**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY



**Prenos dát**

**Semestrálna práca**

Katedra technickej kybernetiky

Bc. Jaroslav Bútora

Žilina, 2017 Bc. Lukáš Malatinský

# Špecifikácia zadania, konkretizácia problému

Zadanie semestrálnej práce spočívalo vo vytvorení webovej aplikácie, ktorá by spracovávala IOT dáta. Preferované nástroje pre vytvorenie tejto semestrálnej práce bolo využitie jazyka Java, webová platforma DropWizard, databáza MySQL a použitie ORM Hibernate. Na tejto semestrálnej práci môže pracovať tím maximálne troch študentov.

Semestrálna práca má pozostávať z troch komponentov:

1. REST Web servis
2. Prezentačná časť
3. Koncové zariadenie

## REST Web servis

Tento servis 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 má byť možné vykonať CRUD operácie.

## Prezentačná časť

Webové rozhranie, bude zobrazovať namerané údaje, zoznam zariadení atď. Pre lepšie hodnotenie je potrebné implementovať prihlásenie sa do systému pomocou užívateľského mena a hesla.

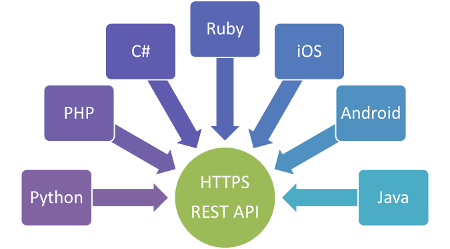
## Koncové zariadenie

Buď reálne zariadenie postavené na ľubovoľnom MCU, Raspberry Pi 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.

Jedným z najpopulárnejších typov API je REST, niekedy nazývané aj RESTful API. REST alebo RESTful API boli navrhnuté tak, aby výhodne stavali na existujúcich protokoloch. Zatiaľ čo REST alebo Representational State Transfer môže byť použitý skoro s každým protokolom, v spojení s webovými API sa zvyčajne využívajú výhody HTTP. To znamená, že vývojári pri vytváraní REST API nemusia inštalovať prídavný software alebo knižnice.

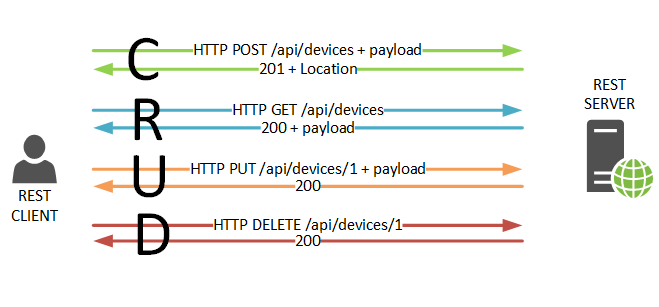
Jednou z hlavných výhod REST API je, že poskytuje vynikajúcu flexibilitu. Dáta nie sú viazané na zdroje alebo metódy, takže REST dokáže spracovať rôzne typy volaní a vrátiť rôzne dátové formáty. Táto flexibilita dovoľuje vývojárom vytvoriť API podľa potreby v kombinácii so spĺňaním požiadaviek veľmi odlišných zákazníkov.

Na rozdiel od SOAP REST nie je viazaný na XML, ale dokáže vrátiť XML, JSON, YAML alebo ktorýkoľvek iný formát v závislosti na tom, ktorý klient vyžaduje. A na rozdiel od RPC nie sú užívatelia nútení vedieť názvy procedúr alebo špecifické parametre v danom poradí.



# Krátka analýza

Ako už bolo spomenuté, jednou z hlavných výhod **REST API** je, že poskytuje vynikajúcu flexibilitu. Dáta nie sú viazané na zdroje alebo metódy, takže REST dokáže spracovať rôzne typy volaní a vrátiť rôzne dátové formáty. Vďaka tejto flexibilite je REST API celkom populárne. Toto REST rozhranie má poskytovať CRUD operácie. Tieto operácie sú skratkou C- Create, R-Read, U-Update a D-Delete.



## 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. Obsahuje v sebe niekoľko ďalších známych knižníc ako napríklad:

### Jetty pre HTTP

Pretože web aplikácia bez HTTP by nemala zmysel, Dropwizard používa Jetty HTTP knižnici aby ju zakomponoval do projektu. Namiesto predávaniu aplikácie komplikovanému aplikačnému serveru má Dropwizard hlavnú metódu main, v ktorej spúšťa HTTP server. Spustenie aplikácie ako jednoduchý proces eliminuje početné množstvo problémových prvkov a dovoľuje použiť existujúce Unixové nástroje pre správu procesov.

### Jersey pre REST

Umožňuje napísať prehľadné, testovateľné triedy, ktoré mapujú HTTP requesty na jednoduché Java obejkty. Podporuje Streamovací výstup, matičné URI parametre, GET requesty s podmienkou a omnoho viac.

### Jackson

Jackson je populárna Java knižnica ktorá umožnuje mapovať JSON objekty na Java objekty.

### Metrics

Knižnica Metrics všetko zaobaľuje a poskytuje jedinečný náhľad do správania sa kódu vo vývoji.

### Iné knižnice

Ako Guava, Logback, Freemarker, Joda Time

## Analýza semestrálnej práce

Aplikácia bude bežať na localhoste teda budeme využívať program WampServer a bude bežať na adrese http://localhost:8080/. Koncové zariadenie budeme mať emluovaná senzory pre meranie teploty vzduchu, tlaku vzduchu a výšky hladiny rieky. Prezentačná časť bude webové rozhranie ktoré poskytuje REST služby. Tu sa bude môcť užívateľ zaregistrovať, prihlásiť sa. Ďalej si bude môcť pozrieť všetky namerané hodnoty od daného senzora, jednu konkrétnu hodnotu. Takisto bude možné vymazať jednu hodnotu alebo všetky hodnoty. Užívateľ bude môcť taktiež pridať hodnotu alebo upraviť už prijatú hodnotu.

# Implementácia

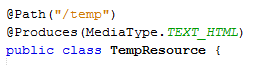
Semestrálne práca nadväzuje na prácu na cvičeniach, teda je vyvíjaná v jazyku JAVA a v prostredí NetBeans 8.2. Tu sme už len projekt upravili a doplnili o požadovanú funkčnosť.

V súbore hibernate.cfg.xml sme si nastavili vlastnú databázu a prístup na ňu:



Ďalej sme vytvorili triedy jednotlivých zariadení a užívateľa. Pre každé zariadenie sme taktiež vytvorili tabuľku v databáze: Temp, Pressure, Level, USER. Ďalej sme tie triedy doplnili o požadované parametre. Pri zariadení ide len o ID a hodnotu, pri užívateľovi to je ID, meno a heslo.

V triedach Resource sme zadefinovali cesty ku daným CRUD funkciam. Pre každé zariadenie sme najskôr určili hlavnú cestu pre prístup ku celej triede:



A ďalej cesty ku jednotlivým funkciám:

 - získanie všetkých hodnôt

- získanie hodnoty podľa ID

- vymazanie hodnoty podľa ID

- vymazanie všetkých hodnôt

- pridanie novej hodnoty

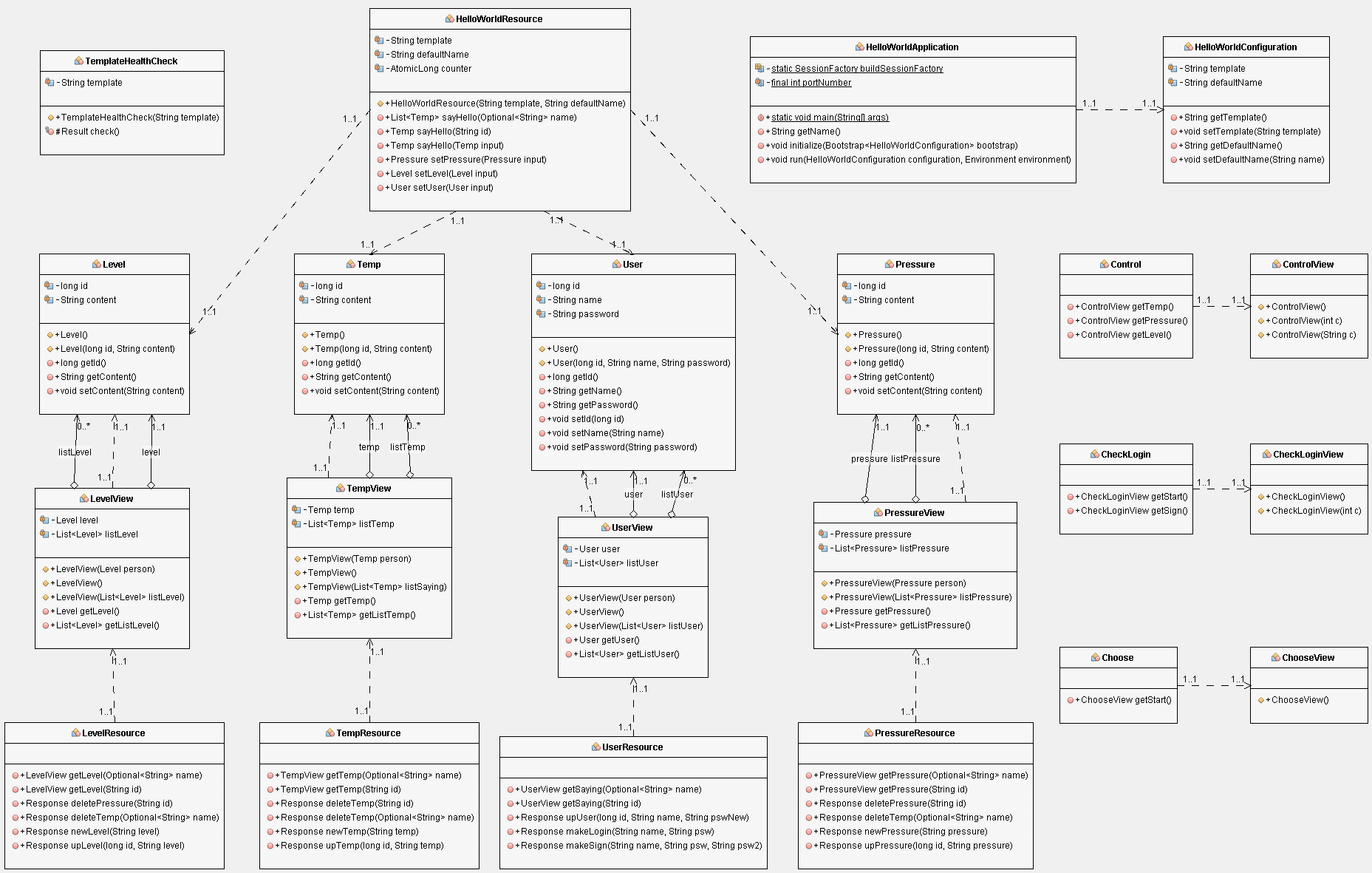
- aktualizácie hodnoty

Podobne je to pri užívateľovi, avšak tu je ešte funkcia pre overenie či zadaný užívateľ je už zaregistrovaný, ak áno tak ho prihlási.



A funkciu pre registráciu, ktorá overuje či náhodou užívateľ s daným menom už náhodou nie je zaregistrovaný.





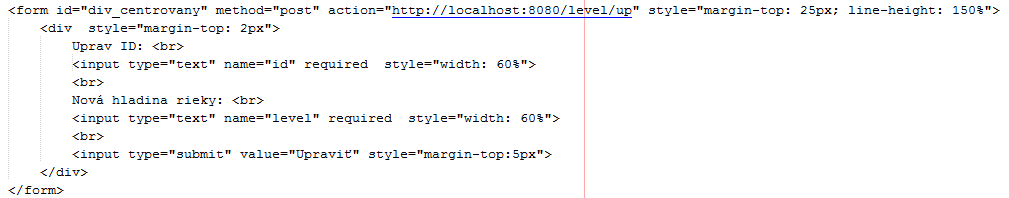
Ďalej pokračujeme vytvorením emulovaných senzorov. Je to nový Java projekt v ktorom je len jedna trieda. Tá obsahuje funkciu postTemperature(), ktorá zabezpečuje nadviazanie spojenia a odoslanie Json-u.



Main metóda obsahuje nekonečný ciklus v ktorom sa generujú náhodne hodnoty, ktoré sa potom odosielajú v určenom intervale.

Ďalej nasledoval návrh prezentačnej časti a teda Web rozhrania. Pre zjednodušenie práce sme pri návrhy používali nástroj PSPad, v ktorom bolo možne jednoducho zobraziť návrh stránky.

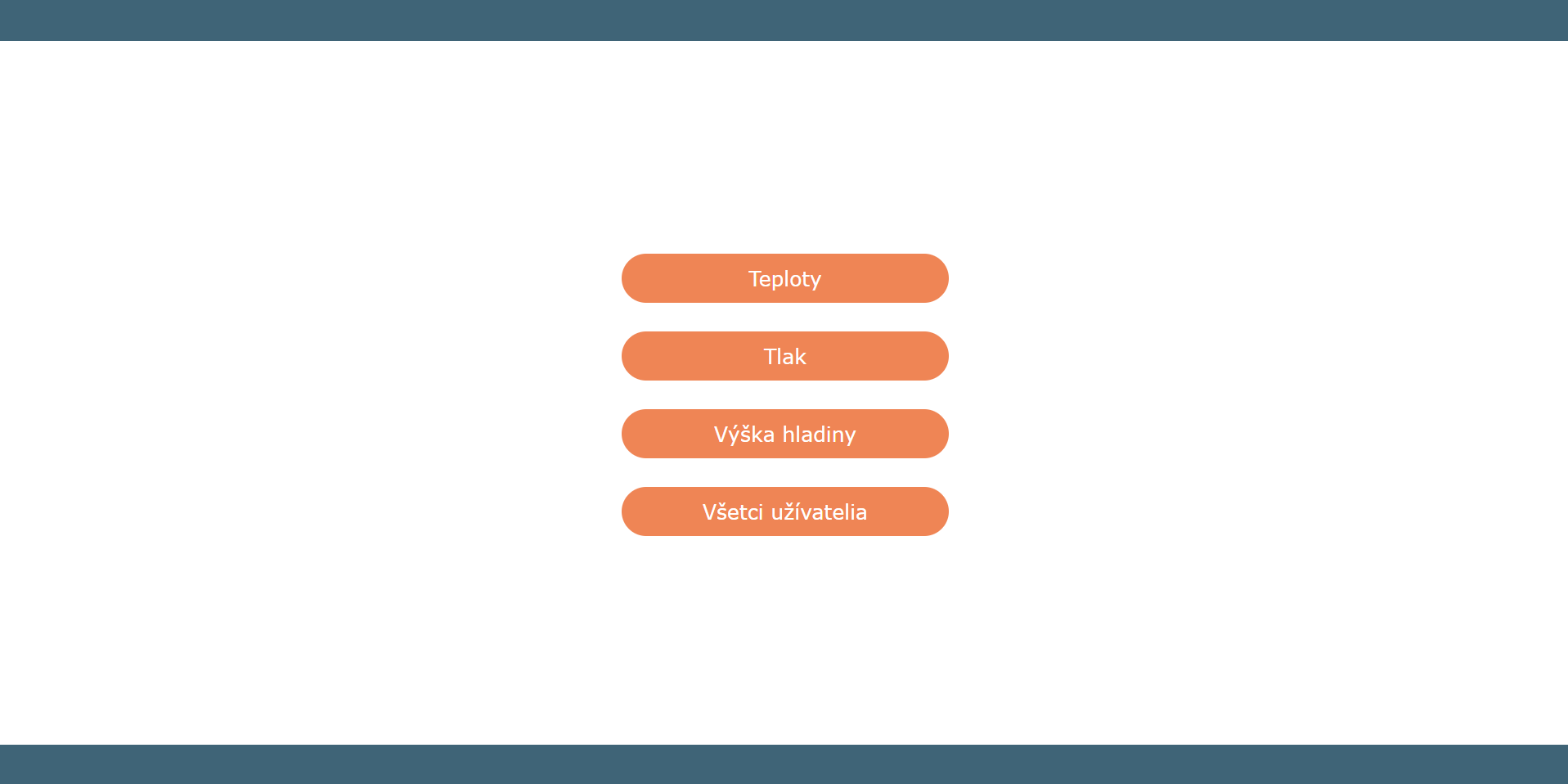
Stránka sa skladá zo štýlu, v ktorom sú pripísané objekty na stránke, ich rozmer tvar, farba, typ a veľkosť písma, rozloženie a podobne. A z formulárov ktoré určujú metódu, teda buď POST alebo GET a akciu ktorá sa má vykonať.



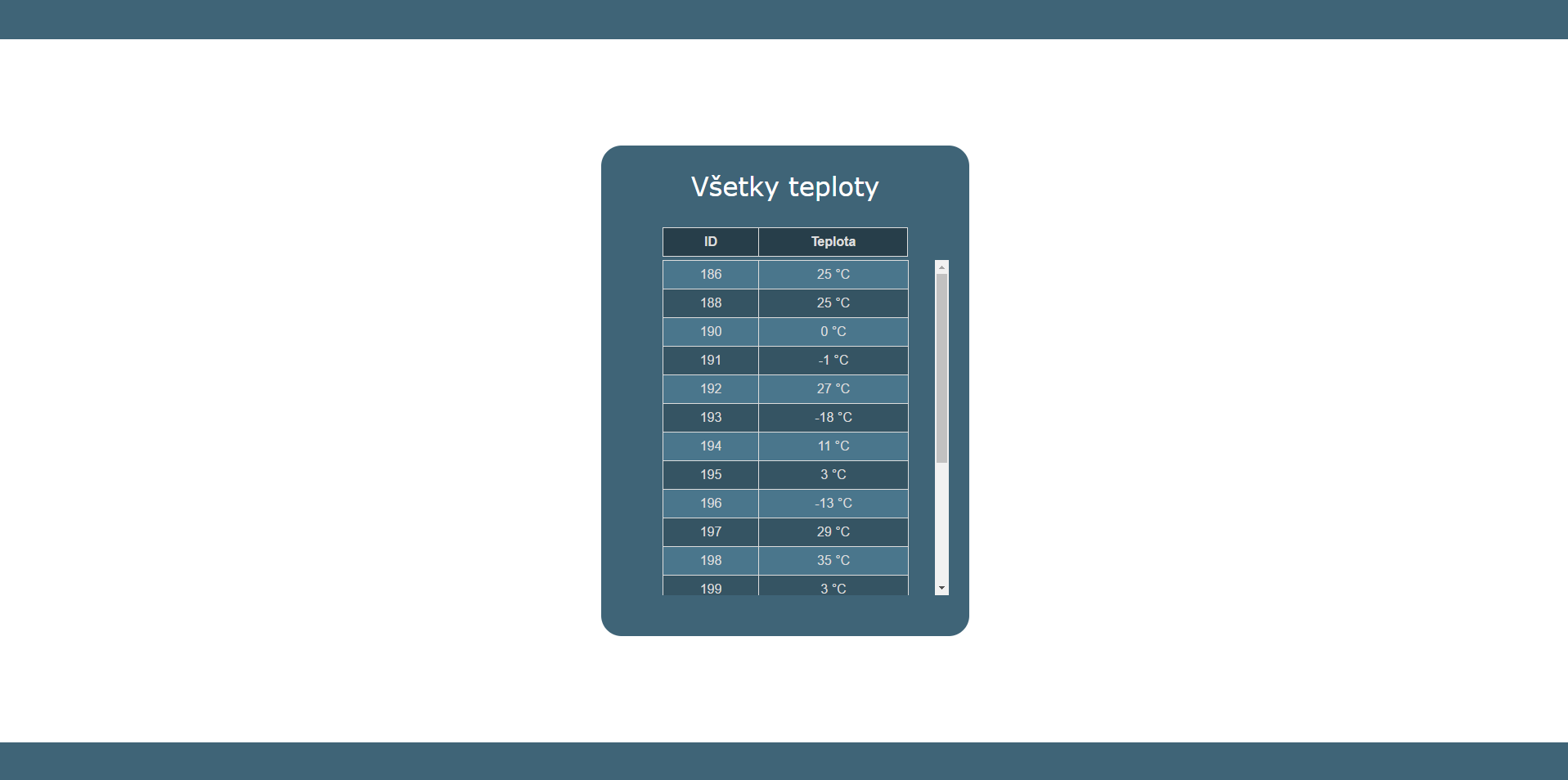
Každá stránka má takto vytvorený svoj \*.ftl súbor, ktoré sa volajú v jednotlivých triedach View.

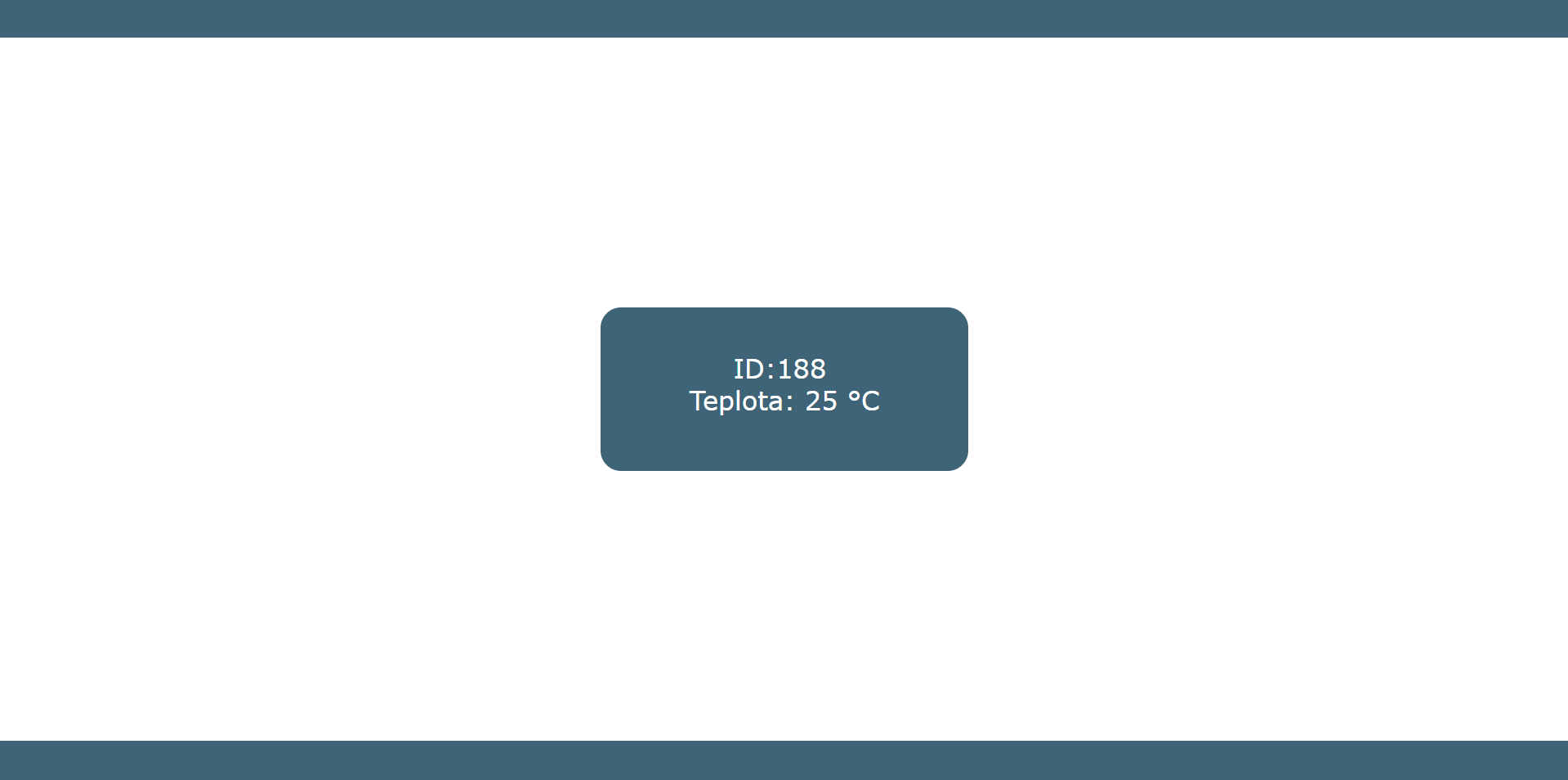














# Záver

Cieľom semestrálnej práce bolo overenie nadobudnutých vedomostí z predmetu Prenos dát. Vytvorili sme semestrálnu prácu, ktorá vie zobrazovať na Web aplikácie pomocou REST služieb namerané údaje z IOT zaradení.