

# Vacuum control

- **Pregled zadatka**

Cilj rada je implementacija WEB aplikacije koja će predstavljati simulaciju upravljanja usisivačima robotima. Aplikacija je potrebno da stvori utisak da je korisnik u stanju da dodaje usisivače u sistem i kontroliše stanja usisivača.

Projekat će biti podeljen po glavnim domenskim objektima koje je potrebno implementirati: upravljanje korisnicima i upravljanje usisivačima.

Upravljanje korisnicima je definisano u trećem domaćem zadatku. Dodatak su nove permisije za upravljanje usisivačima.

Takođe, potrebno je omogućiti zakazivanje operacija nad usisivačima (START , STOP ili DISCHARGE) kao i evidenciju grešaka nad zakazanim operacijama.

- **Zahtevi**

Usisivač kao entitet sadrži sledeće obavezne attribute:

Ime	Tip	Opis
id	Long	Jedinstvena identifikacija entiteta
status	Enum(ON, OFF, DISCHARGING)	Trenutni status usisivača.
addedBy	FK_UserTable(id)	Referenca na korisnika koji je dodao usisivač u sistem.
active	Boolean	Da li je usisivač uključen ili isključen iz sistema.

Ovo su osnovni atributi koje je potrebno podržati. Dozvoljeno je njihovo proširivanje i modifikacija po potrebi. Svaki usisivač pripada korisniku koji ga je dodao u sistem i korisnik ima uvid samo u usisivače koje je on dodao u sistem.

Akcije koje se mogu izvršiti nad usisivačem:

- SEARCH
- START
- STOP
- DISCHARGE
- ADD
- REMOVE

Svaka akcija je ispraćena permisijom. Potrebno je proširiti skup permisija i podržati još 6 novih, za svaku od mogućih akcija nad usisivačem: `can_search_vacuum`, `can_start_vacuum`, `can_stop_vacuum`, `can_discharge_vacuum`, `can_add_vacuum`, `can_remove_vacuums`. Shodno tome, korisnik bez specifične permisije nema pristup, ni mogućnost za izvršavanja odgovarajuće akcije.

Operacije `START`, `STOP` i `DISCHARGE` nisu momentalne i za njihovo izvršavanje je potreban neki vremenski period. Dok se neka od tih operacija izvršava nad usisivačem, obezbediti da nije moguće izvršiti ni jednu drugu operaciju nad tim istim usisivačem.

Svaki uspešni poziv ka nekoj od funkcionalnosti mora da vrati odgovor odmah sa statusom `2xx`, a ne nakon `X` sekundi - operacije `START`, `STOP` i `DISCHARGE` nakon poziva odmah vraćaju odgovor, a njihovo izvršavanje na odabranom usisivaču će se nastaviti u pozadini.

## Akcije/operacije nad usisivačima:

### SEARCH

Iz baze vraća niz aktivnih usisivača za ulogovanog korisnika.

Sljedeća tabela ne predstavlja entitet već moguće parametre pretrage. Svi parametri su opcionii.

Ime	Tip	Opis
<code>name</code>	<code>String</code>	Vratiti usisivače čije ime je jednako ili sadrži vrednost ovog parametra. Case insensitive komparacija.
<code>status</code>	<code>List&lt;String&gt;</code>	Vratiti usisivače čiji je status u jednom od poslaih.
<code>dateFrom</code>	<code>Date</code> (datum bez vremena)	Predstavlja početak vremenskog opsega. Samo ukoliko je i <code>dateTo</code> definisan onda će vratiti usisivače čiji je momenat kreiranja između <code>dateFrom</code> i <code>dateTo</code> .
<code>dateTo</code>	<code>Date</code> (datum bez vremena)	Predstavlja kraj vremenskog opsega. Pročitati <code>dateTo</code> .

### START

Usisivač koji je u bilo kom stanju koje nije `STOPPED` ne može da se startuje. Proces starta traje 15+ sekundi sa implementiranom vremenskom devijacijom po nahođenju.

Nakon proteklog navedenog vremena usisivač je potrebno prebaciti u stanje `RUNNING`.

## **STOP**

Usisivač koji je u bilo kom stanju koje nije RUNNING se ne može isključiti. Proces isključivanja traje 15+ sekundi sa implementiranom vremenskom devijacijom po nahođenju.

Nakon proteklog navedenog vremena usisivač prebaciti u stanje STOPPED.

## **DISCHARGE**

Usisivač koji je u bilo kom stanju koje nije STOPPED ne može biti ispražnjen (Discharged). Proces pražnjenja traje 30+ sekundi sa implementiranom vremenskom devijacijom po nahođenju.

Nakon protekle polovine navedenog vremena usisivač prebaciti u stanje DISCHARGING , a po završetku druge polovine prebaciti ponovo u STOPPED.

Nakon svaka tri perioda rada usisivača, tj. nakon svaka tri ciklusa RUNNING-STOPPED, potrebno je da se usisivač isprazni, tj. da se, i bez da korisnik zada komandu, izvrši operacija DISCHARGE (ovo je potrebno obezbediti i za usisivače čiji korisnici nemaju DISCHARGE permisiju).

## **ADD**

Ova akcija je momentalna i služi za dodavanje novog usisivača u sistem. Prilikom dodavanja usisivača potrebno je proslediti sve obavezne parametre. Rezultat je usisivač koji je u stanju STOPPED.

Prilikom dodavanja usisivača u sistem potrebno je povezati ulogovanog korisnika sa njim.

## **REMOVE**

Remove akcija je momentalna i može se izvršiti samo nad usisivačima koji su u stanju STOPPED. Ova akcija samo označava usisivač kao uklonjenog iz sistema, ali ga ne briše iz baze.

### **Zakazivanje operacije nad usisivačem:**

Operacije START, STOP i DISCHARGE se mogu zakazati - korisnik može da odredi datum i vreme kada će željena operacije da se izvrši. U odgovarajuće vreme, potrebno je da sistem samostalno pokuša da izvrši operaciju, ukoliko je usov za njeno izvršavanje ispunjen. Ukoliko operacija nije uspela, potrebno je to evidentirati u novoj tabeli - ErrorMessage. Entitet

ErrorMessage mora da sadrži datum, id usisivača, operaciju koja je bila zakazana i poruku o grešci koja se dogodila.

## Frontend

Na frontendu, pored grafičkog interfejsa za upravljanje korisnicima iz domaćeg 3, potrebno je implementirati tri nove stranice:

- Stranicu za pretragu usisivača
- Stranicu za dodavanje usisivača
- Stranicu sa istorijom grešaka

### 1. Pretraga usisivača:

Implementirati da se upotrebom SEARCH akcije prikažu svi usisivači, bez prosleđenih parametara kada se stranica učitava. Pored toga, potrebno je implementirati formu iznad tabele koja sadrži potrebna polja da pokrije sve funkcionalnosti za pretragu koje nudi Backend. Filteri će se primeniti submitom te forme.

### 2. Dodavanje usisivača

Napraviti jednostavnu formu koja će sadržati ime koje je potrebno da bi se usisivač dodao u sistem na Backend strani. Ova forma neće sadržati id, status, niti korisnika, to će Backend zaključiti sam.

### 3. Stranica sa istorijom grešaka

Tabela u kojoj je potrebno prikazati greške koje su se dogodile pri izvršavanju zakazane operacije. Potrebno je prikazati samo greške koje su se dogodile nad usisivačima ulogovanog korisnika.

Korisnik bez specifične permisije nema pristup, ni mogućnost za izvršavanja odgovarajuće akcije na frontend-u.

Ukoliko korisnik nema nijednu permisiju, nakon uspešnog login-a, obavestiti ga alertom.

## • Tehnologije

Backend:

- Spring ili JBoss
- Relaciona baza podataka

Frontend:

- Angular 2+