EHotelServices

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 10.4.2021. | 1.0 | Inicijalna verzija | Nikola Petrović, 16288  Lazar Veljković, 16503 |
| 17.04.2021 | 2.0 | Pisanje sadržaja i ubacivanje dijagrama | Nikola Petrović, 16288  Lazar Veljković, 16503  Milan Stevanović, 16906 |
|  |  |  |  |
|  |  |  |  |

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 6

6.1 Dijagram slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 7

6.2.1 Prijavljivanje 7

6.2.2 Kreiranje profila 7

6.2.3 Brisanje profila 7

6.2.4 Ažuriranje podataka o sebi 7

6.2.5 Zahtevanje i rezervacija usluge 7

6.2.6 Pregled usluga i ponude hotela

6.2.7 Otkazivanje rezervisane usluge 7

6.2.8 Obrada zahteva gostiju 7

6.2.9 Ažuriranje računa gostiju 7  
 6.2.10 Ažuriranje rasporeda korišćenja hoteslih usluga i prostorija 7  
 6.2.11 Kreiranje i ažuriranje ponude hotela 7

7. Pogled na logičku arhitekturu sistema 8

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 8

7.1.1 Korisnički interfejs 8

7.1.2 Aplikaciona logika 9

7.1.3 Pristup podacima 9

7.1.4 HTML5 tehnologija 9

7.1.5 Bootstrap tehnologija 9

7.1.6 JavaScript ES6 tehnologija 9

7.1.7 ASP.NET tehnologija 9

7.1.8 MS SQL DBMS 9

8. Pogled na procese 9

8.1 Procesi 10

8.1.1 Web čitač 10

8.1.2 Web server 10

8.1.3 ASP.NET 10

8.1.4 MS SQL Server 10

9. Pogled na raspoređivanje sistema 11

9.1 Klijent 11

9.2 Web server 11

9.3 Baza podataka 11

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 13

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 15

10.3.3 Komponente za pristup podacima 15

11. Performanse 16

12. Kvalitet 16

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture aplikacije „EHotelServices“.

# Opseg dokumenta

Dokument se odnosi na aplikaciju „EHotelServices“. koja će biti razvijena od strane LNM\_CodingTeam-a. Namena sistema je efikasno prezentovanje i reklamiranje događaja u gradu. Namena sistema je efikasno, jednostavno i lako naručivanje usluga hotela kao i brzo ispunjavanje usluga koje je korisnik zahtevao.

# Reference

Spisak korišćene literature:

1. LNM\_CodingTeam – Predlog projekta, LNM\_CodingTeam-EHotelServices-01, V1.0, 2019, LNM\_CodingTeam.
2. LNM\_CodingTeam – Raspored aktivnosti, V1.0, 2019, LNM\_CodingTeam.
3. LNM\_CodingTeam – Plan realizacije projekta, V1.0, 2019, LNM\_CodingTeam.
4. LNM\_CodingTeam – Vizija sistema, V1.0, 2019, LNM\_CodingTeam.
5. LNM\_CodingTeam – Specifikacija zahteva, V1.0, 2019, LNM\_CodingTeam.

# 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. Aplikacija „EHotelServices“ će biti implementiran kao Web aplikacija zasnovana na .NET platformi i Entity Framework-u.
2. Klijentski deo aplikacije će biti optimizovan za sledeće Web čitače: MS Edge, kao i Firefox (Mozilla) i Google Chrome .
3. Svi zahtevi u pogledu performansi 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].

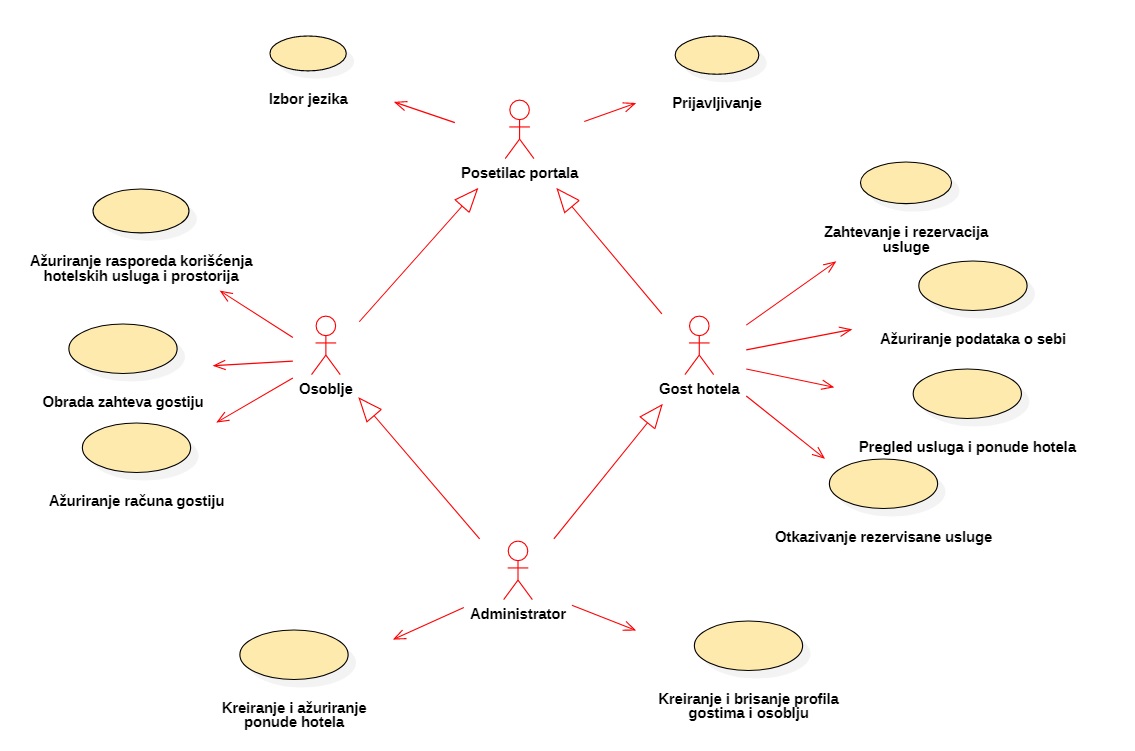
Slučajevi korišćenja aplikacije „EHotelService“ su:

* Prijavljivanje
* Kreiranje profila
* Brisanje profila
* Ažuriranje podataka o sebi
* Zahtevanje i rezervacija usluge
* Pregled usluga i ponude o sebi
* Otkazivanje rezervisane usluge
* Obrada zahteva gostiju
* Ažuriranje računa gostiju
* Kreiranje i ažuriranje rasporeda korišćenja hotelskih usluga i prostorija
* Kreiranje i ažuriranje ponude hotela

Ove slučajevi korišćenja mogu da iniciraju gost, osoblje ili administrator.

## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja aplikacije „EHotelServices“ prikazan je na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika u cilju omogućavanja funkcija .

Akteri koji iniciraju slučaj korišćenja: Administrator, osoblje, gost.

### Kreiranje profila

Kratak opis: Administrator ima mogućnost kreiranja profila gosta i osoblja.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje profila

Kratak opis: Administrator ima mogućnost brisanja profila iz sistema.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje podataka o sebi

Kratak opis: Gost hotela vrši uredjivanje svog profila.

Akteri koji iniciraju slučaj korišćenja: Gost.

### Zahtevanje i rezervacija usluge

Kratak opis: Korisniku je omogućeno zahtevanje odnosno rezervacija usluga koje hotel nudi.

Akteri koji iniciraju slučaj korišćenja: Gost.

### Pregled usluga i ponude hotela

Kratak opis: Korisniku je omogućeno da ima uvid u usluge koje hotel nudi.

Akteri koji iniciraju slučaj korišćenja: Gost.

### Otkazivanje rezervisane usluge

Kratak opis: Korisniku je omogućeno da otkaže rezervisanu uslugu..

Akteri koji iniciraju slučaj korišćenja: Gost.

### Obrada zahteva gostiju

Kratak opis: Osoblje ima mogućnost potvrde ili odbijanja zahteva gosta..

Akteri koji iniciraju slučaj korišćenja: Osoblje.

### Ažuriranje računa gostiju

Kratak opis: Odobravanje naloga od strane administratora. Osoblje ažurira račun gostima za svoje usluge.

Akteri koji iniciraju slučaj korišćenja: Osoblje.

### Ažuriranje rasporeda korišćenja hotelskih usluga i prostorija

Kratak opis: Osoblje ima mogućnost dodavanja i uklanjanja rezervisanih termina u raspored.

Akteri koji iniciraju slučaj korišćenja: Osoblje.

### Kreiranje i ažuriranje ponude hotela

Kratak opis: Administrator ima mogućnost kreiranja i ažuriranja ponude hotela.

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. MVC ASP.NET Framework sam po sebi nameće MVC arhitekturu, koja se inace široko koristi u razvoju Web aplikacija.

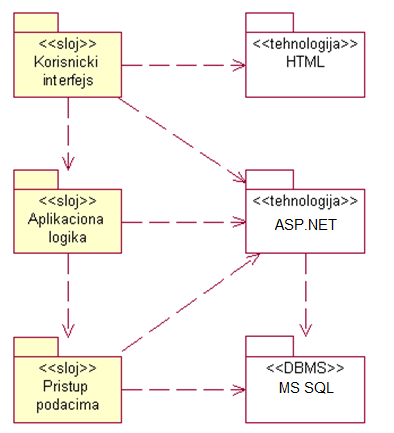
Logički pogled na aplikaciju „EHotelServices“ obuhvata 3 glavna paketa: Korisnički interfejs (view), Aplikaciona logika (controller), Pristup podacima (model).

Paket *Korisnički interfejs* sadrži Web stranice, JavaScript 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 ASP.NET module zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži ASP.NET module 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

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržane sve HTML, multimedijalni sadržaji, JavaScript skripte i ASP.NET kod koji generiše HTML stranice preko kojih korisnici komuniciraju sa sistemom. Ovaj sloj pretstavlja View aplikacije.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i ASP.NET.

### Aplikaciona logika sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi aplikacije. Sadrži ASP.NET module koji realizuju funkcionalnost karakterističnu za domen primene aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima. Ovaj sloj predstavlja kontroler aplikacije.

Takođe, ovaj sloj zavisi od sloja za pristup podacima i ASP.NET modula.

### Pristup podacima sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži ASP.NET funkcije zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MS SQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa ASP.NET i MS SQL baza podataka.

### HTML5 tehnologija

Tehnologija HTML-a 5 definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### Bootstrap tehnologija

Tehnologija Bootstrap Framework-a biće korišćena uz HTML5 i JavaScript da bi se dobilo dizajn koji omogućava dinamične Web stranice aplikacije i Responsive dizajn.

### JavaScript ES6 tehnologija

Tehnologija JavaScript ES6 će se, takođe, koristiti za Front-end deo, kao script programski jezik koji je podržan i čitljiv od strane Web pretraživača. ES6 je znatno napredniji i savremeniji od starijeg ES5 JavaScripta, s toga je on izabran za tehnologiju koja će se primenjivati za Front-end.

### ASP.NET tehnologija

Tehnologija ASP.NET obezbeđuje mehanizam za pisanje i izvršavanje C# koda na strani servera. ASP.NET može da generiše HTML kod koji realizuje korisnički interfejs i da pristupa bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka. ASP.NET Framework je trenutno jako popularan, i široko zastupljen u razvoju Web aplikacija, što je dovelo do odluke da se on koristi i za razvoj ove aplikacije. Još jedna prednost ovog Framework-a jeste njegovo razvojno okruženje – Visual Studio.

### MS SQL DBMS

MS SQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju asplikacije „EHotelServices“. Kao još jedan proizvod Microsoft-a on je kompatibilan sa ASP.NET-om.

# Pogled na procese

Procesi

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka 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.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju aplikacije „EHotelServices“. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ASP.NET-u i MS SQL bazi podataka.



### Web čitač

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

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

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Kada se zahteva određena web stranica, inicira se izvršenje ASP.NET procesa koji poziva odgovarajuće funkcije i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj ASP.NET procesa.

### ASP.NET

ASP.NET proces obavlja posao obrade zadate ASP.NET funkcije i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje ASP.NET funkcija može da se zahtevaju usluge MS SQL baze. Komunikacija između ASP.NET procesa i MS SQL baze se obavlja preko prosleđivanja upita i vraćanja rezultat.

### MS SQL Server

MS SQL Server je proces koji izvršava funkcionalnost MS 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 EHotelServices web aplikacije.



## Klijent

Pristup aplikaciji „EHotelServices“ 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 ASP.NET procesi.

## Baza podataka

Baza podataka se nalazi na istoj mašini kao i WEB server.

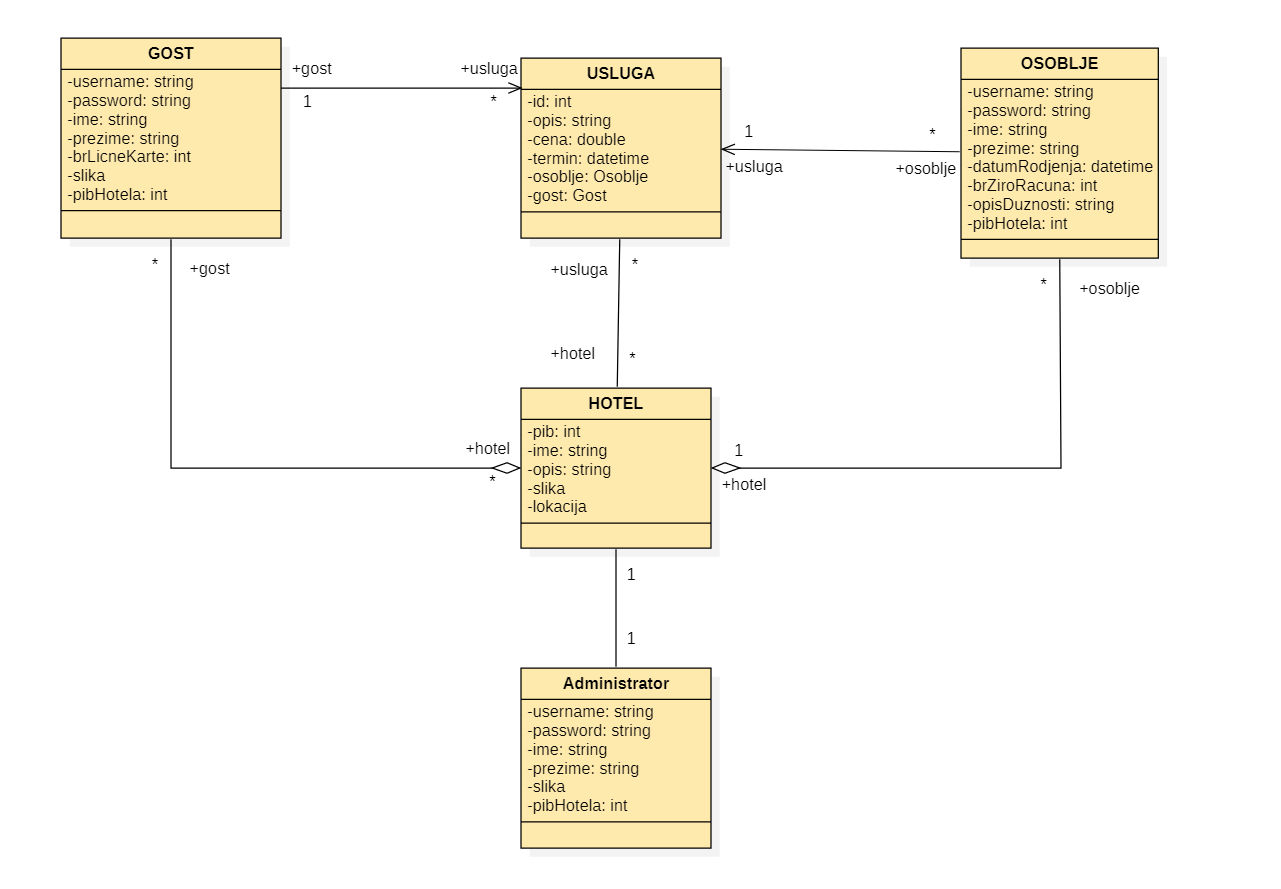
# Pogled na implementaciju sistema

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

## Model domena

Model domena za koji se aplikacija „EHotelServices“ 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.

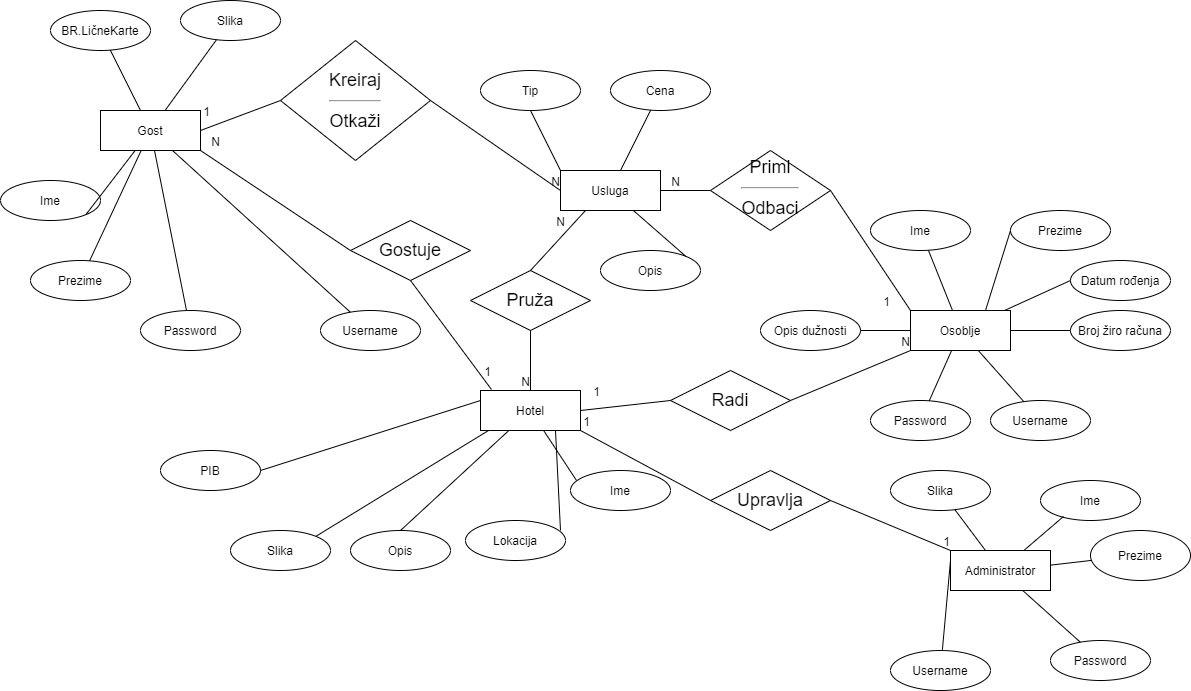
Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti koje će biti implementirane.

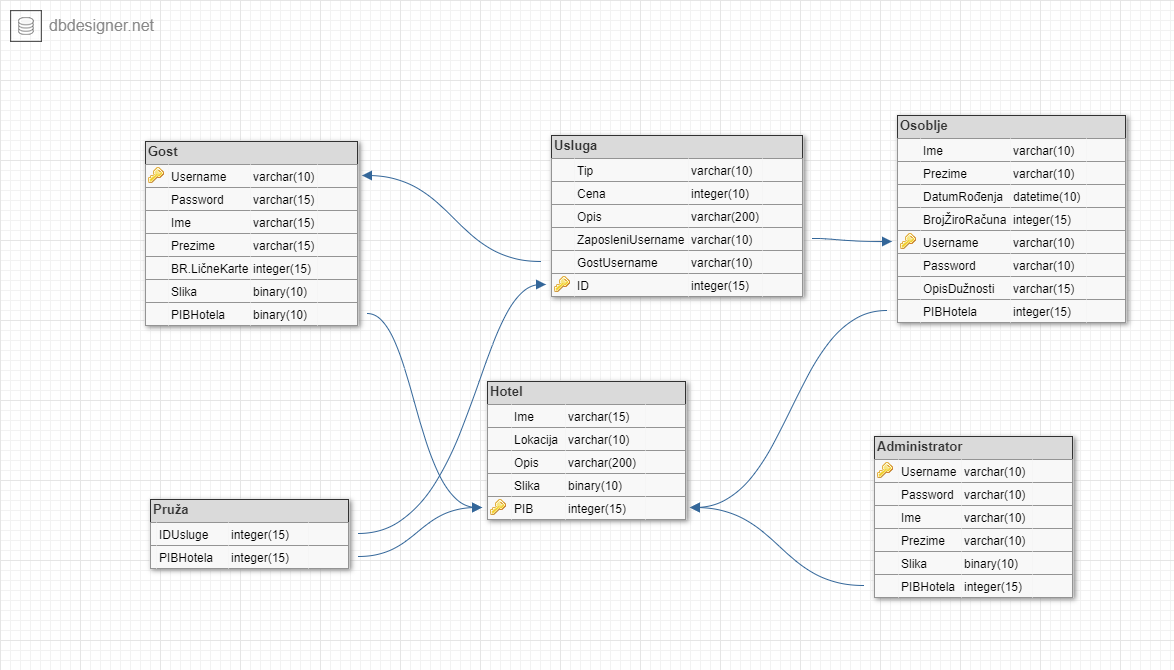


Model domena je urađen korišćenjem *Eclipse Neon Papyrus*.

## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. ER dijagram baze podataka nacrtan je korišćenjem online alata *app.diagrams.net.* Relacioni model baze podataka nacrtan je korišćenjem online alata *DbDesigner*.



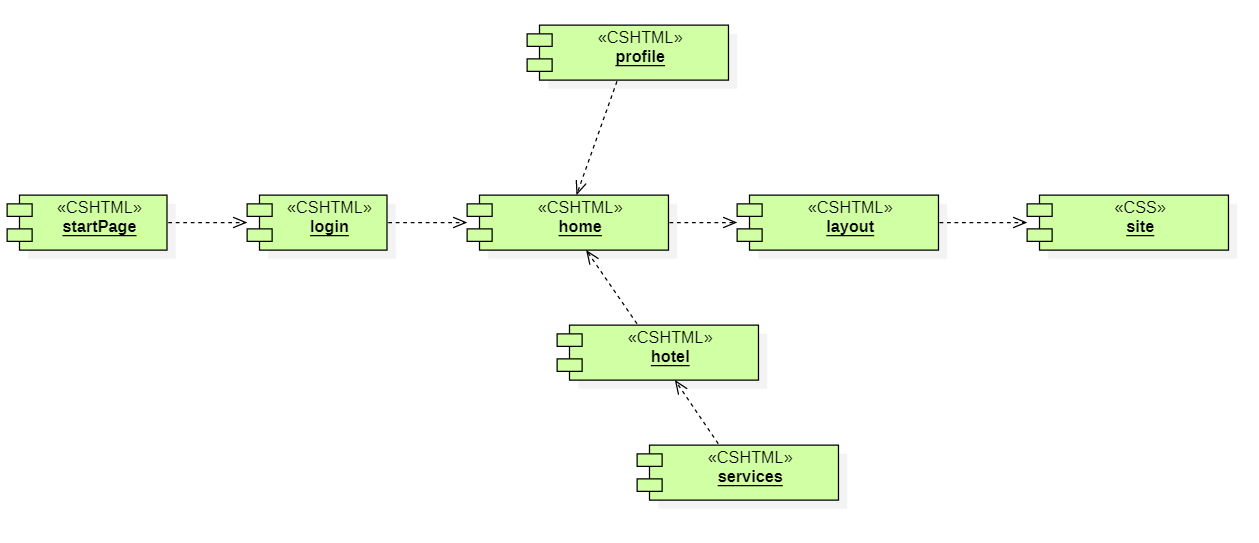


## Komponente sistema

Komponente sistema aplikacije „EHotelServices“ su prikazane u okviru ovog poglavlja po slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa:



Komponenta **startPage.cshtml** implementira početnu stranicu aplikacije koja predstavlja prezentaciju same aplikacije i koja sadrži linkove ka formi za login.

Komponenta **login.cshtml** implementira stranicu za prijavljivanje na sistem.

Komponenta **home.cshtml** predstavlja glavnu stranicu na kojoj je prikazan hotel i koja sadrži linkove ka stranici korisnika i usluga.

Komponenta **profile.cshtml** implementira stranicu za prikaz profila gosta ili osoblja.

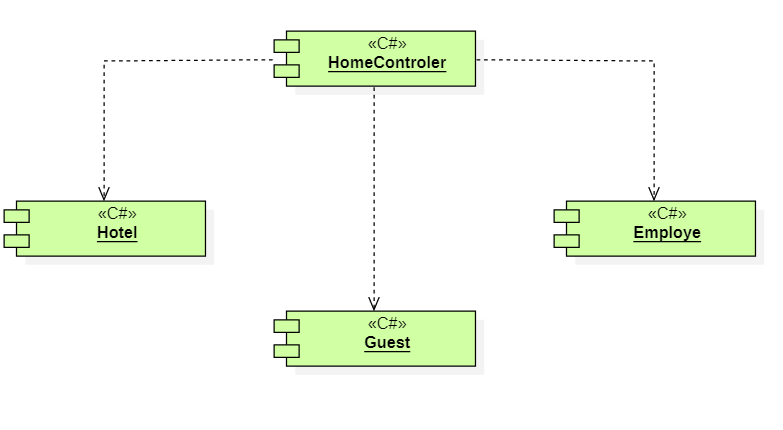
Komponenta **services.cshtml** implementira stranicu za prikaz informacija o uslugama hotela.

Komponenta **layout.cshtml** predstavlja osnovnu komponentu koja određuje izgled svih ostalih stranica.

Komponenta **site.css** predstavlja css fajl kojim se dodatno ukrašava sadržaj stranica.

### Komponente aplikacione logike

Komponente koje realizuju sloj aplikacione logike su kontroleri koji će hendlovati izabrane zahteve korisnika.



Komponenta **HomeControler.cs** implementira aplikacionu logiku osnovne stranice aplikacije.

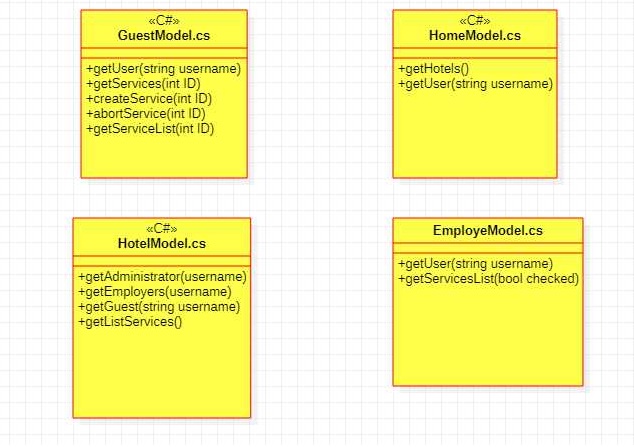
Komponenta **Hotel.cs** predstavlja kontroler zadužen za logiku stranice za prikaz ponude hotela.

Komponenta **Guest.cs** implementira aplikacionu logiku stranice gosta hotela.

Komponenta **Employe.cs** implementira aplikacionu logiku stranice osoblja hotela.

### Komponente za pristup podacima

Pristup bazi podataka je zatvoren u funkcije koje su definisane u okviru modela. Funkcije u okviru modela se pozivaju iz određenih kontrolera.



# 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 [4]:

1. Sistem će da podrži do 1000 simultanih pristupa korisnika aplikacije.
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 [4].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [4]:

1. Aplikacija „EHotelServices“ 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 150 sati