

# Napredne veb tehnologije – Specifikacija predmetnog projekta

Softversko inženjerstvo i informacione tehnologije 2023/2024

## 1. Uvod

Vaš zadatak je da projektujete i implementirate softversko rešenje koje omogućava monitoring pametnih kuća unutar pametnih gradova. Zamislite jednu pametnu kuću (ili stan) - ona ima veliki broj pametnih uređaja koji mogu da mere različite vrednosti poput temperature, vlažnosti vazduha, potrošnju električne energije itd. Zatim, zamislite neke druge pametne uređaje koji mogu da imaju manuelne komande (poput uključivanje klime ili uključivanje prskalica za navodnjavanje bašte) ili automatske (kada uveče bude jeftina struja automatski uključiti veš mašinu ili kada je jako sunce automatski spusti roletne). Bilo bi lepo da postoji platforma u oblaku na koju Vi kao vlasnik pametne kuće možete da se prijavite i da u realnom vremenu pratite šta se dešava u Vašoj kući. Tu istu platformu možete da koristi veći broj korisnika različitih nekretnina i da svako prati svoje nekretnine. Administratori te platforme mogu da je koriste kako bi pratili potrošnju (i proizvodnju) električne energije u četvrtima ili celim gradovima kojima pripadaju nekretnine.

## 2. Uloge korisnika

Platforma razlikuje sledeće vrste korisnika:

- Redovan registrovan korisnik: može da registruje novu pametnu nekretninu (kuću ili stan) i pametne uređaje unutar svoje nekretnine, zatim da nadzire rad različitih uređaja unutar svoje pametne nekretnine i upravlja njima;
- Administrator: ima pravo nadzora i upravljanja sistema u realnom vremenu pomoću većeg broja grafičkih elemenata; **administrator ne može da izvršava nijednu funkcionalnost koja pripada redovnom korisniku, dok redovni korisnik ne može da izvršava nijednu funkcionalnost koja pripada administratoru;**
- Neautentifikovani korisnik: Ima mogućnost registracije i prijave na sistem.

## 3. Pametni uređaji

Za potrebe ovog projekta i podele posla prilikom izrade softverskog rešenja podelićemo sve pametne uređaje u sledeće 3 grupe:

1. Pametni kućni aparati (PKA) - pametni uređaji koji se nalaze unutar stana/kuće
2. Spoljni pametni uređaji (SPU) - pametni uređaji koji se nalaze izvan stana/kuće (na balkonu, u bašti, itd.)

### 3. Veliki elektroenergetski uređaji (VEU) - pametni uređaji koji predstavljaju velike proizvođače ili potrošače električne energije

Kako ne bismo koristili veliki broj pametnih fizičkih uređaja koji su često skupi u teško i dugo se nabavljaju da testiramo rad našeg sistema, potrebno je simulirati pametne uređaje pomoću malih i jednostavnih aplikacija (npr. konzolnih) koji imaju ulogu simulatora pametnog uređaja (u nastavku ćemo ove aplikacije zvati **simulatori**). U zavisnosti od tipa pametnog uređaja, simulator može imati različito ponašanje.

Ako simulatorom simuliramo rad npr. nekog jednostavnog senzora poput termometra, simulator će periodično (npr. svake sekunde) slati nasumično generisanu vrednost temperature. Nasumično generisane vrednosti treba da budu iz nekog smislenog opsega i da što približnije simuliraju ponašanje fizičkog uređaja. Npr. ako simuliramo rad termometra, u sredini dana bi generisao vrednosti u rasponu od 20 do 25 stepeni celzijusa (pošto su to npr. tipične temperature za sredinu dana u toku kasnog leta ili rane jeseni), dok bi u toku noći generisao vrednosti u rasponu od 10 do 15 stepeni celzijusa (pošto su to npr. tipične temperature za to doba dana u isto doba godine).

Ako simulatorom simuliramo rad pametnog uređaja koji ima mogućnost izvršavanja komandi, onda treba simulirati i odziv sistema na te komande. Ako npr. simulatorom simuliramo rad klima uređaja i ako korisnik putem platforme u oblaku uključi hlađenje, simulator treba da odgovori da je akcija uspešna. Zatim, ako korisnik definiše da je nova temperatura na koju treba uređaj da hladi npr. 17 stepeni celzijusa, simulator treba da odgovori da je akcija uspešna ako simulirani uređaj ima mogućnost rada sa tom temperaturom ili da je akcija neuspešna ako simulirani uređaj ima minimalnu temperaturu rada koja je veća od tražene (npr. simulira se uređaj koji radi minimalno sa 18 stepeni celzijusa, a korisnik traži da radi na 17 stepeni celzijusa - akcija se ne može izvršiti).

Događaje poput dolaska automobila na kapiju ili dolazak automobila na punjenje (od strane punjača) simulirati kao nasumične događaje koji se dešavaju u nasumično vreme prema nekom algoritmu koji osmislite.

Softversko rešenje treba da podrži rad sledećih pametnih uređaja:

1. Zajedničko za sve uređaje
  - a. Periodično javlja platformi da je online - u slučaju da se neko duže vreme uređaj nije javio da je online (npr. 30 sekundi), platforma treba da smatra da taj uređaj više nije na mreži i to treba da zabeleži u bazi podataka
  - b. Komuniciraju sa platformom putem MQTT protokola
2. Pametni kućni aparati
  - a. Senzor ambijentalnih uslova
    - i. periodično javlja koja je trenutna sobna temperatura
    - ii. periodično javlja koja je trenutna sobna vlažnost vazduha
  - b. Klima uređaj
    - i. Uređaj ima spisak podržanih režima rada kao i minimalnu i maksimalnu temperaturu sa kojom radi

- ii. Korisnik može da započne ili zaustavi rad ovog uređaja
- iii. Korisnik može da promeni režim rada ovog uređaja (hlađenje, grejanje, automatski ili ventilacija)
- iv. Korisnik može da podesi temperaturu sa kojom uređaj radi
- v. Korisnik može da definiše specijalan način rada klima uređaja, npr. može da definiše da svakog dana između 10h i 14h radi režim hlađenja na 20 stepeni ili da preko dana radi automatski režim koji održava temperaturu na 23 stepena, a da noću od 22h do 6h radi automatski režim koji održava temperaturu na 20 stepeni

c. Veš mašina

- i. Uređaj ima spisak podržanih režima rada (ciklusa)
- ii. Korisnik može da započne rad mašine prema jednom od podržanih načina rada (ciklusa)
- iii. Korisnik može da zakaže početak rada nekog ciklusa, npr. može definisati da se uređaj uključi večeras u 23h ili sutradan u 22:30h i započne određeni ciklus pranja

3. Spoljni pametni uređaji

a. Lampa

- i. Korisnik može da uključi i isključi sijalicu lampe (jedna lampa ima tačno jednu sijalicu)
- ii. Uređaj ima senzor osvetljenja i periodično javlja koliko je očitano osvetljenje
- iii. Lampa ima automatski režim rada - u zavisnosti od nivoa osvetljenja lampa automatski uključi ili isključi sijalicu

b. Kapija za vozila

- i. Korisnik može otvori i zatvori kapiju
- ii. Korisnik može da uključi i isključi javan ili privatan režim kapije; ne mogu istovremeno biti oba režima aktivna
- iii. Kapija ima kameru sa čitačem registarskih oznaka te može da očita koja je registarska oznaka vozila koje je ispred kapije
- iv. Kapija poseduje senzor blizine te može očitati kada je u blizini vozilo
- v. Kapija ima javan režim rada - propušta svako vozilo koje dođe pred kapiju (kapija se otvori i zabeleži registarsku oznaku vozila i vreme kada ulazi u dvorište, a kada vozilo prođe kapija se zatvori)
- vi. Kapija ima privatan režim rada - korisnik može da definiše spisak registarskih oznaka koje imaju dopuštenje da prođu kroz kapiju i kapija propušta svako vozilo čija registarska oznaka je na tom spisku (kapija se otvori i zabeleži registarsku oznaku vozila i vreme kada ulazi u dvorište, a kada vozilo prođe kapija se zatvori)
- vii. Kada vozilo želi da napusti dvorište, kapija se otvori i zabeleži registarsku oznaku vozila i vreme kada izlazi iz dvorišta, a kada vozilo prođe kapija se zatvori

c. Sistem prskalice

- i. Korisnik može da uključi i isključi sistem

- ii. Korisnik može da definiše specijalan način rada sistema, npr. može da definiše da se automatski uključi ili isključi u određeno doba dana (npr. uključi u 20h i isključi u 7h) ili određeni dan (ili određenim danima) u nedelji
4. Veliki elektroenergetski uređaji
- a. Sistem solarnih panela
    - i. Sistem solarnih panela sadrži veći broj panela koji zajedno proizvode električnu energiju kada su pogodni vremenski uslovi. Paneli imaju veličinu ( $m^2$ ) i efikasnost. Na svaki minut javljaju koliko je električne energije (u kW) proizvedeno u prethodnom minutu. Na [sledećem linku](#) se nalazi formula za izračunavanje proizvodnje električne energije za dan i mesec, a potrebno je izraditi formulu za obračunavanje proizvodnje za jedan minut. U zavisnosti od doba dana količina proizvedene električne energije može se drastično razlikovati.
    - ii. Korisnik može da isključi sistem da bi se izvršilo održavanje panela i da ga ponovo uključi.
  - b. Kućna baterija
    - i. Električna energija koja je proizvedena od strane sistema solarnih panela biva sačuvana unutar kućne baterije (unutar jedne ili je ravnomerno raspodeljena unutar više njih); svaka baterija ima svoju veličinu
    - ii. Svi pametni uređaji unutar nekretnine koji su povezani na mrežu troše električnu energiju prvo iz baterije. U slučaju da je baterija prazna, uzimaju struju iz mreže (od elektrodistribucije). Stoga, baterija periodično javlja kakvo je njeno stanje (koliko je trenutno zauzeta). U slučaju da se proizvede više struje nego što baterije mogu da sačuvaju, višak struje proslediti elektrodistribuciji.
    - iii. Zabeležiti svaki minut koja količina električne energije je preuzeta od elektrodistribucije ili vraćena elektrodistribuciji.
  - c. Punjač za električna vozila
    - i. Punjač za električna vozila ima određenu snagu punjenja, kao i ukupan broj priključaka (na svaki priključak može se povezati po jedno vozilo)
    - ii. Kada vozilo započne punjenje, zabeleži se trenutak kada se to desilo. Takođe, kada se završi punjenje, zabeleži se i taj trenutak kao i koliko energije je potrošeno u toku punjenja. Električna energija koja je potrošena na punjenje vozila se uzima iz kućne baterije (ako ima sačuvane energije) ili iz mreže ako je baterija prazna.
    - iii. Kada se vozilo poveže na punjač, punjač od vozila dobija informaciju koliko je baterija vozila puna (kolika je veličina baterije vozila i koliko je baterija puna trenutno). Korisnik ima mogućnost da definiše do kog procenta popunjenosti želi da punjač dopunjava vozila i kada se taj trenutak desi, punjač treba da stane sa dopunjavanjem vozila (npr. definiše da kada se vozilo napuni do 90% da prestane dalje punjenje)

## 4. Funkcionalni zahtevi

### 4.1. Inicijalno stanje sistema

Kada se softversko rešenje pokrene, potrebno je automatski (prilikom prvog podizanja aplikacije) generisati superadmin nalog (sa username-om **admin**). Lozinka superadmina treba da bude nasumično generisana tekstualna vrednost koja sadrži veći broj karaktera. Superadmin može da dodaje druge admina u sistem i da radi sve što i oni mogu da rade u sistemu. Kada superadmin doda novog admina, nalog novododatog admina biva automatski aktivan. Obični admini ne mogu da dodaju druge admina.

### 4.2. Registracija i prijava korisnika

Dizajnirati početnu stranu koju vidi neprijavljeni korisnik sa linkovima na stranice za registraciju i prijavu. Omogućiti registraciju i prijavu novih korisnika na sistem. Nakon što korisnik popuni formu za registraciju (u sklopu koje **upload-uje svoju profilnu sliku**) i uspešno se registruje, njegov nalog nije aktivan, već sistem šalje email na unetu adresu korisnika sa linkom za aktivaciju naloga. Nakon što se nalog aktivira, korisnik može da se prijavi. Prijavljeni korisnik u bilo kom trenutku može da se odjavi sa sistema.

### 4.3. Registracija pametne nekretnine

Običan korisnik može da registruje pametnu nekretninu (kuću ili stan). Pored osnovnih informacija (poput adrese, grada, kvadrature, broja spratova itd.) i slike (koja se **upload-uje**), može da navede lokaciju nekretnine tako što je označi na klikom na mapi (koristiti bilo koje dostupno rešenje: npr. OpenLayers map, Leaflet, Modest maps, Polymaps itd.). Za izbor grada ponuditi izbor jednog od gradova iz predefinisanoeg spiska gradova koji se nalazi u bazi podataka (bitno zbog funkcionalnosti u nastavku; kako svaki grad pripada nekoj državi, potrebno je imati i predefinisani spisak država u bazi podataka). Kada se kreira nova nekretnina, ona nije automatski aktivna, već je na čekanju da je administrator odobri ili odbije. Takođe, korisnik može da vidi spisak svih svojih nekretnina i njihov status. Jedan korisnik može da poseduje više nekretnina. Stranica sa spisakom svih nekretnina je početna stranica običnog korisnika nakon što se prijavi na sistem.

### 4.4. Obrada zahteva za registraciju nekretnine

Administrator ima pregled svih novokreiranih nekretnina na čekanju i može da ih prihvati ili odbije uz obrazloženje. Jednom kada je nekretnina odbijena ili prihvaćena, njen status se ne može menjati. Ako administrator prihvati, nekretnina biva aktivna i sistem korisniku čija je nekretnina aktivirana šalje email sa potvrdom. Ako administrator odbije, mora da navede obrazloženje zašto odbija i korisniku čija je nekretnina odbijena šalje email sa informacijom da

je nekretnina odbijena uz obrazloženje zašto. Stranica sa pregledom svih zahteva je početna stranica administratora nakon što se prijavi na sistem.

## 4.5. Registracija pametnih uređaja

Običan korisnik može da registruje novi pametni uređaj (bilo koji od nabrojanih tipova uređaja iz poglavlja 3) unutar svoje nekretnine. Nakon registracije uređaja (u sklopu koje **upload-uje sliku uređaja**), taj uređaj je offline (nije na mreži). Svaki uređaj ima autonomno napajanje (npr. na male alkalne baterije) ili kućno (koristi struju iz mreže ili kućne baterije) kao i brojne dodatne podatke koji zavise od tipa uređaja koje korisnik bira prilikom kreiranja. U slučaju uređaj da koristi struju iz mreže, značajna je informacija koja je njegova potrošnja struje. Sistem treba novokreiranom uređaju da automatski dodeli podatke koji su potrebni za njegovo povezivanje i komunikaciju sa platformom u oblaku.

## 4.6. Konekcija pametnih uređaja i platforme

Implementirati zajedničko ponašanje za sve pametne uređaje. Kada pametni uređaj ostvari konekciju sa platformom, on periodično javlja da je na mreži. U slučaju da duže vreme uređaj ne javi da je na mreži, platforma treba automatski da označi da uređaj nije na mreži. Komunikacija uređaja i platforme se izvodi putem MQTT protokola sa posrednikom (broker). Demonstrirati pokretanjem 3 različite instance pametnog uređaja koji nezavisno jedan od drugog komuniciraju sa platformom.

## 4.7. Davanje permisija za upravljanje pametnim uređajima

Vlasnik nekretnine može da dodeli permisije drugim korisnicima da upravljaju jednom ili više uređaja uređajima. To može da uradi na dva načina:

- Vlasnik može nekom korisniku da dodeli permisije za upravljanje celom nekretninom i tom prilikom korisnik koji je dobio permisije može da vidi sve uređaje u sistemu i koristi sve njihove funkcionalnosti;
- Vlasnik može nekom korisniku da dodeli permisije za korišćenje tačno određenog uređaja i tom prilikom korisnik koji je dobio permisije može da vidi isključivo one pametne uređaje unutar nekretnine za koje ima permisije.

U bilo kom trenutku vlasnik nekretnine može da uskrati permisije korisniku za celu nekretninu ili za određeni uređaj.

## 4.8. Pregled dostupnosti pametnih uređaja

Običan korisnik može da vidi sve uređaje unutar neke nekretnine. Za izabrani uređaj može da vidi grafik dostupnosti. Korisnik može da izabere vremenski period u kom želi da vidi dostupnost uređaja (npr. poslednjih 6h, poslednjih 12h, poslednja 24h, poslednjih nedelju dana, mesec dana ili da izabere datume od-do pri čemu razlika između ta dva datuma ne sme da bude veća od mesec dana). Za izabrani vremenski period, korisnik vidi koliko je procentualno i vremenski uređaj bio na mreži i koliko nije bio na mreži, (koliko je bio online, a koliko offline). U slučaju da

je u pitanju kraći vremenski period (npr. poslednjih 6h), prikazati dostupnost uređaja unutar svakog sata, dok ako je izabran duži vremenski period (npr. duži od 2 dana), prikazati dostupnost uređaja unutar svakog dana. Dostupnost uređaja prikazati pomoću grafika (bar chart, plot-ovi itd.).

## 4.9. Pregled potrošnje električne energije

Omogućiti administratoru da prati potrošnju električne energije. Administrator ima spisak svih registrovanih nekretnina u sistemu i za svaku može da vidi koliko električne energije je potrošeno i proizvedeno u različitim danima, mesecima, dobu dana itd. Takođe, vidi i spisak svih gradova u kojima se nalaze registrovane nekretnine i za svaki grad može da vidi koliko su sve nekretnine koje pripadaju tom gradu zajedno potrošile i proizvele električne energije. Korisnik može da izabere vremenski period u kom želi da vidi pregled potrošnje (npr. poslednjih 6h, poslednjih 12h, poslednja 24h, poslednjih nedelju dana, mesec dana ili da izabere datume od-do pri čemu razlika između ta dva datuma ne sme da bude veća od mesec dana).

## 4.10. Upravljanje PKA uređajem - Senzor ambijentalnih uslova

Implementirati funkcionalnosti koje podržava senzor ambijentalnih uslova (sekcija 3). Vrednosti koje senzor očitava potrebno je sačuvati u bazi podataka. Korisnik može da posmatra pročitane vrednosti senzora za svaki tip merenih podataka u dva režima:

- Pregled vrednosti u realnom vremenu - korisnik putem grafikona gleda vrednosti u poslednjih sat vremena; svaki put kada pristigne nova vrednost ona treba automatski biti prikazana na grafikonu; implementirati dobavljanje najnovijih vrednosti putem WebSocket protokola;
- Pregled istorijskih vrednosti - korisnik može da izabere vremenski period u kom želi da vidi vrednosti na grafikonu (npr. poslednjih 6h, poslednjih 12h, poslednja 24h, poslednjih nedelju dana, mesec dana ili da izabere datume od-do pri čemu razlika između ta dva datuma ne sme da bude veća od mesec dana).

## 4.11. Upravljanje PKA uređajem - Klima uređaj

Implementirati funkcionalnosti koje podržava klima uređaj (sekcija 3). Za uređaj je moguće videti istorijat izvršenih akcija (izuzetno je bitan podatak koji korisnik je izvršio akciju). Korisnik može da vidi sve akcije izvršene nad nekim uređajem (uključujući i one koje uređaj samostalno izvrši na osnovu zakazivanja) i da ih filtrira prema tome ko je izvršio akciju i u kom periodu (datumi od-do).

## 4.12. Upravljanje PKA uređajem - Veš mašina

Implementirati funkcionalnosti koje podržava veš mašina (sekcija 3). Za uređaj je moguće videti istorijat izvršenih akcija (izuzetno je bitan podatak koji korisnik je izvršio akciju). Korisnik može da vidi sve akcije izvršene nad nekim uređajem (uključujući i one koje uređaj samostalno izvrši

bez da korisnik nešto naredi) i da ih filtrira prema tome ko je izvršio akciju i u kom periodu (datumi od-do).

#### 4.13. Upravljanje SPA uređajem - Lampa

Implementirati funkcionalnosti koje podržava lampa (sekcija 3). Vrednosti osvetljenja koje lampa očitava potrebno je sačuvati u bazi podataka. Korisnik može da posmatra istorijske vrednosti - korisnik može da izabere vremenski period u kom želi da vidi vrednosti na grafikonu (npr. poslednjih 6h, poslednjih 12h, poslednja 24h, poslednjih nedelju dana, mesec dana ili da izabere datume od-do pri čemu razlika između ta dva datuma ne sme da bude veća od mesec dana).

#### 4.14. Upravljanje SPA uređajem - Kapija za vozila

Implementirati funkcionalnosti koje podržava kapija za vozila (sekcija 3). Korisnik može da u realnom vremenu posmatra događaje (akcije) koji se izvršavaju od strane kapije i da ih filtrira prema tome ko je izvršio akciju i u kom periodu (datumi od-do). Dobavljanje događaja radi prikaza akcija u realnom vremenu ostvariti pomoću WebSocket protokola.

#### 4.15. Upravljanje SPA uređajem - Sistem prskalice

Implementirati funkcionalnosti koji podržava sistem prskalice (sekcija 3). Za sistem je moguće videti istorijat izvršenih akcija (izuzetno je bitan podatak koji korisnik je izvršio akciju). Korisnik može da vidi sve akcije izvršene nad nekim sistemom (uključujući i one koje uređaj samostalno izvrši bez da korisnik nešto naredi) i da ih filtrira prema tome ko je izvršio akciju i u kom periodu (datumi od-do).

#### 4.16. Upravljanje VEU uređajem - Sistem solarnih panela

Implementirati funkcionalnosti koje podržava sistem solarnih panela (sekcija 3). Za sistem je moguće videti istorijat izvršenih akcija. Korisnik može da vidi sve akcije izvršene nad nekim sistemom i da ih filtrira prema tome ko je izvršio akciju i u kom periodu (datumi od-do). Posebno obratiti pažnju na implementaciju formule za računanje proizvedene energije i pravilnu simulaciju proizvedene energije (u toku dana kada je sunčano je veća proizvodnja, dok je noću ne proizvode).

#### 4.17. Upravljanje VEU uređajem - Kućna baterija

Implementirati funkcionalnosti koje podržava kućna baterija (sekcija 3). Platforma treba da prati potrošnju i proizvodnju električne energije unutar cele nekretnine. U slučaju da pametna kuća ima instaliran sistem solarnih panela, postojanje baterije pomaže čuvanju proizvedene energije i samim tim smanjenje troškova kupovine energije od elektrodistribucije. Korisnik može prati da potrošnju električne energije u celoj nekretnini u dva režima:

- Pregled vrednosti u realnom vremenu - korisnik putem grafikona gleda potrošnju energije od strane cele nekretnine u poslednjih sat vremena; svaki minut kada pristignu



nova očitavanja potrošne energije, ona treba automatski biti prikazana na grafikonu; implementirati dobavljanje najnovijih vrednosti putem WebSocket protokola;

- Pregled istorijskih vrednosti - korisnik može da izabere vremenski period u kom želi da vidi vrednosti potrošnje energije od strane cele nekretnine na grafikonu (npr. poslednjih 6h, poslednjih 12h, poslednja 24h, poslednjih nedelju dana, mesec dana ili da izabere datume od-do pri čemu razlika između ta dva datuma ne sme da bude veća od mesec dana).

## 4.18. Upravljanje VEU uređajem - Punjač za električna vozila

Implementirati funkcionalnosti koje podržava punjač za električna vozila (sekcija 3). Za punjač je moguće videti istorijat izvršenih akcija (izuzetno je bitan podatak koji korisnik je izvršio akciju). Korisnik može da vidi sve akcije izvršene nad nekim sistemom (uključujući i one koje uređaj samostalno izvrši bez da korisnik nešto naredi) i da ih filtrira prema tome ko je izvršio akciju i u kom periodu (datumi od-do).

# 5. Nefunkcionalni zahtevi

## 5.1. Serverske platforme

Za realizaciju projekta može se izabrati serverska platforma po želji. Neke od platformi mogu biti:

- Java + Spring Boot (koristi se na vežbama)
- Java + Play framework
- Java + Spark framework
- Python + Django
- Ruby on Rails
- .NET
- ...

## 5.2. Klijentske tehnologije

Za realizaciju klijentske aplikacije može se izabrati jedan od sledećih frontend tehnologija:

- Angular
- Vue
- React

## 5.3. Slanje email-a

Za slanje email-a nije obezbeđen poseban servis. Možete koristiti sopstveni email nalog. Email-ovi koje šalje vaša platforma treba da imaju jasno definisan template koji izgledom podseća na izgled vaše platforme.

## 5.4. API dizajn

API glavne aplikacije treba da bude dizajniran i dokumentovan u skladu sa [OpenAPI specifikacijom](#).

## 5.5. Perzistencija podataka

Sve podatke koji se kreiraju u toku rada platforme potrebno je trajno perzistirati. Podatke koje generišu pametni uređaji (poput vrednosti različitih merenja, promena stanja/statusa itd.) potrebno je čuvati u time series bazi podataka po izboru (npr. InfluxDb). Ostale podatke (poput informacija o korisnicima, nekretninama, pametnim uređajima itd.) potrebno je čuvati u relacionoj bazi podataka po izboru (npr. MySQL, MariaDB, PostgreSQL itd.).

## 5.6. Serviranje statičkog sadržaja

Iako mnoge serverske platforme imaju mogućnost da serviraju statički sadržaj i sadržaj koji korisnici upload-uju (poput slika), takav način serviranja sadržaja nije preporučljiv. Umesto toga, tehnologije poput nginx-a mogu da se koriste kao reverse proxy i da bi servirali statički sadržaj. Česta upotreba npr. nginx-a je serviranje statičkog sadržaja koji čini korisnički interfejs aplikacije (html, css, js itd.). U tom slučaju, ako je klijentska aplikacija izrađena pomoću nekog radnog okvira i/ili skupa biblioteka (npr. Angular, React, Vue), ona se pretvori u skup statičkog sadržaja (tj. izgradi za produkciju - *production build*) koji nginx servira.

## 5.7. Reverse proxy

Kada imate softversko rešenje koje ima više komponenti ili resursa koji su dostupni na različitim rutama, a pri čemu je potrebno definisati način pristupa, dobra praksa je koristiti reverse proxy (npr. nginx, Apache Traffic Server, HAProxy itd.). [Reverse proxy](#) može da reši i neke druge probleme sa kojima se možete susresti prilikom izrade projektnog zadatka.

## 5.8. Prikaz grafikona

Za prikaz grafikona na korisničkom interfejsu mogu se koristiti third party biblioteke.

## 5.9. Testiranje performansi sistema

Potrebno je izvršiti dva tipa testiranja performansi sistema:

1. Testiranje opterećenja sistema (*load testing*) za česte scenarije korišćenja;
2. Testiranje performansi sistema usled povećanja broja uređaja (simulatora) koji komuniciraju sa platformom.

Testiranje opterećenja sistema za česte scenarije korišćenja - Svaki student treba da izabere 10 različitih scenarija korišćenja koje je implementirao u sklopu funkcionalnosti koje mu pripadaju (jedan scenario bi npr. bio da se korisnik prijavi u sistem i registruje novu nekretninu). Za svaki od scenarija treba da se provere performanse sistema i ispita njihova promena sa povećanjem količine podataka sa kojim aplikacija rukuje i povećanjem broja korisnika koji istovremeno pokušavaju da izvrše isti scenario (npr. veliki broj korisnika istovremeno pokuša da se prijavi na sistem i registruje novu nekretninu). Potrebno je istražiti koji alati se mogu koristiti za ovaj vid testiranja (JMeter, Locust, Gatling itd.) i koristiti jedan i/ili više njih kako bi se izvršilo potpuno testiranje izabranih scenarija korišćenja.

Testiranje performansi sistema prilikom rada sa simulatorima - Svaki student treba da testira performanse sistema za svaku funkcionalnost koju podržavaju simulatori pametnih uređaja koje je taj student implementirao. Potrebno je identifikovati sve delove sistema u kojima može da se desi problem sa performansama prilikom povećanja broja uređaja koji komuniciraju sa platformom u sklopu određene funkcionalnosti (npr. proveriti kako se ponaša sistem sa povećanjem broja senzora ambijentalnih uslova koji istovremeno šalju veliki broj očitanih vrednosti).

Potrebno je da svaki student sastavi izveštaj (u slobodnoj formi) kako je izvršio testiranje svog dela sistema. Dokument može da sadrži tekst, slike i grafikone koji dokumentuju proces testiranja. U slučaju da prilikom testiranja uočite da određeni delovi sistema imaju loše performanse, potrebno je uložiti napore u unapređenje tih delova sistema i to bi trebalo takođe dokumentovati u izveštaju. Konačnu verziju izveštaja konvertovati u **PDF dokument** i okačiti na udaljeni git repozitorijum.

Kod oba tipa testiranja savetuje se da platformu (sa svim njenim komponentama) pokrenete na jednom računaru, a da na drugom računaru (ili više njih) pokrenete alate za testiranje performansi i/ili simulatore pametnih uređaja. Računari bi trebalo da se nalaze u istoj mreži kako bi mogli međusobno komunicirati. Na ovaj način ćete platformi dozvoliti da koristi što više resursa koje ima računar na kom je platforma pokrenuta, dok će drugi računar (ili više njih) služiti za pokretanje što više aplikacija koje će komunicirati sa platformom.

## 5.10. Kontrola verzija i kontinualni rad

Neophodno je koristiti **Git** za kontrolu verzija i repozitorijum mora biti na GitLab-u. Asistentskom nalogu Vanja-Mijatov ([vanja.mijatov@uns.ac.rs](mailto:vanja.mijatov@uns.ac.rs)) i dati pristup repozitorijumu i ulogu **Maintainer**-a. Takođe, potrebno je u **README.md** napisati koje tehnologije su korišćene, ko je učestvovao u izradi rešenja i navesti tačno uputstvo za pokretanje projekta i priložiti skriptu za popunu baze testnim podacima.

**Napomena:** Na GitLab-u će se kroz commit-ove pratiti kontinualni rad. Projekti koji budu kačeni na GitLab u svega nekoliko komitova neće biti podložni ocenjivanju. Nije neophodno raditi na projektu prema nekoj strogoj metodologiji (npr. Scrum) niti voditi tablu sa zadacima (kanban

board ili nešto slično), ali je neophodno da se nove verzije koda šalju redovno u toku semestra na udaljeni repozitorijum.

## 6. Ocenjivanje

### 6.1. Bodovanje - opšte

Projekat ukupno nosi **50 bodova**. Minimalan broj bodova koji je potreban za polaganje projekta je **26**. Projekat se ocenjuje u 3 etape:

1. Prva kontrolna tačka - nosi 5 bodova i izvodi se u terminu vežbi sredinom novembra
2. Druga kontrolna tačka - nosi 10 bodova i izvodi se u terminu vežbi sredinom decembra
3. Finalna odbrana - nosi 35 bodova i izvodi se u 3 roka:
  - a. Januarsko-februarski rok
  - b. Junski rok
  - c. Septembarski rok

U sklopu odbrane projekta u istom terminu održaće se i usmeni ispit koji takođe nosi **50 bodova**.

### 6.2. Podela zadataka u timu

#### Student 1

4.1, 4.2, 4.7, 4.10, 4.11, 4.12

#### Student 2

4.3, 4.4, 4.8, 4.13, 4.14, 4.15

#### Student 3

4.5, 4.6, 4.9, 4.16, 4.17, 4.18

### 6.3. Prva kontrolna tačka

Prva kontrolna tačka će se održati u terminu vežbi u nedelji od 20.11. do 24.11. i vrednuje se sa 5 bodova. Za ovu kontrolnu tačku potrebno je implementirati sledeće funkcionalnosti:

- Student 1 - 4.1. i 4.2.
- Student 2 - 4.3. i 4.4.
- Student 3 - 4.5. i 4.6. (demonstrirati funkcionalnost 4.6 na jednom pametnom uređaju po izboru; nije potrebno implementirati sve funkcionalnosti tog uređaja već samo demonstrirati da radi komunikacija sa platformom u oba smera, tj. da simulator šalje

poruke platformi i platforma simulatoru)

## 6.4. Druga kontrolna tačka

Druga kontrolna tačka će se održati u terminu vežbi u nedelji od 18.12. do 22.12. i vrednuje se sa 10 bodova. Za ovu kontrolnu tačku potrebno je implementirati sledeće funkcionalnosti:

- Student 1 - 4.10. i 4.11.
- Student 2 - 4.13. i 4.14.
- Student 3 - 4.16. i 4.17.

## 6.5. Finalna odbrana

Implementacija svih funkcionalnosti je preduslov za uspešno polaganje projekta. Da biste osvojili bodove za implementaciju neke funkcionalnosti nije bitno samo da li radi funkcionalnost, već i kako radi. Kako aplikacija treba da podrži rad sa velikim brojem korisnika, nekretnina i uređaja, potrebno je da za svaku funkcionalnost nađete optimalno rešenje. To bi značilo da, ako je očekivan sporiji rad sistema ili njegovih delova zbog dobavljanja ili obrade veće količine podataka, koristeći neke od metoda obrade i keširanja podataka ubrzate rad sistema. Keširanje podataka se često radi na strani serverske aplikacije (obično uz korišćenje baze za keširanje poput redis-a). Zatim, u slučaju prikazivanja veće količine podataka u vidu tabela ili grafikona, potrebno je primeniti dobru praksu koja podrazumeva paginaciju, downsampling i slične pristupe. U tu svrhu dozvoljeno je koristiti gotova rešenja (aplikacije, biblioteke, dodatne baze podataka).

Korisničko iskustvo (UX) je takođe značajna stavka prilikom ocenjivanja projekta. Nije toliko bitno koliko neka stranica lepo izgleda (boje, raspored elemenata), već da se intuitivno koristi i da akcije koje zahtevaju dobavljanje ili obradu veće količine podataka ne učine pretraživač da se zamrzne ili ugasi. Email-ovi koje šalje vaša platforma treba da imaju jasno definisan template koji izgledom podseća na izgled vaše platforme.

Na finalnoj odbrani će biti provereno ispunjavanje i nefunkcionalnih zahteva sa akcentom na zahteve iz sekcija od 5.3. do 5.10. pri čemu je potrebno posebnu pažnju posvetiti zahtevu 5.9. (zahtev koji nosi najveći deo ocene).

## 7. Akademska čestitost

Akademska čestitost podrazumeva samostalno pisanje radova (teksta, programskog koda i slično) uz striktno poštovanje tuđih autorskih prava. Ovaj pojam i obaveza su sastavni deo Zakona o visokom obrazovanju:

[http://www.parlament.gov.rs/upload/archive/files/lat/pdf/predlozi\\_zakona/3048-14\\_Lat.pdf](http://www.parlament.gov.rs/upload/archive/files/lat/pdf/predlozi_zakona/3048-14_Lat.pdf)

Više o ovoj temi možete pronaći, na primer, na sledećim linkovima:

- [https://en.wikipedia.org/wiki/Academic\\_honor\\_code](https://en.wikipedia.org/wiki/Academic_honor_code)
- <https://www.akademsko-pisanje.uniri.hr/akademsko-cestitost/>

U okviru ovog predmeta to podrazumeva da studenti samostalno pišu sopstveni rad. Pomoć drugih studenata u obliku direktno preuzetih delova teksta ili programa nije dozvoljena. **Kako se ovaj projekat radi u timu, odgovornost za nepridržavanje principa akademske čestitosti snose svi članovi tima.**

Sistemi za otkrivanje plagijarizma su vremenom postali prilično efektivni. Više informacija se može naći ovde:

<https://theory.stanford.edu/~aiken/moss/>

ili recimo ovde:

<https://github.com/genchang1234/How-to-cheat-in-computer-science-101>

Kako će se na ovom predmetu prilikom analize projekata pristupati originalnim udaljenim repozitorijumima, treba imati u vidu da se na njima vidi kompletna istorija izmena u toku rada na projektu. Okolnosti u kojima se na repozitorijumu pojavljuje nešto neregularno odmah dovode u sumnju dati studentski tim.

Zajedničko učenje studenata ili rad na projektu je lepa praksa, ali to ne podrazumeva da se do rezultata koji se ocenjuje dolazi zajednički.