

**ŽILINSKÁ UNIVERZITA V ŽILINE**

**FAKULTA RIADNIA A INFORMATIKY**

**SEMESTRÁLNA PRÁCA**

**PRENOS DÁT**

Adrián Bednár

Luboš Fukas

Štefan Sliacky

Žilina, 2016

## *Špecifikácia zadania, konkretizácia problému*

Úlohou semestrálnej práce je navrhnuť webovú aplikáciu pre posielanie a spracovanie dát. Preferovaný programovací jazyk je JAVA a webová platforma DropWizard, databáza MySQL s použitím ORM Hibernate.

**Aplikácia bude pozostávať z daných komponentov:**

- **Web servis (REST)**, ktorý bude prijímať a poskytovať dáta. Daný web servis bude pozostávať z minimálne troch koncových bodov. (užívatelia, zariadenia a dáta pre zariadenia). Nad jednotlivými koncovými bodmi bude možné vykonať CRUD operácie.
- **Prezentačná časť** – webové rozhranie, bude zobrazovať namerané údaje, zoznam zariadení atď. Pre lepšie hodnotenie bude potrebné implementovať prihlásenie sa do systému pomocou užívateľského mena a hesla.
- **Koncové zariadenie** – môže byť buď reálne zariadenie postavené na ľubovoľnom MCU, raspberriPi atď. alebo emulované zariadenie, ktoré bude spúšťané na PC. Toto koncové zariadenie bude predstavovať senzor(napr. teplota, tlak, svetelnosť, rýchlosť vetra atď. ), ktorý bude odosielať dáta cez REST rozhranie na server.

# Krátka analýza

## Dropwizard

Je Java framework určený pre vysoko výkonné RESTful webové servisy. Balancuje na hranici medzi knižnicou a frameworkom. Jeho cieľom je poskytovať výkonné, spoľahlivé implementácie všetkého, čo potrebuje webová aplikácia.

## REST (Representational State Transfer)

Je architektúra rozhrania, navrhnutá pre distribuované prostredie. Na rozdiel od XML-RPC či SOAP, je REST orientovaný dátovo, nie procedurálne. Webové služby definujú vzdialené procedúry a protokol pre ich volanie, REST určuje, ako sa pristupuje k dátam a je použiteľný pre jednotný a ľahký prístup ku zdrojom (resources). Zdrojom môžu byť dáta, rovnako ako stavy aplikácie (ak sa dajú popísať konkrétnymi dátami). Všetky zdroje majú vlastný identifikátor URI a REST služba definuje štyri základné metódy pre prístup k nim.

## Metódy pre prístup ku zdrojom

REST implementuje štyri základné metódy, ktoré sú známe pod označením CRUD, vytvorenie dát (Create), získanie požadovaných dát (Retrieve), zmenu (Update) a zmazanie (Delete). Tieto metódy sú implementované pomocou odpovedajúcich metód HTTP protokolu.

- **POST (Create)**

Získavanie dát je jednoduché a priamočiare. Pre vytvorenie dát slúži totiž metóda POST. Pri metóde POST nie je vo chvíli volania známy presný identifikátor zdroja (pretože zdroj ešte neexistuje). Preto sa pre POST používa identifikátor („endpoint“).

- **GET (Retrieve)**

Základnou metódou pre prístup ku zdrojom je získanie zdroja – metóda GET. Každý zdroj podľa rozhraní REST vlastní identifikátor (URI). Pomocou HTTP GET požiadavky získame dáta konkrétného zdroja.

- **PUT (Update)**

Operácia zmeny je podobná ako operácia vytvorenia POST, s tým rozdielom, že voláme konkrétnu URI konkrétného zdroja, ktorý chceme zmeniť. V ňom predávame novú hodnotu (ako pri metóde POST). Na rozdiel od POST je pri úpravách zdroja jeho URI už známe, takže ju môžeme zadať

- **DELETE**

Zdroj je možné zmazať pomocou volania URI HTTP metódy DELETE. Volanie je obdobné ako volanie metódy GET.



CREATE

C



READ

R



UPDATE

U



DELETE

D

## Návrh aplikácie

Aplikácia bola navrhovaná na základe aplikácie navrhutej počas cvičení. Celá aplikácia bola napísaná v jazyku JAVA v prostredí NetBeans.

Prácu sme doplnili o nové triedy pre autentifikáciu užívateľa a pre 2 senzory. Každý senzor má vlastnú Resource triedu, kde sme zdefinovali cesty ku daným CRUD funkciám ako GET, POST, DELETE a PUT.

```
@GET
public SensorTemperatureView getTemperature() {

@GET
@Path("/{id}")
public SensorTemperatureView getTemperatureById(@PathParam("id") String id)

@POST
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
public LoginCheckView postPutTemperature(@FormParam("postId") int id, @FormParam("postContent") String content)

POST
Path("/{id}")
public LoginCheckView postDeleteTemperature(@PathParam("id") int id)
```

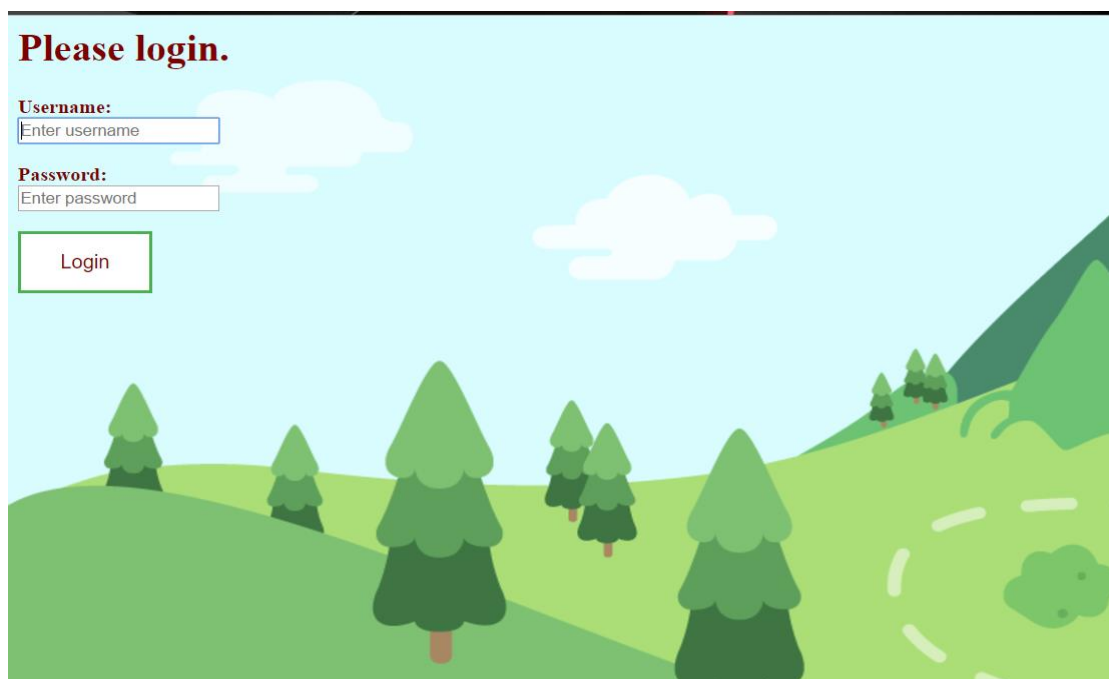
Pre každé zariadenie sme vytvorili databázu na servery <http://www.phpmyadmin.co>. V databáze sa nachádza ID a hodnota záznamu.

Ďalšou časťou bolo vytvoriť senzory, ktoré nám budú odosielať dáta do databázy. My sme sa rozhodli tieto senzory emulovať v jazyku JAVA. Vytvorili sme 2 senzory pre vlhkosť vzduchu a teplotu. Dáta zo senzorov sú náhodne generované a posielané v pravidelných intervaloch cez náš server do databázy. Dáta sú odosielané pomocou jednoduchých http requestov.

```
// Zakladne nastavenie pre POST request
con.setRequestMethod("POST");
con.setRequestProperty("User-Agent", USER_AGENT);
con.setRequestProperty("Accept-Language", "en-US,en;q=0.5");
con.setRequestProperty("Content-Type", "application/json");
String postJsonData = "{\"content\":\"" + value + "\"}";
```

Súčasťou práce je aj prezentačná časť. Tá je tvorená pomocou html stránok a štýl jednotlivých objektov na stránke je tvorený pomocou css. Na stránkach sa užívateľ prihlási pomocou prihlasovacieho mena a hesla a následne si môže vybrať požadovaný senzor. Pri každom senzore je možné pridávať nové hodnoty, mazať jednotlivé údaje, prípadne zobrazit' všetky hodnoty zo senzora.

Každá stránka má takto vytvorený svoj \*.ftl súbor, ktoré sa volajú v jednotlivých triedach View.

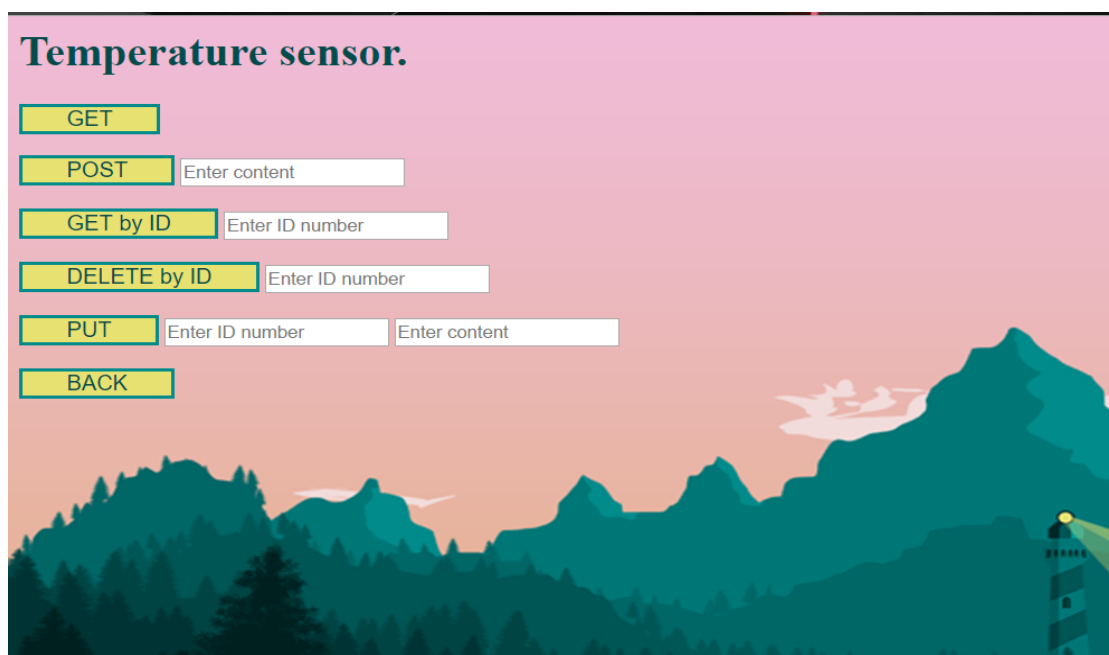


**Please login.**

**Username:**

**Password:**

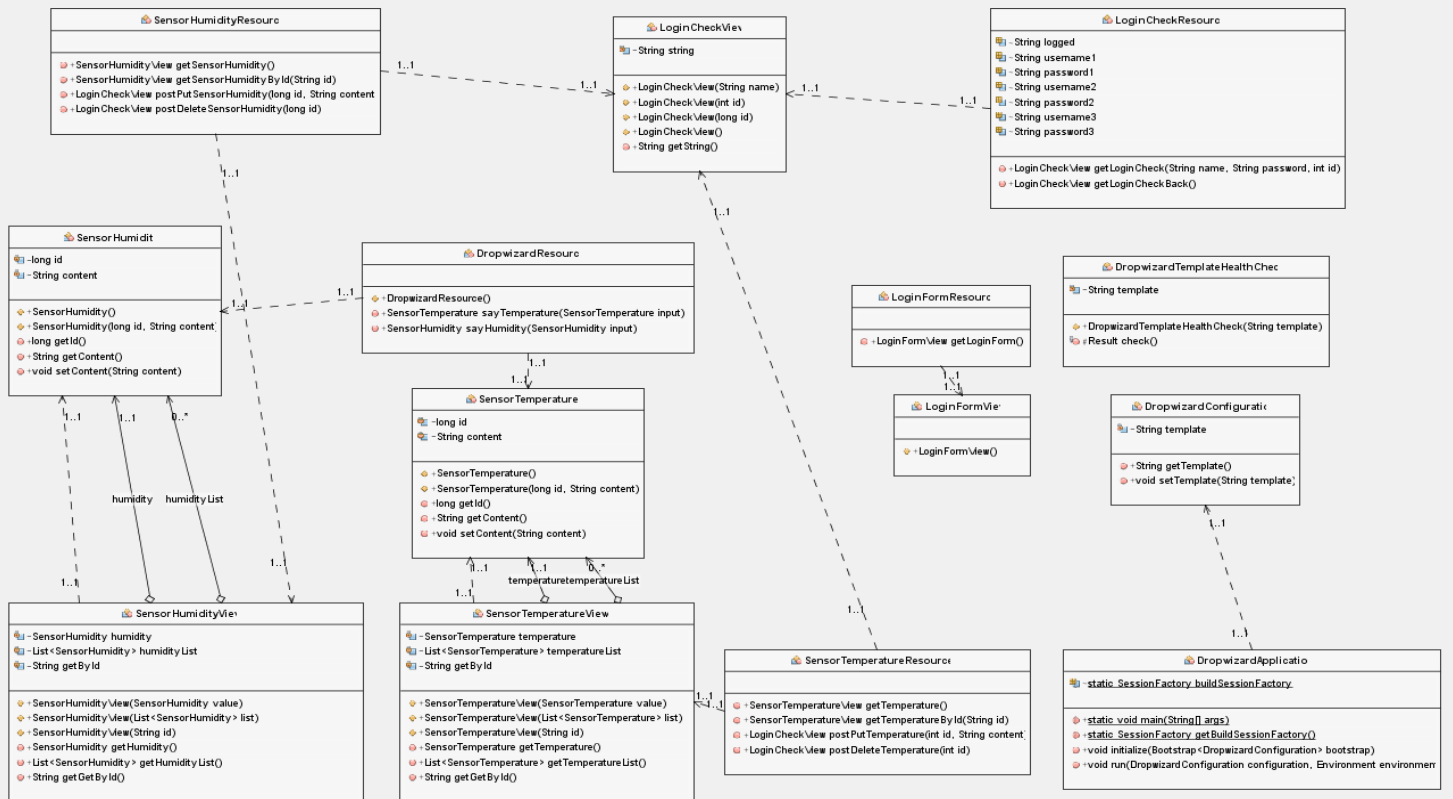
The login page features a light blue sky with white clouds and a green landscape with rolling hills and several dark green coniferous trees. A dashed white line indicates a path leading towards the right side of the screen.



**Temperature sensor.**

The temperature sensor management page has a pink background. It contains several yellow buttons for different actions: GET, POST, GET by ID, DELETE by ID, PUT, and BACK. The POST, GET by ID, DELETE by ID, and PUT buttons are accompanied by text input fields for content or ID numbers. The background image shows a dark green mountain range under a sunset or sunrise sky with orange and pink hues. A small lighthouse is visible on the right side of the mountains.

### *UML diagram*



## **ZÁVER**

Cieľom semestrálnej práce bolo navrhnuť webovú aplikáciu pre posielanie a spracovanie dát. Vytvorili sme aplikáciu, ktorá využíva službu REST na ukladanie prijatých dát zo senzorov do databázy. Okrem toho obsahuje prezentačnú časť vo forme HTML stránok s jednoduchou autentifikáciou. Na posielanie dát do databázy sme simulovali jednoduché senzory, ktoré posielajú dáta v pravidelných intervaloch na server do databázy.