WUN

WhatsUpNeighbours

Arhitekturni projekat

Verzija 3.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 06.05.2021. | 1.0 | Inicijalna verzija | Anastasija, Emilija, Milica |
| 08.05.2021. | 2.0 | Dorađivanje | Anastasija, Emilija, Milica |
| 09.05.2021. | 3.0 | Provera dokumenta | Anastasija, Emilija, Milica |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 7

7. Pogled na logičku arhitekturu sistema 10

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 11

7.1.1 Korisnički interfejs 11

7.1.2 Aplikaciona logika 11

7.1.3 Pristup podacima 11

7.1.4 Javascript (React) 11

7.1.5 C# 11

7.1.6 Azure Data Studio 11

8. Pogled na procese 12

8.1 Procesi 12

8.1.1 Web čitač 12

8.1.2 Web server 12

8.1.3 Javascript-CGI 12

8.1.4 Azure Server 12

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 13

9.2 Web server 13

9.3 DBMS server 13

10. Pogled na implementaciju sistema 13

10.1 Model domena 13

10.2 Šema baze podataka 14

10.3 Komponente sistema 15

10.3.1 Komponente korisničkog interfejsa 15

10.3.2 Komponente aplikacione logike 15

10.3.3 Komponente za pristup podacima 16

11. Performanse 17

12. Kvalitet 17

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture WUN aplikacije.

# Opseg dokumenta

Dokument se odnosi na WUN web aplikaciju koja će biti razvijena od strane MuffinsFromCanteen. WUN predstavlja skraćenicu za WhatsUpNeighbours. Namena sistema je efikasno informisanje, komuniciranje medju stanarima zgrade kao i pregled statusa uplate zajedničkih troškova.

# Reference

Spisak korišćene literature:

1. WUN – Predlog projekta, SWE-WUN-01, V1.0, 2021, MuffinsFromCanteen.
2. WUN – Planirani raspored aktivnosti na projektu, V1.0, 2021, MuffinsFromCanteen.
3. WUN – Plan realizacije projekta, V1.0, 2021, MuffinsFromCanteen.
4. WUN – Vizija sistema, V1.0, 2021, MuffinsFromCanteen.
5. WUN – Specifikacija zahteva, V1.0, 2021, MuffinsFromCanteen.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. WUN web aplikacija će biti implementiran kao Web aplikacija zasnovana na Javascript jeziku u React okruženju i Azure SQL bazi podataka.
2. Klijentski deo WUN ortala će biti optimizovan za sledeće Web čitače: Google Chrome, Opera, Microsoft Edge, kao i Firefox (Mozilla).
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

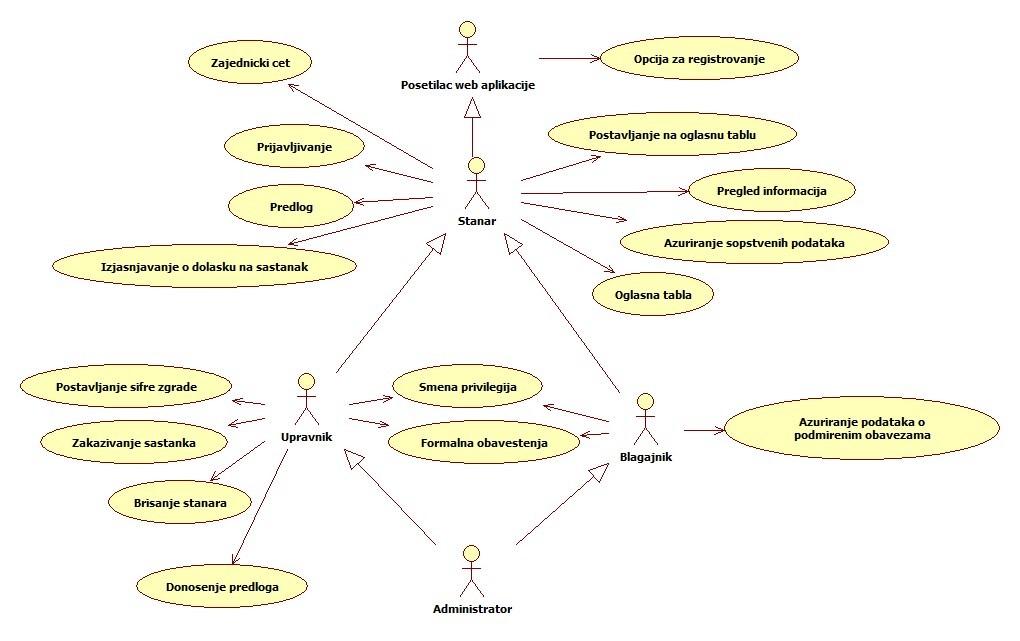
Slučajevi korišćenja WUN portala su:

* *Pregled informacija*
  + Pregled zakazanih sastanaka
  + Pregled oglasne table
    - Pretrazivanje
    - Sortiranje
    - Filtriranje
  + Pregled liste stanara
  + Pregled zajednickog četa
  + Pregled plaćenih obaveza
  + Pregled obaveštenja
  + Pregled predloga
  + Pregled sopstvenih podataka
* Prijavljivanje
* Ažuriranje sopstvenih podataka
* Izjasnjavanje o dolasku na sastanak
* Postavljanje na oglasnu tablu
* Predlog
  + - Postavljanje predloga
    - Glasanje za predlog
* Donošenje predloga
* Brisanje stanara
* Zakazivanje sastanaka
* Postavljanje šifre zgrade
* Smena privilegija
* Formalna obaveštenja
* Ažuriranje podataka o podmirenim obavezama

Ove slučajevi korišćenja mogu da iniciraju posetilac web aplikacije, stanari, upravnik, blagajnik ili administrator.

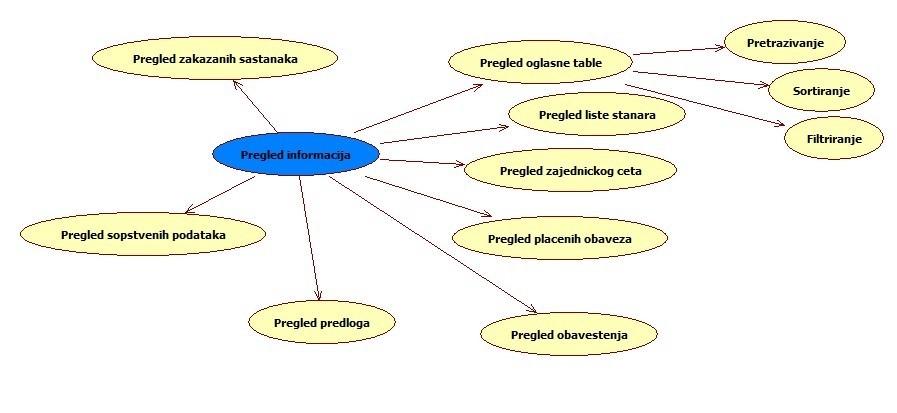
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja WUN portala prikazan je na sledećoj slici:

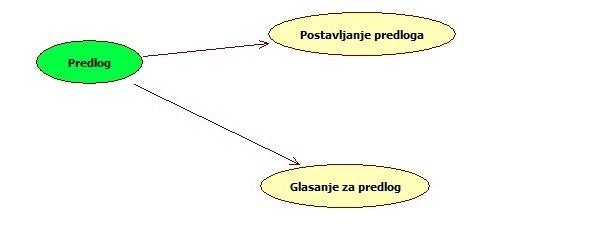


Slučajevi korišćenja *pregled informacija* i *predloga,* obuhvataju složenije radnje koje se mogu razložiti dalje na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *pregled informacija* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *predloga* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

## Pregled zakazanih sastanaka

Kratak opis: Stanari imaju uvid sve buduće sastanake zakazane od strane upravnika.

Akteri koji iniciraju slučaj korišćenja: Stanar, Blagajnik, Upravnik i Administrator.

* + 1. Pregled sopstvenih podataka

Kratak opis: Stanari imaju mogućnost da pregledaju sve lične podatake koji su na njihovom profilu.

Akteri koji iniciraju slučaj korišćenja: Stanar, Blagajnik, Upravnik i Administrator.

* + 1. Pregled predloga

Kratak opis: Stanari imaju uvid u liste predloga postavljenim od strane drugih stanara.

Akteri koji iniciraju slučaj korišćenja: Stanar, Blagajnik, Upravnik i Administrator.

## Pregled oglasne table

Kratak opis: Prikaz sekcije aplikacije sa svim obaveštenjima namenjenim stanarima.

Akteri koji iniciraju slučaj korišćenja: Stanari, Upravnik, Blagajnik, Administrator.

## Pregled zajednickog chat-a

Kratak opis: Prikaz sekcije aplikacije sa svim porukama koje stanari međusobno razmenjuju.

Akteri koji iniciraju slučaj korišćenja: Stanari, Upravnik, Blagajnik, Administrator.

## Pregled liste stanara

Kratak opis: Prikaz sekcije aplikacije sa listom svih stanara stambene zgrade.

Akteri koji iniciraju slučaj korišćenja: Stanari, Upravnik, Blagajnik, Administrator.

## Pregled plaćenih obaveza

Kratak opis: Prikaz sekcije aplikacije sa statusom podmirenih obaveza prema zgradi prijavljenog stanara.

Akteri koji iniciraju slučaj korišćenja: Stanari, Upravnik, Blagajnik, Administrator.

## Pregled obavestenja

Kratak opis: Prikaz sekcije aplikacije sa svim obaveštenjima namenjenim stanarima.

Akteri koji iniciraju slučaj korišćenja: Stanar, Upravnik, Blagajnik, Administrator.

## Prijavljivanje

Kratak opis: Prijavljivanje korisnika na web aplikaciju u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: Stanar, Upravnik, Blagajnik i Administrator.

## Registracija

Kratak opis: Registracija korisnika na web aplikaciju u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: Stanar, Upravnik, Blagajnik.

## Ažuriranje podataka o članu

Kratak opis: Ažuriranje podataka o sebi od strane prijavljenog stanara zgrade.

Akteri koji iniciraju slučaj korišćenja: Stanar, Upravnik, Blagajnik i Administrator.

## Brisanje postojećeg člana

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg stanara zgrade.

Akteri koji iniciraju slučaj korišćenja: Upravnik i Administrator.

## Ažuriranje podataka o podmirenim obavezama prema zgrad

Kratak opis: Azuriranje podataka nakon placanja mesečnih trškova.

Akteri koji iniciraju slučaj korišćenja: Blagajnik i Administrator.

## Oduzimanje privilegija upravnika/blagajnika jednom stanaru i dodela drugom

Kratak opis: Pri izboru novog upravnika/blagajnika, stari upravnik/blagajnik mora dodeliti novom svoje privilegije

Akteri koji iniciraju slučaj korišćenja: Upravnik, Blagajnik i Administrator.

## Formalna obaveštenja

Kratak opis: Poseban deo web aplikacije na kome se postavljaju formalna obaveštenja upravnika i blagajnika.

Akteri koji iniciraju slučaj korišćenja: Upravnik, Blagajnik i Administrator.

## Sortiranje, filtriranje, pretraživanje stanara

Kratak opis: Prikaz stranice sa informacijama svih članova web aplikacije, tj. svih stanara zgrade.

Akteri koji iniciraju slučaj korišćenja: Stanar, Upravnik, Blagajnik i Administrator.

## Organizovanje sastanaka

Kratak opis: Po potrebi, upravnik zakazuje sastanke (savete) stanara. Nakon zakazivanja sastanka, stanari se izjašnjavaju da li žele da prisustvuju sastanku.

Akteri koji iniciraju slučaj korišćenja: Upravnik i Administrator.

## Izjašnjavanje o dolasku na sastanak

Kratak opis: Nakon sto upravnik zakaže sastanak, on je vidljiv svim stanarima koji se mogu izjasniti o tome da li će prisustvovati ili ne.

Akteri koji iniciraju slučaj korišćenja: Stanar, Blagajnik, Upravnik i Administrator.

## Zajednički chat

Kratak opis: Zajednički chat služi za direktnu komunikaciju svih stanara o bitnim i manje bitnim temama vezanim za dešavanja u stambenoj zgradi.

Akteri koji iniciraju slučaj korišćenja: Stanar,Upravnik, Blagajnik i Administrator.

## Postavljanje predloga

Kratak opis: Svaki stanar ima pravo da iznese predlog za koji potom mogu glasati ostali stanari

Akteri koji iniciraju slučaj korišćenja: Stanar,Upravnik, Blagajnik i Administrator.

## Glasanje za predlog

Kratak opis: Svi postojeći predlozi za koje nije doneta konačna odluka od strane upravnika, dostupni su drugim stanarima za izjašnjavanje o njemu (glasanje za i protiv).

Akteri koji iniciraju slučaj korišćenja: Stanar,Upravnik, Blagajnik i Administrator.

## Usvajanje predloga

Kratak opis: Odluku da li će postojeći predlog stanara biti usvojen ili ne donosi upravnik zgrade. Kada upravnik donese odluku, ukida mogućnost glasanja o predlogu tako što ga uklanja iz liste, a odluku o usvajanju postavlja u sekciji za obaveštenja.

Akteri koji iniciraju slučaj korišćenja: Upravnik i Administrator.

## Oglasna tabla

Kratak opis: U okviru web aplikacije postoji oglasna tabla na koju svi stanari mogu postavljati obaveštenja, upozorenja, primedbe koje se tiču zgrade.

Akteri koji iniciraju slučaj korišćenja: Stanar, Upravnik, Blagajnik i Administrator.

## Postavljanje šifre zgrade

Kratak opis: Upravnik ima mogućnost postavljanja i menjanja sifre zgrade.

(Namena ove šifre: Pri registraciji na aplikaciju, potrebno je uneti pored korisničke šifre i šifru datu od strane upravnika koja sluzi za validaciju pri registraciji. )

Akteri koji iniciraju slučaj korišćenja: Upravnik i Administrator.

## Pregled zakazanih sastanaka

Kratak opis: Stanari imaju mogućnost u uvid svih budućih sastanaka zakazanih od strane upravnika.

Akteri koji iniciraju slučaj korišćenja: Stanar, Blagajnik, Upravnik i Administrator.

## Pregled sopstvenih podataka

Kratak opis: Stanari imaju mogućnost u uvid ličnih podataka koji se odnose na njihov profil. Akteri koji iniciraju slučaj korišćenja: Stanar, Blagajnik, Upravnik i Administrator.

## Pregled predloga

Kratak opis: Stanari imaju mogućnost u uvid liste predloga postavljenim od strane drugih stanara. Akteri koji iniciraju slučaj korišćenja: Stanar, Blagajnik, Upravnik i Administrator.

## Pregled zajednickog chat-a

Kratak opis: Prikaz sekcije aplikacije sa svim porukama koje stanari međusobno razmenjuju.

Akteri koji iniciraju slučaj korišćenja: Stanari, Upravnik, Blagajnik, Administrator.

## Pregled liste stanara

Kratak opis: Prikaz sekcije aplikacije sa listom svih stanara stambene zgrade.

Akteri koji iniciraju slučaj korišćenja: Stanari, Upravnik, Blagajnik, Administrator.

## Pregled plaćenih obaveza

Kratak opis: Prikaz sekcije aplikacije sa statusom podmirenih obaveza prema zgradi prijavljenog stanara.

Akteri koji iniciraju slučaj korišćenja: Stanari, Upravnik, Blagajnik, Administrator.

## Pregled obavestenja

Kratak opis: Prikaz sekcije aplikacije sa svim obaveštenjima namenjenim stanarima.

Akteri koji iniciraju slučaj korišćenja: Stanar, Upravnik, Blagajnik, Administrator.

.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

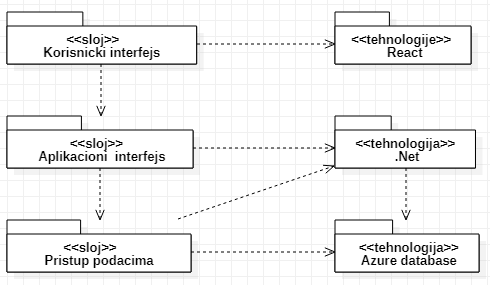
Logički pogled na WUN portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži React komponente koje predstavljaju forme, menu-je i tabele koje realizuju grafički dizajn preko kojeg korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema zadužen za funkcionalnosti sistema koji se razvija.

Paket *Pristup podacima* predstavlja interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs portala i sadrži React komponente koje predstavljaju forme, menu-je i tabele koje realizuju grafički dizajn preko kojeg korisnici sistema komuniciraju sa sistemom. Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i Javascript.

### Aplikaciona logika

sloj

Sloj aplikacione logike predstavlja srednji sloj sistema zadužen za funkcionalnosti sistema koji se razvija.

Ovaj sloj zavisi od sloja za pristup podacima i C# paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i predstavlja interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka. Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa C# i MySQL baza podataka.

### Javascript (React)

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### C#

tehnologija

.NET aplikacija koja je napisana u C# jeziku koja predstavlja serverski deo web aplikacije i sadrži model podataka, kao i metode za komuniciranje sa frontend-om. //?????

### Azure Data Studio

DBMS

Azure Data Studio predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju WUN portala.

# Pogled na procese

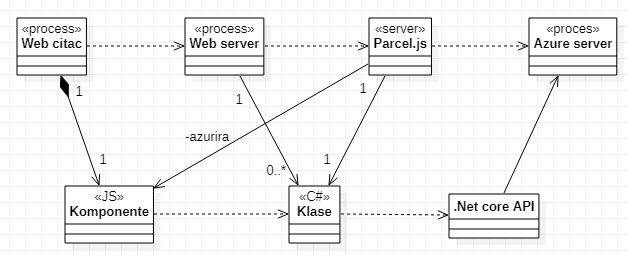
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Web aplikacije zasnovane na Javascript-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta Javascript Web aplikacije nije potrebno voditi računa o načinu rada Web servera i načinu izvršavanja skripti.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje WUN portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju WUN portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na Javascript-u i Azure bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz React komponenta. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje neograničeni broj React komponenti.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana stranica JS script , Web server inicira izvršenje JS-CGI procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj JS-CGI procesa.

### Parcel.js

Pokreće razvojni server, koji automatski obnavlja aplikaciju dok se nešto menja i podržava brzu zamenu modula za brzi razvoj. Ali mi imamo svoj server, tako da Parcel pokrećemo u režimu praćenja. Ovim se i dalje automatski obnavljaj aplikacija kada ima promena i podržava zamenu modula za vruće promene, ali ne pokreće web server.

#### Komponente

Ove komponente su su deo React aplikacije, koje služe da se izvrsava kod u web browser-u.

#### C#

Komunicira sa .Net API-jem kako bi pribavo željene podatke iz baze podataka

#### .Net Core API

.Net API sadrzi metode koje služe za manipulaciju nad podacima iz baze.

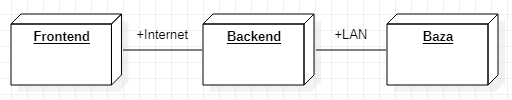
Azure Server

Azure Server je proces koji sluzi kao sistem za upravljanje bazama podataka. Baza je sloj perzistencije koji cuva podatke. Ovaj proces može konkurentno da izvrši pretraživanje baze podataka i vrati željene rezultate.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja WUN portala.



## Klijent

Pristup PeNcIL portal se obavlja preko klijentskih računara na kojima se izvršava Web čitač. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi PHP-CGI koji vrše obradu zadatih PHP skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server

DBMS server je računar na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

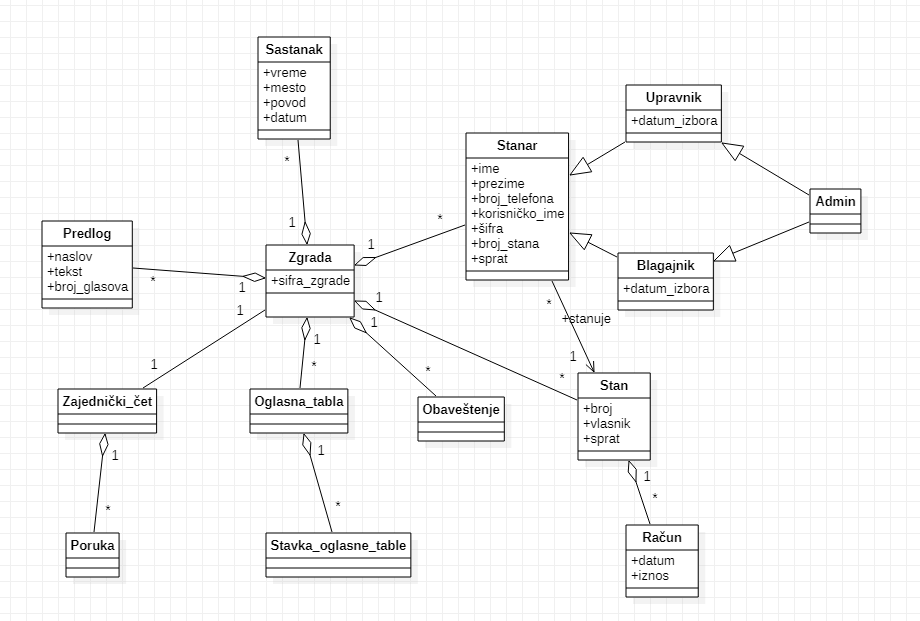
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju WUN aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

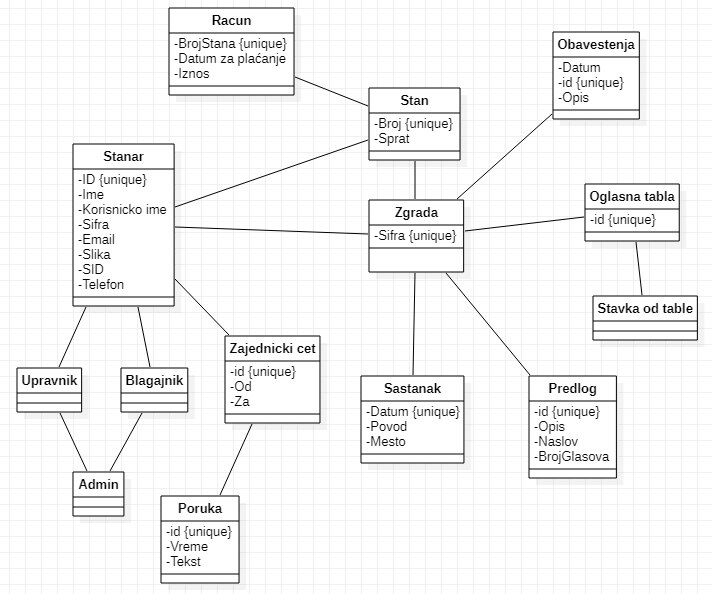
Model domena za koji se WUN portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (Javascript skripti) koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu.



## Komponente sistema

Komponente sistema WUN portala su React komponente čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je React kompoenta prikazana kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

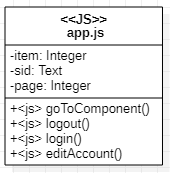
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



Komponenta **Komponenta.js**  implementira komponentu web aplikacije čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **Komponenta.css** predstavlja opis stilova za pojedine React komponente koji se javljaju na različitim stranicama.

Parametri koji utiču na izbor komponente ilustrovani su sledećim dijagramom klasa:

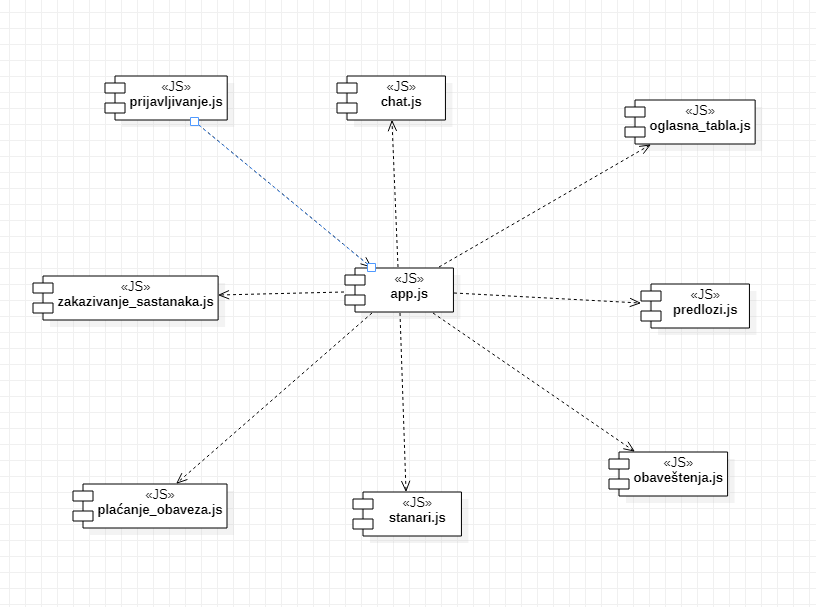


Značenje atributa je sledeće:

* page – interni identifikator stranice (oglasna tabla, sastanci, predlozi, troskovi,...)
* item – identifikator stavke koja se detaljno prikazuje na stranici (određeni stanar, satstanak, predlog...)
* sid – identifikator sesije kada je korisnik ulogovan

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.js** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:

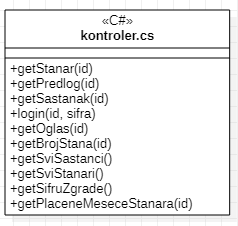


### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u metode kontrolerske klase. Svaka metoda ima jedinstvenu rutu pomoću koje React komponente fetch-uju zahtev za pristup podacima u bazi.



Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* getStanar-vraća stanara sa datim id-jem
* getPredlog-vraća predlog sa datim id-jem
* getSastanak-vraća sastanak sa datim id-jem
* login-loguje korisnika i vraća SID
* getOglas-vraćaoglas sa datim id-jem
* getBrojStana-vraća broj stana sa datim id-jem
* getSviSastanci-vraća listu svih sastanaka do sada
* getSviStanari-vraća listu svih stanara u zgradi
* getSifruZgrade-vraća trenutnu šifru zgrade
* getPlaceneMeseceStanara-vraća sve plaćene mesece stanara sa datim id-jem

# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će da podrži do 1000 simultanih pristupa korisnika portalu.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. WUN web aplikacija će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 15%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.