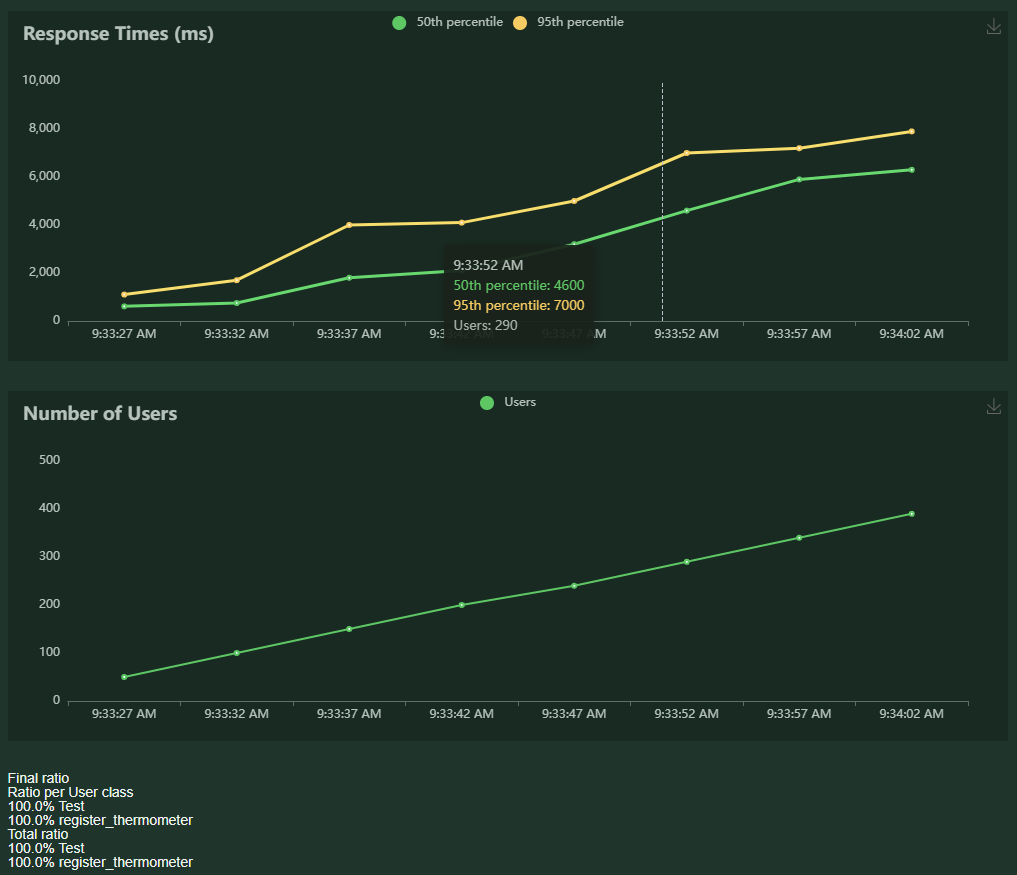
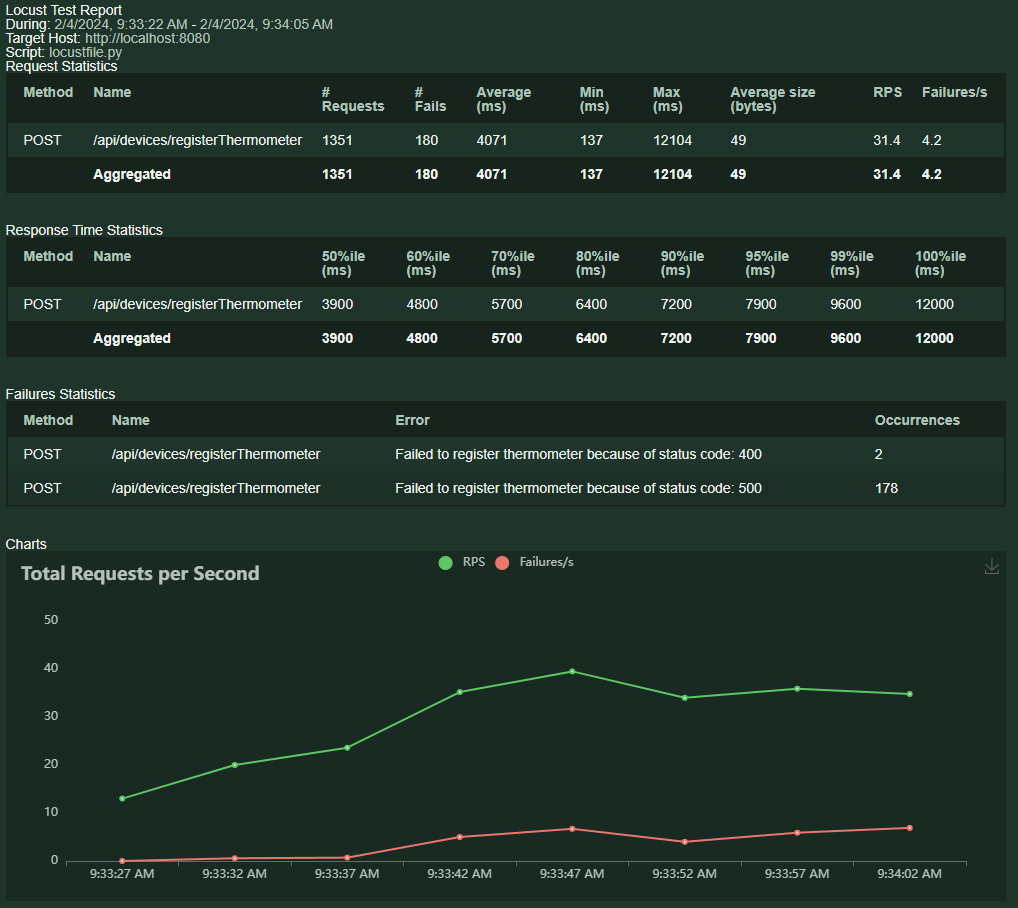
Testiranje performansi sistema - student 3

# Testiranje opterećenja sistema (*load testing*)

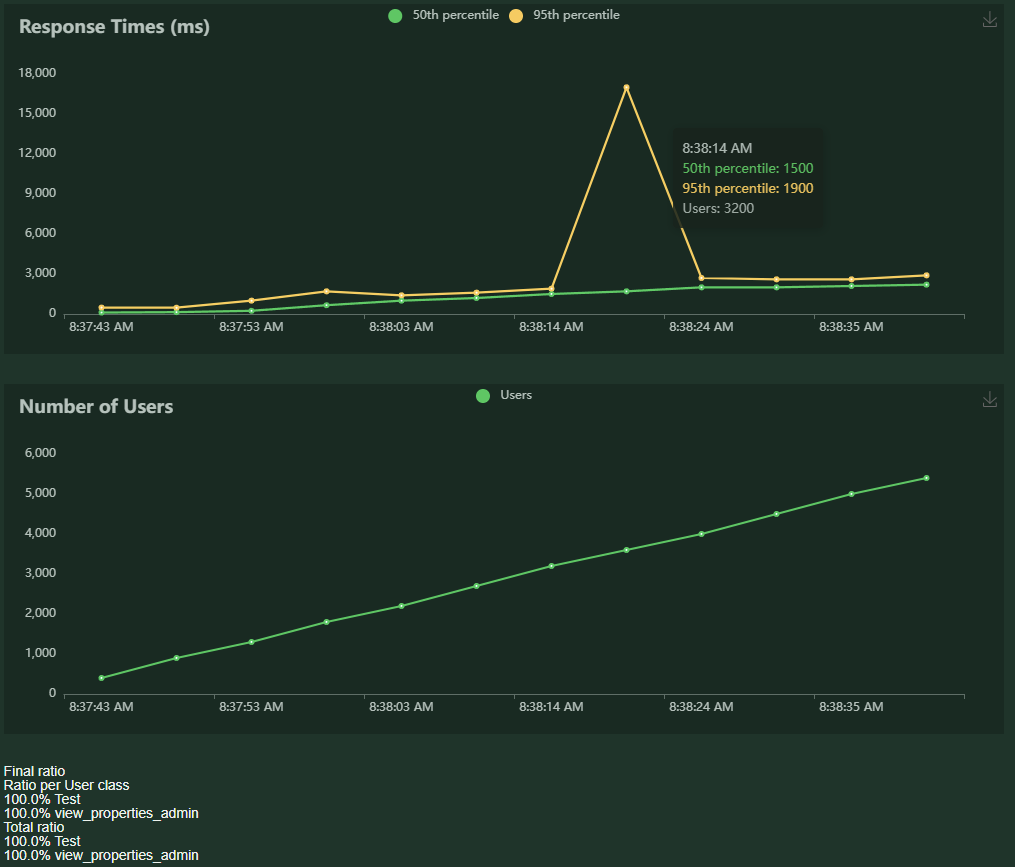
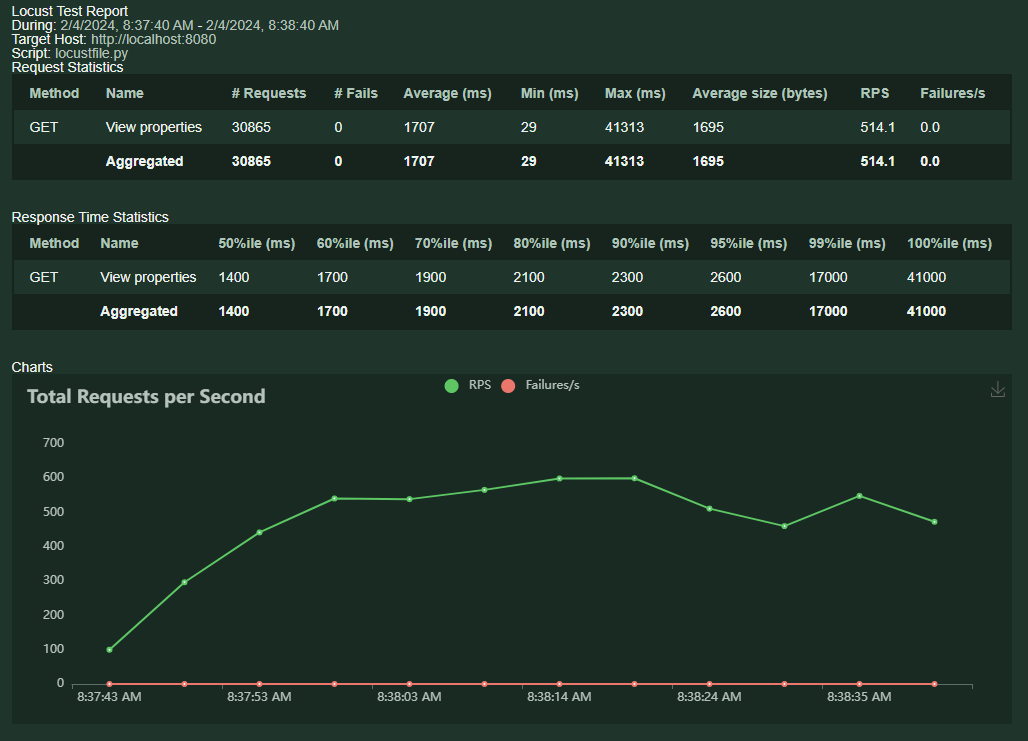
Testiranje opterećenja sistema za česte scenarije korišćenja izvršeno je upotrebom alata Locust za slanje zahteva backendu. Proverene su performanse sistema za 10 različitih scenarija korišćenja i ispitana je njihova promena sa povećanjem količine podataka sa kojim aplikacija rukuje kao i povećanjem broja korisnika koji istovremeno pokušavaju da izvrše neki isti scenario. Sledećih 10 scenarija je ispitano:

## Registracija termometra kao korisnik



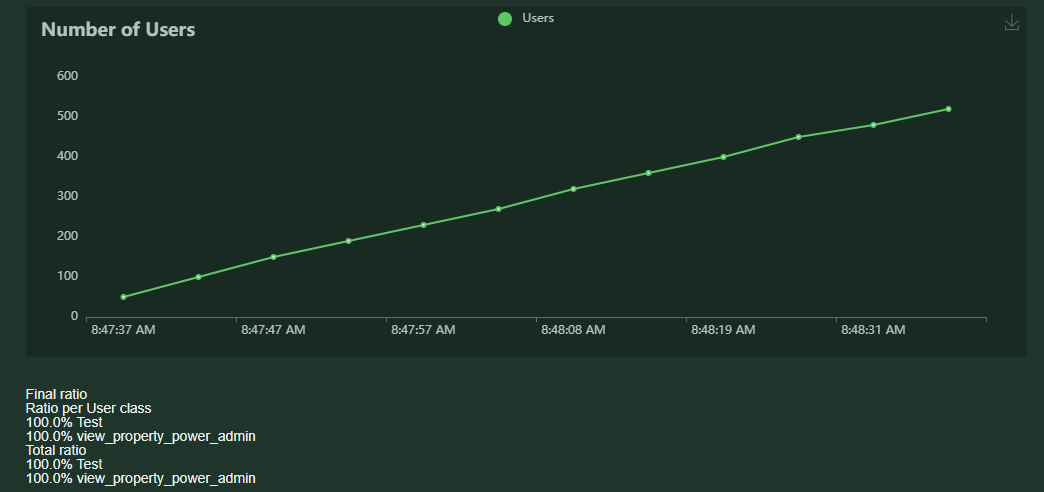
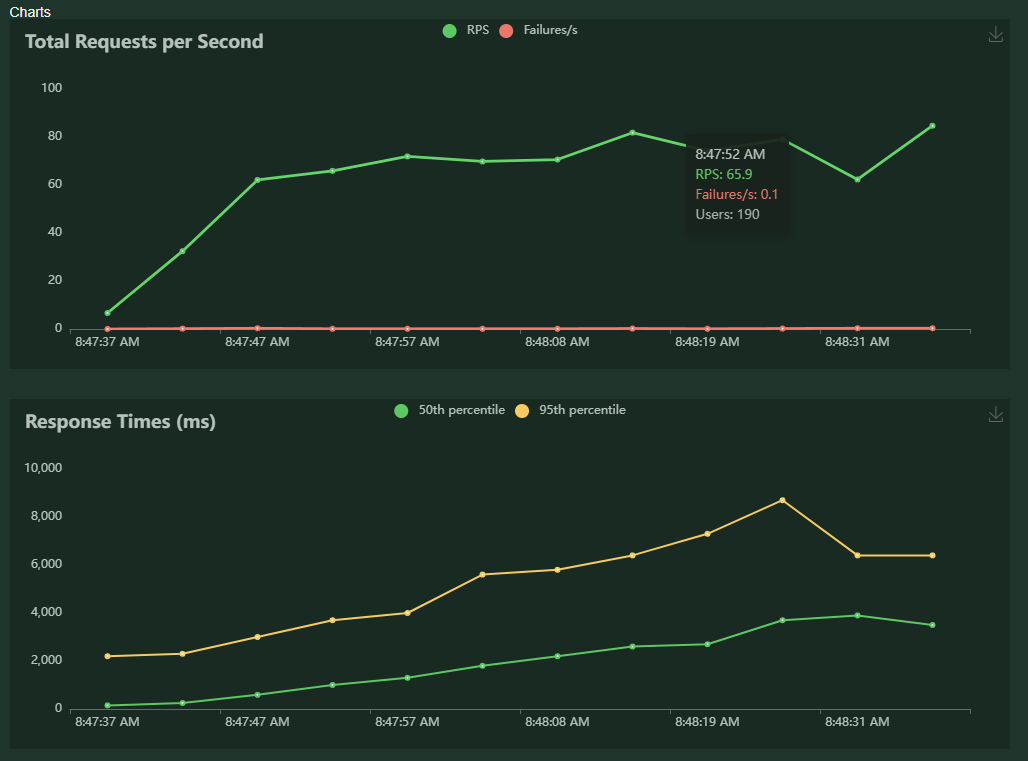
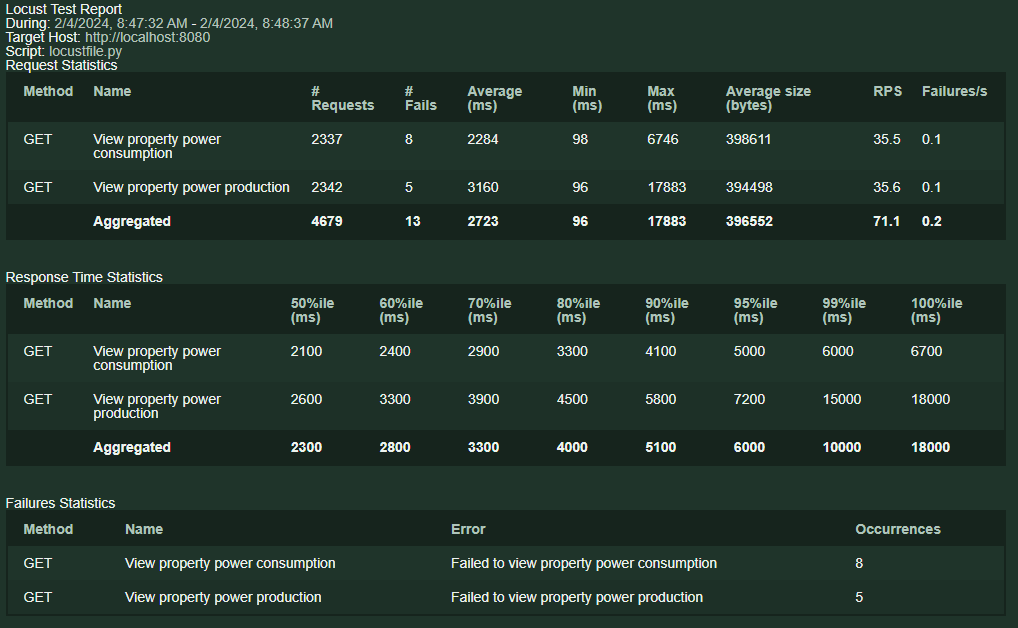
## Pregled svih nekretnina u sistemu kao administrator

Testiranje je izvršeno sa 10 nekretnina u sistemu.



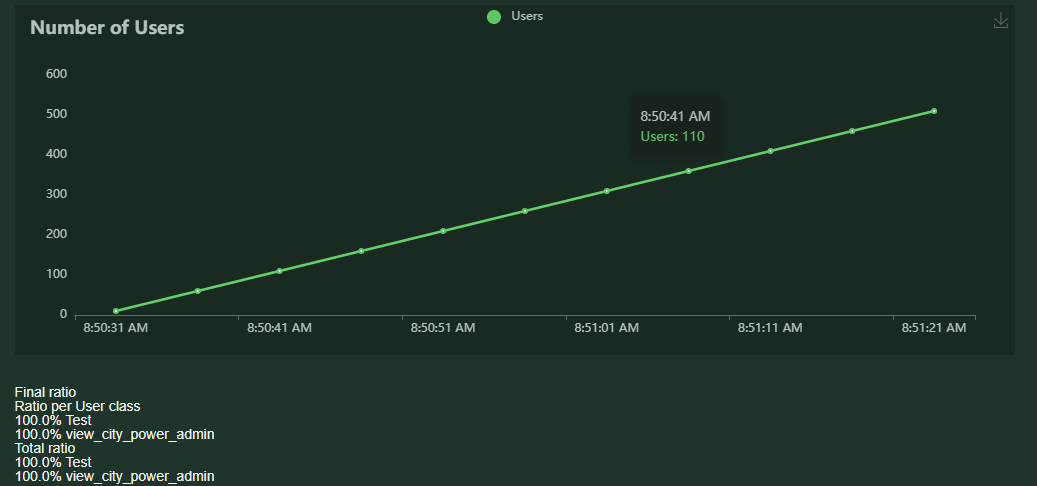
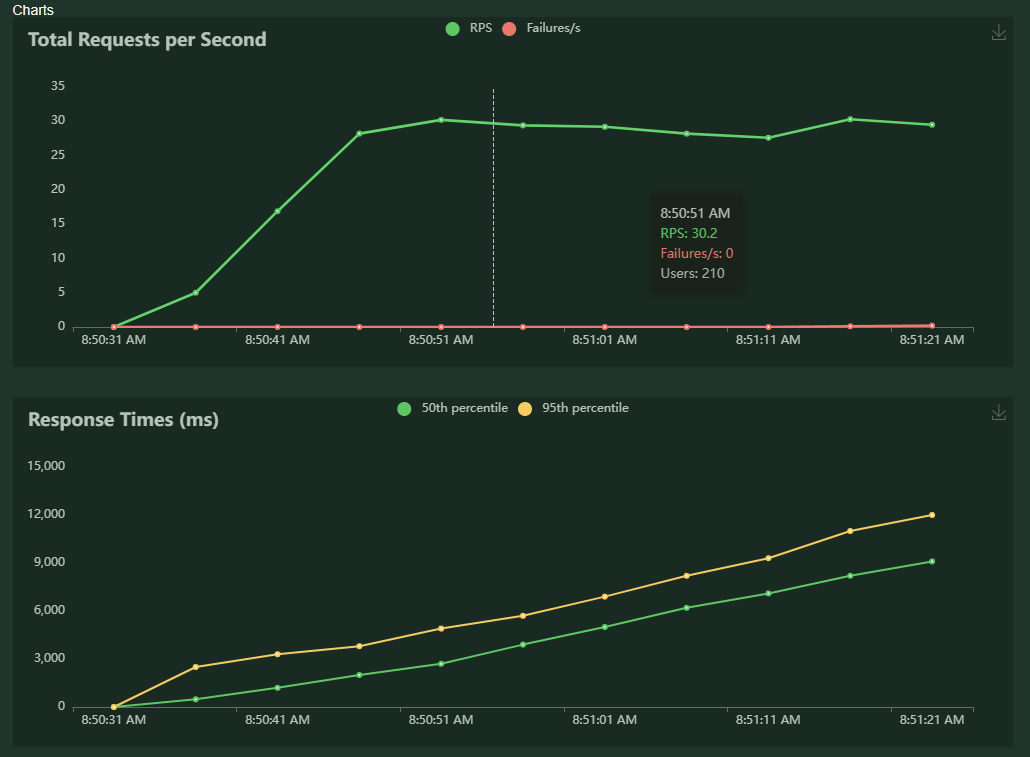
## Pregled potrošnje i proizvodnje struje unutar jedne nekretnine kao administrator

Testiranje je izvršeno sa oko nedelju dana podataka u sistemu.



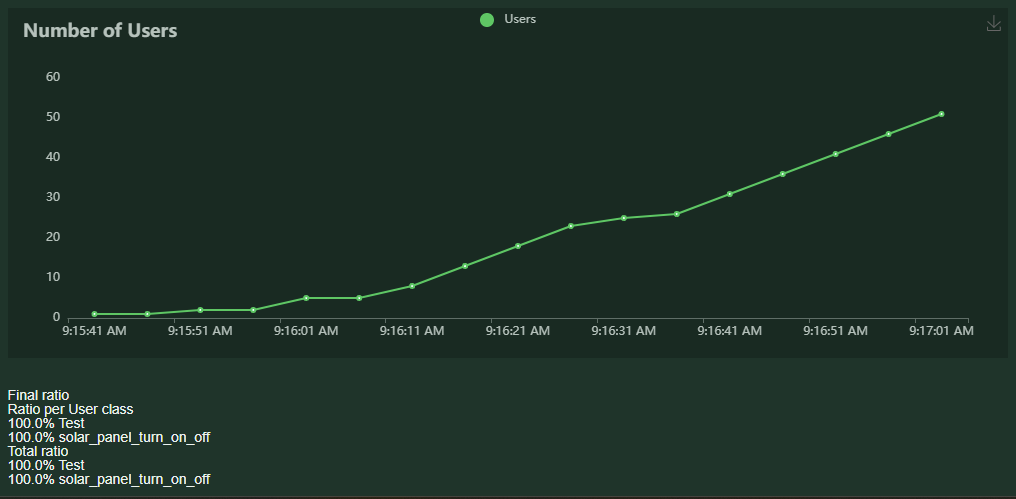
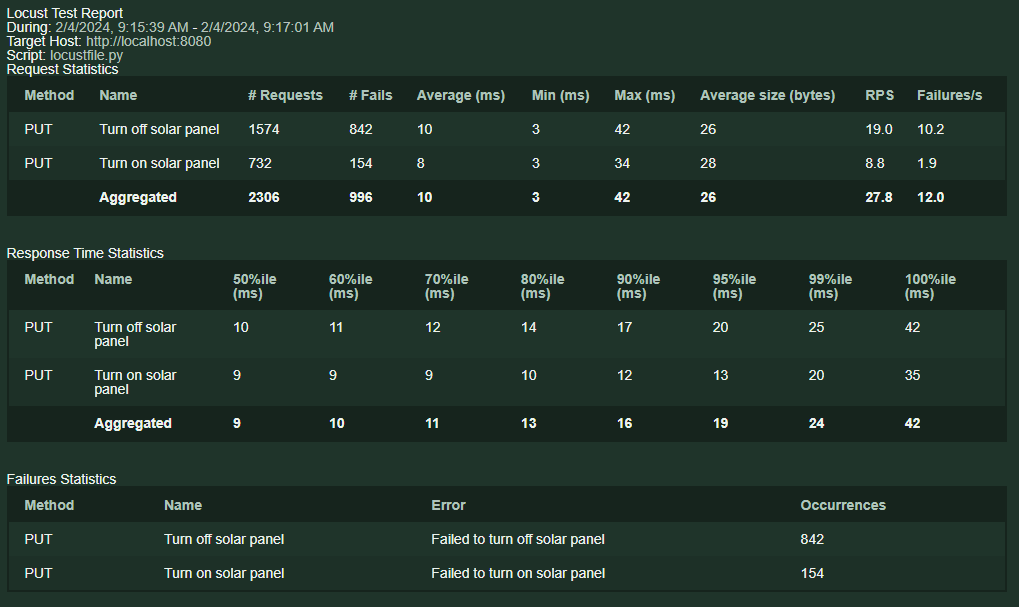
## Pregled potrošnje i proizvodnje struje unutar jednog grada kao administrator

Testiranje je izvršeno sa oko nedelju dana podataka u sistemu.



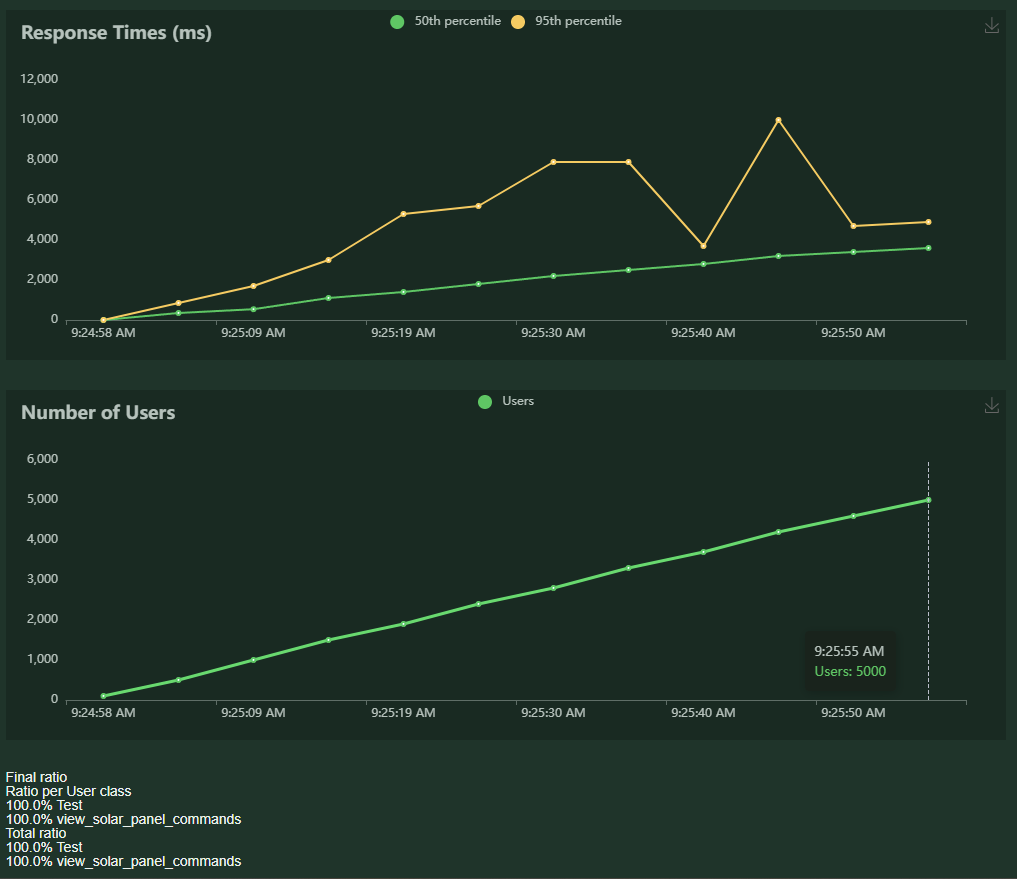
## Paljenje i gašenje sistema solarnih panela kao korisnik

Ekstreman slučaj gde veliki broj korisnika konkurentno pokušava da pali i gasi 2 solarna panela, pa postoji veliki broj neuspeha (jedan korisnik ugasi solarni panel pa želi da ga ponovo upali, ali ga je neko upalio u međuvremenu i slični slučajevi)



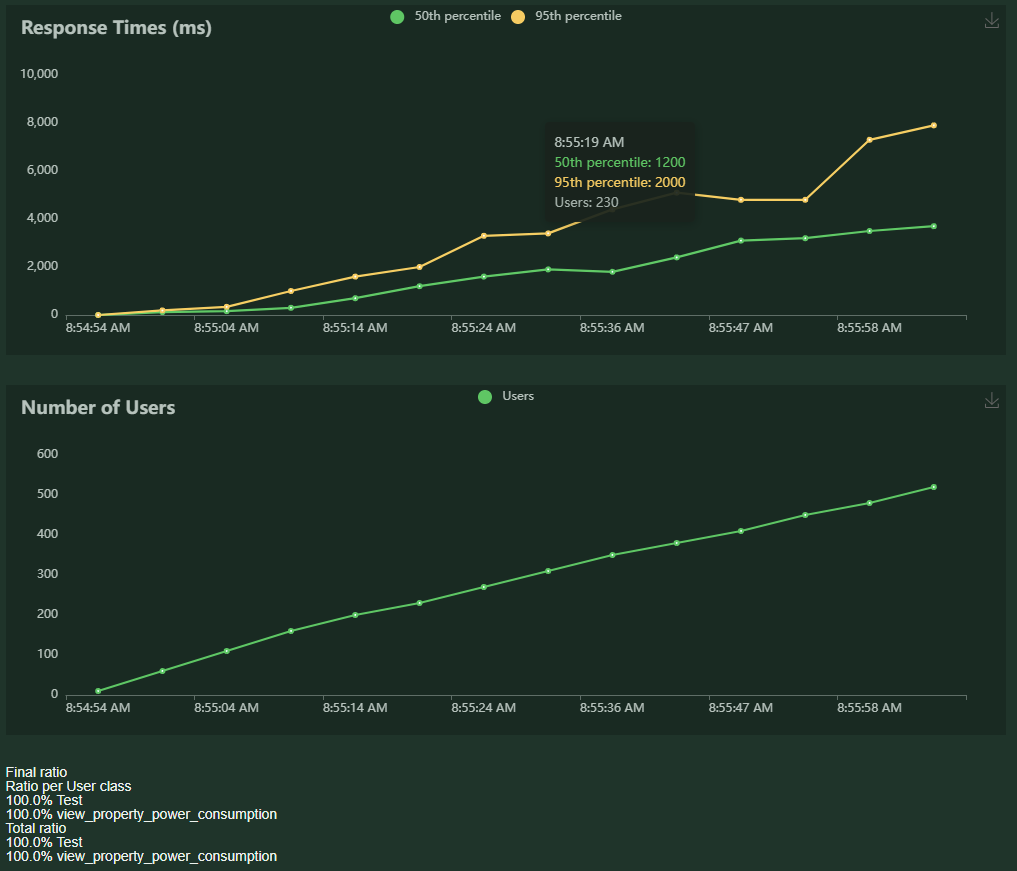
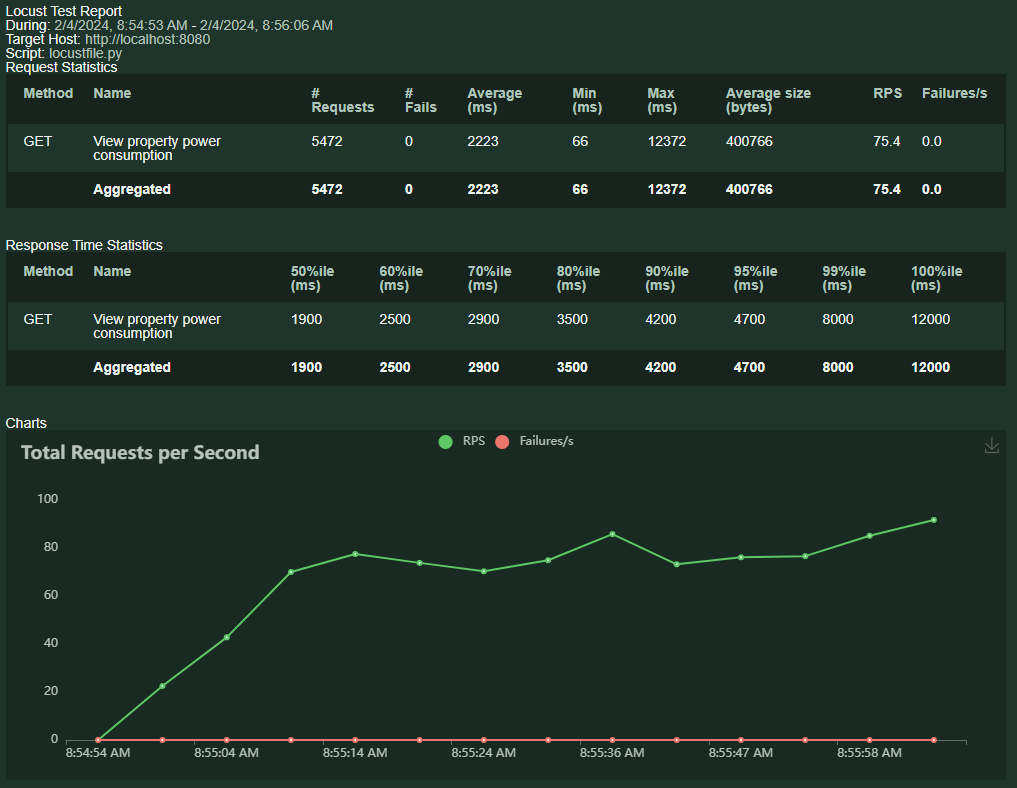
## Pregled akcija izvršenih nad sistemom solarnih panela kao korisnik

Testiranje je izvršeno sa preko 100 unesenih komandi.



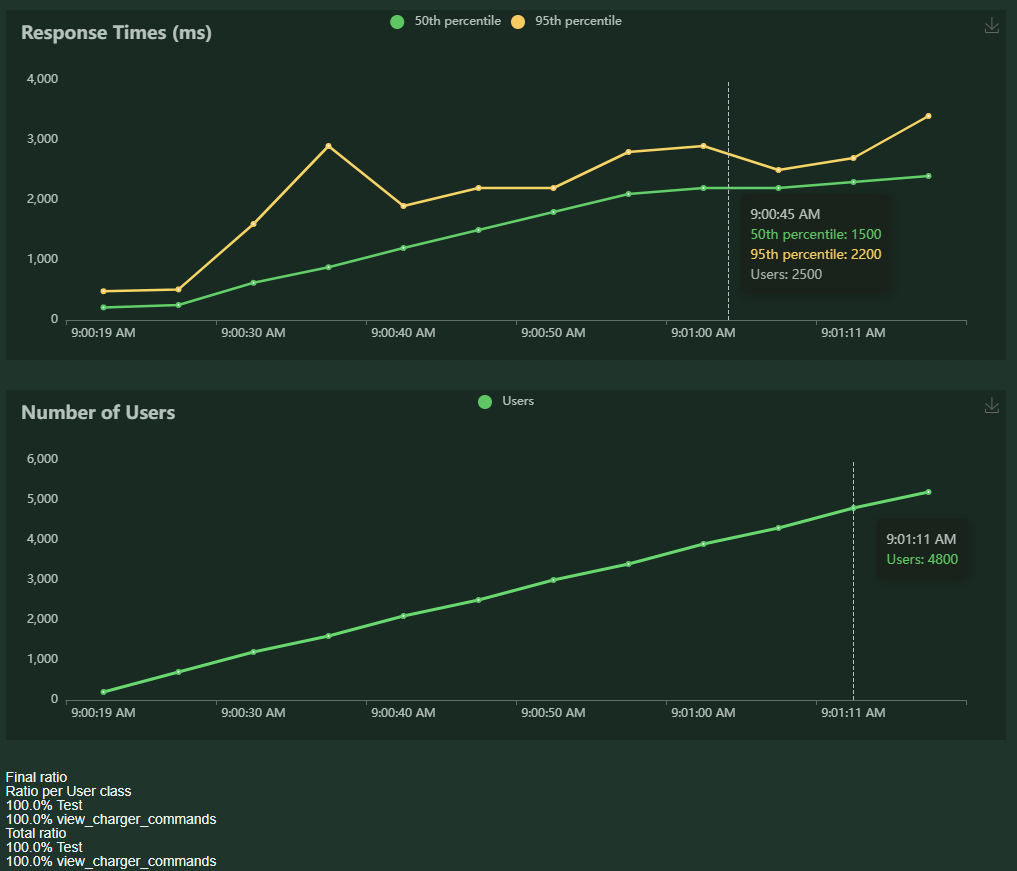
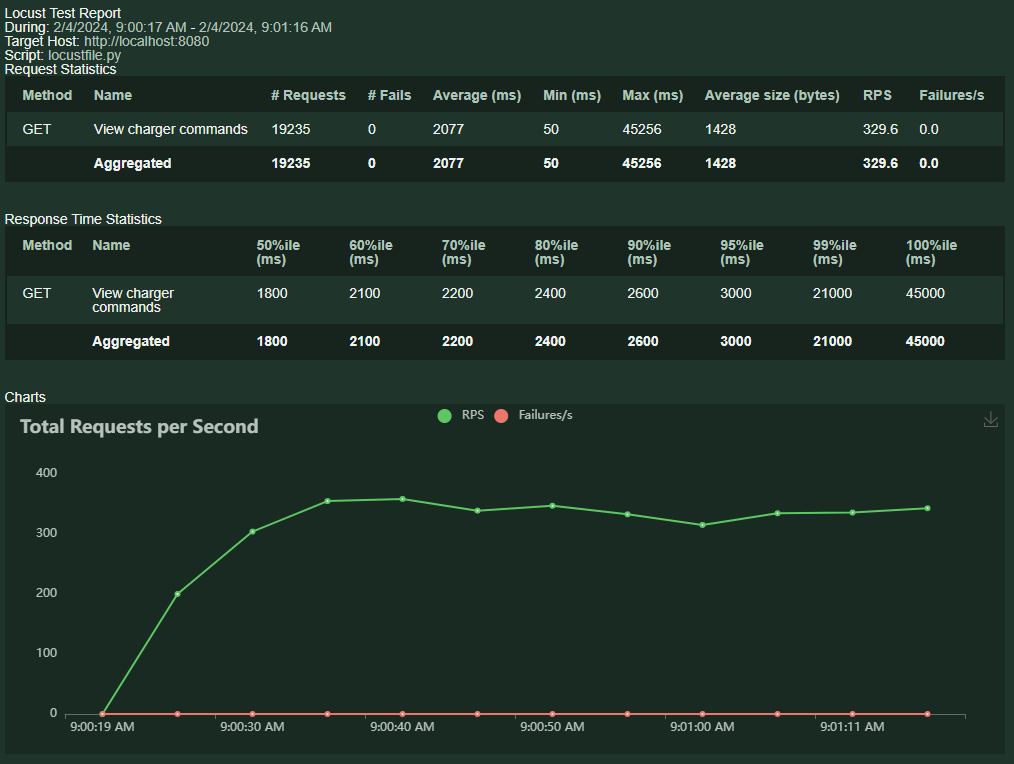
## Pregled istorijske potrošnje struje unutar jedne nekretnine kao korisnik

Testiranje je izvršeno sa oko nedelju dana podataka u sistemu.

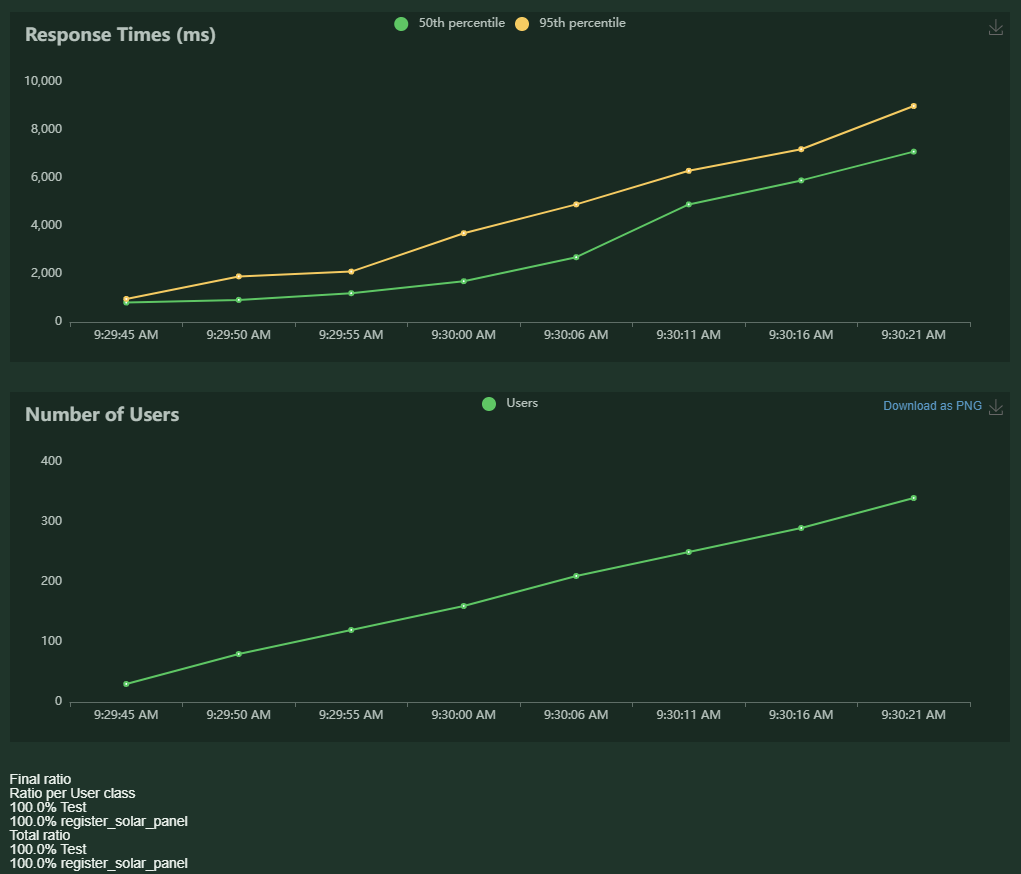
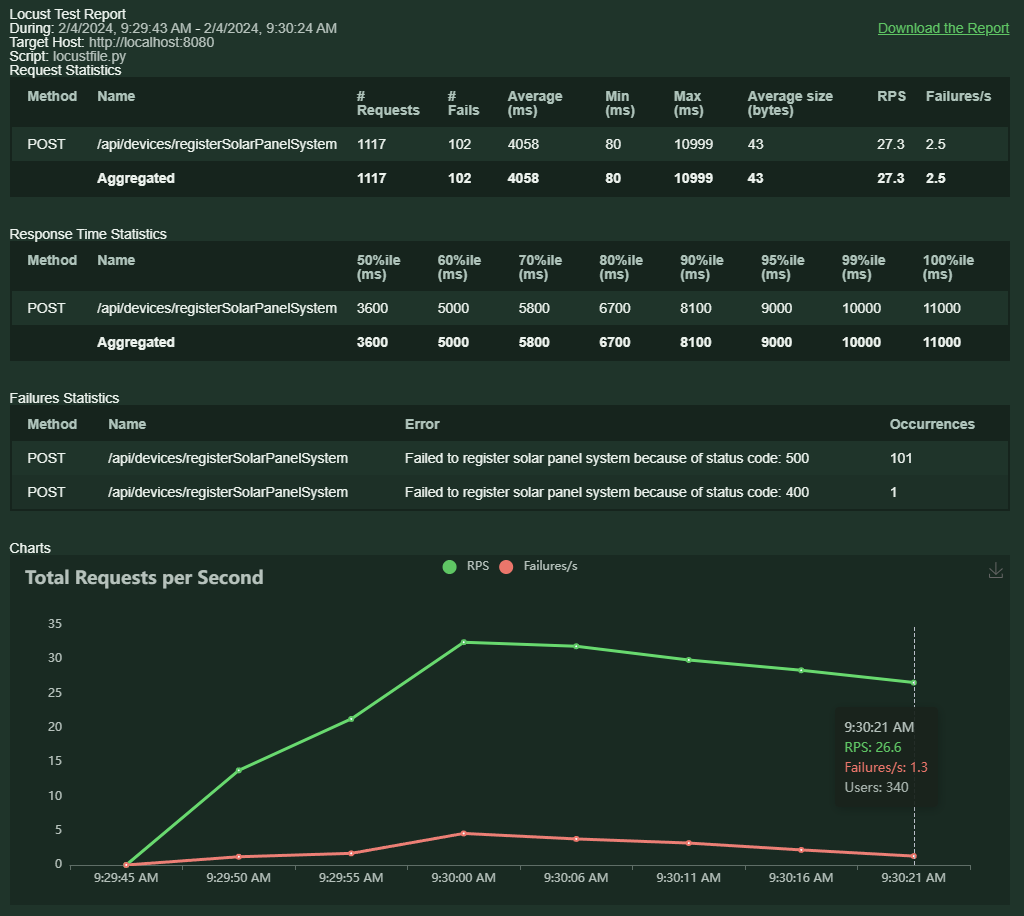


## Pregled akcija izvršenih nad punjačem za električna vozila kao korisnik

Testiranje je izvršeno sa oko nedelju dana podataka u sistemu.

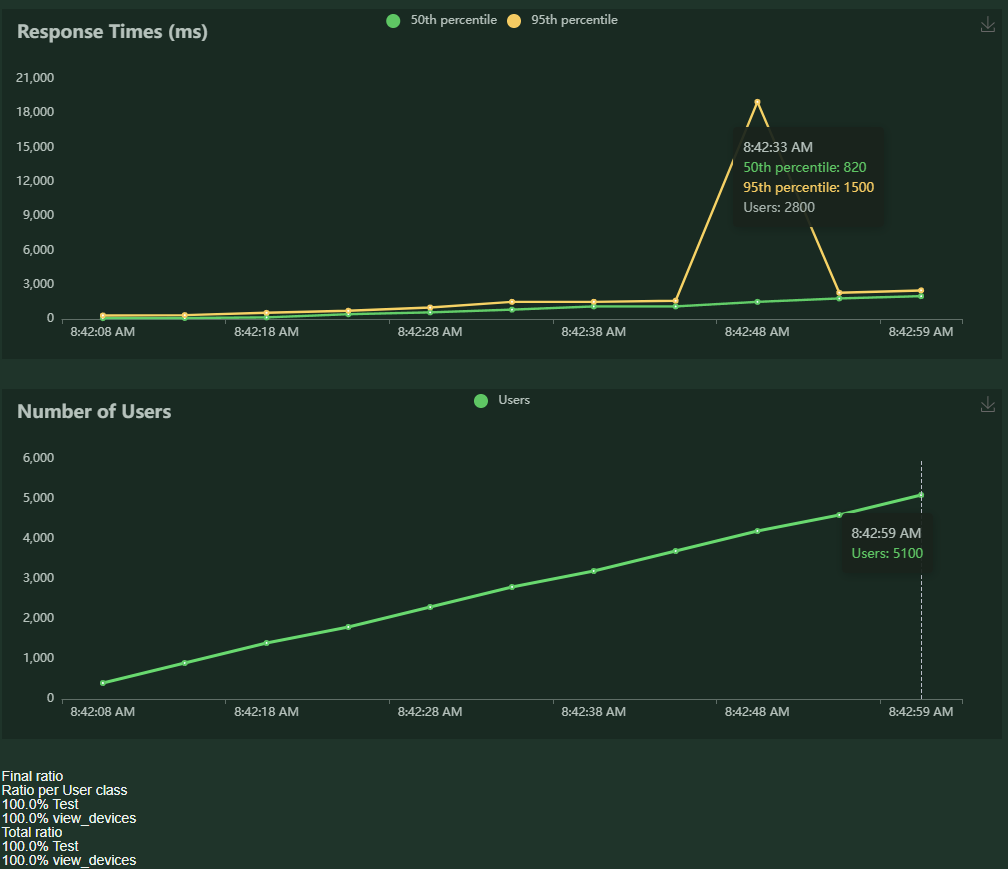
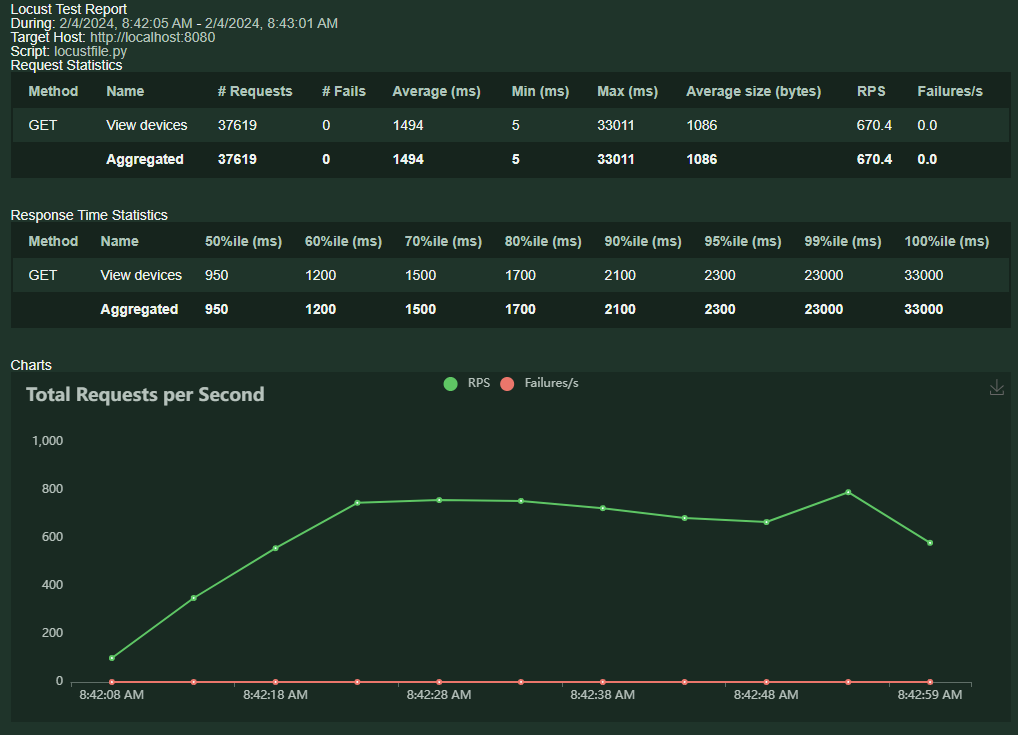


## Registracija solarnog panela kao korisnik



## Pregled uređaja kao korisnik

Testiranje je izvršeno sa 6 uređaja vezano za nekretninu.



# Testiranje performansi sistema usled povećanja broja uređaja (simulatora) koji komuniciraju sa platformom

Izvršeno je testiranje performansi sistema prilikom rada sa simulatorima, specifično simulatora za punjač, bateriju i sistem solarnih panela. Testirano je kako se te performanse menjaju pri povećanju broja simulatora koji su istovremeno pokrenuti.

Prvo je testiran sistem sa 1000 simulatora električnih punjača. Bez obzira na broj simulatora, komunikacija između simulatora i servera tekla je praktično instantno. Ne treba očekivati da bi se ovo promenilo sa većim brojem simulatora, uzimajući u obzir brzinu prenosa poruke i činjenicu da punjači imaju mali broj retkih promena stanja.



Zatim je testiran sistem sa 1000 simulatora baterija. Veći deo procesa smo imali praktično instantno slanje poruka, ali se viđa usporenje sa vremenom kako se unosi veći broj baterija i prilazi broju od 1000. Ovo se dešava zbog relativno dosta češće frekvencije slanja poruka baterije u odnosu na punjač. Naravno, i sa usporenjem kašnjenje je samo 200ms što nije značajno pogotovo u kontekstu naše aplikacije.





Na kraju je testiran sistem sa 1000 sistema solarnih panela. Od početka se vidi malo, neznatno kašnjenje. Kako vreme odmiče i veći broj panela se uključuje, kašnjenje se povećava ali bez problema ostaje u neznatnim granicama, sa oko 50 ms kašnjenja pri kraju.



Primećuje se da za konkretne uređaje koji su bili testirani nije potrebna neka specijalna optimizacija. Korišćeni MQTT protokol je već bio prilično efikasan u brzom prenosu podataka bez puno overheada.

Ipak, jedna potencijalna metoda optimizacije koja je bila generalno razmatrana i pre testiranja jeste keširanje. Funkcionalnosti koje su pogodne za optimizaciju keširanjem su one koje uključuju često pristupanje podacima ali retko menjanje. Utvrđeno je da su sledeće funkcionalnosti pogodne za keširanje: dobavljanje svih vlastitih kao i deljenih uređaja korisnika, dobavljanje odobrenih nekretnina korisnika i dobavljanje informacija o korisnicima.

Implementiranjem keširanja performanse za te funkcionalnosti su značajno poboljšane. Kao primer, dobavljanje 10 odobrenih nekretnina bez keširanja traje oko 10ms, dok sa keširanjem manje od 1ms. Kada god se odobri nova nekretnina tom korisniku, potrebno je ponovo oko 10ms za dobavljanje tih nekretnina pre nego što se podaci keširaju i performanse se ponovo poboljšaju. Treba imati u vidu da je odobrenje nove nekretnine veoma redak slučaj i da je zbog toga ovo uspešna i efikasna optimizacija.

* Vuk Radmilović SV73/2020