ikono „X“ pa poljubno odstranjujemo intervalne vrednosti zapisov.

Po vnosu podatkov zaključimo s klikom na gumb „Generate data“ ali „Reset“. V prvem primeru se po uspešno generirani meritvi prikaže obvestilo s povezavo do ogleda meritve, v drugem pa se vsi podatki brišejo.



Slika 4.6: Zgornja slika prikazuje spletno okno za prikaz seznama meritev.

Na spletnem oknu za prikaz seznama meritev (slika 4.6) iz razdelka „Filters“ lahko iščemo meritve po različnih kriterijih. Te so:

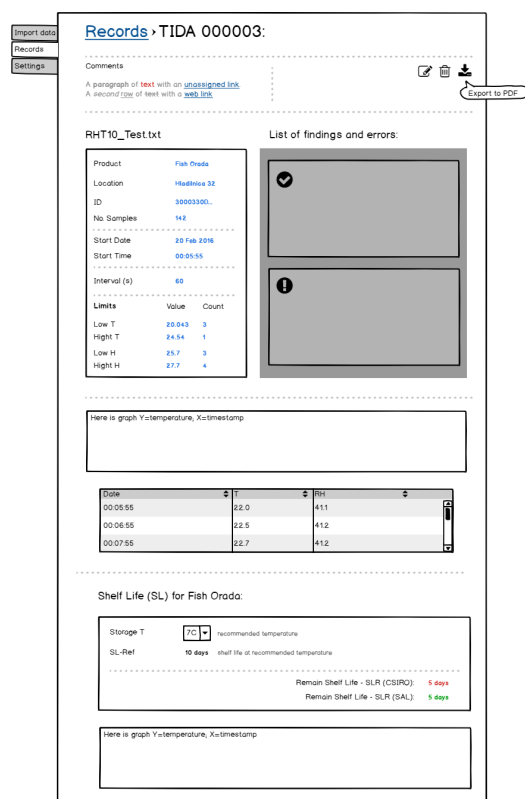
- polje „search“ po ključni besedi,
- gumbi „All“, „TIDA“, „RHT10“, „Generated“ po tipu naprave,
- spustni seznam „Product“ po imenu živila,
- izpustni seznam „Location“ po imenu lokacije.

S klikom na gumb „Reset“ pobrišemo trenutne iskalne kriterije.

Pod razdelkom s filtri se v tabelarnem seznamu prikazujejo filtrirane meritve z nekaj osnovnimi informacijami in možnostjo ogleda podrobnosti (gumb „Details“) ali brisanjem (gumb „Delete“).

Gumb „Details“ omogoča prehod na stran s podrobno analizo meritve.

Privzeto imamo prikazano deset meritev na eni strani. Ostale si lahko ogledamo s klikom na ikoni puščice naprej ali nazaj.



Slika 4.7: Zaslonski predlog z podrobnimi informacijami in analizo meritve.

Na sliki 4.7 v skrajnem desnem robu imamo s kliki na ikone možnost urejanja, brisanja ali izvoza meritve v dokument PDF. Poleg že omenjenih osnovnih informacij meritve se nam v razdelku ugotovitve in opozorila („Findings and Alerts“) izpišejo morebitna opozorila, celoten čas izven priporočenih temperatur in skupni čas po lokacijah.

Vsi zapisi meritve se poleg tabelarične izpišejo tudi v grafični obliki. Graf po Y osi prikazuje temperaturo (levo) in relativno vlago (desno), po X osi pa celotni čas meritve. Črti na grafu določata temperaturo (modra črta) in relativno vlago (črna črta) v odvisnosti od časa.

Ob kliku na ikono znotraj grafa v skrajnem desnem robu imamo možnost pogleda čez celoten ekran, izvoz v različne formate ali tiskanje grafa.

V zadnjem razdelku sta vizualizirana dva modela napovedi preostale dobe



uporabnosti živila. Podatek „Storage-T“ prikaže priporočeno temperaturo hranjenja živila, ki je del naše meritve. Podatek „SL-Ref“ sporoča referenčno dobo uporabnosti živila pri poročni temperaturi v število dneh.

Na desni strani nam z dvema različnima barvama sistem izračuna preostalo dobo uporabnosti po modelu CSIRO in SAL.

Graf pri dobi uporabnosti je podoben prvemu, le da vsebuje še dve linearni črti, ki ponazarjata kako se je doba uporabnosti spreminjala skozi čas meritve.

**Settings**

Products: [Add Product](#)

Name	SLT	Storage T (interval)	Description	Edit / Delete
Fish Orada	4	-2,4	Lorem ipsum	

Locations: [Add Location](#)

Name	Storage T (interval)	Description	Color	Edit / Delete
Hladilnica 32	(-20,0)	Lorem ipsum	Yellow	

Devices

ID	Name
1	TIDA
2	RHT10
3	Generated

Users

Name	Role	Approved
Admin user admin@gmail.com	Admin	01-01-2020
Manager user manager@gmail.com	Manager	<a href="#">Approve</a>

Slika 4.8: Spletno okno predstavlja stran s seznamom živil, lokacij, naprav in uporabnikov.

Seznam živil je predstavljen v razdelku „Products“ v obliki tabele. Vsako živilo lahko urejamo ali brišemo. Imamo tudi možnost dodajanja novega živila.

Seznam lokacij je predstavljen v razdelku „Locations“ v obliki tabele. Vsako lokacijo lahko urejamo ali brišemo. Imamo tudi možnost dodajanja nove lokacije.

Seznam naprav je predstavljen v razdelku „Devices“. Seznam naprav lahko samo pregledujemo, saj so že pred-nastavljene v spletni aplikaciji.

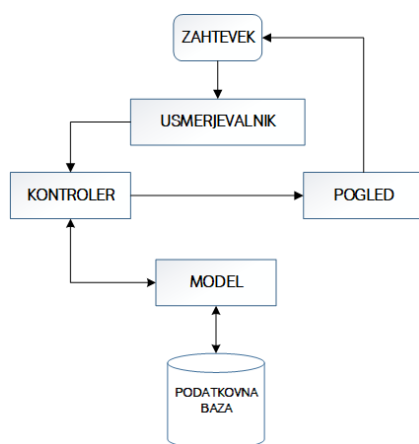
Seznam uporabnikov je predstavljen v razdelku „Users“. Administrator ima pri neodobrenih uporabnikih možnost odobritve s klikom na gumb „Approve“, ostali uporabniki ne vidijo tega stolpca.

### 4.2.3 Arhitektura

Spletna stran je narejena z odprtokodnim ogrodjem Laravel [8], ki temelji na programskem jeziku PHP [23]. Ogrodje je arhitekturno zasnovano po vzorcu MVC.

Princip arhitekturnega vzorca MVC temelji na delitvi aplikacije na tri medsebojno povezane komponente. Model (ang. model) ima odgovornost za podatke, kontroler (ang. controller) vsebuje poslovno logiko in pogled (ang. view) skrbi za prikaz podatkov uporabniku [1].

Spodnja slika 4.9 prikazuje tok MVC v ogrodju Laravel. Tok se začne z uporabniškim zahtevkom (ang. request), ki potuje preko usmerjevalnika oziroma naslova URL (ang. route) do kontrolerja. Kontroler običajno pridobi podatke iz podatkovne baze preko modela. Vsa komunikacija od kontrolerja potuje tudi v obratni smeri do pogleda, kjer se uporabniku prikažejo podatki [7].



Slika 4.9: Tok MVC v ogrodju Laravel.

---

Dinamični deli aplikacije delujejo po principu spletne storitve REST. Za upravljanje in prikaz dinamičnih delov uporabniku so nam v pomoč različne knjižnice Javascript kot na primer bootstrap-table.js [3] za prikaz podatkov v tabeli in hightcharts.js [5] za prikaz podatkov v grafu.



## Poglavje 5

# Razvoj

Poglavje podrobno opisuje bistvene funkcionalnosti spletne aplikacije. V začetku opišemo kako smo se lotili implementacije podatkovne baze in zasnove spletne aplikacije ter storitve, v nadaljevanju pa opišemo posamezne funkcionalnosti z zaslonskimi slikami in pomembnimi odseki izvirne kode.

### 5.1 Podatkovna baza

Za vir podatkov naše spletne aplikacije smo izbrali podatkovno bazo MySQL in v njej kreirali shemo. Znotraj sheme smo definirali tabele na podlagi modelov, ki pa so bili narejeni po vzoru ER diagrama iz prejšnjega poglavja.

Za implementacijo tabel smo si pomagali s tehnologijo Migrations znotraj ogrodja Laravel. Migrations je zapis definicije tabele v obliki datoteke PHP, ki nam omogoča lažje verzijoniranje, spreminjanje in deljenje med razvojnimi okolji [10].

Primer programske kode za kreiranje tabele uporabnik s pomočjo tehnologije Migrations Laravel:

```
1 class CreateUsersTable extends Migration {
2     public function up() {
3         if (!Schema::hasTable('users')) {
4             Schema::create('users', function (Blueprint
5                 $table) {
6                     $table->bigIncrements('id');
7                     $table->string('name');
8                     $table->string('email')->unique();
9                     $table->timestamp('email_verified_at')->
10                        nullable();
11                     $table->string('password');
12                     $table->timestamp('approved_at')->nullable();
13                     $table->rememberToken();
14                     $table->timestamps();
15                 });
16     }
17 }
```

Ogrodje Laravel nam omogoča tudi vmesnik ukazne vrstice imenovane Artisan, ki nam olajša razvojno delo z številnimi uporabnimi ukazi, kot so ustvari kontroler (`php artisan make:controller`), počisti predpomnilnik (`php artisan cache:clear`), pokaži listo povezav (`php artisan route:list`) [9].

Tako smo tudi samo kreacijo tabel izvedli z vmesnikom Artisan spodnje ukazne vrstice:

```
php artisan migrate
```

## 5.2 Spletna aplikacija in storitev

Vsaka komponenta MVC spletne aplikacije vsebuje usmerjevalnik, ki sprejme uporabnikove zahteve HTTP in jih posreduje do ustreznega kontrolerja. V tabeli 5.1 je seznam in opis vseh usmerjevalnikov aplikacije.

naslov URL	metoda HTTP	opis
/LOGIN	GET	Prijava v aplikacijo.
/LOGOUT	GET	Odjava iz aplikacijo.
/REGISTER	GET	Registracija v aplikacijo.
/APPROVAL	GET	Opozorilna stran pred odobritvijo.
/IMPORT	GET	Prikaz strani z uvozom meritve.
/IMPORT/PARSE	POST	Preverba in razčlenitev meritve.
/IMPORT/SAVE	POST	Shranitev meritve.
/IMPORT/GENERATE	POST	Shranitev generirane meritve.
/RECORDS	GET	Seznam meritev.
/RECORD/{ID}	GET	Ogled posamezne meritve.
/RECORD/EDIT/{ID}	POST	Shranitev sprememb meritve.
/RECORD/DELETE/{ID}	DELETE	Izbris meritve.
/SETTINGS	GET	Naslov za ogled živil, lokacij, naprav in uporabnikov.
/SETTINGS/APPROVE/{USER_ID}	GET	Naslov za odobritev novega uporabnika.
/PRODUCT/STORE	POST	Shranitev novega živila.
/PRODUCT/EDIT/{ID}	POST	Shranitev sprememb živila.
/PRODUCT/DELETE/{ID}	DELETE	Izbris živila.
/LOCATION/STORE	POST	Shranitev nove lokacije.
/LOCATION/EDIT/{ID}	POST	Shranitev sprememb lokacije.
/LOCATION/DELETE/{ID}	DELETE	Izbris lokacije.

Tabela 5.1: Seznam in opis vseh usmerjevalnikov aplikacije.

## 5.3 Prijava, odjava, registracija in potrditev uporabnika

Funkcionalnosti povezane z avtentikacijo in avtorizacijo smo povezali v eno smiselno komponento MVC, ki vsebuje dva modela; uporabnik (ang. User) in pravilo (ang. Role), tri kotrolerje in poglede; prijava (`LoginController`, `login.blade.php`), registracija (`RegisterController`, `register.blade.php`) in potrditev (`ApprovalController`, `approval.blade.php`).

The image shows two side-by-side web forms for the 'COLDCHAIN' application. The left form is for registration, featuring fields for Name, E-Mail Address, Password, and Confirm Password, with a green 'REGISTER' button and a link 'Sign In' for existing users. The right form is for login, featuring fields for E-Mail Address and Password, a 'Remember Me' checkbox, a green 'LOGIN' button, and a link 'Sign Up Here' for new users.

Slika 5.1: Zaslonska posnetka registracije (levo) in prijave (desno). Novo registrirani uporabnik privzeto dobi vlogo obiskovalca. Le-tega pa mora administrator, pred nadaljnjo uporabo aplikacije, odobriti.

Primer glavne funkcije potrditvenega konrolerja:

```

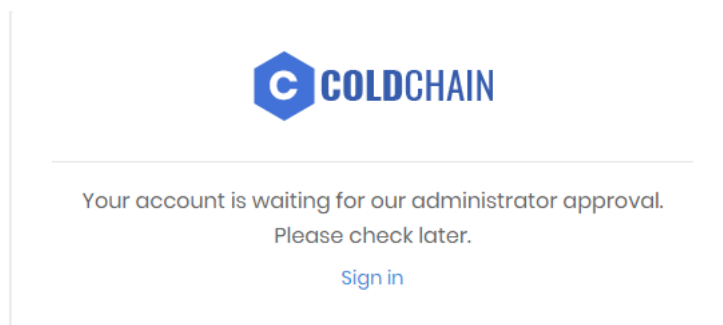
1 public function handle($request, Closure $next)
2 {
3     if (!auth()->user()->approved_at) {
4         return redirect()->route('approval');
5     }
6     return $next($request);
7 }

```

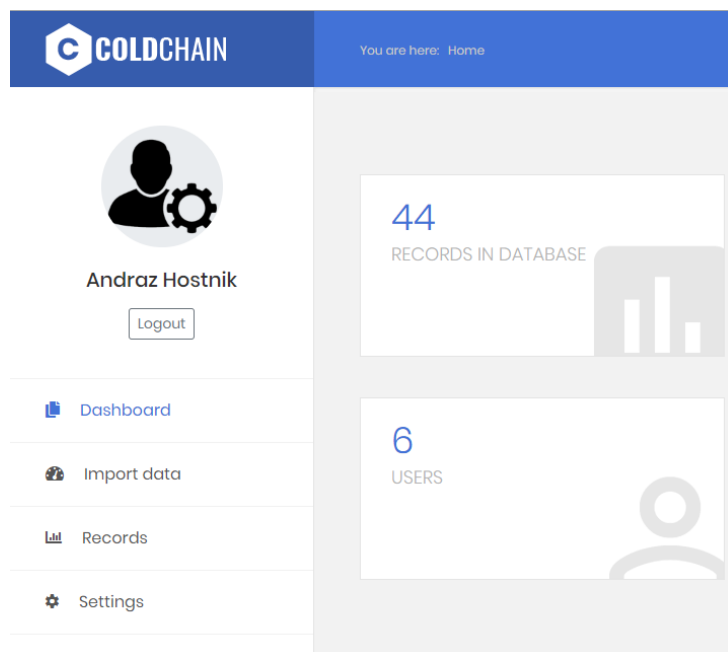


Primer pogleda pri neodobrenem uporabniku:

```
1 @extends('layouts.login')
2 @section('content')
3 <div class="container">
4     <div class="login-logo">
5         <a href="{{ route('login') }}">
6             
7         </a>
8     </div>
9     <div class="login-form text-center">
10         <p>Your account is waiting for our administrator approval.</p>
11         <p>Please check later.</p>
12         <p><a href="{{ route('login') }}" class="btn btn-link ">Sign in</a></p>
13     </div>
14 </div>
15 @endsection
```



Slika 5.2: Zaslonski posnetek pogleda pri neodobrenem uporabniku.



Slika 5.3: Po prijavi odobrenega uporabnika se nam odpre prva avtorizirana stran, ki tako kot vse preostale, vsebuje navigacijsko vrstico, logotip, prikazno sliko, ki ponazarja vlogo uporabnika, in povezavo do odjave.

## 5.4 Uvoz in generiranje meritev

Ena iz med glavnih funkcionalnosti urednika je uvoz ali generiranje meritev. Za uvoz meritve smo razvili posebno logiko prepoznavne glavnih meritvenih podatkov uvožene datoteke. Na podlagi determinatorja, števila in imena stolpcev v fazi shranjevanja ponudimo uporabniku možnost konfiguracije meritvenih podatkov.

Primer privatne funkcije znotraj kontrolerja ImportController za prepoznavo meritev tipa TIDA:

```

1 // P A R S E   T I D A   files
2 // input - data of file
3 // output - object record
4 private function parseTIDAFiles($data){
5     $record = new Records();
6     $record->device_id = 'TIDA';
7     // check if string ;; exist and remove this
8     $lastEl = key(array_slice($data[0], -1, 1, true));
9     if(array_key_exists(0, $data) && strpos($data[0][$lastEl
10 ], ';;') !== false){
11         $data = $this->removeWhiteSpace($data);
12     }
13     // parse headers data
14     foreach($data as $key => $row){
15         if(array_key_exists(1, $row)){
16             if($row[1] == 'Title') $record->title = $row[2];
17             if($row[1] == 'Description') $record->comments =
18 $row[2];
19             if($row[1] == 'Location') $record->location_id =
20 $row[2];
21             if($row[1] == 'StartDate') $record->start_date =
22 $row[2] . ' ' . $row[3];
23             if($row[1] == 'EndDate') $record->end_date = $row
24 [2] . ' ' . $row[3];
25             if($row[1] == 'LogInterval(s)') $record->
26 intervals = $row[2];
27             if($row[1] == 'Measurements') $record->samples =
28 $row[2];
29         }
30         if($row[0] != '#' ){
31             $record->nr_header_rows = $key;
32             break;
33         }
34     }
35     return $record;
36 }

```

Na sliki 5.4 V razdelku „Define headers data“ smo pri stolpcu „light“ iz izpustnega seznama izbrali „ignore–“, kar pomeni ignoriranje omenjene vrednosti pri uvozu v bazo. V spodnjem desnem delu je med drugim razvidno tudi, da gre za tip naprave TIDA, sami pa smo izbrali, da gre za produkt Ribe - tuna na lokaciji Hladilnik 1. Z klikom na modri gumb „Import Data“ se meritev shrani.

**Complete record data**

Define headers data:

date	time	temperature	light	humidity
2018-10-11	13:24:28	25.30	445.0	61.0
2018-10-11	13:50:00	24.60	199.0	58.0

**Title: \***

**Comment:**

**File report:**

File name: **LeM\_Lcsv**

Device: **TIDA**

Product:

Location:

Start Date: **2018-10-11 13:24:28**

End Date: **2018-10-12 15:10:00**

No.Samples: **154**

Interval (s): **600**

Slika 5.4: Zaslonski posnetek shranjevanja meritve z imenom „TIDA1-2“.

Slika 5.5 prikazuje primer generiranja meritve pri kateri smo pod razdelkom „Required data“ izbrali produkt Piščanec – file, začetek meritve ob 16:36:54 dne 7.2.2020 z intervalom 3600 sekund. V nadaljevanju smo določili tri generirane sklope. Prvi vsebuje 100 zapisov z temperaturo 3°C, vlažnostjo 33% na lokaciji Hladilnik 1, drugi 50 zapisov z temperaturo 6°C, vlažnostjo 45% na lokaciji Hladilnik 2 in tretji 20 zapisov z temperaturo 2°C, vlažnostjo

**Generate record data**

**Required data:**

Product: Piščanec - file

Start timestamp: 2020-02-07 18:36:54

Interval (s): 3600

Title: \* Testna generirana meritev

Comment: Preverjamo delovanje.

Records: \* 100 - Temp: \* 3 Hum: \* 33 Hladilnik 1 +

Records: \* 50 - Temp: \* 6 Hum: \* 45 Hladilnik 2 + x

Records: \* 20 - Temp: \* 2 Hum: \* 30 Hladilnik 1 + x

Generate Data Cancel

Soba  
Hladilnik 2  
Hladilnik 1

Slika 5.5: Slika prikazuje primer generiranja meritev.

30% na lokaciji Hladilnik 1. Z klikom na modri gumb „Generate Data“ se meritev shrani.

**Success** You successfully save record data with ID 155. [See details on this link.](#) x

**Import record data**

Support csv, txt files

Choose File No file chosen

Parse CSV or Generate data

Slika 5.6: Primer sporočila, ki se pojavi po uspešno shranjeni meritvi. Odebeljeni del sporočila omogoča prehod do strani z analizo uvožene ali generirane meritve.



Slika 5.7: V primeru napačnega formata datoteke se uporabniku izpiše rdeče opozorilo z besedilom „File is not in proper format.“.

## 5.5 Pregled, iskanje in filtriranje meritev

Glavna funkcionalnost v spletni aplikaciji je ravno pregled podrobnosti in vizualizacija z grafično analizo meritve. V njej prikažemo uvožene, izračunane, ugotovljene in predvidene podatke o izbrani meritvi v obliki alinej, tabel ali grafov.

Najprej izpišemo podatke uvožene iz datoteke, med njimi tudi robne vrednosti, kot prikazuje slika 5.8 v spodnjem delu desno.

Robne vrednosti so zapisane v obliki tabele in pomenijo naslednje podatke:

- Limits – oznaka vrednosti, Value – dejanska vrednost, Count – število ponovitev,
- Low T(°C) – najnižja temperaturna vrednost,
- Hight T(°C) – najvišja temperaturna vrednost,
- Low H(%) – najnižja vrednost relativne vlage,
- Hight H(%) – najvišja vrednost relativne vlage.

Levi del slike prikazuje ime osebe in čas od uvoza ter komentar meritve.

Na zeleni podlagi so izpisane splošne ugotovitve hladne verige iz podatkov. Podatek „Measurements (Location, Time)“ prikazuje čas meritve na

You are here: [Home](#) / [Records](#) / [Record 155](#)

### Record 155 – Generated test data 3

Upload: 6 days ago, Andraz Hostnik  
Comments:

**Findings!**

**Measurements** (Location, Time):  
Hladilnik 1, 10 hours  
Hladilnik 2, 5 hours

**Remained Shelf Life:**  
CSIRO: 9 days, 20 hours  
SAL: 9 days, 1 hours

**Alarms!**

**Temperature is out of range:**  
0 (°C) - 4.0 (°C): 5 hours

**Relative humidity is incorrect:**  
< 0(%) or > 100(%): 1 samples

File name: **Generated file**

Product: **Svinina**

Location: **Hladilnik 1, Hladilnik 2**

Device: **Generated**

Samples: **15**

Start Date: **2020-02-07 10:13:25**

End Date: **2020-02-08 01:13:25**

Interval (s): **3600**

Limits	Value	Count
Low T (°C)	2	10
Hight T (°C)	6	5
Low H (%)	30	10
Hight H (%)	105	1

Slika 5.8: Del zaslonskega posneteka strani z analizo in podrobnosti meritve.

posamezni lokaciji. Čas je zapisan v človeku berljivem načinu. Iz slike 5.8 lahko izberemo, da je meritev bila izvedena na dveh lokacijah, in sicer Hladilnik 1 z časom 10 ur ter Hladilnik 2 z časom 5 ur.

Primer izpis skupnega časa po lokacijah meritve. Metoda `forHumans` je del knjižnice `Carbon`, ki služi kot razširitev programskemu jeziku PHP pri uporabi s časom (ang. `DateTime`) [4]:

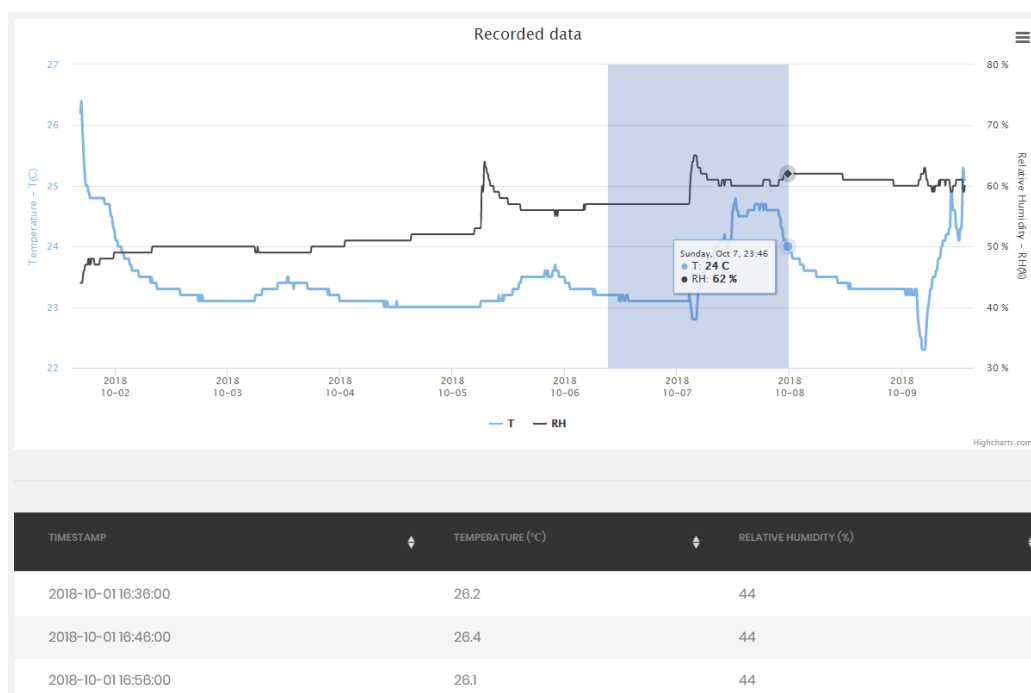
```

1 <div class="alert alert-success" role="alert">
2   <h4 class="alert-heading">Findings!</h4>
3   <p><b>Measurements</b> (Location, Time):</p>
4   @foreach($locationsPerTime as $location)
5     <p>{{ $location->name }}, {{ $location->sumTime->
6       forHumans() }}</p>
7   @endforeach
8 </div>

```

Pod izpisom skupnega časa po lokacijah se izpiše podatek o preostali dobi uporabnosti živila po modelu CSIRO in SAL. Podrobno o omenjenih dveh podatkih je opisano v naslednjem poglavju.

V primeru, da meritev vsebuje opozorila, se nam le-ta izpišejo na rdeči podlagi. Podatek „Temperature is out of range“ kot območje prikaže čas pri katerem je bilo živilo izven priporočenih temperatur hranjenja. V primeru iz slike 5.8 razberemo, da je bilo živilo 5 ur izven priporočenega območja temperature (od 0 °C do 4°C). Če meritev vsebuje neveljavne vrednosti relativne vlage, izpišemo število takih vzorcev pri podatku „Relative humidity is incorrect“. Iz zgornje slike 5.8 vidimo, da je en vzorec meritve vseboval vrednost manjšo od 0% ali večjo od 100%.



Slika 5.9: Graf in tabela prikaza meritvenih podatkov.

V osrednjem delu strani z podrobnostjo meritve se nahaja graf, ki prikazuje podatke o temperaturi in relativni vlagi v odvisnosti od časa. Za grafom je prikazan tudi izpis omenjenih podatkov v obliki tabele. Zgoraj na sliki 5.9



vidimo, da smo označili časovni interval, ki ga želimo povečati. Z miško smo se ustavili na vrednosti, ki je prikazana v malem pojavnem oknu (čas 23:46, temperatura 24°C, relativna vlaga 62%). V primeru, da želimo večjo preglednost, lahko s klikom na posamezno postavko v agendi na dnu grafa skrijemo vrednosti označene postavke.

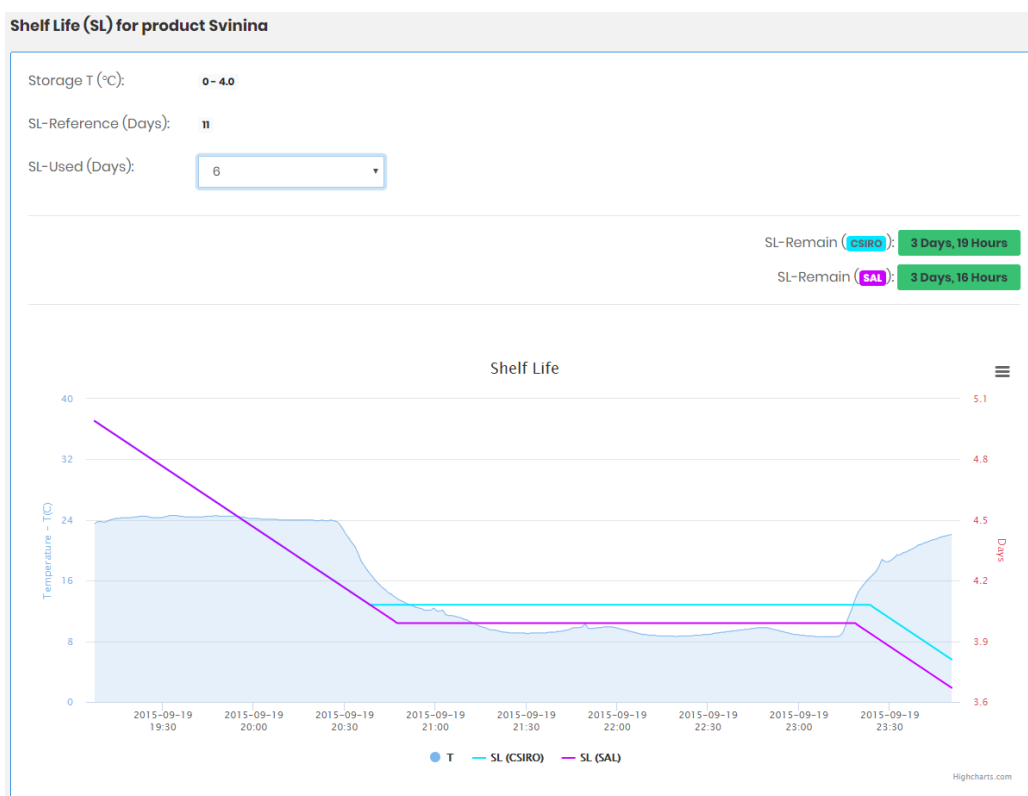
## 5.6 Napoved preostale dobe uporabnosti izbranega živila

V aplikaciji smo za izračun dobe uporabnosti živila implementirali modela, ki sta opisana v poglavju 4.1.2.

Funkciji za izračun preostale dobe uporabnosti živila po modelu CSIRO in SAL:

```
1 // izracun za model CSIRO
2 public function getS1_CSIRO($s1, $interval, $temperature){
3
4     $t = floatval($interval) / 86400; // v casu t
5     $k = pow(1 + round($temperature) * 0.1, 2); // koeficient
6     return round($s1 - $t * $k, 2); // preostala doba
       uporabnosti
7 }
8 // izracun za model SAL
9 public function getS1_SAL($s1, $s1_ref, $interval,
       $temperature){
10
11     $t = floatval($interval) / 86400; // v casu t
12     $t_sal = $this->getT_SALfromTable( round($temperature) );
       // doba uporabnosti pri temperaturi T prebrana iz tabele
       SAL
13     $k = intval($s1_ref / $t_sal); // koeficient iz
       referencne dobe uporabnosti deljeno z $t_sal
14     return round($s1 - $t * $k, 2); // preostala doba
       uporabnosti
15 }
```

Za vsak zapis meritve kličemo zgornji funkciji z parametri preostale dobe uporabnosti ( $\$sl$ ), interval beleženja ( $\$interval$ ) in vrednost temperature ( $\$temperature$ ). Pri modelu SAL dodatno še parameter referenčne dobe uporabnosti ( $\$sl\_ref$ ). Rezultat vsake funkcije je preostala doba uporabnosti.



Slika 5.10: Zaslonski posnetek s podatki in grafom dobe uporabnosti.

Na sliki 5.10 v zgornjem odseku zaslonska posnetka vidimo, da se graf dobe uporabnosti nanaša na živilo svinina („Shelf Life (SL) for product Svinina“), ki ima priporočeno temperaturo hranjenja med 0 in 4 stopinje Celzija („Storage T (°C)“) in referenčno dobo uporabnosti 11 dni („SL-Reference (Days)“).

Iz izpustnega seznama smo izbrali, da je pred meritvijo minilo že 6 dni od izvora živila („SL-Used (Days)“).

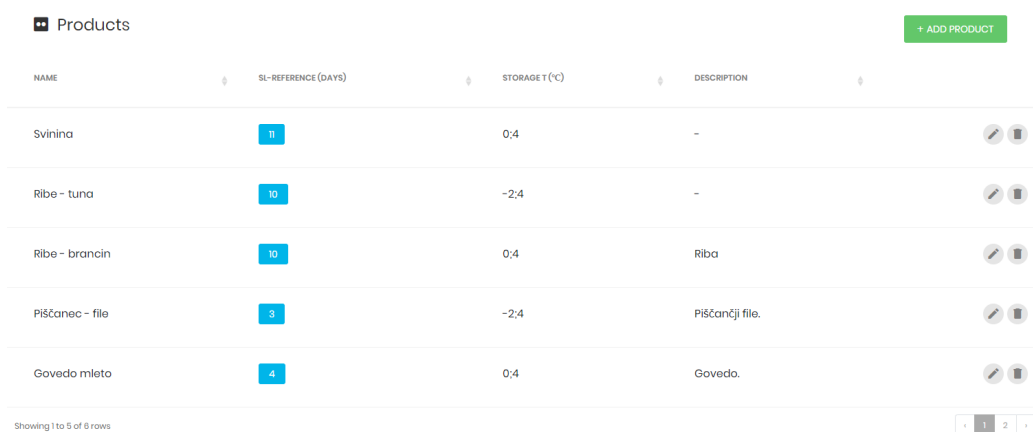
Na grafu modro obarvano območje prikazuje temperaturo, turkizno modra črta dobo uporabnosti po modelu CSIRO in vijoličasta črta dobo uporabnosti po modelu SAL v odvisnosti od časa (x os na grafu).

Iz zgornjega primera lahko razberemo, da je doba uporabnosti živila padala linearno navzdol, kadar so bile vrednosti temperature nad 16°C. Pri vrednosti temperature približno 10°C je bila tudi doba uporabnosti približno konstantna. Doba uporabnosti pri modelu SAL je tam znašala nekoliko manj kot za model CSIRO.

Na koncu meritve pa je preostala doba uporabnosti znašala 3 dni in 19 ur za model CSIRO in 3 dni 16 ur za model SAL („SL-Remain“).

## 5.7 Pregled in urejanje živil

Do strani z nastavitvami („Settings“) imajo dostop uporabniki z vlogo administrator ali urednik. Na omenjeni strani je v prvem razdelku pregled vseh živil v obliki tabele („Products“). Uporabniki imajo možnost dodajanja, urejanja ali brisanja živil.



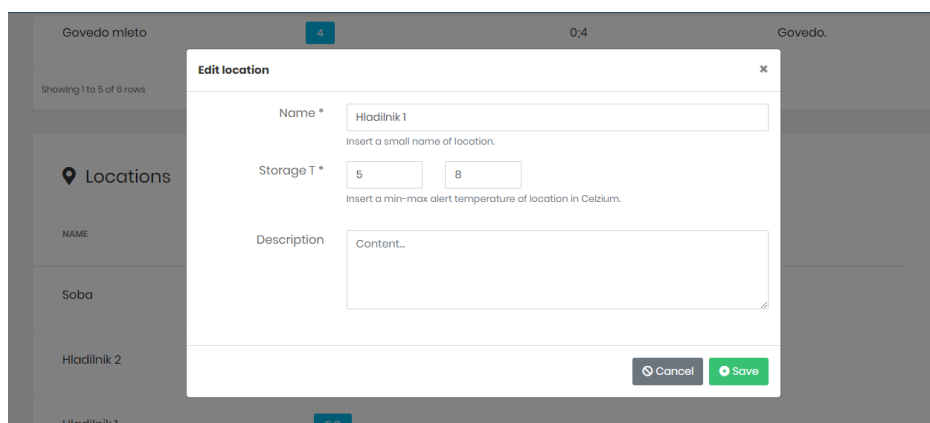
NAME	SL-REFERENCE (DAYS)	STORAGE T (°C)	DESCRIPTION
Svinina	11	0;4	-
Ribe - tuna	10	-2;4	-
Ribe - brancin	10	0;4	Riba
Piščanec - file	9	-2;4	Piščanci file.
Govedo mleto	4	0;4	Govedo.

Slika 5.11: Na sliki so prikazana živila, ki smo jih predhodno vnesli v aplikacijo v obliki tabele razdelku („Products“).

Na sliki 5.11 prvem stolpcu je definirano ime („Name“), v drugem referenčna doba uporabnosti v dnevih („SL-Reference (Days)“), v tretjem območje priporočene temperature hranjenja („Storage T (°C)“) in v zadnjem opis živila. Z modro barvo smo odebelili pomembnejše informacije. Na koncu vsake vrstice se nahajata ikoni za urejanje in brisanje živila.

## 5.8 Pregled in urejanje lokacij

V tabelarnem pogledu so na strani z nastavitvami („Settings“) v drugem razdelku prikazane vse lokacije v aplikaciji („Locations“). Uporabniki imajo možnost pregleda, urejanja ter dodajanja novih lokacij (slika 5.12).



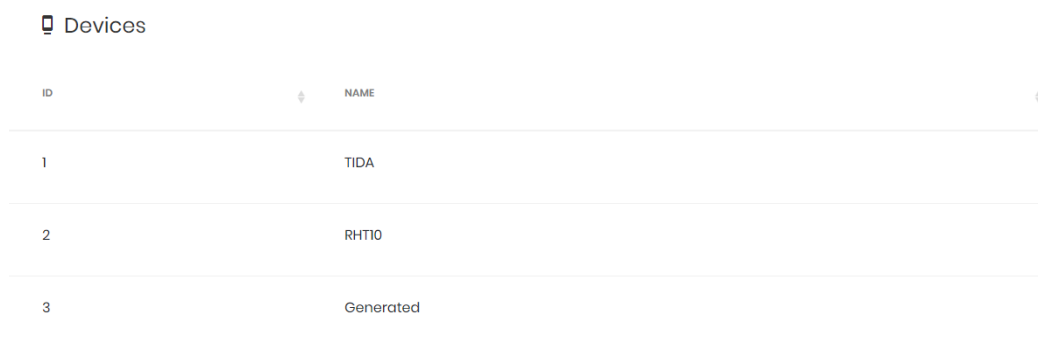
Slika 5.12: Primer dialoga za urejanje lokacije.

Dialog iz slike 5.12 odpremo s klikom na ikono za urejanje pri posamezni lokaciji. Spreminjamo lahko podatke kot so ime („Name\*“), območje delovne temperature („Storage T\*“) in opcijsko opis lokacije. S klikom na gumb shrani („Save“) se zapre dialog in prikaže besedilo o uspešnosti shranitve.

## 5.9 Pregled senzorskih naprav

V aplikaciji smo pri uvozu meritev upoštevali dva različna formata datoteke, ki sta rezultat meritvenih naprav TIDA [?] in RHT10 [20].

Tretji format smo določili za potrebe simulacije pogojev hladne verige in testiranja. Zaradi ročno ustvarjenih podatkov smo ta tip naprave poimenovali kar 'Generated'. Tako imamo v aplikaciji podprte naslednje tri tipe naprav: TIDA, RHT10 in Generated (slika 5.13).

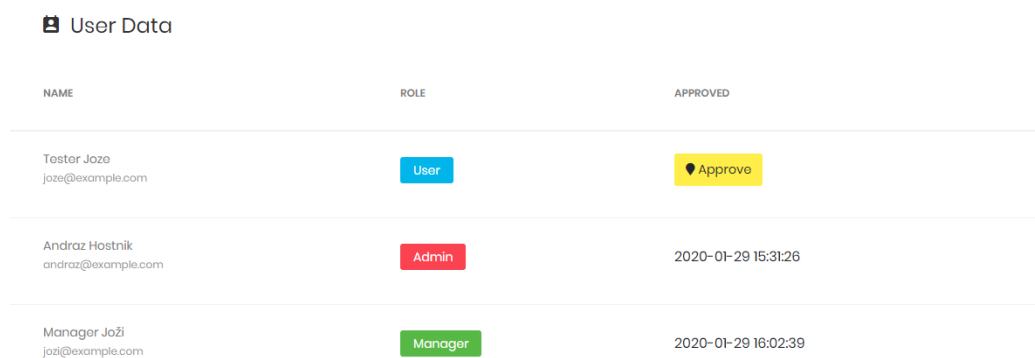



Devices	
ID	NAME
1	TIDA
2	RHT10
3	Generated

Slika 5.13: Na strani z nastavitvami imamo pregled senzorskih naprav („Devices“).

## 5.10 Pregled in potrditev uporabnikov

Administrator ima možnost pregleda in odobritve uporabnikov na strani z nastavitvami v razdelku „User data“. V prvem stolpcu sta izpisana ime in e-pošta, v drugem vloga in v tretjem gumb za potrditev („Approve“) ali čas potrditve v primeru, da uporabnik že ima dostop (slika 5.14).



NAME	ROLE	APPROVED
Tester Jozo joze@example.com	User	
Andraz Hostnik andraz@example.com	Admin	2020-01-29 15:31:26
Manager Joži jozi@example.com	Manager	2020-01-29 16:02:39

Slika 5.14: Pregled uporabnikov, njihovih vlog in status potrditve.

## Poglavje 6

# Testiranje

Spletno aplikacijo smo najprej testirali v lokalnem okolju, kjer smo tudi razvijali nove funkcionalnosti. Kasneje smo spletno aplikacijo, vključno s podatkovno bazo in testnimi podatki, prenesli na zakupljeno gostovanje.

### 6.1 Testiranje spletne storitve

Tehnično smo si pri testiranju pomagali z orodjem za razvijalce, ki je del brskalnika Google Chrome („DevTools“). V njem smo preverjali zapis kode HTML in posamezne klice spletnih storitev [15].

Testirali smo tudi s pomočjo paketa Dump Server, ki je del ogrodja Laravel. Omenjeni paket nam v ozadju delovanja aplikacije zbira podatke, ki smo jih v izvorni kodi določili da se pošiljajo preko funkcije `dump()`, in izpisujejo preko ukazne vrstice [11].

Za vsebinsko testiranje meritev pa smo razvili funkcionalnost posebej namenjeno temu. Simulirali smo podatke, ki jih vsebujejo meritvene datoteke preko generatorja podatkov, ki je bil podrobneje opisan v poglavju 5.4.

## 6.2 Postavitev aplikacije na testno okolje

Za uporabo spletne aplikacije ostalim uporabnikom, smo aplikacijo prenesli na javno dostopno gostovanje.

Najprej smo v administrativnem vmesniku ponudnika gostovanja kreirali podatkovno bazo, kjer smo z pomočjo orodja **phpMyAdmin** [14] uvozili podatkovno shemo in podatke iz lokalnega okolja (slika 6.1). Za tem smo preko protokola za prenos datotek (FTP) prenesli še ostale datoteke spletne strani.

**Web Domain**

Domain	Redirect	SSL	Statistics	Backup
<p>IPv4-Address: *</p> <p>IPv6-Address:</p> <p>Domain: fri.djandro.net</p> <p>Document Root: /var/www/clients/client54/web308</p> <p>Harddisk Quota: 600 MB</p> <p>Traffic Quota: -1 MB</p> <p>Auto-Subdomain: None</p> <p>SSL: <input checked="" type="checkbox"/></p> <p>Let's Encrypt SSL: <input checked="" type="checkbox"/></p> <p>PHP: PHP-FPM</p> <p>PHP Version: php7.2</p> <p>Active: <input checked="" type="checkbox"/></p> <p><b>Save</b> <b>Cancel</b></p>				

Slika 6.1: Administrativni vmesnik ponudnika gostovanja.



## Poglavje 7

# Sklepne ugotovitve

V diplomskem delu smo razvili spletno aplikacijo za vnos, shranjevanje, analizo in vizualizacijo podatkov iz hladne verige. Za podatke, ki so v različnih podatkovnih formatih, smo uspešno razvili enoten uvoz podatkov z možnostjo konfiguracije posameznih senzorskih meritev. Razvili smo generator podatkov, ki pomaga uporabnikom pri ustvarjanju podatkov v simuliranih pogojih hladne verige za potrebe optimizacije procesa, hkrati pa je pomagal tudi nam pri samem testiranju aplikacije.

Osrednja funkcionalnost aplikacije je pregled nad posamezno meritvijo, kjer uporabniku podrobno analiziramo in vizualiziramo ključne podatke ter po modelu CSIRO in SAL napovemo preostalo dobo uporabnosti. Izpostavili bi še nekaj funkcionalnosti, ki uporabnikom olajšajo uporabo aplikacije:

- omogočanje prijave, odjave, registracije in potrditve uporabnika,
- tri uporabniške vloge glede na tip uporabe (administrator, urednik, obiskovalec),
- možnost spreminjanja števila dni med meritvijo in izvora živila,
- iskanje in filtriranje meritev,
- dodajanje, urejanje in izbris živil,
- dodajanje, urejanje in izbris lokacij.

Z izbiro ogrodja Laravel in spletne predloge HTML smo pohitrili razvoj, saj so bili osnovni spletni gradniki in grafične komponente že del omenjenih dveh ogrodij.

Naša spletna aplikacija uspešno rešuje zgoraj omenjene probleme iz področja senzorskih meritev hladne verige. Vidimo pa še kar nekaj možnosti za razširitev in izboljšav na omenjenem področju. Smiselno bi bilo nadaljevati z razvojem in razširiti trenutne funkcionalnosti z naslednjimi:

- poleg ročnega uvoza meritev bi aplikacija lahko imela povezavo z ostalimi podatkovnimi bazami ali spletnimi storitvami s področja hladne verige,
- pri dobi uporabnosti živila, bi lahko aplikacija sama preračunala preostalo dobo uporabnosti glede na trenutni dan ogleda meritve,
- pri napovedovanju preostale dobe uporabnosti živila bi lahko upoštevali še mikrobiološke podatke živila,
- pri analizi meritve, bi lahko sistem sam ugotovil različne lokacije, glede na meritvene podatke,
- na grafu meritve bi lahko prikazali še območje priporočene temperature hranjenja živila,
- dodali bi urejanje uporabniških podatkov kot so: ime, priimek in e-pošta.

# Literatura

- [1] Arhitektura mvc. Dosegljivo: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. [Dostopano: 23. 02. 2020].
- [2] Github. Dosegljivo: <https://sl.wikipedia.org/wiki/GitHub>. [Dostopano: 25. 02. 2020].
- [3] Knjižnica bootstrap table. Dosegljivo: <https://bootstrap-table.com/>. [Dostopano: 02. 02. 2020].
- [4] Knjižnica carbon. Dosegljivo: <https://carbon.nesbot.com/>. [Dostopano: 07. 02. 2020].
- [5] Knjižnica javascript highcharts®. Dosegljivo: <https://www.highcharts.com/blog/products/highcharts/>. [Dostopano: 23. 02. 2020].
- [6] Klemen Marolt. Napovedovanje dobe uporabnosti na podlagi senzorskih podatkov v hladni verigi. Magistrsko delo/naloga, FRI - Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2016.
- [7] Fernando Monteiro. The mvc flow. In *Hands-On Full Stack Web Development with Angular 6 and Laravel 5*, pages 23–64. Packt, 2018.
- [8] Ogrodje laravel. Dosegljivo: <https://laravel.com/>. [Dostopano: 14. 01. 2020].

- 
- [9] Orodje laravel artisan console. Dosegljivo: <https://laravel.com/docs/5.8/artisan>. [Dostopano: 23. 02. 2020].
- [10] Orodje laravel database: Migrations. Dosegljivo: <https://laravel.com/docs/5.8/migrations>. [Dostopano: 23. 02. 2020].
- [11] Orodje laravel dump. Dosegljivo: <https://laravel.com/docs/5.8/telescope#dump-watcher>. [Dostopano: 23. 02. 2020].
- [12] Orodje laravel logging. Dosegljivo: <https://laravel.com/docs/5.8/logging>. [Dostopano: 23. 02. 2020].
- [13] Organizacija csiro. Dosegljivo: <https://en.wikipedia.org/wiki/CSIRO>. [Dostopano: 23. 02. 2020].
- [14] Orodje phpmyadmin. Dosegljivo: <https://en.wikipedia.org/wiki/PhpMyAdmin>. [Dostopano: 07. 02. 2020].
- [15] Orodje za razvijalce google devtools. Dosegljivo: <https://developers.google.com/web/tools/chrome-devtools>. [Dostopano: 23. 02. 2020].
- [16] Podatkovna baza mysql. Dosegljivo: <https://sl.wikipedia.org/wiki/MySQL>. [Dostopano: 23. 02. 2020].
- [17] Predloga html cooladmin. Dosegljivo: <https://github.com/puikinsh/CoolAdmin>. [Dostopano: 23. 02. 2020].
- [18] Programsko orodje intellij idea. Dosegljivo: [https://en.wikipedia.org/wiki/IntelliJ\\_IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA). [Dostopano: 23. 02. 2020].
- [19] Protokol rest. Dosegljivo: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm). [Dostopano: 23. 02. 2020].
- [20] Merilna naprava rht10: Humidity and temperature usb datalogger. Dosegljivo: <http://www.extech.com/products/RHT10>. [Dostopano: 13. 02. 2020].

- 
- [21] Dean Črnigoj. Senzorski moduli nfc in varnost podatkov. Magistrsko delo/naloga, FRI - Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2018.
- [22] Sistem za nadzor različic git. Dosegljivo: <https://git-scm.com/>. [Dostopano: 02. 02. 2020].
- [23] Skriptni programski jezik php. Dosegljivo: <https://sl.wikipedia.org/wiki/PHP>. [Dostopano: 23. 02. 2020].
- [24] Spletni strežnik. Dosegljivo: [https://sl.wikipedia.org/wiki/Spletni\\_stre%C5%BEnik](https://sl.wikipedia.org/wiki/Spletni_stre%C5%BEnik). [Dostopano: 25. 02. 2020].
- [25] Uvod v računalniške informacijske sisteme / baze podatkov. Dosegljivo: [https://sl.wikibooks.org/wiki/Uvod\\_v\\_ra%C4%8Dunalni%C5%A1ke\\_informacijske\\_sisteme/\\_baze\\_podatkov](https://sl.wikibooks.org/wiki/Uvod_v_ra%C4%8Dunalni%C5%A1ke_informacijske_sisteme/_baze_podatkov). [Dostopano: 25. 02. 2020].
- [26] Varnostni koncept iaaa. Dosegljivo: <https://thorteaches.com/cissp-iaaa/>. [Dostopano: 23. 02. 2020].