

SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

*Borna Kovačević*

**DIJAGNOSTIKA  
VENTILOKONVEKTORA  
KORIŠTENJEM ANALIZE  
VREMENSKIH SLIJEDOVA  
PODATAKA**

Zagreb, 2019.

Ovaj rad izrađen je uz potporu Europskog fonda za regionalni razvoj kroz projekt 3Smart (Smart Building – Smart Grid – Smart City, <http://www.interreg-danube.eu/3smart>) koji se provodi u sklopu Programa transnacionalne suradnje Dunav, na Zavodu za automatiku i računalno inženjerstvo pod vodstvom prof. dr. sc. Marija Vašaka te asistencijom mag. ing. Anite Martinčević i mag. ing. Danka Marušića, kojima se ovim putem zahvaljujem na pomoći prilikom realizacije ove aplikacije. Rad je predan na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2018./2019.

# Skraćenice

FER – *Fakultet elektrotehnike i računarstva*

HVAC – *Heating, ventilation, and air conditioning*

MVC – *Model View Controller*

JSP – *Java Server Pages*

ORM – *Object Relation Model*

HQL – *Hibernate Query Language*

SQL – *Structured Query Language*

Java EE – *Java Enterprise Edition*

FCU – *Fan coil unit*

HTML – *Hypertext markup language*

# SADRŽAJ

Skraćenice .....	III
Uvod.....	1
1. Opis razmatranog sustava .....	3
2. Sustav za detekciju neispravnog rada ventilokonvektora .....	4
3. Opis aplikacije .....	7
3.1. Povezivanje entiteta u bazi podataka s objektima u Javi .....	7
3.2. Korištenje stvorenih entiteta za prikaz pojedinog mjerenja unutar prostorije .....	8
3.3 Opis algoritama .....	9
3.3.1. Algoritam 1 .....	9
3.3.2. Algoritam 2.....	11
3.3.3 Pronalazak stacionarnog stanja .....	13
4. Opis baze podataka .....	14
4.1. Entitet zgrade .....	14
4.2. Entitet kalorimetar .....	15
4.3. Entitet ventilokonvektor.....	15
4.4. Entiteti prostorije .....	16
5. Korištene tehnologije .....	18
5.1. Java .....	18
5.2. Model-View-Controller (MVC).....	19
5.3. Java Server Pages (JSP) .....	20
5.4. Hibernate .....	21
6. Kontroleri i stranice za prikaz .....	22
6.1. Prikaz i opis dobivenih podataka.....	25
7. Zaključak.....	28
LITERATURA .....	29
Popis slika.....	31
Popis tablica .....	32
Sažetak .....	33
Summary.....	34

# Uvod

Sustavi grijanja i hlađenja zastupljeni su u gotovo svim zgradama i veliki su potrošači energije [1], [2]. Održavanje i nadgledanje sustava grijanja i hlađenja od velike je važnosti za održavanje komfora. Stoga su za te poslove u velikim zgradama najčešće zadužene vanjske tvrtke ili zasebne organizacijske jedinice unutar tvrtke. Glavni cilj ove aplikacije jest olakšati posao operaterima zgrade zaduženim za održavanje sustava grijanja i hlađenja koji se temelje na sustavu ventilokonvektora [3]. Razmatrana konfiguracija sustava je neboderska zgrada Sveučilišta u Zagrebu Fakulteta elektrotehnike i računarstva (FER) koja je u sklopu projekta 3Smart opremljena dodatnom mrežom senzora s ciljem ostvarivanja pametnog upravljanja komforom zgrade ostvarenog kroz hijerarhijsku koordinaciju sustava za grijanje i hlađenje s ostalim podsustavima zgrade što u konačnici zgradu čini fleksibilnim potrošačem energije tehnički spremnim da pruži usluge upravljanja potrošnjom prema različitim entitetima u energetske mrežama. Razvijena aplikacija omogućuje daljinsko nadgledanje sustava grijanja i hlađenja, praćenje važnih varijabli vezanih uz unutarnju klimu u prostorijama zgrade te ono najvažnije detekciju djelomičnog ili potpunog začepljenja izmjenjivača topline ventilokonvektora zrakom. Začepljenje zrakom događa se prilikom ulaska zraka u sustav prilikom radova ili redovnog održavanja sustava primjerice kod prelaska sustava s grijanja na hlađenje i obratno.

Pametni mobiteli, pametni satovi, pametni televizori, sve postaje "pametnije" pa tako i zgrada FER-a. U njoj se mjeri velika količina podataka koji omogućavaju izradu brojnih korisnih aplikacija.

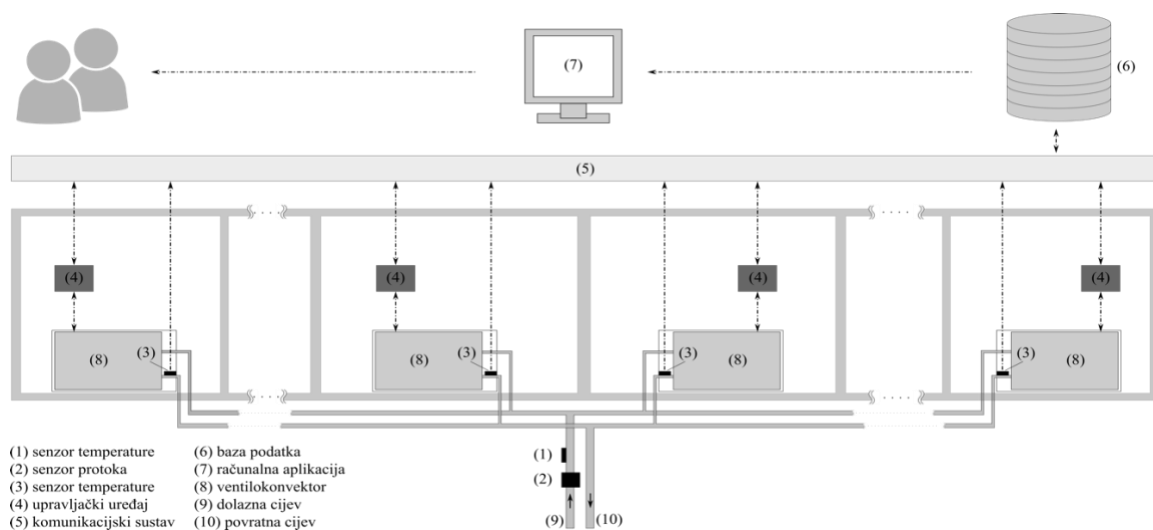
Neispravan rad rashladnih uređaja narušava komfor korisnika koji borave u prostoru. Kako komfor uvelike utječe na produktivnost samih korisnika pravovremena detekcija neispravnih uređaja veoma je bitna. Neboderska zgrada FER-a ima preko 300 ventilokonvektora stoga je aplikacija za detekciju neispravnih ventilokonvektora na temelju mjerenih podataka iz zgrade neophodna za pravovremenu detekciju neispravnih uređaja.

Za izradu aplikacije odabran je programski jezik Java te pripadajuće tehnologije za izradu web aplikacije.

U poglavlju 1. opisana je konfiguracija sustava nad kojim je izrađena ova aplikacija, sustav za detekciju opisan poglavljem 2. objašnjava formule korištene u aplikaciji te način detekcije neispravnog ventilokonvektora, a rad aplikacije i algoritmi korišteni prilikom izrade opisani su u 3. poglavlju. Baza podataka na kojoj se temelji cijeli sustav te korišteni entiteti objašnjeni su u 4. poglavlju. Kako bi se lakše shvatila pozadina rada ovakve aplikacije u poglavlju 5. opisane su tehnologije koje se koriste. Poglavlje 6. prikazuje izgled aplikacije, klase koje aplikacija koristi za prikaz te se uz opis klasa nalaze i slike aplikacije iz preglednika. Na kraju se donose zaključci te opisuje budući rad.

# 1. Opis razmatranog sustava

Razmatrana konfiguracija sustava je neboderska zgrada Sveučilišta u Zagrebu Fakulteta elektrotehnike i računarstva. U nastavku su opisana samo mjerenja potrebna za izradu ove aplikacije. U svakoj prostoriji nalazi se jedan ili više paralelno postavljenih ventilokonvektora. Svi ventilokonvektori opremljeni su dodatnim sensorima za mjerenje temperature povratnog medija. Ventilokonvektori imaju mogućnost rada s četiri različite brzine ventilatora označene redom kao {0,L,M,H}. Dodatno, za svaku prostoriju dostupna su mjerenja temperature prostorije te brzine ventilatora ventilokonvektora. Uz centralni kalorimetar, koji služi za mjerenje ukupnog protoka i temperature medija, svi katni razvodi ventilokonvektora opremljeni su kalorimetrima koji mjere protok i temperaturu medija na pojedinim razvodima. Uz pretpostavku dobro izoliranih katnih razvoda temperatura medija na ulazu u pojedini ventilokonvektor jednaka je temperaturi mjerenoj na kalorimetru koji se nalazi na istom katnom razvodu kao i promatrani ventilokonvektor.



Slika 1. Konfiguracija razmatranog sustava

## 2. Sustav za detekciju neispravnog rada ventilokonvektora

U slučaju stvaranja zračnog čepa unutar izmjenjivača topline ventilokonvektora, uslijed ograničenog strujanja medija kroz izmjenjivač topline ventilokonvektor radi sa smanjenom efikasnošću koja u ekstremnim slučajevima može pasti i na nulu.

Toplinska snaga preuzeta iz vode  $P_w$  izražena je kao:

$$P_w = q_w \cdot C_w \cdot (T_w^{in} - T_w^{out}), \quad (1)$$

pri čemu je  $q_w$  maseni protok medija,  $C_w$  označava specifični toplinski kapacitet medija te za slučaj kada je medij voda iznosi  $4186 \text{ J}/(\text{kg} \cdot \text{K})$  [4]. Temperaturu dovodnog medija u ventilokonvektor  $T_w^{in}$  i temperaturu odlaznog medija iz ventilokonvektora  $T_w^{out,raw}$  očitavamo pomoću senzora temperature na kalorimetru odnosno na odlaznoj cijevi medija na izmjenjivaču topline ventilokonvektora. Nad očitanim podacima potrebno je napraviti kalibraciju prema formuli:

$$T_w^{out} = a \cdot T_w^{out,raw} + b, \quad (2)$$

$T_w^{out}$  označava kalibriranu temperaturu. Za dobivanje granične temperature ispravnog rada ventilokonvektora provjerava se toplinska snaga predana zraku  $P_t$ :

$$P_t = U(x, q_w) \cdot \left( \frac{T_w^{in} + T_w^{out}}{2} - T_a \right), \quad (3)$$

pri čemu je  $T_a$  temperatura prostorije, a  $U(x, q_w)$  koeficijent prijelaza topline izražen kao funkcija brzine ventilatora  $x$  i protoka medija kroz ventilokonvektor  $q_w$ .

U stacionarnom stanju vrijedi:

$$P_t = P_w. \quad (4)$$



Pretpostavlja se uniformna distribucija masenog protoka medija  $q_w$  kroz sve ventilokonvektore u zgradi:

$$q_w = \frac{q_{total}}{N_{fcu}}, \quad (5)$$

gdje  $q_{total}$  označava ukupni maseni volumen cjelokupnog medija u sustavu mjeren pomoću centralnog kalorimetra, a  $N_{fcu}$  označava ukupan broj ventilokonvektora u zgradi. Maksimalna snaga koja se može prenijeti s vode na zrak ovisi o maksimalnoj vrijednosti koeficijenta prijenosa topline u pojedinoj brzini ventilokonvektora  $U_{MAX}^x$  izraženog kao:

$$U_{MAX}^x = \lim_{q_w \rightarrow \infty} U(x, q_w). \quad (6)$$

Uvrštavanjem (1), (3) i (6) u (4) slijedi:

$$q_w \cdot C_w \cdot (T_w^{in} - T_w^{out,LIM}) = U_{MAX}^x \cdot \left( \frac{T_w^{in} + T_w^{out,LIM}}{2} - T_a \right), \quad (7)$$

pri čemu  $T_w^{out,LIM}$  označava graničnu temperaturu pomoću koje se određuje ispravnost ventilokonvektora definiranu kao:

$$T_w^{out,LIM} = \frac{2 \cdot q_w \cdot C_w - U_{MAX}^x}{2 \cdot q_w \cdot C_w + U_{MAX}^x} \cdot T_w^{in} + \frac{2 \cdot U_{MAX}^x \cdot T_a}{2 \cdot q_w \cdot C_w + U_{MAX}^x}. \quad (8)$$

Postoje ukupno četiri brzine ventilatora  $x$  za koje je proračunati maksimalni koeficijent prijenosa topline (6) (Tablica I.). Tako definirani koeficijenti vrijede za Trane ventilokonvektore tip FCC06 [5] koji se koriste u razmatranom sustavu.

Tablica I. Iznosi koeficijenta prijenosa topline za različite brzine ventilatora

Brzina ventilatora $x$	Koeficijent prijenosa topline $U_{MAX}^x$
H	200
M	150
L	100
0	5

Uz pretpostavku maksimalnih performansi izmjenjivača topline, konstantnog ukupnog protoka, uniformne distribucije protoka (5) i srednje temperature zraka u prostorijama jednake  $T_a = 24 \text{ }^\circ\text{C}$  uvrštavanjem vrijednosti iz Tablice I. u jednadžbu (8), slijede 4 jednadžbe za računanje vrijednosti granične temperature izlaznog medija iz ventilokonvektora  $T_w^{\text{out,LIM}}$ :

$$T_w^{\text{out,LIM}} = 0.44 \cdot T_w^{\text{in}} + 13.51 \quad (9)$$

$$T_w^{\text{out,LIM}} = 0.55 \cdot T_w^{\text{in}} + 10.89 \quad (10)$$

$$T_w^{\text{out,LIM}} = 0.67 \cdot T_w^{\text{in}} + 7.86 \quad (11)$$

$$T_w^{\text{out,LIM}} = 0.98 \cdot T_w^{\text{in}} + 0.47 \quad (12)$$

Aplikacija u jednostavnom sučelju dostupnom na web stranici prikazuje neispravne ventilokonvektore koje pronalazi pomoću dva algoritma. Algoritam 1 radi s ventilokonvektorima kod kojih su ventilatori isključeni te uzima fiksnu granicu ispravnog rada isključenog uređaja, dok Algoritam 2 radi i s uključenim ventilatorima ventilokonvektora te, ovisno o brzini ventilatora, izračunava granicu temperature izlaznog medija za ispravan rad ventilokonvektora.

## 3. Opis aplikacije

### 3.1. Povezivanje entiteta u bazi podataka s objektima u Javi

Entiteta *building* u bazi podataka, označava zgradu FER-a na kojoj se želi dohvatiti mjerenje. Iako ovaj dio nije toliko bitan u trenutnom radu aplikacije jer postoji samo jedna zgrada u kojoj se provode mjerenja, napravljen je kako bi se u budućnosti olakšalo dodavanje novih zgrada s ovom tehnologijom. *Calorimeter* i *calorimeterMeasurementHistory* su dva entiteta povezana s katnim kalorimetrima i mjerenjima koje oni bilježe. Za algoritam je bitno mjerenje *supply\_temperature* pomoću kojeg dohvaćamo temperaturu medija kojom se pojedini kat opskrbljuje. Također iz istog entiteta bitno je vrijeme te orijentacija kalorimetra jer se na svakom katu nalaze dva kalorimetra koja opslužuju sjeverni odnosno južni dio zgrade pa se njihova temperatura uspoređuje s prostorijama koje se nalaze na njihovoj strani. Nadalje entitet *zoneMeasurements* jako je važan prilikom prikaza podataka o trenutnim mjerenjima unutar prostorije, dok se pomoću entiteta *zone* dobivaju podaci iz entiteta *fcu* koji povezuju pojedinu prostoriju s ventilokonvektorom ili više njih ovisno koliko ih se nalazi u pojedinoj prostoriji. Nakon što se iz prostorije dohvati ventilokonvektor unutar nje, potrebno je dohvatiti njegova mjerenja iz entiteta *FcuWaterSideMeasurementHistory* te ih usporediti s podacima dobivenim od kalorimetra. Trenutno se u aplikaciji nevaljani rad ventilokonvektora detektira pomoću dva algoritma objašnjena u poglavljima 3.3.1 i 3.3.2.

Na primjeru entiteta *zone* objašnjen je postupak spajanja klase u Javi s entitetom u bazi podataka pomoću Hibernate okvira. Stvara se nova klasa s proizvoljnim imenom, nije bitno da se zove isto kao i u bazi. Iznad klase moramo definirati `@Entity` koji označava da se radi o entitetu te ispod toga deklarirati na koji se entitet u bazi podataka odnosi novostvorena klasa na način `@Table(name="zone")` čime se java klasu povezuje s tablicom *zone* iz baze. Nakon toga potrebno je definirati što se sve nalazi u entitetu. Iznad varijabli proizvoljnog imena, koje se koristi umjesto naziva te varijable u bazi, mora se dodati `@Column(name="zone_id")` te ukoliko se radi o primarnom ključu kao u ovom primjeru još se mora dodati `@Id`. Nakon definiranja svih entiteta koji su bitni, odnosno nabrojani u prvom dijelu, gotovo je povezivanje baze podataka s klasama u Javi [6].

```

@Entity
@Table(name="zone")
public class Zone {

    @Id
    @Column(name="zone_id")
    private int zoneID;

    @Column(name="timestamp")
    private String timestamp;

    @Column(name="floor")
    private int floor;

    @Column(name="name")
    private String name;
}

```

Slika 2. Primjer iz aplikacije

### 3.2. Korištenje stvorenih entiteta za prikaz pojedinog mjerenja unutar prostorije

*Data Access Object* (DAO) model omogućuje razdvajanje podataka iz baze od rada s tim podacima. Aplikacija sadrži implementaciju klasa koje služe za dohvat podataka, obradu te prosljeđivanje kontroleru koji ih onda pomoću JSP tehnologije prikazuje u obliku HTML stranice. Svaka klasa implementira jednostavno sučelje s jednom metodom. Ovaj dio možda je pretjeran no on je dobra programerska praksa jer nam opisuje sve metode koje se nalaze unutar klase koja implementira to sučelje. *BuildingPostDAO* implementira sučelje *BuildingDAO* s metodom *getBuildings*. Metoda pomoću ranije definiranih parametara za spajanje na bazu podataka uspostavlja vezu sa njom. Putem te veze i Hibernate okvira dohvaćamo podatke o zgradama te ih prosljeđujemo kontroleru. Kao što je ranije navedeno ovo trenutno i nije toliko bitno jer se programska podrška testira samo na jednoj zgradi. Sljedeća jednostavna klasa za dohvat broja katova je *FloorPostDAO* koja implementira sučelje *FloorDAO*. Metoda *getFloors* unutar klase vraća broj katova pojedine zgrade. Ove dvije klase s pripadajućim sučeljima omogućuju dohvat zgrade i broj kata kojeg koristimo prilikom odabira pojedine prostorije za prikaz mjerenja unutar njih.

Sljedeće dvije klase opisuju prostorije i dohvat mjerenja unutar njih. *ZoneDAO* je sučelje s metodom *getZones(int floor)* koja prima kat, dohvaća sve zone s tog kata

i vraća ih u obliku liste, a klasa koja implementira to sučelje je *ZonePostDAO*. Za dohvaćanje mjerenja unutar prostorije koristimo *MeasurementsDAOImpl* klasu. Ona također implementira sučelje s jednom metodom *getMeasurements(String zoneName)* pod nazivom *MeasurementsDAO*. Metoda prima ime prostorije koju korisnik odabire pomoću padajućih izbornika kojima odabire zgradu, kat te nakraju ime prostorije na web stranici *building/list*. Ti parametri potrebni su nam za jednoznačno određivanje prostorije. Ukratko kontroler *Zone*, koji prikazuje JSP stranicu *list-buildings* na adresi *building/list*, prosljeđuje prostoriju koju je korisnik odabrao te se onda pomoću *MeasurementsDAOImpl* prvo dohvaća ključ prostorije putem alternativnog ključa *ZoneName*, a nakon toga putem ključa *zoneID* dohvaćaju mjerenja koja se vrše u toj prostoriji iz tablice *ZoneMeasurement* i prosljeđuju nazad kontroleru koji poziva novu stranicu koja pomoću jednostavne tablice prikazuje ID prostorije, vrijeme mjerenja, temperaturu, brzinu ventilatora, status katnog ventila, status 3Smart, status sklopke te željenu temperaturu prikazano Slikom 8 [7].

## 3.3 Opis algoritama

### 3.3.1. Algoritam 1

Prvi algoritam provjerava ispravnost isključenih ventilokonvektora. Ukoliko je ventilator isključen ( $x = 0$ ), gibanje zraka kroz ventilokonvektor će biti slabo i uzrokovano isključivo prirodnim strujanjem toplog zraka prema gore, a hladnog prema dolje, što povlači malu izmjenu energije dovodnog medija unutar ventilokonvektora sa zrakom unutar prostorije. Stoga slijedi:

$$T_w^{in} \sim T_w^{outLIM}. \quad (13)$$

odnosno temperatura ulaznog medija tek neznatno odstupa od temperature izlaznog medija [8]. Ispravnost rada određuje se usporedbom temperature dovodnog medija katnog kalorimetra s temperaturom odlaznog medija iz ventilokonvektora. Ukoliko je apsolutna razlika temperatura manja od fiksne granice od 1°C, ventilokonvektor je ispravan, inače se sprema na listu neispravnih i prikazuje korisniku putem web stranice. Fiksna granica od 1°C određena je tako da

se uključi pad temperature uslijed prirodnog strujanja zraka te eventualne greške koje mogu nastati zbog nesavršenosti mjernog uređaja, odnosno smetnji. U slučaju da je ventilator isključen, a razlika između dovodne i odlazne temperature medija veća od  $1^{\circ}\text{C}$  odnosno manja od  $-1^{\circ}\text{C}$  sugerira se kako je ventilokonvektor napunjen zrakom te da bi ga trebalo odzračiti [9].

Princip rada prvog algoritma, tj. prolazak i provjera svakog ventilokonvektora, objašnjen je pseudokodom, (Algoritam 1). Nakon pokretanja algoritam prolazi sve katove zgrade i vraća listu neispravnih ventilokonvektora te prekida s radom. Na svakom katu postoje dva kalorimetra, sjeverni i južni. Algoritam na svakom katu najprije dohvaća mjerenja sa sjevernog katnog kalorimetra te uspoređuje njegov *pipeline\_id* s *pipeline\_id*-om prostorija na istom katu. Ako se id cjevovoda prostorije i katnog kalorimetra poklapaju, algoritam kreće s detekcijom. U slučaju da id-ovi nisu isti, prostorija se preskače te se ispravnost ventilokonvektora u tim prostorijama provjerava kod učitavanja južnog kalorimetra. Nakon provjere položaja, provjerava se brzina ventilatora, ako je ventilator ventilokonvektora uključen, algoritam se prekida te se učitava nova prostorija. Iako postoji više ventilokonvektora unutar prostorije, nije potrebno provjeravati brzinu ventilatora svakog ventilokonvektora jer je brzina svih ventilatora u istoj prostoriji sinkronizirana. Kad je pronađen ventilokonvektor u stacionarnom stanju – zbog svoje kompleksnosti pronalazak stacionarnog stanja objašnjen je u poglavlju 3.3.3 – uspoređuje se temperatura na odlazu iz ventilokonvektora s temperaturom dovodnog medija s katnog kalorimetra u trenutku stacionarnog stanja. Razlika veća od  $1^{\circ}\text{C}$  znači neispravan rad te se važni parametri za identifikaciju ventilokonvektora spremaju u listu i nakon završetka prikazuju na web stranici.

Algoritam 1. Pseudokod za detekciju neispravnog rada isključenog ventilokonvektora

1:	učitaj ukupan broj katova zgrade $N_k$
2:	$j = 1$
3:	<b>sve dok je:</b> $j \leq N_k$
4:	učitaj ukupan broj kalorimetara na $j$ -tom katu $N_c$
5:	$k = 1$
6:	<b>sve dok je:</b> $k \leq N_c$
7:	učitaj protok medija kroz $k$ -ti kalorimetar $V_c$
8:	<b>ako je:</b> $V_c > 0$
9:	učitaj temperaturu dovodnog medija kalorimetra $T_w^{cal}$
10:	učitaj pripadni id cjevovoda $P_{id}^c$
11:	učitaj broj prostorija na katu $N_p$
12:	$l = 1$
13:	<b>sve dok je:</b> $l \leq N_p$

14:	učitaj pripadni id cjevovoda prostorije $P_{id}^p$
15:	<b>ako je:</b> $P_{id}^p = P_{id}^c$
16:	učitaj trenutnu brzinu rada ventilokonvektora $x$
17:	<b>ako je:</b> $x = 0$
18:	pronađi 3 stacionarna stanja $N_t$
19:	učitaj ukupan broj ventilokonvektora u prostoriji $N_{vk}$
20:	$m = 1$
21:	<b>sve dok je:</b> $m \leq N_{vk}$
22:	$n = 1$
23:	<b>sve dok je:</b> $n \leq N_{stac}$
24:	-učitaj temperaturu povratnog medija ventilokonvektora u
25:	stacionarnom stanju $T_w^{out,raw}$
26:	kalibriraj temperaturu $T_w^{out} = a \cdot T_w^{out,raw} + b$
27:	-odredi razliku temperature povrata i polaza $\Delta T_w = T_w^{cal} - T_w^{out}$
28:	<b>ako je:</b> $ \Delta T_w  \geq 1$
29:	spremi ventilokonvektor na listu neispravnih
30:	<b>kraj</b>
31:	$n = n + 1$
32:	<b>kraj</b>
33:	$m = m + 1$
34:	<b>kraj</b>
35:	<b>kraj</b>
36:	<b>kraj</b>
37:	$l = l + 1$
38:	<b>kraj</b>
39:	<b>kraj</b>
40:	$k = k + 1$
41:	<b>kraj</b>
42:	$j = j + 1$
43:	<b>kraj</b>
44:	<b>kraj</b>

### 3.3.2. Algoritam 2

Drugi algoritam provjerava ispravnost ventilokonvektora usporedbom mjerene povratne temperature ventilokonvektora u stacionarnom stanju s izračunatom graničnom temperaturom  $T_w^{outMAX}$  (9) – (11). Ukoliko je temperatura prostorije  $T_a$  manja od temperature dovodnog medija, a razlika veća od  $1^\circ\text{C}$ , ventilokonvektor je neispravan. U drugoj situaciji, kada je temperatura prostorije veća od temperature dovodnog medija (režim hlađenja), razlika mora biti manja od  $-1^\circ\text{C}$  kako bi mogli prepoznati neispravan ventilokonvektor. Navedeni algoritam može raditi sa svim brzinama ventilatora. Princip rada za slučaj drugog algoritma dan je u pseudokodu (Algoritam 2).

Algoritam 2. Pseudokod za detekciju neispravnog rada ventilokonvektora pomoću granične temperature

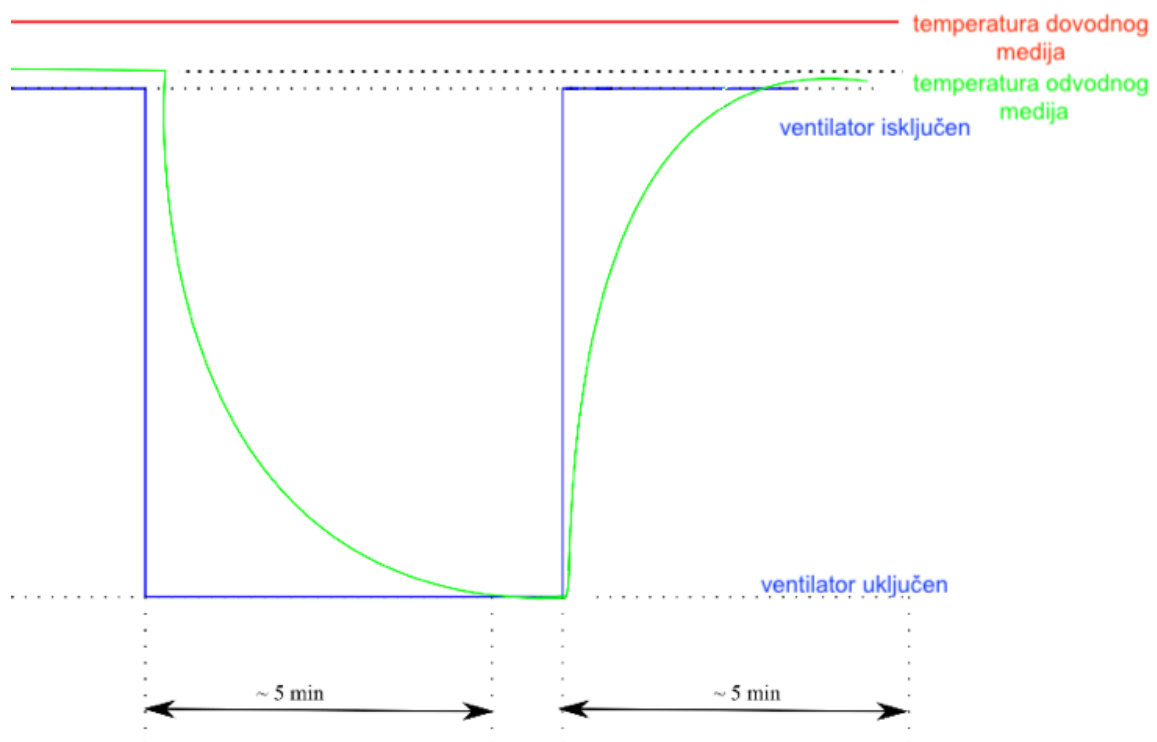
1:	učitaj ukupan broj katova zgrade $N_k$
2:	$k = 1$
3:	<b>sve dok je:</b> $k \leq N_k$
4:	učitaj ukupan broj kalorimetara na $k$ -tom katu $N_c$
5:	$l = 1$
6:	<b>sve dok je:</b> $l \leq N_c$
7:	učitaj protok medija kroz $l$ -ti kalorimetar $V_c$
8:	<b>ako je:</b> $V_c > 0$
9:	učitaj temperaturu dovodnog medija kalorimetra $T_w^{cal}$
10:	učitaj pripadni id cjevovoda kalorimetra $P_{id}^c$
11:	učitaj broj prostorija na katu $N_p$
12:	$m = 1$
13:	<b>sve dok je:</b> $m \leq N_p$
14:	učitaj pripadni id cjevovoda prostorije $P_{id}^p$
15:	<b>ako je:</b> $P_{id}^p = P_{id}^c$
16:	učitaj trenutnu temperaturu prostorije $T_a$
17:	učitaj ukupan broj ventilokonvektora u prostoriji $N_{vk}$
18:	$n = 1$
19:	<b>sve dok je:</b> $n \leq N_{vk}$
20:	$o = 1$
21:	<b>sve dok je:</b> $o \leq N_{stac}$
22:	učitaj temperaturu povratnog medija ventilokonvektora
23:	$T_w^{out,raw}$ u stacionarnom stanju
24:	kalibriraj temperaturu $T_w^{out,raw}$ (2)
25:	izračunaj graničnu temperaturu povratnog medija $T_w^{out,LIM}$
26:	(8)
27:	odredi razliku $\Delta T_w = T_w^{out,LIM} - T_w^{out}$
28:	<b>ako je:</b> $T_w^{cal} \geq T_a$
29:	<b>ako je:</b> $\Delta T_w > 1$
30:	spremi ventilokonvektor na listu neispravnih
31:	<b>kraj</b>
32:	<b>kraj</b>
33:	<b>ako je:</b> $T_w^{cal} < T_a$
34:	<b>ako je:</b> $\Delta T_w < -1$
35:	spremi ventilokonvektor na listu neispravnih
36:	<b>kraj</b>
37:	<b>kraj</b>
38:	<b>kraj</b>
39:	$o = o + 1$
40:	<b>kraj</b>
41:	$n = n + 1$
42:	<b>kraj</b>
43:	<b>kraj</b>
44:	$m = m + 1$
45:	<b>kraj</b>
46:	<b>kraj</b>
47:	<b>kraj</b>



48:	$l = l + 1$
49:	<b>kraj</b>
50:	$k = k + 1$
51:	<b>kraj</b>

### 3.3.3 Pronalazak stacionarnog stanja

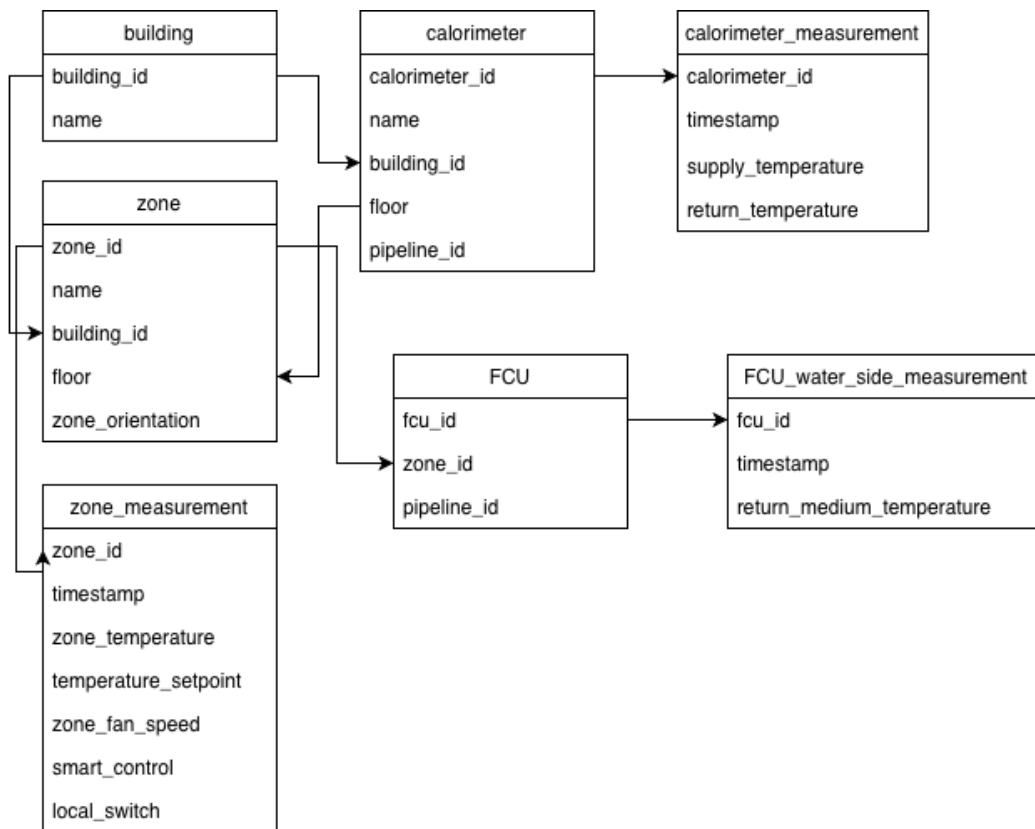
Stacionarno stanje na ventilokonvektoru nastupa približno 5 minuta nakon promjene stanja ventilatora Slika 3. Prilikom izračuna stacionarnog stanja koriste se povijesna mjerenja. Kako bi bili sigurni da je ventilokonvektor uistinu neispravan algoritam moramo provesti nad tri različita stacionarna stanja.



Slika 3. Primjer stacionarnog stanja

## 4. Opis baze podataka

Od cijele baze podataka objašnjeni su samo oni entiteti koji se koriste u izradi aplikacije. Navedeni su opisi pojedinih varijabli, a njihov tip podatka i vrsta ključa dati su u tablicama [10].



Slika 4. Entiteti i njihova međusobna povezanost

### 4.1. Entitet zgrade

Sastoji se od primarnog ključa *building\_id*, zapisa vremena dodavanja zgrade u bazu podataka *timestamp* i naziva zgrade *name*.

Tablica II. Entitet *building*

Naziv	Tip podatka	Tip ključa	Primjer
<i>building_id</i>	Integer	Primarni ključ	1
Name	Varchar(30)	Alternativni ključ	C

## 4.2. Entitet kalorimeter

Pomoću entiteta *calorimeter* definiranog u Tablici III. i *calorimeter\_measurement\_history* definiranog u Tablici IV., određuje se veza između pojedinih ventilokonvektora i katnih kalorimetara. Ukoliko su ventilokonvektori i kalorimeter povezani nalaze se na istom cjevovodu (*pipeline\_id*). Osim *id*-a cjevovoda, za povezivanje entiteta *calorimeters* s entitetima *zone* i *FCU* (*ventilokonvektor*, eng. *fan coil unit*) potreban nam je kat, strani ključ *calorimeter\_id* te vrijeme stacionarnog stanja kako bi u entitetu *calorimeter\_measurements\_history*, mogli jednoznačno pronaći mjerenje temperature dolaznog medija ventilokonvektora.

Tablica III. Entitet *calorimeter*

Naziv	Tip podatka	Tip ključa	Primjer
<i>calorimeter_id</i>	Integer	Primarni ključ	1
<i>name</i>	Varchar(30)	Alternativni ključ	C00N
<i>building_id</i>	Integer	Strani ključ	1
<i>floor</i>	Integer		12
<i>pipeline_id</i>	Integer		1

Tablica IV. Entitet *calorimeter\_measurements\_history*

Naziv	Tip podatka	Tip ključa	Primjer
<i>id</i>	Long	Primarni	9899830
<i>calorimeter_id</i>	Integer		1
<i>timestamp</i>	Timestamp		2018-03-20 17:29:18.3
<i>supply_temperature</i>	Real		8.8
<i>return_temperature</i>	Real		9.3

## 4.3. Entitet ventilokonvektor

Za povezivanje katnog kalorimetra i ventilokonvektora koristimo *pipeline\_id* koji se nalazi u *FCU* entitetu (Tablica V). Nakon provjere cjevovoda, parametri važni za jednoznačno određivanje ventilokonvektora su *zone\_id* za određivanje prostorije u kojoj se nalazi ventilokonvektor te *fcu\_id* za određivanje ventilokonvektora unutar prostorije. Nakon pronalaska *id*-a ventilokonvektora koji pripada zadanoj prostoriji mjerenje temperature povratnog medija pronalazi se unutar entiteta

FCU\_water\_side\_measurement\_history (Tablica VI). Kalibrirano mjerenje temperature računa se pomoću formule (2), a koeficijente  $a$ , koji odgovara  $fcu\_sensor\_slope$ , te  $b$  koji odgovara retku  $fcu\_sensor\_offset$  u entitetu *fcu\_sensors\_calibration\_curve* Tablica VII., ukoliko koeficijenti ne postoje u tablici su označeni s *null* vrijednošću te se pretpostavlja da mjerenja s tog senzora nije potrebno dodatno korigirati.

Tablica V. Entitet *fcu*

	Tip podatka	Tip ključa	Primjer
<i>fcu_id</i>	Integer	Primarni	1
<i>zone_id</i>	Integer	Strani	5
<i>pipeline_id</i>	Integer		1

Tablica VI. Entitet *fcu\_water\_side\_measurements\_history*

Naziv	Tip podatka	Tip ključa	Primjer
<i>id</i>	Long	Primarni	12455293
<i>fcu_id</i>	Integer		1
<i>timestamp</i>	Timestamp		2018-03-20 17:24:11
<i>return_medium_temperature</i>	Real		22.4

Tablica VII. Entitet *fcu\_sensors\_calibration\_curve*

Naziv	Tip podatka	Tip ključa	Primjer
<i>fcu_id</i>	Long	Primarni	1
<i>fcu_sensor_slope</i>	Real		0.98
<i>fcu_sensor_offset</i>	Real		1.07

## 4.4. Entiteti prostorije

Za prikaz prostorije s neispravnim ventilokonvektorom koristimo varijablu *name* iz entiteta *zone*. Osim prikaza imena prostorije ovaj entitet (Tablica VIII.) koristi se i za određivanje prostorije u kojoj prikazujemo važne parametre poput trenutne mjerene temperature prostorije, željene temperature prostorije definirane od strane korisnika preko zidne jedinice ili brzine ventilatora ventilokonvektora, spremljene u entitetu *zone\_measurement\_history* (Tablica IX).

Tablica VIII. Entitet zone

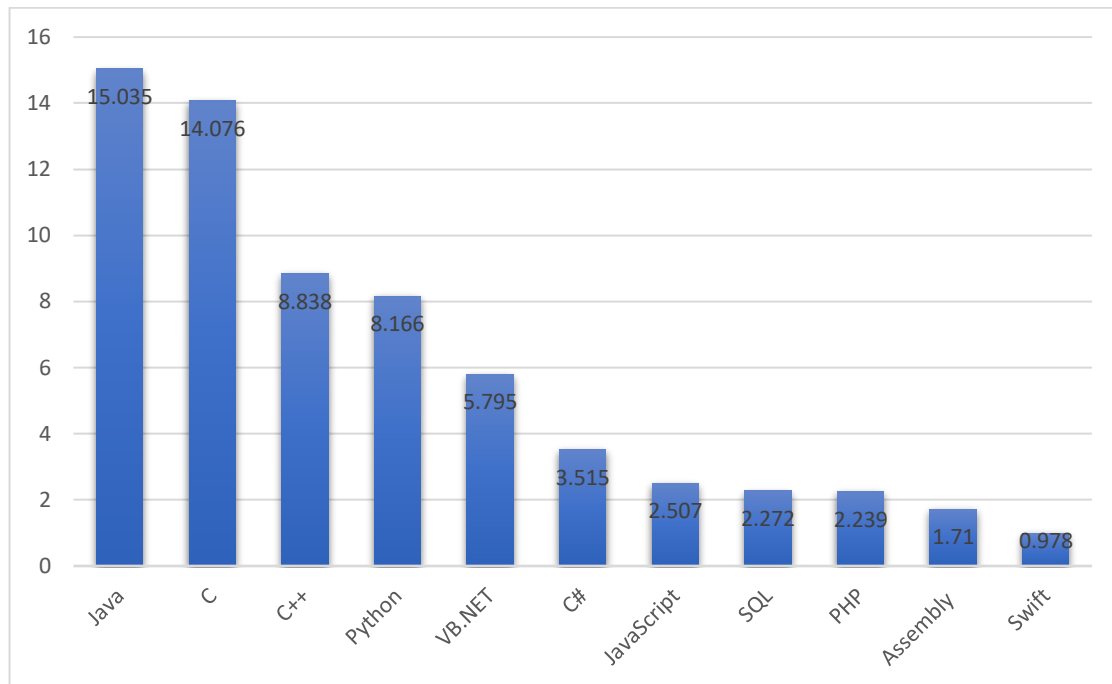
Naziv	Tip podatka	Tip ključa	Primjer
zone_id	Integer	Primarni	1
name	Varchar(30)	Alternativni	C01-10
building_id	Integer	Strani	1
floor	Real		10
zone_orientation	Smallint		0

Tablica IX. Entitet zone\_measurement\_history

Naziv	Tip podatka	Tip ključa	Primjer
zone_id	Integer	Primarni/strani	1
timestamp	Timestamp		2018-03-20 17:29:18.36
zone_temperature	Real		22.3
zone_fan_speed	Real		100
smart_control	Boolean		True
local_switch	Real		0.5
temperature_setpoint	Real		35.0

## 5. Korištene tehnologije

Jedan od glavnih zahtjeva aplikacije je mogućnost pristupa putem računala i pametnog mobitela pa se samim time nameće web aplikacija kako bi se izbjeglo pisanje posebne aplikacije za različite platforme. Prilikom odabira programskog jezika glavna vodilja je bila popularnost jezika. To je dovelo do odabira Jave jer prema Tiobe istraživanjima već nekoliko godina zauzima prvo mjesto [11].



Slika 5. Zastupljenost programskih jezika prema udjelu u tržištu

### 5.1. Java

Programski jezik Java nastao je iz programskog jezika Oak. To je objektno orijentirani programski jezik jer se prilikom razvoja programske opreme dijelovi aplikacije grupiraju u objekte koji su često jednostavnog dizajna, ali kada ih se međusobno poveže tvore kompleksnu cjelinu.

Nakon što je 1995. u internet preglednik Netscape Navigator ugrađena podrška za Javu dolazi do njegovog daljnjeg razvoja i širenja i opće prihvaćenosti. Osnovna ideja u razvoju bila je "napiši jednom, pokreni bilo gdje." Nakon što je program kodiran u Javi preveden u strojni kôd, može se pokrenuti na bilo kojoj platformi sa Java virtualnim strojem, bez obzira na operacijski sustav ili sklopovsku platformu.

To je dovelo do potpore za više-platformnost što programeru nudi apstraktni pogled na računalo. Programski jezik Java definira apstraktni računski strojni kôd kod kojeg je sve propisano i opisano specifikacijom odnosno ne daje se platformski specifične detalje npr. širine podatkovne riječi centralnog procesora, direktan pristup upravljačkim programima operacijskog sustava ili podržanom sklopovlju. Programeru nije bitno da li će se program izvoditi na 16-bitnom, 32-bitnom ili 64-bitnom procesoru.

Prilikom programiranja u Javi programi se pišu za Javin virtualni stroj koji je definiran specifikacijom i na kojem se izvode svi programi pisani u Javi. U specifikaciji definira se način pisanja literala, ključne riječi jezika, njihova semantika te način pisanja valjanog kôda. "Valjani kôd" podrazumijeva sintaksno i semantički ispravni kôd sa stajališta jezičnog prevodioca koji izvorni Java kôd treba obraditi i stvoriti izvršni kôd za Javin virtualni stroj.

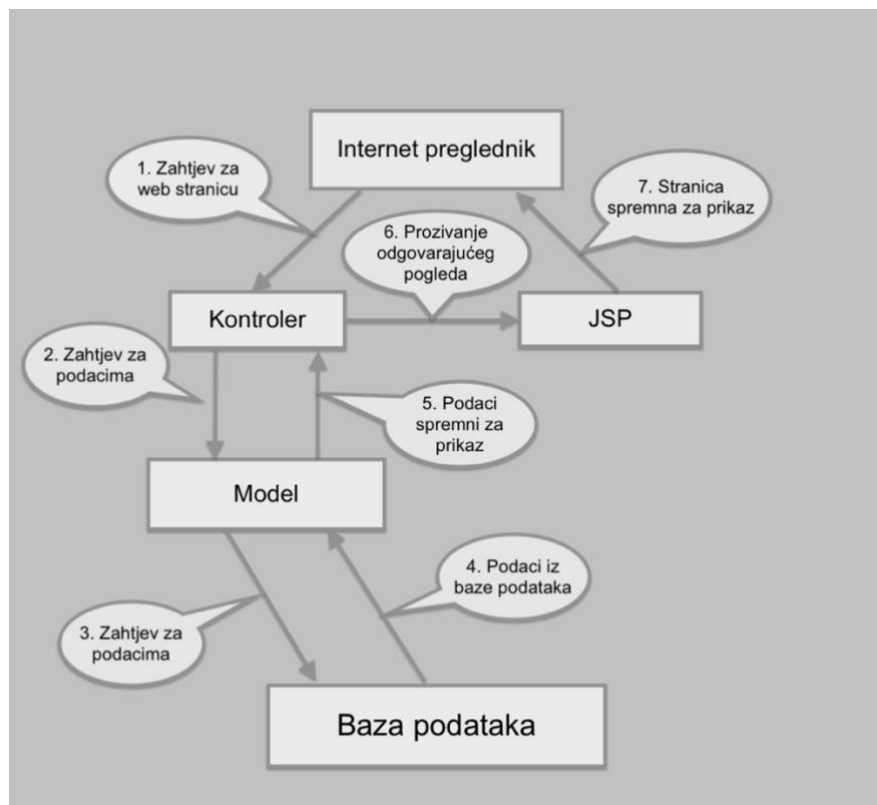
Java ima najširi spektar mogućnosti i definitivno je programski jezik koji je programerima broj jedan. Jedna od bitnih značajki jezika je „neovisnost o platformi“ što omogućava da se kôd napisan u Javi izvodi bez unošenja promjena u različitim okruženjima na računalima. To dovodi i do njenog osnovnog nedostatka jer je sporija od drugih konkurentskih jezika i tehnologija [12].

## **5.2. Model-View-Controller (MVC)**

MVC predstavlja obrazac prema kojem se odvajaju pojedini dijelovi aplikacije u komponente ovisno o njihovoj namjeni. Sastoji se od tri dijela modela koji sadrži podatke i poslovnu logiku odnosno što sve možemo raditi s podacima, pogleda koji predstavlja korisničko sučelje te upravitelja koji prihvaća ulazne podatke i uspostavlja komunikaciju između modela odnosno pogleda. Ovaj obrazac je koristan jer smanjuje međuzavisnost slojeva aplikacije i odvaja poslovnu logiku od korisničkog sučelja, tj. nama olakšava snalaženje u aplikaciji tako što odvajamo klase prema namjeni odnosno sve modele spremamo u direktorij za modele, poglede u direktorij za poglede i kontrolere u direktorij za kontrolere [13].

## 5.3. Java Server Pages (JSP)

Budući da je za poslovnu logiku izabran programski jezik Java, za prikaz korisničkog sučelja odnosno *view* komponentu ovog obrasca odabrana je Java Server Pages tehnologija (JSP). Java Server Pages ne miješa Java kôd i HTML te sadrži naredbe za dohvat podataka od upravljača [14].



Slika 6. Primjer komunikacije MVC modela

Primjer komunikacije prilikom prikaza mjerenja odabrane zone

- a) pogled zadužen za prikaz stranice za odabir zone prima podatke u obliku forme te ih prosljeđuje kontroleru
- b) kontroler prima podatke iz pogleda te ih šalje modelu
- c) model izvršava upit nad bazom, obrađuje podatke te ih vraća kontroleru
- d) kontroler prosljeđuje podatke prema pogledu koji je zadužen za prikaz podataka o mjerenjima



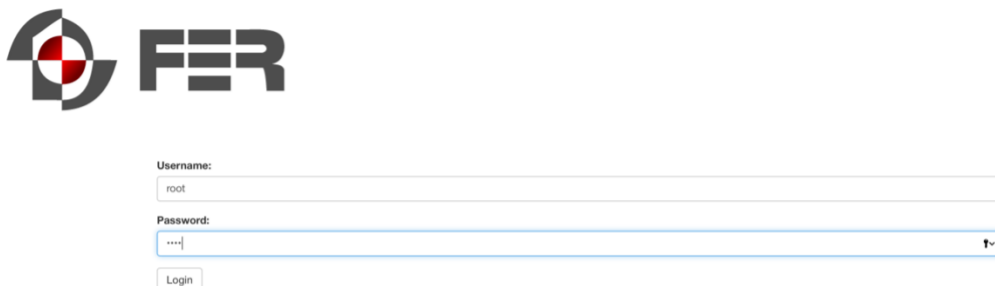
## 5.4. Hibernate

Hibernate je okvir za mapiranje objekata u relacije baze podataka Object-relational-mapping (ORM). Korištenjem hibernate tehnologije stvara se neovisnost o vrsti baze podataka, a Hibernate Query Language (HQL), čija je sintaksa inspirirana SQL jezikom, nam omogućava upite nad Hibernate objektima. [15] Hibernate radi na principu mapiranja java objekata u entitete baze podataka nad kojima se onda obavljaju zadani upiti.

Kako bi se što manje opteretilo bazu podataka potrebni podaci ne učitavaju se lokalno nego se oni dohvaćaju dinamički kroz rad algoritama. Uzme li se u obzir da trenutno samo za podatke o mjerenjima kalorimetara postoji preko 100 milijuna zapisa, odnosno dnevno oko 370 tisuća ovakav način je znatno efikasniji. Dio povijesnih mjerenja sprema se lokalno, također se spremaju svi podaci za neispravne ventilokonvektore.

## 6. Kontroleri i stranice za prikaz

Aplikacijom upravlja šest kontrolera. Polovina ih služi za kretanje između stranica odnosno za implementaciju jednostavnog sučelja za prijavu. Klase *Login* i *Logout* služe upravo za to, a rade na principu da u obliku jednostavnog tekstualnog ulaza na početnoj stranici čitaju podatke pomoću kojih se prijavljujemo u aplikaciju i uspoređuje ih s podacima unutar klase *LoginData* u kojoj se nalaze korisničko ime „root“ i lozinka „root“, trenutno zbog jednostavnosti aplikacije a i samog mehanizma za prijavu podatci nisu enkriptirani.



The image shows a login form for FER. At the top left is the FER logo, which consists of a stylized 'F' and 'E' in a circle next to the letters 'FER'. Below the logo is a form with two input fields. The first field is labeled 'Username:' and contains the text 'root'. The second field is labeled 'Password:' and contains the text '....'. Below the password field is a button labeled 'Login'.

Slika 7. Stranica za prijavu

Za prikaz stranice prikazane na slici iznad koristimo *index.jsp* koji se sastoji od dva dijela. Jedan predstavlja zaglavlje[16], slika 7. prikazano plavom bojom, i koristi se u svim dijelovima aplikacije, a sadrži logo FER-a te još nekoliko poveznica ovisno na kojoj se stranici nalazimo. Drugi dio sadrži formu koja je odgovorna za prikupljanje podataka za prijavu.

Ukoliko smo se uspješno ulogirali u aplikaciju ona nas šalje na početnu stranicu [17].



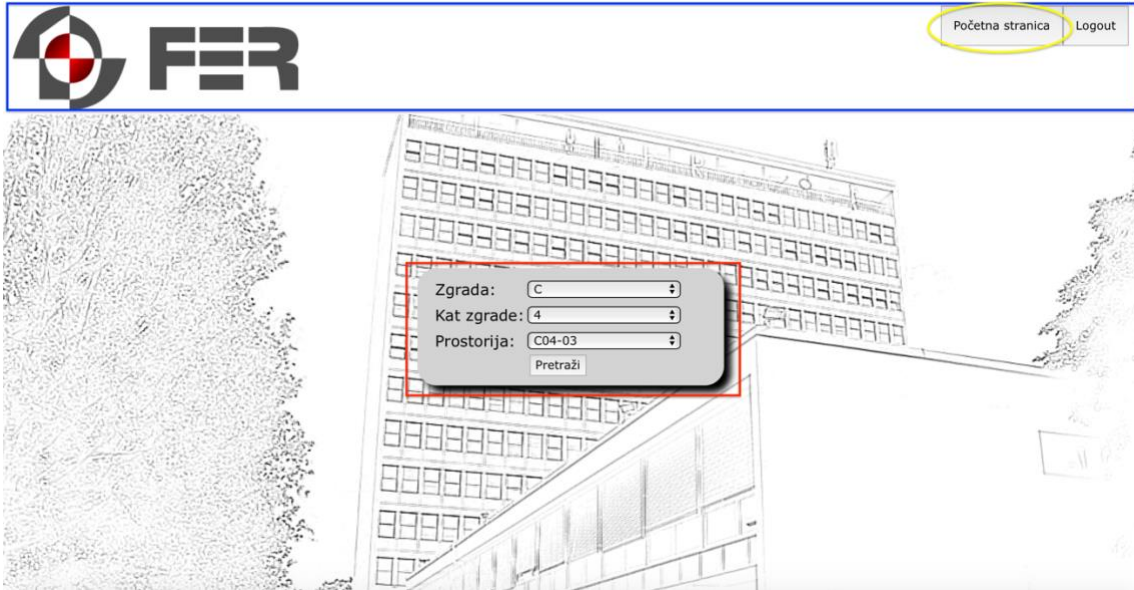
Slika 8. Početna stranica

Početna stranica sastoji se također od zaglavlja. Stranica je podijeljena na 3 jednaka djela sa slikama koje su ujedno i linkovi na stranice [18][19]. Ispod svake slike se nalazi kratki opis što se prikazuje nakon pritiska na nj.

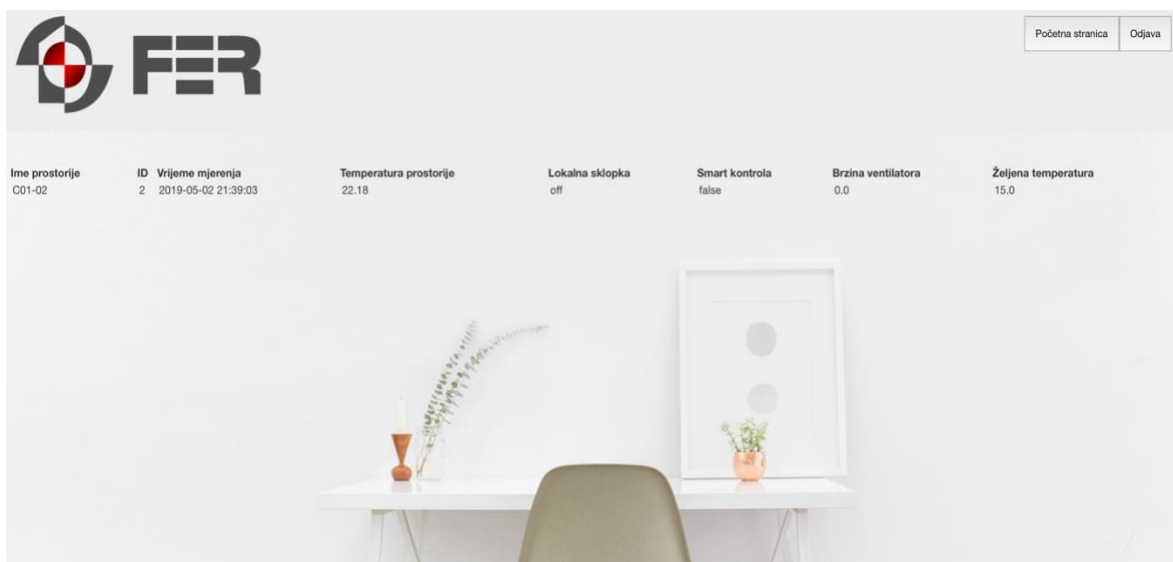
*Logout* kontroler jednostavno je izveden te samo zatvara otvorenu sjednicu i preusmjerava nas na stranicu za prijavu.

Kontroler *Main* služi za povratak na početnu stranicu ukoliko smo uspješno prijavljeni. Poveznicu na njega možemo vidjeti na slici 7. označeno žutom bojom.

Kontroler koji se odnosi na prikaz podataka o prostorijama zove se *Zone*. On upravlja s dvije stranice. Adresa prve stranice je „/building/list“ pomoću koje se korisniku prikazuje izbornik [20] za odabir zgrade, kata te prostorije, Slika 9. označeno crvenom bojom. Aplikacija dohvaća podatke o odabranoj prostoriji, a izgled web stranice s prikazom podataka možemo vidjeti na slici 10.



Slika 9. Stranica za odabir prostorije



Slika 10. Stranica za prikaz mjerenja unutar prostorije

## 6.1. Prikaz i opis dobivenih podataka

Nakon odabira prikaza neispravnih ventilokonvektora na početnoj stranici koriste se dva kontrolera. Za prikaz prvog algoritma koristimo sučelje *DefectiveFcu* koji pomoću DAO klase *DefectiveFcuImpl* dohvaća neispravne ventilokonvektore te ih prikazuje na stranici u obliku tablice, prikazano na slici 11. Navedenu tablicu moguće je preuzeti u PDF obliku za lakši ispis slika 13. Drugi kontroler pod nazivom *FindDefectiveFcu* također zahtjeva listu neispravnih ventilokonvektora pomoću DAO klase *DefectiveFcuSecondImpl* modela te ih prikazuje na stranici.

Svi su ventilokonvektori u zgradi trenutno ispravni, dakle nema zračnih čepova i neispravnih senzora. Za potrebe testiranja rada aplikacije, algoritam je promijenjen tako da je smanjena granica temperature prirodnih pojava koje nastaju uslijed protoka zraka unutar prostorije te su na slici 10. prikazani ventilokonvektori kod kojih su takve pojave izraženije, no to ne znači da je on neispravan. Slika 9. pokazuje trenutno stanje svi ventilokonvektori su ispravni. U prvom stupcu tablice ispisuje se ime prostorije, drugi stupac označava njezin identifikacijski broj ,treći stupac prikazuje brzinu ventilatora {0,L,M,H}, sljedeći stupac točno označava ventilokonvektor pomoću njegovog identifikacijskog broja, sljedeći stupac je razlika temperature dovodnog medija katnog kalorimetra i temperature povratnog medija s ventilokonvektora. Stupac *Zona Aktivna* sadrži vrijednost *true* ili *false*, što označava spremaju li se podatci o toj prostoriji u bazu podataka. Kalibracija je stupac koji pomoću „\*“ označava obrađuju li se kalibrirana mjerenja ili podaci o temperaturi dohvaćeni sa senzora bez dodatnih korekcija.



Početna stranica Odjava

\* - nema podataka o kalibraciji senzora

Ukupan broj neispravnih ventilokonvektora - 0

PDF

Ime zone	ID zone	Brzina ventilatora	ID ventilokonvektora	Razlika temperature supply kalorimetar i return s ventilokonvektora	Datum	Vrijeme	Zona Aktivna	Kalibracija
----------	---------	--------------------	----------------------	---	-------	---------	--------------	-------------

Slika 11. Prikaz liste ventilokonvektora

Slika 12 prikazuje modificirani algoritam s popisom „neispravnih“ ventilokonvektora. Tamno plavom bojom označena je jedna prostorija (C10-01) unutar koje je algoritam pronašao odstupanja od idealnih. Zelenom bojom označena su mjerenja prvog (id = 258) ventilokonvektora unutar prostorije, a crvenom bojom drugog (id = 259). Za svaki ventilokonvektor prikazana su tri mjerenja prikazujući tako pronalazak stacionarnog stanja, odnosno pet minuta ventilator radi istom brzinom, a kako bi bili sigurni da nije došlo da greške u sustavu tražimo tri stacionarna stanja.

\* - nema podataka o kalibraciji senzora

Ukupan broj neispravnih ventilokonvektora - 6

PDF

Ime zone	ID zone	Brzina ventilatora	ID ventilokonvektora	Razlika temperature supply kalorimetar i return s ventilokonvektora	Datum	Vrijeme	Zona Aktivna	Kalibracija
C10-02	178	0	258	1.52	2019-05-02	10:05:02	true	*
C10-02	178	0	259	1.15	2019-05-02	10:05:02	true	*
C10-02	178	0	258	1.52	2019-05-02	10:00:02	true	*
C10-02	178	0	259	1.15	2019-05-02	10:00:02	true	*
C10-02	178	0	258	1.58	2019-05-02	09:55:02	true	*
C10-02	178	0	259	1.21	2019-05-02	09:55:02	true	*
C10-03	179	0	260	1.4	2019-05-02	10:05:02	true	*
C10-03	179	0	261	1.02	2019-05-02	10:05:02	true	*
C10-03	179	0	260	1.4	2019-05-02	10:00:02	true	*
C10-03	179	0	261	1.08	2019-05-02	10:00:02	true	*
C10-03	179	0	260	1.46	2019-05-02	09:55:02	true	*

Slika 12. Prikaz liste ventilokonvektora

Ime zone	ID zone	Brzina ventilatora	ID ventilokonvektora	Razlika temperature supply i return s ventilokonvektora	Datum	Vrijeme	Zona Aktivna	Kalibracija
C10-02	178	0	258	1.52	2019-05-02	10:05:02	true	*
C10-02	178	0	259	1.15	2019-05-02	10:05:02	true	*
C10-02	178	0	258	1.52	2019-05-02	10:00:02	true	*
C10-02	178	0	259	1.15	2019-05-02	10:00:02	true	*
C10-02	178	0	258	1.58	2019-05-02	09:55:02	true	*
C10-02	178	0	259	1.21	2019-05-02	09:55:02	true	*
C10-03	179	0	260	1.4	2019-05-02	10:05:02	true	*
C10-03	179	0	261	1.02	2019-05-02	10:05:02	true	*

Slika 13. Prikaz liste u obliku pogodnom za ispis

## 7. Zaključak

Uz potporu projekta 3Smart te korištenjem sustava prikupljanja podataka iz zgrade koji je izrađen nad neboderskom zgradom Sveučilišta u Zagrebu Fakulteta elektrotehnike i računarstva razvijena je aplikacija sa sučeljem za odabir prostorije te prikazom važnih parametara poput trenutne temperature prostorije, željene temperature u prostoriji ili brzine ventilatora ventilokonvektora u pojedinoj prostoriji. Aplikacija sadrži dva algoritma za detekciju neispravnog rada ventilokonvektora. Neispravan rad ventilokonvektora znači začepljenje cijevi izmjenjivača topline zrakom ventilokonvektora na strani vode. Ova aplikacija znatno olakšava pronalazak neispravnih ventilokonvektora, a samim time i njihov brži popravak. S obzirom na robusnost sustava detekcije i dostupnost varijabli koje se mjeri u zgradi, aplikacija se može proširiti različitim dodatnim informacijama poput prikaza potrošnje energije unutar prostorije ili informacijama o vremenskim uvjetima u području zgrade, ali i drugim funkcionalnostima kao što su daljinsko zadavanje brzine ventilatora ili postavljanje željene temperature.



# LITERATURA

- [1] – Renewable Energy Policy Network for the 21st Century (REN21). Renewables 2014, Global status report, 2014.
- [2] – Europe's buildings under the microscope, Buildings Performance Institute Europe, 2011.
- [3] – M. Price, „Fan coil units“, CIBSE, 2008
- [4] – D. Horvat, „Fizika 1: Mehanika i toplina“, Hinus, 2002
- [5] <https://www.trane.com/download/equipmentpdfs/untprc001en.pdf>, 15. srpnja. 2018.
- [6] – <https://www.udemy.com/spring-hibernate-tutorial/learn/v4/content>, 17. travnja 2019
- [7] – [https://www.tutorialspoint.com/hibernate/hibernate\\_query\\_language.htm](https://www.tutorialspoint.com/hibernate/hibernate_query_language.htm) 22. travnja 2019.
- [8] A. Martinčević, F. Rukavina, V. Lešić, M. Vašak, „Comfort Control in Buildings with Adherence to the Required Thermal Energy Input in Zones“, Proceedings of the 2017 IEEE International Symposium on Industrial Electronics, pp. 1477-1482. Edinburgh, Scotland, United Kingdom, 2017.
- [9] – S Pourarian, J. Wen, D. Veronica, A. Pertzborn, X. Zhou, R. Liu, „A tool for evaluating fault detection and diagnostic methods for fan coil units“, United States, 2016.
- [9] – R. Manger, „Baze podataka“, Element, 2012
- [11] – <https://www.tiobe.com/tiobe-index/> 27. travnja 2019.
- [12] – Čupić M. *Programiranje u Javi*, Zagreb, 2012.
- [13] – <http://www.ucim-programiranje.com/2013/02/mvc-model-view-controller/>, 5. kolovoza 2018.
- [14] – Predavanja iz predmeta Otvoreno računarstvo
- [15] – <http://hibernate.org>, 5. kolovoza 2018
- [16] – <https://jdbc.postgresql.org/documentation/head/connect.html>, 5. kolovoza. 2018
- [17] – [https://www.w3schools.com/bootstrap/bootstrap\\_examples.asp](https://www.w3schools.com/bootstrap/bootstrap_examples.asp), 20. listopada 2018
- [18] – <https://www.w3schools.com/css/default.asp>, 10. prosinca 2018

- [19] – <https://www.w3schools.com/html/default.asp>, 10. prosinca 2018
- [20] – [https://www.w3schools.com/js/js\\_ajax\\_intro.asp](https://www.w3schools.com/js/js_ajax_intro.asp), 21. svibnja 2018

# Popis slika

Slika 1. Konfiguracija razmatranog sustava .....	3
Slika 2. Primjer iz aplikacije.....	8
Slika 3. Primjer stacionarnog stanja.....	13
Slika 4. Entiteti i njihova međusobna povezanost .....	14
Slika 5. Zastupljenost programskih jezika prema udjelu u tržištu .....	18
Slika 6. Primjer komunikacije MVC modela .....	20
Slika 7. Stranica za prijavu.....	22
Slika 8. Početna stranica.....	23
Slika 9. Stranica za odabir prostorije .....	24
Slika 10. Stranica za prikaz mjerenja unutar prostorije .....	24
Slika 11. Prikaz liste ventilokonvektora.....	25
Slika 12. Prikaz liste ventilokonvektora.....	26
Slika 13. Prikaz liste u obliku pogodnom za ispis .....	27

# Popis tablica

Tablica I. Iznosi koeficijenta prijenosa topline za različite brzine ventilatora .....	6
Tablica II. Entitet building .....	14
Tablica III. Entitet calorimeter.....	15
Tablica IV. Entitet calorimeter_measurements_history .....	15
Tablica V. Entitet fcu.....	16
Tablica VI. Entitet fcu_water_side_measurements_history .....	16
Tablica VII. Entitet fcu_sensors_calibration_curve .....	16
Tablica VIII. Entitet zone .....	17
Tablica IX. Entitet zone_measurement_history .....	17

# Sažetak

## Dijagnostika ventilokonvektora korištenjem analize vremenskih slijedova podataka

**Borna Kovačević**

U sklopu 3Smart projekta (Smart Building – Smart Grid – Smart City, Program transnacionalne suradnje Dunav), za nebodersku zgradu Sveučilišta u Zagrebu Fakulteta elektrotehnike i računarstva izrađen je sustav koji omogućuje praćenje važnih varijabli sustava grijanja i hlađenja zgrade poput potrošnje energije po katovima ili temperature svake prostorije. Ovaj sustav redovito pohranjuje zapise navedenih varijabli u bazu podataka te je moguće razviti različite aplikacije koje bi iskoristile te podatke i unaprijedile rad sustava. Jedna od takvih aplikacija je i nadzor grijanja i hlađenja zgrade po prostorijama s ventilokonvektorima. Svrha aplikacije je olakšati posao operaterima zgrade zaduženim za ispravan rad sustava grijanja i hlađenja tako što im se unutar jednostavnog sučelja, dostupnog na web stranici, prikazuje neispravne ventilokonvektorske uređaje u prostorijama, a putem složenijih algoritama i točne podatke o iskoristivosti svakog uređaja. Također, aplikacija služi i za prikaz važnih varijabli unutar odabrane prostorije, npr. trenutne temperature, željene temperature ili brzine ventilatora ventilokonvektora. Dizajn aplikacije omogućuje jednostavno proširenje njenih funkcija, primjerice dodavanje sučelja za upravljanje grijanjem i hlađenjem prostorije.

**Ključne riječi:** HVAC, ventilokonvektor, klimatizacija, 3Smart.

# Summary

## Diagnostic of faulty fan coil units using data time analyzes

**Borna Kovačević**

Within the 3Smart project (Smart Building – Smart Grid – Smart City, the Interreg Danube Transnational Programme), a system for monitoring the important variables of heating and cooling systems, such as energy consumption per floor or temperature in zones, for the skyscraper building of the University of Zagreb Faculty of Electrical Engineering and Computing was developed. This system regularly stores the records of the mentioned variables in the database and it is possible to develop different applications that would use this data and improve the system performance. One of such applications is also the supervision of heating and cooling in individual building rooms with fan coils. The purpose of the application is to assist the building operators responsible for the proper operation of the heating and cooling system by displaying defective fan coil devices on a simple web interface. Also, the application displays important variables within a selected room, such as current temperature, setpoint temperature or fan coil unit fan speed. Application design allows for easy expansion of its functions, such as adding an interface for room heating and cooling control.

**Keywords:** HVAC, fan coil units, air conditioning, 3Smart.