RoomRover

Web aplikacija za rezervaciju hotelskog smeštaja

Detaljni arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 23.11.2023. | 1.0 | Inicijalna verzija | Amina |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 8

6.2.1 Izbor jezika 8

6.2.2 Registacija 8

6.2.3 Pretraga hotelskog smeštaja po datumu boravka 8

6.2.4 Pretraga hotelskog smeštaja na osnovu cene 9

6.2.5 Pretraga hotelskog smeštaja na osnovu broja gostiju 9

6.2.6 Pretraga hotelskog smeštaja na osnovu destinacije 9

6.2.7 Pregled informacija o hotelu 9

6.2.8 Rezervacija hotelskog smeštaja 9

6.2.9 Pregled rezervacija 9

6.2.10 Prijavljivanje 9

6.2.11 Otkazivanje rezervacije 9

6.2.12 Pregled recenzija i ocena hotela 9

6.2.13 Unos ocene i recenzije za hotel 9

6.2.14 Povlastice za prijavljene korisnike 10

6.2.15 Ažuriranje podataka o prijavljenom korisniku 10

6.2.16 Dodavanje novog hotelskog smeštaja 10

6.2.17 Brisanje postojećeg hotelskog smeštaja 10

6.2.18 Ažuriranje osnovnih podataka o dostupnim hotelskim smeštajima 10

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 HTML 11

7.1.5 ReactJS 11

7.1.6 Microsoft SQL Server Management Studio 12

8. Pogled na procese 12

8.1 Procesi 12

8.1.1 Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju RoomRover web aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ReactJs-u i Microsoft SQL Server Management Studio bazi podataka. 12

8.1.2 Web čitač 12

8.1.3 Web server 12

8.1.1 Microsoft SQL Server Management Studio 13

9. Pogled na raspoređivanje sistema 13

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 16

10.3.3 Komponente za pristup podacima 17

11. Performanse 18

12. Kvalitet 18

Detaljni arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture RoomRover web aplikacije.

# Opseg dokumenta

Dokument se odnosi na RoomRover web aplikaciju koja će biti razvijena od strane RoverTeam-a. RoomRover predstavlja skraćenicu za Web aplikaciju za rezervaciju hotelskog smeštaja. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za rad jedne pretraživačke stranice.

# Reference

Spisak korišćene literature:

1. RoomRover – Predlog projekta, RRT-RoomRover-01, V1.0, 2023, RoverTeam.
2. RoomRover – Planirani raspored aktivnosti na projektu, V1.0,2023, RoverTeam.
3. RoomRover – Plan realizacije projekta, V1.0, 2023, RoverTeam.
4. RoomRover – Vizija sistema, V1.0, 2023, RoverTeam.
5. RoomRover – Specifikacija zahteva, V1.0, 2023, RoverTeam.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, 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. RoomRover web aplikacija će biti implementiran kao Web aplikacija zasnovana na ReactJS skripting jeziku i Microsoft SQL Server Management Studio bazi podataka [4].
2. Klijentski deo RoomRover web aplikacije će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, Firefox (Mozilla), kao i GoogleChrome [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Klasifikaciju publikacija treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za turizam.

# Pogled na slučajeve korišćenja

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

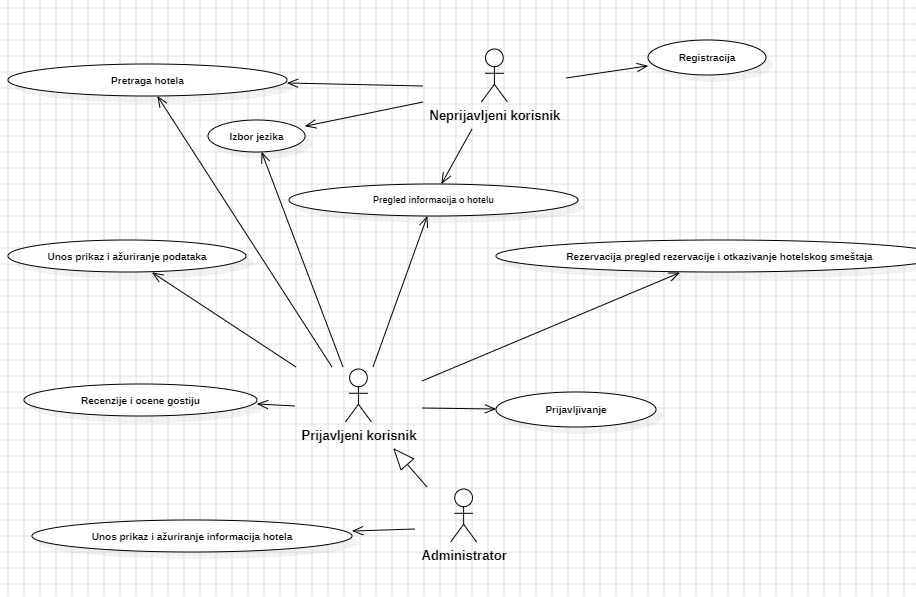
Slučajevi korišćenja RoomRover web aplikacije su:

* Izbor jezika
* Registacija
* Pretraga hotelskog smeštaja
  + Pretraga hotelskog smeštaja po datumu boravka
  + Pretraga hotelskog smeštaja na osnovu destinacije
* Pregled informacija o hotelu
* Rezervacija, pregled rezervacije i otkazivanje rezervacije hotelskog smeštaja
  + Rezervacija hotelskog smeštaja
  + Pregled rezervacija
  + Otkazivanje rezervacije
* Pregled recenzija i ocena hotela
* Unos ocene i recenzije za hotel
* Prijavljivanje
* Ažuriranje podataka o prijavljenom korisniku
* Unos, prikaz i ažuriranje podataka hotela
  + Dodavanje novog hotelskog smeštaja
  + Brisanje postojećeg hotelskog smeštaja
  + Ažuriranje osnovnih podataka o dostupnim hotelskim smeštajima

Ove slučajevi korišćenja mogu da iniciraju neprijavljeni korisnik, prijavljeni korisnik ili administrator.

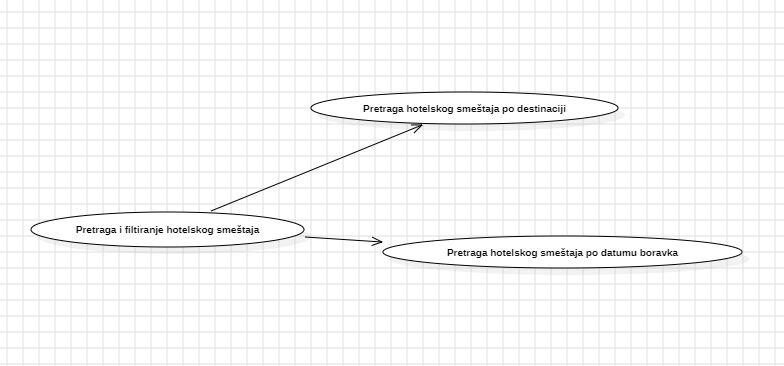
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja RoomRover web aplikacije prikazan je na sledećoj slici:

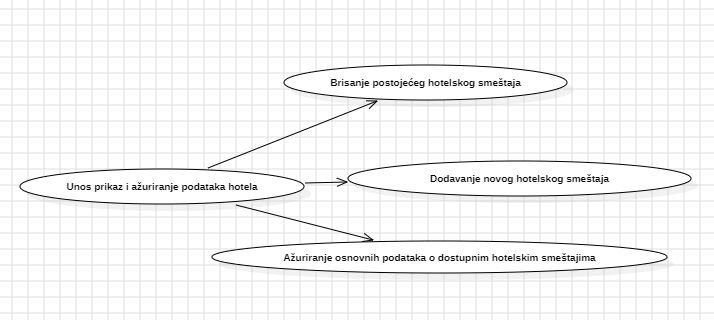


Slučajevi korišćenja *pretraga i filtriranje hotela, Unos prikaz i ažuriranje informacija hotela, Rezrvacija, pregled rezervacije i otkazivanje hotelskog smeštaja i Recenzije i ocene gostiju* obuhvataju složenije radnje koje se mogu razložiti na pojedinačne slučajeve korišćenja.

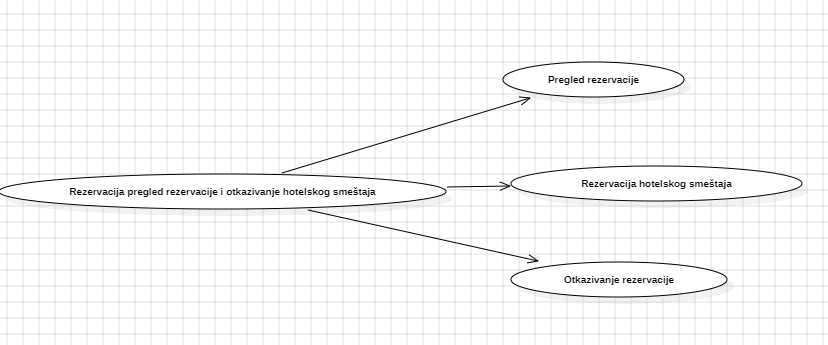
Detaljni UML dijagram za slučaj korišćenja *pretraga i filtriranje hotela* je prikazan na sledećoj slici:



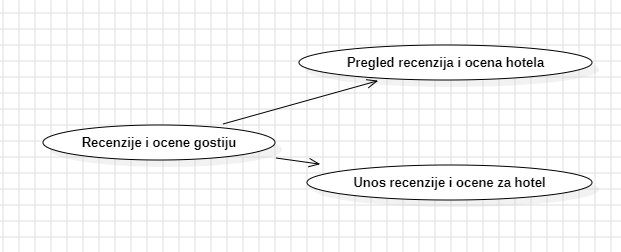
Detaljni UML dijagram za slučaj korišćenja *Unos prikaz i ažuriranje podataka hotela* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *Rezeracija, pregled rezervacije i otkazivanje rezervacije hotelskog smeštaja* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *Recenzije i ocene gostiju* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Izbor jezika

Kratak opis: Izbor između srpskog i engleskog jezika za prikaz informacija na web aplikaciji.

Akteri koji iniciraju slučaj korišćenja:Neprijavljeni korisnik, Prijavljeni korisnik, Administrator.

### Registacija

Kratak opis: Proces registracije omogućava posetiocu web aplikacije da postane prijavljeni korisnik, stvaranjem korisničkog naloga.

Akteri koji iniciraju slučaj korišćenja: Neprijavljeni korisnik.

### Pretraga hotelskog smeštaja po datumu boravka

Kratak opis: Korisnicima se omogućava pretraga hotela na osnovu datuma boravka, čime se pruža mogućnost pronalaženja smeštaja dostupnog u određenom vremenskom periodu.

Akteri koji iniciraju slučaj korišćenja: Posetilac web aplikacije (neprijavljeni korisnik), Prijavljeni korisnik, Administrator.

### Pretraga hotelskog smeštaja na osnovu destinacije

Kratak opis: Ova funkcionalnost omogućava korisnicima da pretražuju hotele na osnovu željene destinacije, pružajući mogućnost pronalaženja smeštaja u određenom geografskom području.

Akteri koji iniciraju slučaj korišćenja: Posetilac web aplikacije (neprijavljeni korisnik), Prijavljeni korisnik

### Pregled informacija o hotelu

Kratak opis: Ova funkcionalnost omogućava korisnicima da pregledaju detaljne informacije o određenom hotelu, uključujući opis, slike, ocene i dostupnost soba.

Akteri koji iniciraju slučaj korišćenja: Posetilac web aplikacije (neprijavljeni korisnik), Prijavljeni korisnik.

### Rezervacija hotelskog smeštaja

Kratak opis: Ova funkcionalnost omogućava korisnicima da rezervišu sobu u odabranom hotelu za određeni period.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Pregled rezervacija

Kratak opis: Ova funkcionalnost omogućava korisnicima, uključujući i administratore, da pregledaju postojeće i prošle rezervacije.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Administrator.

### Prijavljivanje

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

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Administrator.

### Otkazivanje rezervacije

Kratak opis: Ova funkcionalnost omogućava korisnicima da otkažu svoju postojeću rezervaciju, pružajući fleksibilnost u promeni planova.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Pregled recenzija i ocena hotela

Kratak opis: Ova funkcionalnost omogućava korisnicima, kako neprijavljenim posetiocima tako i prijavljenim korisnicima, da pregledaju recenzije i ocene drugih gostiju o odredjenom hotelu.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Administrator.

### Unos ocene i recenzije za hotel

Kratak opis: Ova funkcionalnost omogućava prijavljenim korisnicima da ostave svoje ocene i recenzije nakon boravka u određenom hotelu.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Ažuriranje podataka o prijavljenom korisniku

Kratak opis: Kreiranje korisničkog naloga za novog člana laboratorije.

Akteri koji iniciraju slučaj korišćenja: Šef laboratorije, Administrator.

### Dodavanje novog hotelskog smeštaja

Kratak opis: Dodavanje novog hotelskog smeštaja od strane administratora sistema.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje postojećeg hotelskog smeštaja

Kratak opis: Biranje postojećeg hotelskog smeštaja od strane administratora koji je taj hotelski smeštaj uneo.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje osnovnih podataka o dostupnim hotelskim smeštajima

Kratak opis: Izmena osnovnih podataka o dostupnim hotelskim smeštajima koji se prikazuju na početnoj stranici web aplikacije.

Akteri koji iniciraju slučaj korišćenja: 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.

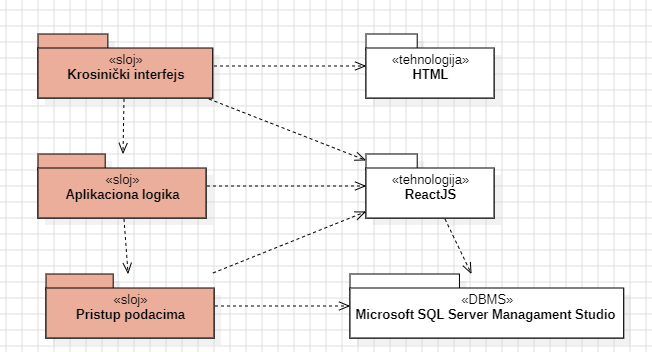
Logički pogled na RoomRover web aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, ReactJS skripte i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži ReactJs skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži ReactJs skripte koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Sloj korisničkog interfejsa u ReactJS arhitekturi realizuje vizuelni deo portala. Ovde se nalaze sve React komponente, HTML, multimedijalni sadržaji, i JavaScript skripte koje generišu korisnički interfejs. React komponente omogućavaju dinamičko ažuriranje korisničkog interfejsa na osnovu interakcije korisnika. Ovaj sloj zavisi od sloja aplikacione logike, kao i od React paketa.

### Aplikaciona logika

sloj

Sloj aplikacione logike u ReactJS arhitekturi sadrži JavaScript skripte koje realizuju funkcionalnosti karakteristične za domen primene portala. Ove skripte uspostavljaju vezu između korisničkog interfejsa (React komponenti) i sloja za pristup podacima. Ovaj sloj zavisi od sloja za pristup podacima i React paketa.

### Pristup podacima

sloj

Sloj za pristup podacima u ReactJS arhitekturi sadrži JavaScript skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka. Ovaj sloj ne zavisi od drugih slojeva, ali može biti zavisan od korišćenih HTTP biblioteka ili React konteksta.

### HTML

tehnologija

Tehnologija HTML-a i dalje definiše gradivne elemente stranica, ali u ReactJS kontekstu, HTML se generiše dinamički putem React komponenti. JS se koristi za opisivanje strukture korisničkog interfejsa unutar JavaScript koda. React komponente se potom prevode u HTML kod koji se prikazuje u web čitaču.

### ReactJS

tehnologija

ReactJS je JavaScript biblioteka koja omogućava izgradnju korisničkih interfejsa na strani klijenta. Njegove komponente omogućavaju efikasno upravljanje stanjem interfejsa i omogućavaju dinamičko ažuriranje korisničkog iskustva. React komponente mogu sadržavati HTML, stilove i JavaScript logiku, sve integrisane u jednom komponentnom modelu.

### Microsoft SQL Server Management Studio

DBMS

Microsoft SQL Server Management Studio predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju RoomRover web aplikacije.

# Pogled na procese

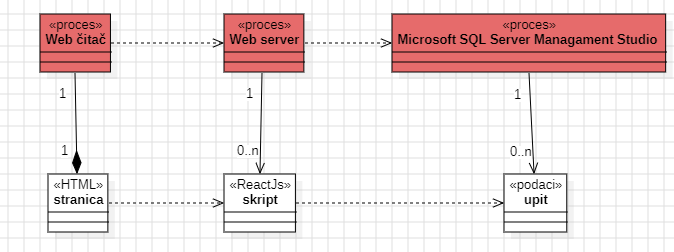
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na ReactJs-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta ReactJs 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 RoomRover projekta kao Web aplikacije.

## Procesi

### Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju RoomRover web aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ReactJs-u i Microsoft SQL Server Management Studio bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz ReactJS stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu ReactJS stranicu.

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću ReactJs stranicu na zahtev.

### Web server

ReactJS je JavaScript biblioteka koja se izvršava na strani klijenta (u pregledaču korisnika). Kada korisnik zahteva stranicu, web server isporučuje statičke fajlove (HTML, CSS, JavaScript) koji sadrže React komponente. Ove komponente se potom izvršavaju u pregledaču, gde se manipuliše virtuelnim DOM-om radi dinamičkog ažuriranja korisničkog interfejsa. ReactJS ne zahteva posebne procese na strani servera, i umesto toga, omogućuje interaktivne korisničke interfejse putem izvršavanja JavaScript koda direktno u pregledaču.

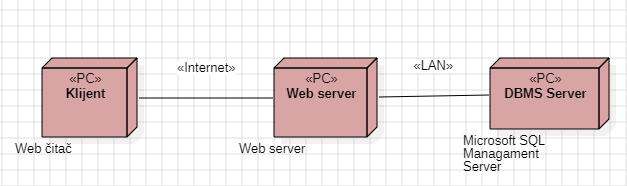
### Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio je proces koji izvršava funkcionalnost SQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# 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 RoomRover web aplikacije.



## Klijent

Pristup RoomRover web aplikaciji 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. 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 Microsoft SQL Managament 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).

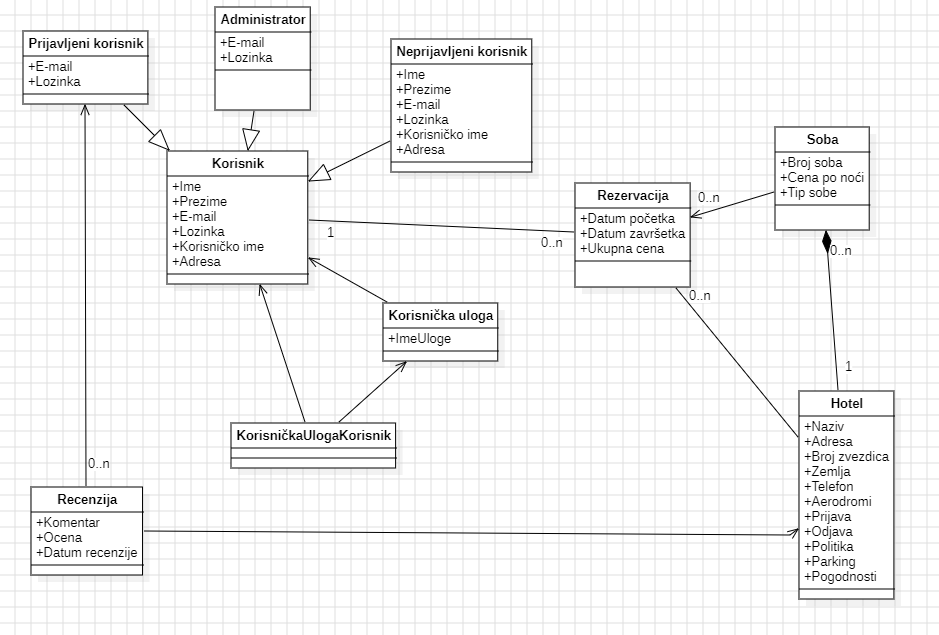
# Pogled na implementaciju sistema

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

## Model domena

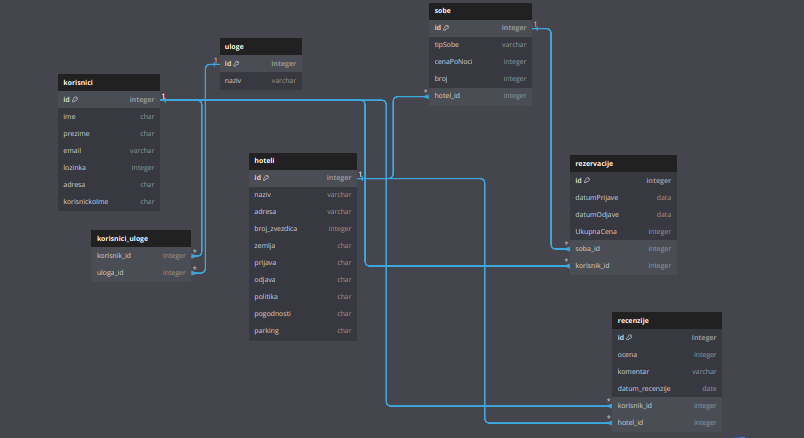
Model domena za koji se RoomRover web aplikacijom projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *Microsoft SQL Server Managment Studio-a*.

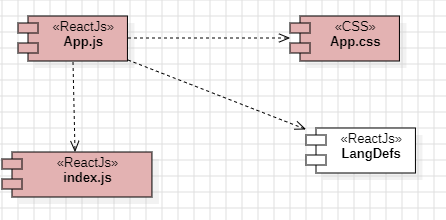


## Komponente sistema

Komponente sistema RoomRover web aplikacije su ReactJs skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



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

Komponenta **App.css** predstavlja opis stilova za pojedine ReactJs elemente koji se javljaju na različitim stranicama.

Komponenta **langDefs** predstavlja grupu ReactJs skriptova kojima se definišu konstante na različitim jezicima. Konvencija je da se ovi skriptovi nazivaju po ID-u jezika. Primer skripta koji definiše konstante za srpski jezik (ćirilica) je sledeće sadržine:

const LANG\_SR = {

L\_USERNAME: "Корисничко име:",

L\_PASSWORD: "Лозинка:",

L\_LOGIN: "Пријави се",

L\_LOGOUT: "Одјави се",

L\_SAVECHANGES: "Сачувајте измене",

L\_HOME: "О нама",

L\_HOTELS: "Доступни хотели",

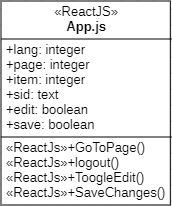
L\_DESTINATION: "Популарне дестинације",

};

export default LANG\_SR;

Komponenta **index.js** je ulazna tačka u web aplikaciju gde se koristi **ReactDOM** za montiranje glavne komponente (u ovom slučaju, **App** komponente) u HTML element sa ID-om "root".

Parametri koji utiču na izbor i jezik za prikaz stranice ilustrovani su sledećim dijagramom klasa:

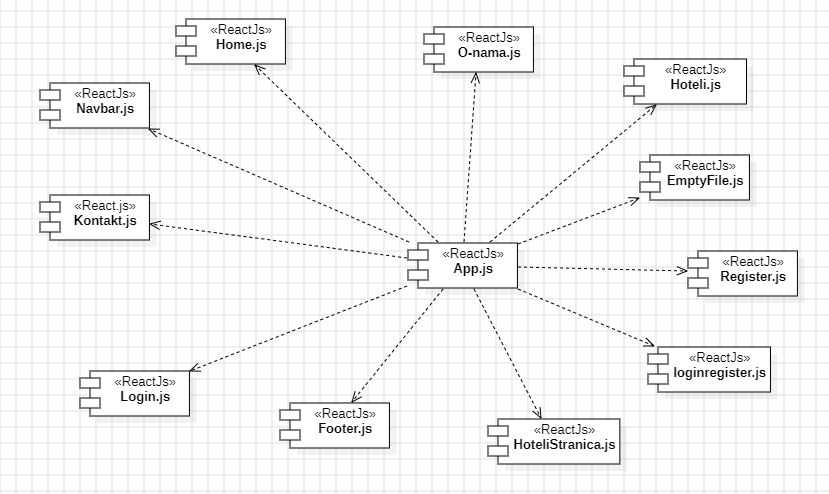


Značenje atributa je sledeće:

* lang – identifikator jezika (JID iz tabele Jezik)
* page – interni identifikator stranice (o laboratoriji, članovi, publikacije, projekti)
* item – identifikator stavke koja se detaljno prikazuje na stranici (određeni član ili projekat)
* sid – identifikator sesije kada je korisnik ulogovan
* edit – definiše da li je uključeno uređivanje
* save – definiše da li treba izvršiti snimanje podataka sa tekuće stranice/stavke

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **App.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:

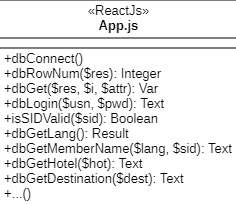


### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru PHP skripta **db.php**. Pomenuti skript se uključuje na početku index.php-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



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



Navedene funkcije obavljaju sledeće zadatke:

* dbConnect – povezivanje na bazu podataka, poziva se u samom skriptu
* dbRowNum – vraća broj slogova sadržanih u rezultatu upita
* dbGet – vraća vrednost polja iz rezultata upita
* dbLogin – loguje korisnika i vraća SID
* isSIDValid – vrši proveru validnosti SID-a
* dbGetLang – vraća rezultat upita nad jezicima definisanim u bazi podataka
* dbGetMemberName – vraća ime prijavljenog korisnika po jeziku i SID-u
* dbGetHotel – vraća traženi hotel
* dbGetDestination – vraća traženu destinaciju
* ... – ostale funkcije

# 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. PeNcIL portal će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.