Hotel Petsylvania

Rezervacija usluga u hotelu za kućne ljubimce

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 09.05.2021 | 1.0 | Inicijalna verzija | Tea Mitić, Milja Pejčić, Dimitrije Mitić |
| 27.6.2021 | 2.0 | Konacna verzija | Tea Mitić, Milja Pejčić, Dimitrije Mitić |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc75682260)

[2. Opseg dokumenta 5](#_Toc75682261)

[3. Reference 5](#_Toc75682262)

[4. Predstavljanje arhitekture 5](#_Toc75682263)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc75682264)

[6. Pogled na slučajeve korišćenja 5](#_Toc75682265)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc75682266)

[6.2 Kratak opis slučajeva korišćenja 7](#_Toc75682267)

[6.2.1 Registracija nove mušterije 7](#_Toc75682268)

[6.2.2 Pregled osnovnih informacija o hotelu 8](#_Toc75682269)

[6.2.3 Pregled galerije slika 8](#_Toc75682270)

[6.2.4 Pregled ocena i recenzija 8](#_Toc75682271)

[6.2.5 Pregled cenovnika usluga 8](#_Toc75682272)

[6.2.6 Pregled odeljka za pitanja 8](#_Toc75682273)

[6.2.7 Pregled notifikacija 8](#_Toc75682274)

[6.2.8 Kreiranje novog zahteva 8](#_Toc75682275)

[6.2.9 Postavljanje pitanja 8](#_Toc75682276)

[6.2.10 Postavljanje ocene i recenzije 8](#_Toc75682277)

[6.2.11 Ažuriranje poslatog zahteva 8](#_Toc75682278)

[6.2.12 Brisanje poslatog zahteva 8](#_Toc75682279)

[6.2.13 Pregled poslatih zahteva 8](#_Toc75682280)

[6.2.14 Prijavljivanje na sistem 8](#_Toc75682281)

[6.2.15 Odjavljivanje sa sistema 8](#_Toc75682282)

[6.2.16 Ažuriranje ličnih podataka korisnika 9](#_Toc75682283)

[6.2.17 Odgovaranje na pitanja 9](#_Toc75682284)

[6.2.18 Pretraga i filtriranje zahteva 9](#_Toc75682285)

[6.2.19 Pregled zahteva 9](#_Toc75682286)

[6.2.20 Prihvatanje zahteva za obradu 9](#_Toc75682287)

[6.2.21 Slanje obaveštenja mušteriji o obrađenom zahtevu 9](#_Toc75682288)

[6.2.22 Pregled radnika hotela 9](#_Toc75682289)

[6.2.23 Pregled usluga hotela 9](#_Toc75682290)

[6.2.24 Brisanje postojeće usluge 9](#_Toc75682291)

[6.2.25 Dodavanje nove usluge 9](#_Toc75682292)

[6.2.26 Ažuriranje cene postojeće usluge 9](#_Toc75682293)

[6.2.27 Postavljanje zvaničnih obaveštenja 9](#_Toc75682294)

[6.2.28 Ažuriranje kapaciteta hotela 9](#_Toc75682295)

[6.2.29 Dodavanje novog radnika 9](#_Toc75682296)

[6.2.30 Brisanje radnika 10](#_Toc75682297)

[7. Pogled na logičku arhitekturu sistema 10](#_Toc75682298)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10](#_Toc75682299)

[7.1.1 Korisnički interfejs 10](#_Toc75682300)

[7.1.2 Aplikaciona logika 11](#_Toc75682301)

[7.1.3 Pristup podacima 11](#_Toc75682302)

[7.1.4 HTML 11](#_Toc75682303)

[7.1.5 JavaScript 11](#_Toc75682304)

[7.1.6 Vue.js 11](#_Toc75682305)

[7.1.7 ASP.NET Core 11](#_Toc75682306)

[7.1.8 SQL Server 11](#_Toc75682307)

[8. Pogled na procese 11](#_Toc75682308)

[8.1 Procesi 11](#_Toc75682309)

[8.1.1 Web čitač 12](#_Toc75682310)

[8.1.2 Web server 12](#_Toc75682311)

[8.1.3 ASP.NET Core 12](#_Toc75682312)

[8.1.4 SQL Server 12](#_Toc75682313)

[9. Pogled na raspoređivanje sistema 12](#_Toc75682314)

[9.1 Klijent 12](#_Toc75682315)

[9.2 Web server 12](#_Toc75682316)

[9.3 DBMS server 13](#_Toc75682317)

[10. Pogled na implementaciju sistema 13](#_Toc75682318)

[10.1 Model domena 13](#_Toc75682319)

[10.2 Šema baze podataka 14](#_Toc75682320)

[10.3 Komponente sistema 15](#_Toc75682321)

[10.3.1 Komponente korisničkog interfejsa 16](#_Toc75682322)

[10.3.2 Komponente aplikacione logike 18](#_Toc75682323)

[10.3.3 Komponente za pristup podacima 20](#_Toc75682324)

[○ GET GetSpecificID – Pribavljanje ID-a iz tabeli Radnik/Musterija/Admin na osnovu ID-a iz tabele korisnik 20](#_Toc75682325)

[○ GET GetUserByID – Pribavljanje korisnika na osnovu njegovog ID-a 20](#_Toc75682326)

[11. Performanse 23](#_Toc75682327)

[12. Kvalitet 23](#_Toc75682328)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Web aplikacije Hotel Petsylvania.

# Opseg dokumenta

Dokument se odnosi na Web aplikaciju Hotel Petsylvania koja će biti razvijena od strane tima 101 Dalmatinac. Hotel Petsylvania predstavlja smisleno i kraće ime aplikacije koja izvršava rezervacije usluga u hotelu za ljubimce. Ime je osmišljeno na osnovu animiranog filma Hotel Transylvania u kome se čuvaju razna mitska stvorenja. Namena aplikacije je prevazilaženje trenutnih problema kod rezervisanja usluga i omogućavanje lakšeg slanja rezervacija i uvid u iste u “samo tri klika”.

# Reference

Spisak korišćene literature:

Hotel Petsylvania – Predlog projekta, SWE-Hotel Petsylvania-01, V1.0, 2021, 101 Dalmatinac.

Hotel Petsylvania – Vizija sistema, SWE-Hotel Petsylvania-02, V1.0, 2021, 101 Dalmatinac.

Hotel Petsylvania – Plan realizacije projekta, SWE-Hotel Petsylvania-03, V1.0, 2021, 101 Dalmatinac.

Hotel Petsylvania – Planirani raspored aktivnosti na projektu, V1.0, 2021, 101 Dalmatinac.

Hotel Petsylvania - Specifikacija zahteva, V1.0, 2021, 101 Dalmatinac.

Hotel Petsylvania – ApiPozivi – sporedni dokument

# 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:

Platforma Hotel Petsylvania će biti implementirana kao Web aplikacija zasnovana na Vue.js framework-u i MySQL bazi podataka [2].

Klijentski deo platforme Hotel Petsylvania će biti optimizovana za sledeće Web čitače: Internet Explorer, Opera, FireFox (Mozzila) i Google Chrome.

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].

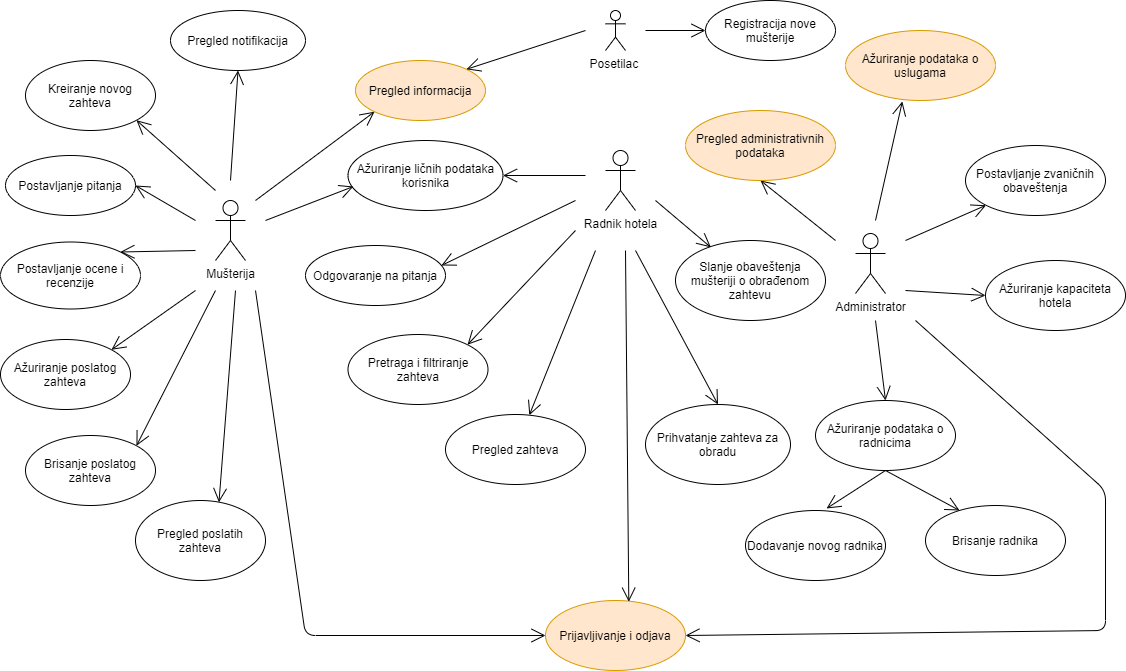
Slučajevi korišćenja platforme Hotel Petsylvania su:

* Registracija nove mušterije
* *Pregled informacija*
  + Pregled osnovih informacija o hotelu
  + Pregled galerije slika
  + Pregled ocena i recenzija
  + Pregled cenovnika usluga
  + Pregled odeljka za pitanja
* Pregled notifikacija
* Kreiranje novog zahteva
* Postavljanje pitanja
* Postavljanje ocene i recenzije
* Ažuriranje poslatog zahteva
* Brisanje poslatog zahteva
* Pregled poslatih zahteva
* *Prijavljivanje i odjava*
  + Prijavljivanje na sistem
  + Odjavljivanje sa sistema
* Ažuriranje ličnih podataka korisnika
* Odgovaranje na pitanja
* Pretraga i filtriranje zahteva
* Pregled zahteva
* Prihvatanje zahteva za obradu
* Slanje obaveštenja mušteriji o obrađenom zahtevu
* *Pregled administrativnih podataka*
  + Pregled radnika hotela
  + Pregled usluga hotela
* *Ažuriranje podataka o uslugama*
  + Brisanje postojeće usluge
  + Dodavanje nove usluge
  + Ažuriranje cena postojeće usluge
* Postavljanje zvaničnih obaveštenja
* Ažuriranje kapaciteta hotela
* *Ažuriranje podataka o radnicima*
  + Dodavanje novog radnika
  + Brisanje radnika

Ove slučajeve korišćenja mogu da iniciraju posetilac platforme, mušterija, radnik hotela ili administrator.

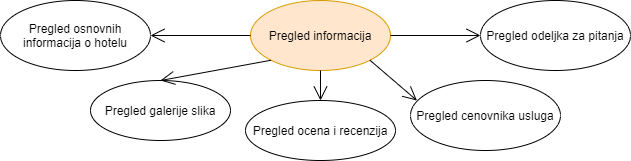
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Web aplikacije Hotel Petsylvania prikazan je na sledećoj slici:

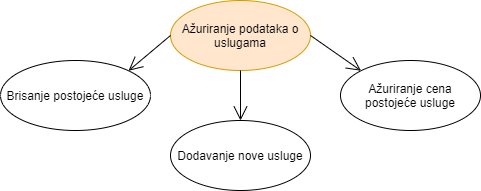


Slučajevi korišćenja *Pregled informacija*, *Ažuriranje podataka o uslugama*, *Pregled administrativnih podataka* i *Prijavljivanje i odjava* obuhvataju složenije radnje koje se mogu razložiti dalje na pojedinačne slučajeve korišćenja.

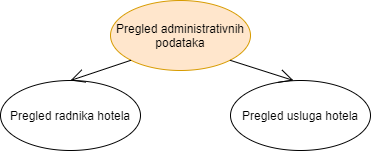
Detaljni UML dijagram za slučaj korišćenja *Pregled informacija* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *Ažuriranje podataka o uslugama* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *Pregled administrativnih podataka* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *Prijavljivanje i odjava* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Registracija nove mušterije

Kratak opis: Kreiranje naloga novom korisniku kako bi kasnije pristupao ostalim funkcionalnostima aplikacije.

Akteri koji iniciraju slučaj korišćenja: Posetilac.

### Pregled osnovnih informacija o hotelu

Kratak opis: Prikaz stranice sa informacijama o hotelu.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije, Mušterija.

### Pregled galerije slika

Kratak opis: Pregled slika o hotelu.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije.

### Pregled ocena i recenzija

Kratak opis: Prikaz stranice na kojoj se nalazi ocena hotela i recenzije mušterija o hotelu.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije, Mušterija.

### Pregled cenovnika usluga

Kratak opis: Prikaz stranice sa cenovnikom usluga.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije, Mušterija.

### Pregled odeljka za pitanja

Kratak opis: Mogućnost pregleda svih prethodno postavljenih pitanja radi bolje informisanosti.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije, Mušterija.

### Pregled notifikacija

Kratak opis: Prikaz forme na kojoj se nalaze sve notifikacije.

Akteri koji iniciraju slučaj korišćenja: Mušterija.

### Kreiranje novog zahteva

Kratak opis: Kreiranje zahteva usluga koje nudi hotel.

Akteri koji iniciraju slučaj korišćenja: Mušterija.

### Postavljanje pitanja

Kratak opis: Mušterija postavlja pitanja na koje radnik treba da odgovori.

Akteri koji iniciraju slučaj korišćenja: Mušterija.

### Postavljanje ocene i recenzije

Kratak opis: Ocenjivanje i ostavljanje utiska od strane mušterije.

Akteri koji iniciraju slučaj korišćenja: Mušterija.

### Ažuriranje poslatog zahteva

Kratak opis: Izmena nekih parametara još neizvršenog zahteva..

Akteri koji iniciraju slučaj korišćenja: Mušterija.

### Brisanje poslatog zahteva

Kratak opis: Brisanje prethodno poslatog zahteva koji još uvek nije obrađen.

Akteri koji iniciraju slučaj korišćenja: Mušterija.

### Pregled poslatih zahteva

Kratak opis: Prikaz svih poslatih neizvršenih zahteva koje je mušterija poslala..

Akteri koji iniciraju slučaj korišćenja: Mušterija.

### Prijavljivanje na sistem

Kratak opis: Prijavljivanje korisnika na platformu radi pristupanja ostalim funkcionalnostima platforme.

Akteri koji iniciraju slučaj korišćenja: Mušterija, Administrator, Radnik hotela.

### Odjavljivanje sa sistema

Kratak opis: Odjavljivanje korisnika iz aplikacije..

Akteri koji iniciraju slučaj korišćenja: Mušterija, Administrator, Radnik hotela.

### Ažuriranje ličnih podataka korisnika

Kratak opis: Izmena ličnih podataka korisnika koji se prikazuju na *Moj profil* stranici Web aplikacije.

Akteri koji iniciraju slučaj korišćenja: Mušterija, Radnik hotela.

### Odgovaranje na pitanja

Kratak opis: Radnik odgovara na pitanje koje je postavila mušterija hotela.

Akteri koji iniciraju slučaj korišćenja: Radnik hotela.

### Pretraga i filtriranje zahteva

Kratak opis: Uvid u sve vrste zahteva i omogućeno njihovo filtriranje po različitim kriterijumima.

Akteri koji iniciraju slučaj korišćenja: Radnik hotela.

### Pregled zahteva

Kratak opis: Uvid u sve vrste rezervisanih zahteva.

Akteri koji iniciraju slučaj korišćenja: Radnik hotela.

### Prihvatanje zahteva za obradu

Kratak opis: Radnik bira zahtev koji želi da obradi iz liste *Neobrađenih zahteva*.

Akteri koji iniciraju slučaj korišćenja: Radnik hotela.

### Slanje obaveštenja mušteriji o obrađenom zahtevu

Kratak opis: Nakon završetka izrade usluge šalje se obaveštenje mušteriji (čiji je zahtev) da je izvršena usluga. Obaveštenje može biti u vidu poruke.

Akteri koji iniciraju slučaj korišćenja: Radnik hotela.

### Pregled radnika hotela

Kratak opis: Uvid u sve radnike hotela.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Pregled usluga hotela

Kratak opis: Uvid u sve usluge hotela.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje postojeće usluge

Kratak opis: Brisanje postojeće usluge koju pruža hotel za kućne ljubimce.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Dodavanje nove usluge

Kratak opis: Dodavanje nove usluge koju će da pruža hotel za kućne ljubimce.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje cene postojeće usluge

Kratak opis: Promena cene postojeće usluge koju pruža hotel za kućne ljubimce.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Postavljanje zvaničnih obaveštenja

Kratak opis: Postavljanje novog zvaničnog obaveštenja o hotelu na početnoj stranici Web aplikacije.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje kapaciteta hotela

Kratak opis: Promena kapaciteta smeštaja hotela i ostalih usluga koje hotel pruža.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Dodavanje novog radnika

Kratak opis: Administrator kreira nalog novom radniku hotela.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje radnika

Kratak opis: Administrator briše postojećeg radnika hotela iz baze podataka.

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.

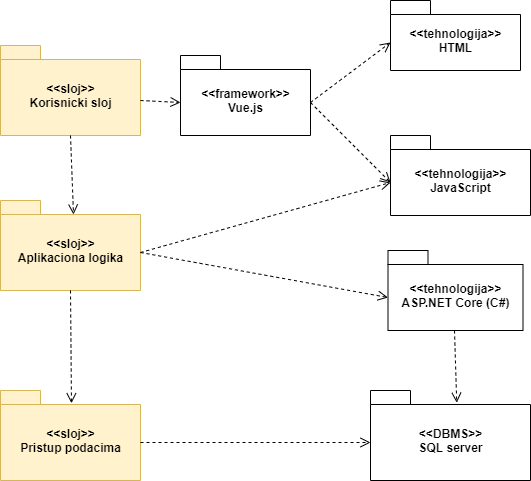
Logički pogled na platformu Hotel Petsylvania obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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 C# skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* pomoću ASP.NET Core framework-a za C# 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 Web aplikacije. U njemu su sadržane sve HTML fajlovi, multimedijalni sadržaji i JavaScript skripte koje generišu dizajn stranice i kojima su definisani načini prikaza stranice prilikom njenog menjanja zavisno od akcija korisnika. Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i Vue.js framework-a.

### Aplikaciona logika

sloj

Funkcije sloja aplikacione logike su pribavljanje podataka iz baze korišćenjem API poziva i njihovo lokalno smeštanje u Web aplikaciju. Nakon pribavljanja, na ovom sloju vrši se obrada podataka u skladu sa funkcionalnostima vezanim za logiku rada aplikacije, i rezultati obrade postaju dostupni sloju korisničkog interfejsa, radi prikaza.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži C# skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SQL Server bazi podataka.

Ovaj sloj zavisi od aplikacionog sloja u kome se kreiraju upiti upućeni bazi podatka.

### HTML

tehnologija

Tehnologija HTML-a definiše osnovne elemente stranica i izgled stranica koje se prikazuju u Web čitaču uz pomoć Vue.js framework-a.

### JavaScript

tehnologija

Tehnologija JavaScript-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs.

### Vue.js

framework

Vue.js framework se koristi za potpuno uređenje Web stranica koje podrazumeva definisanje komponenti na Web stranicama, kao i način na koji one reaguju na akcije koje je preduzeo korisnik. Zavisi od HTML tehnologije uz pomoć koje se definiše izgled Web stranice, kao i od JavaScript kojom se definiše ponašanje.

### ASP.NET Core

framework

ASP.NET Core je višeplatformski framework otvorenog koda za kreiranje modernih web aplikacija.

Služi za kreiranje aplikacione logike u C#-u. Primenom ovog framework-a biće napravljene backend funkcionalnosti za aplikaciju Hotel Petsylvania.

### SQL Server

DBMS

SQL Server predstavlja sistem za upravljanje bazom podataka koja će se koristiti za realizaciju Web aplikacije Hotel Petsylvania.

# Pogled na procese

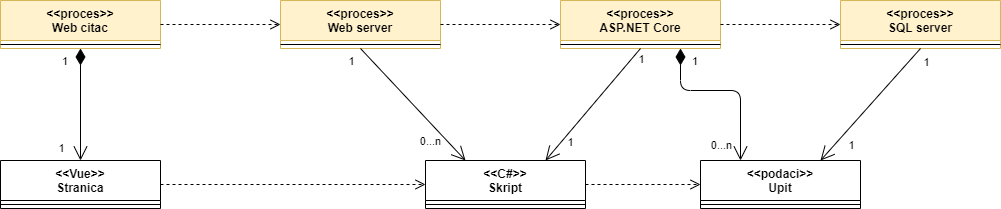
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na Vue.js 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 Hotel Petsylvania platforme kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Hotel Petsylvania platforme.



### 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 koja je generisana od strane Vue biblioteke.

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.

Obradu tih zahteva vrši ASP.NET Core proces i rezultat obrade je je C# skript. Web server na osnovu sadržaja C# skripte formira prikaz i prikazuje ga klijentu u Web čitaču. Web server može paralelno da inicira veći broj ASP.NET Core procesa.

### ASP.NET Core

ASP.NET Core obavlja proces obrade C# skripti. Kao rezultat se generišse odgovarajući tekstualni sadržaj koji Web čitač može da prikaže. Za izvršenje C# skripti ovaj proces može da zahteva usluge SQL Server servera. Komunikacija između ASP.NET Core procesa i SQL Server servera se obavlja preko prosleđivanja upita i vraćanja rezultata.

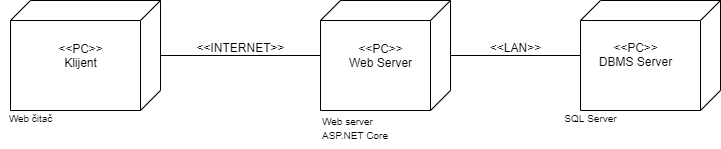
### SQL Server

SQL Server 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 Hotel Petsylvania platforme.



## Klijent

Pristup Hotel Petsylvania 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. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi ASP.NET Core koji vrše obradu zadatih C# skripti. U najopštijoj 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 SQL 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).

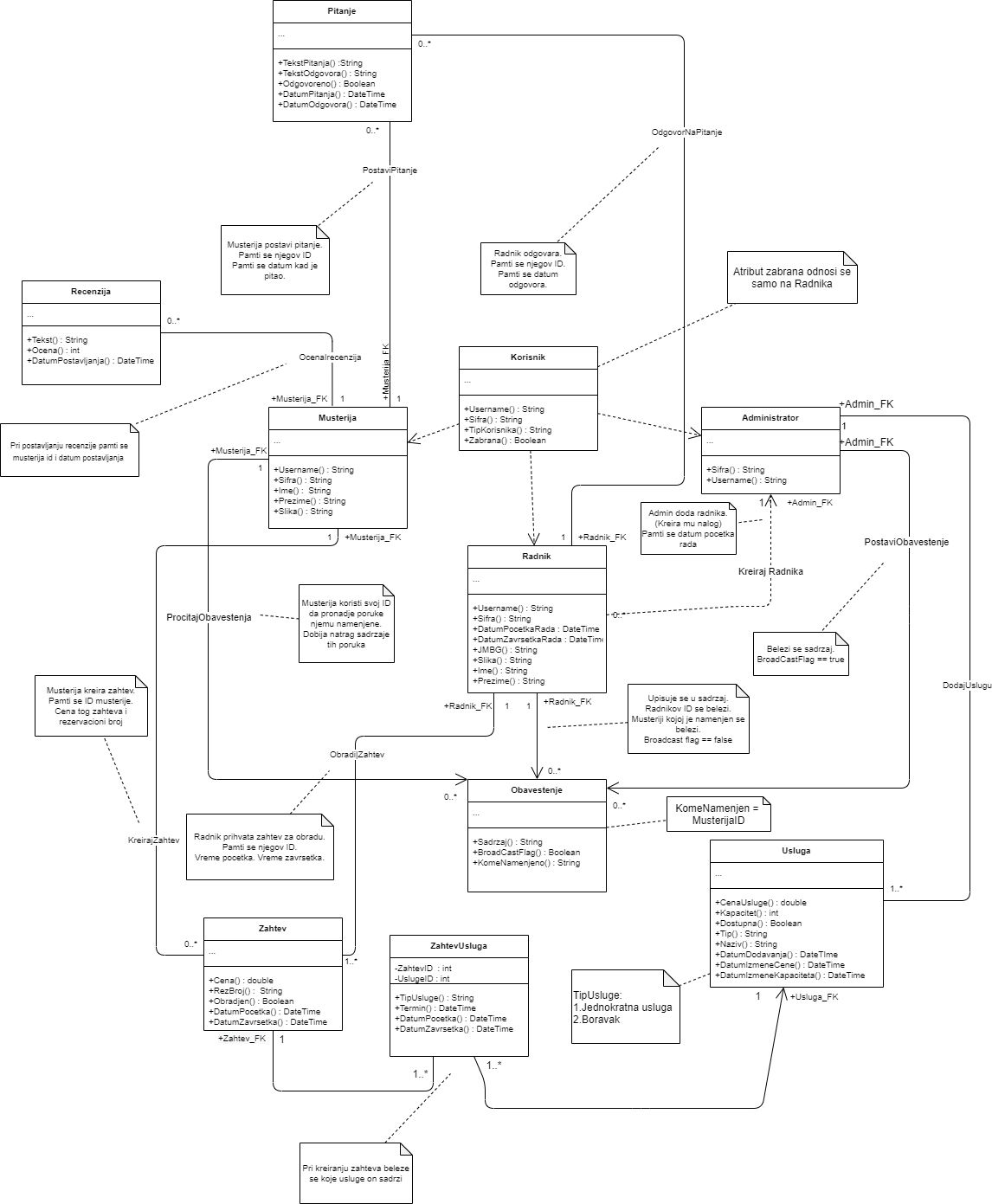
# Pogled na implementaciju sistema

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

## Model domena

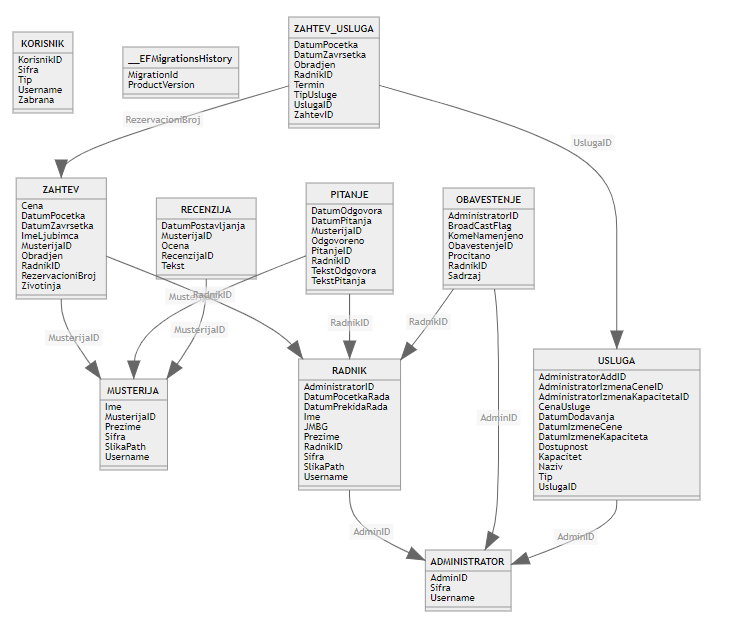
Model domena za koju se Hotel Petsylvania aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

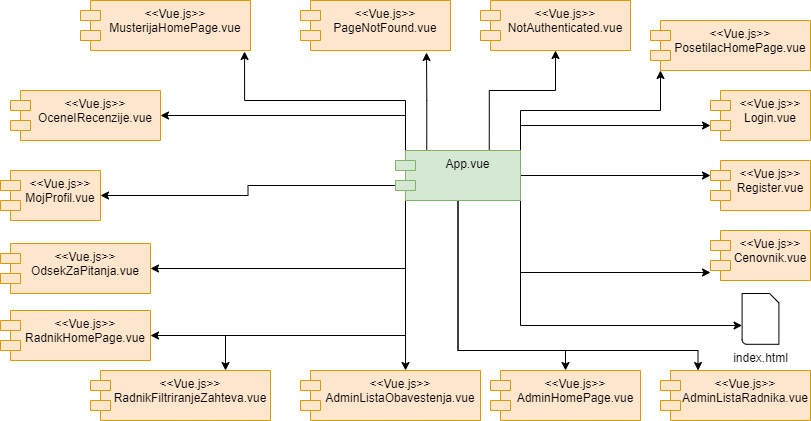
Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem Azure Data Studio i ekstenzije Schema Visualization za Azure Data Studio, dok je migracija na SQLServer obavljena pomoću EF Core CLI naredbi.



## Komponente sistema

Komponente sistema Hotel Petsylvania su PHP skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je PHP skript prikazan 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



Glavna komponenta, **App.vue**, predstavlja „okvir“ u kome se renderuju sve ostale komponente korisničkog interfejsa Web aplikacije. U direktnoj je vezi sa svim stranicama aplikacije i sa fajlom **index.html**. Komponenta **index.html** služi za uključivanje spoljačnjih zavisnosti kao što mogu da budu fontovi, ikonice i ostali linkovi. Takođe služi za uključivanje Bootstrap-a.

**PosetilacHomePage.vue** je početna stranica, prikazana pre nego što se korisnik prijavi. Sadrži osnovne podatke o samom hotelu. Stranice do kojih je moguće doći sa ove jesu stranice za registraciju i prijavu korisnika (mušterija, radnik, administrator), cenovnik, ocene i recenzije, kao i odsek za pitanja.

**Login.vue** sadrži formu za prijavljivanje korisnika sa poljima za unos šifre i username-a za prijavu.

**Register.vue** sadrži formu za registrovanje mušterije hotela, sa poljima za unos ličnih podataka.

**Cenovnik.vue** je stranica na kojoj stoji lista cena svih dostupnih usluga koje pruža hotel za kućne ljubimce. Do ove stranice je moguće doći preko **PosetilacHomePage.vue** i **MusterijaHomePage.vue**.

**OceneIRecenzije.vue** je stranica na kojoj posetilac može da pogleda ocene i recenzije koje su mušterije postavljale. Takođe, mušterija ima pristup ovoj stranici gde, pored pregleda drugih ocena i renzija, dodatno može postaviti svoju ocenu i recenziju.

**MojProfil.vue** je stranica za prikaz detaljnih informacija o korisniku (mušterija, radnik). Na ovoj stranici postoji opcija za izmenu ličnih podataka, pri čemu se u skladu sa tim menja izgled stranice.

**MusterijaHomePage.vue** je prva stranica koja se pojavi mušteriji nakon prijavljivanja. Na njoj je prikazana forma za slanje novih usluga, forma sa zvaničnim obaveštenjima hotela i forma sa listom postojećih zahteva koje je mušterija poslala. Mušterija na formi *Moje Prijave* ima mogućnost izmene i brisanje prijave. Sa ove stranice, mušterija može da pređe na stranice **Cenovnik.vue**, **OceneIRecenzije.vue**, **MojProfil.vue** i **OdsekZaPitanja.vue**.

**OdsekZaPitanja.vue** je stranica na kojoj mušterija može da postavi pitanje vezano za hotel, kao i da čita pitanja drugih mušterija. Pristup ovoj stranici imaju i radnici koji imaju mogućnost da odgovaraju na postavljena pitanja.

**RadnikHomePage.vue** je prva stranica koja se pojavi radniku nakon prijavljivanja. Na njoj su prikazane rezervacije za čuvanje kućnih ljubimaca, kao i neobrađeni zahtevi. Radnik može izabrati jedan zahtev koji želi da obradi, čime će se taj zahtev prebaciti u odeljak namenjen za zahtev koji se trenutno obrađuje. Dok se obrađuje, radnik ima opciju da pošalje komentar mušteriji koja je poslala taj zahtev u vidu poruke. Kada radnik obradi zahtev, prebacuje ga u odeljak završenih zahteva. Sa ove stranice radnik može da pređe na stranicu **RadnikFiltriranjeZahteva.vue**, **MojProfil.vue** i **OdsekZaPitanja.vue**.

**RadnikFiltriranjeZahteva.vue** je stranica na kojoj radnici mogu vršiti pregled i filtriranje svih zahteva koje je hotel primio pomoću odgovarajućih parametara.

**AdminHomePage.vue** je prva stranica koja se pojavi administratoru nakon prijavljivanja. Na njoj su prikazane forme za dodavanje novog radnika, nove usluge i postavljanje zvaničnih obaveštenja o hotelu. Takođe, administrator preko ove stranice ima mogućnost da promeni kapacitete hotela, kao i cene trenutnih usluga. Sa ove stranice adminstrator može da pređe na stranicu **AdminListaRadnika.vue** i **AdminListaObavestenja.vue**.

**AdminListaRadnika.vue** je stranica koja administratoru prikazuje listu svih radnika hotela koji trenutno rade u njemu.

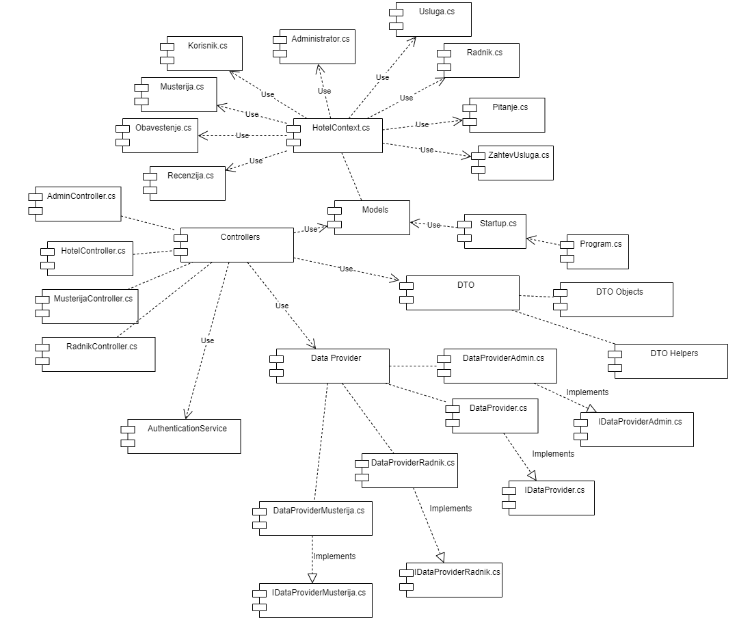
**AdminListaObavestenja.vue** je stranica preko koje administrator ima uvid u sva postavljena zvanična obaveštenja o hotelu i ima opciju da izbriše bilo koje obaveštenje.

**NotAuthenticated.vue** je stranica koja se pojavljuje ukoliko neki korisnik pokuša da pristupi nekoj stranici koja nije namenjena za korisnika njegovih privilegija.

**PageNotFound.vue** je stranica koja se pojavljuje ukoliko neki korisnik pokuša da pristupi stranici koja ne postoji.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **program.cs** fajla aplikacione logike. Na taj način se omogućava pristup bazi i svim dostupnim api funkcijama. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



**Program.cs** je fajl koji predstavlja ulaznu tacku u bekend deo aplikacije.

**Startup.cs** fajl koji se prvi izvrsava nakon ulaska u bekend deo. Ovaj fajl je obavezan.

**HotelContext.cs** najbitnija klasa pri izradi bekenda. Koristi se za uspostavljanje konekcije sa bazom i sadrži reference na klase koje se mapiraju u tabele u bazi podataka. Korišćenjem ove klase mogu se obavljati CRUD operacije (Create, Read, Update, Delete).

**Administrator.cs** klasa koja sadrži podatke administratora. Neke atribute ove klase i klasa navedenih ispod mogu se videti na slici u poglavlju 10. Model Domena.

**Radnik.cs** klasa sadrži podatke o radnicima u hotelu.

**Musterija.cs** klasa sadrži podatke o mušterijama koje koriste usluge hotela.

**Korisnik.cs** klasa sadrži podatke o svim korisnicima (admini, radnici, mušterije) i ovi podaci se koriste pri prijavljivanju na sistem.

**Obaveštenje.cs** klasa sadrži podatke o obaveštenjima namenjenim mušterijama.

**Pitanje.cs** klasa sadrži podatke o postavljenim pitanjima i odgovorima na istih od strane radnika.

**Recenzija.cs** klasa sadrži podatke o recenzijama datim od strane mušterija.

**Usluga.cs** klasa sadrži podatke o svim uslugama koje hotel nudi.

**Zahtev.cs** klasa sadrži podatke o zahtevima koje kreiraju mušterije.

**ZahtevUsluga.cs** klasa sadrži podatke o uslugama koje pripadaju postojećim zahtevima.

**HotelController.cs** klasa sadrži api pozive dostupne front strani.U ovoj klasi su smešteni svi API pozivi namenjeni interfejsu posetilaca Web aplikacije.

**AdminController.cs** klasa sadrži API pozive namenjene interfejsu administratora Web aplikacije.

**MušterijaController.cs** klasa sadrži API pozive namenjene interfejsu mušterije Web aplikacije.

**RadnikController.cs** klasa sadrži API pozive namenjene interfejsu radnika Web aplikacije.

**DataProvider.cs** je klasa koja sadrži implementiranu logiku koja se krije iza poziva API funkcija za HotelController klasu. U ovoj klasi se vrši interakcija bekend strane sa bazom podatka.

**DataProviderAdmin.cs** je klasa koja sadrži implementiranu logiku koja se krije iza poziva API funkcija za AdminController klasu. U ovoj klasi se vrši interakcija bekend strane sa bazom podatka.

**DataProviderMusterija.cs** je klasa koja sadrži implementiranu logiku koja se krije iza poziva API funkcija za MusterijaController klasu. U ovoj klasi se vrši interakcija bekend strane sa bazom podatka.

**DataProviderRadnik.cs** je klasa koja sadrži implementiranu logiku koja se krije iza poziva API funkcija za RadnikController klasu. U ovoj klasi se vrši interakcija bekend strane sa bazom podatka.

**IDataProviderRadnik.cs** je interfejs koji se implementira u klasi **DataProviderRadnik.cs**

**IDataProviderMusterija.cs** je interfejs koji se implementira u klasi **DataProviderMusterija.cs**

**IDataProviderAdmin.cs** je interfejs koji se implementira u klasi **DataProviderAdmin.cs**

**IDataProvider.cs** je interfejs koji se implementira u klasi **DataProvider.cs**

**Models** je folder u kome se nalaze sve funkcije koje sluze za mapiranje na entitete baze.

**Controllers** je folder u kome se nalaze sve Controller klase.

**DTO** je folder u kome se nalaze sve DTO klase za prikaz, i DTO Helper klase.

**DTOObj** je folder u kome se nalaze sve DTO klase za prikaz, View deo obrazca MVC.

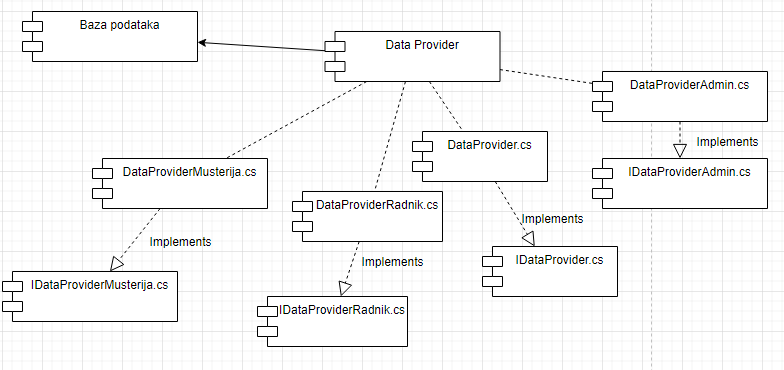
**DTOHelpers**  je folder u kome se nalaze pomocne klase za konvertovanje iz entitetskih objekata u objekte za prikaz frontendu.

**DataProvider** je folder u kome se nalaze DataProvider<imeKontrolera> klase i IDataProvider<imeKontrolera> interfejsi. Ove klase sluze za komunikaciju sa bazom.

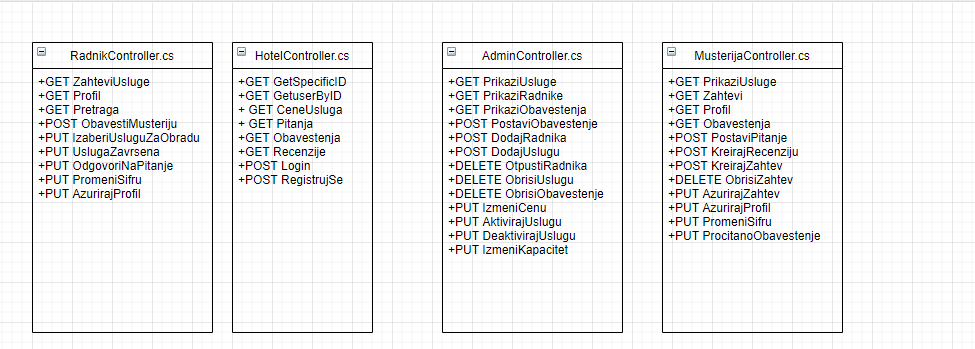
**AuthenticationService** je folder u kome se nalaze klase za autentifikaciju i autorizaciju korisnika u aplikaciji.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u api funkcije unutar *DataProvider, DataProviderAdmin,DataProviderRadnik i DataProviderMusterija* klasa koje su definisane u okviru istoimenih fajlova. U ovim klasama koriste se reference objekata iz *HotelContext.cs* klase koji su mapirani na objekte iz baze podataka. Funckije ovih klasa se pozivaju iz Controller klasa.



Na sledećim UML dijagramima klasa nabrojane su API funkcije iz Controller klasa koje predstavljaju medijum za komunikaciju frontend-a (krajnih korisinika aplikacije) sa podacima iz baze:



U referenciranom dokumentu ApiPozivi.docx se nalaze detaljni opisi svih api funkcija.

Za navedene API funkcije dat je kratak opis:

* HotelController:

## GET GetSpecificID – Pribavljanje ID-a iz tabeli Radnik/Musterija/Admin na osnovu ID-a iz tabele korisnik

* + - APi poziv: https://<imeWebSajta>/Hotel/GetSpecificID/{userID}
    - Parametri: int userID .

## GET GetUserByID – Pribavljanje korisnika na osnovu njegovog ID-a

* + - APi poziv: https://<imeWebSajta>/Hotel/GetUserByID/{userID}
    - Parametri: int userID .
  + GET CeneUsluga - Prikaz podataka o uslugama u vidu cenovnika.
    - APi poziv: https://<imeWebSajta>/Hotel/CeneUsluga
    - Parametri: Nema.
  + GET Obavestenja - Prikaz podataka o obavestenjima
    - APi poziv: https://<imeWebSajta>/Hotel/Obavestenja
    - Parametri: Nema.
  + GET Pitanja - Prikaz podataka sa svim pitanjima koje su druge mušterije postavile.
    - API poziv: https://<imeWebSajta>/Hotel/Pitanja
    - Parametri: Nema.
  + GET Recenzije - Prikaz podataka o ostavljenim recenzijama i ocenama od strane drugih mušterija.
    - API poziv: https://<imeWebSajta>/Hotel/Recenzije
    - Parametri: Nema.
  + POST Login - Prijavljivanje na sistem
    - API poziv: https://<imeWebSajta>/Hotel/Login
    - Parametri: username i sifra se šalje kao parametar iz body-ja.
  + POST RegistrujSe - Registrovanje nove mušterije na sistem.
    - API poziv: https://<imeWebSajta>/Hotel/RegistrujSe
    - Parametri: objekat klase Mušterija sa popunjenim podacima koji se šalje kao parametar iz body-ja.
* AdminController
  + GET PrikaziUsluge - Prikaz svih podataka svih usluga koje postoje u bazi.
    - API poziv: https://<imeWebSajta>/Admin/PrikaziUsluge
    - Parametri: Nema.
  + GET PrikaziObavestenja - Prikaz svih podataka svih usluga koje postoje u bazi.
    - API poziv: https://<imeWebSajta>/Admin/Usluge
    - Parametri: Nema.
  + GET PrikaziRadnike - Prikaz svih podataka o radnicima.
    - API poziv: https://<imeWebSajta>/Admin/PrikaziRadnike
    - Parametri: Nema.
  + POST PostaviObavestenje - Objavljivanje generalnog obaveštenja vidljivog svim mušterijama.
    - API poziv: https://<imeWebSajta>/Admin/PostaviObavestenje
    - Parametri: Objekat tipa Obaveštenje sa popunjenim potrebnim podacima koji se šalju kao parametar iz body-ja.
  + POST DodajRadnika - Kreiranje naloga novom zaposlenom radniku.
    - API poziv: https://<imeWebSajta>/Admin/DodajRadnika
    - Parametri: Objekat tipa Radnik kao parametar iz body-ja.
  + POST DodajUslugu - Kreiranje nove usluge.
    - API poziv: https://<imeWebSajta>/Admin/DodajUslugu
    - Parametri: Objekat tipa Usluga kao parametar iz body-ja.
  + DELETE OtpustiRadnika – Brisanje/Otpustanje radnika iz sistema.
    - API poziv: https://<imeWebSajta>/Admin/OtpustiRadnika/{id}
    - Parametri: ID radnika se šalje kao parametar url-a.
  + DELETE ObrisiUslugu - Brisanje usluge iz sistema.
    - API poziv: https://<imeWebSajta>/Admin/ObrisiUslugu/{id}
    - Parametri: ID usluge kao parametar iz url-a.
  + DELETE ObrisiObavestenje – Brisanje obavestenja iz baze.
    - API poziv: https://<imeWebSajta>/Admin/ObrisiUslugu/{id}
    - Parametri: ID usluge kao parametar iz url-a.
  + PUT IzmeniCenu - Promena cene izabrane usluge
    - API poziv: https://<imeWebSajta>/Admin/ IzmeniCenu /{id}
    - Parametri: id usluge se salje kroz url, cena kroz body
  + PUT IzmeniKapacitet - Promena kapaciteta izabrane usluge
    - API poziv: https://<imeWebSajta>/Admin/ IzmeniKapacitet /{id}
    - Parametri: i usluge se salje kroz url, a kapacitet kroz body.
  + PUT AktivirajUslugu – Usluga postaje dostupna za musterije da je rezervisu
    - API poziv: https://<imeWebSajta>/Admin/ AktivirajUslugu
    - Parametri: id usluge se salje kroz body
  + PUT DeaktivirajUslugu – Usluga postaje nedostupna za musterije da je rezervisu
    - API poziv: https://<imeWebSajta>/Admin/ DeaktivirajUslugu
    - Parametri: id usluge se salje kroz body
* RadnikController
  + GET ZahteviUsluge – Prikaz usluga svih zahteva rasclanjenih posebno
    - API poziv: https://<imeWebSajta>/Radnik/Zahtevi/{radnikID}
    - Parametri: radnik id se salje kroz url, koristi se da se prikazu usluge koje je on krenuo da obradjuje
  + GET Pretraga - Mogućnost prikaza filtriranih zahteva po nekom parametru.
    - API poziv: https://<imeWebSajta>/Radnik/Pretraga/{rezBroj}
    - Parametri: rezBroj se salje kroz url
  + GET Profil - Prikaz ličnih podataka radnika.
    - API poziv: https://<imeWebSajta>/Radnik/Profil/{id}
    - Parametri: ID radnika čiji se profil prikazuje.
  + POST ObavestiMusteriju - Slanje obaveštenja mušteriji.
    - API poziv: https://<imeWebSajta>/Radnik/ObavestiMusteriju/{id}
    - Parametri: ID mušterije kojoj se šalje obaveštenje i objekat tipa Obaveštenje. ID se šalje kao parametar url-a, a obaveštenje se šalje kao parametar body-ja.
  + PUT IzaberiUsluguZaObradu -Biranje usluge koju ce radnik obraditi.
    - API poziv: https://<imeWebSajta>/Radnik/IzaberiUsluguZaObradu
    - Parametri: Objekat koji se salji sadrzi tri atributa: rezervacioni broj zahteva,id usluge za tu uslugu i id radnika koji krece sa obradom.
  + PUT UslugaZavrsena – Prebacivanje usluge iz usluge u obradi u obradjenu uslugu
    - API poziv: https://<imeWebSajta>/Radnik/IzaberiUsluguZaObradu
    - Parametri: Objekat koji se salji sadrzi tri atributa: rezervacioni broj zahteva,id usluge za tu uslugu i id radnika koji krece sa obradom.
  + PUT OdgovoriNaPitanje - Postavljanje odgovora na mušterijino pitanje.
    - API poziv: https://<imeWebSajta>/Radnik/OdogovoriNaPitanje/
    - Parametri Objekat tipa pitanje se salje kroz body.
  + PUT AzurirajProfil - Izmena određenih podataka profila Radnika
    - API poziv: https://<imeWebSajta>Radnik/AzurirajProfil
    - Parametri: Izmenjeni podaci u vidu objekta tipa Radnik. Objekat Radnik se šalje kao parametar iz body-ja.
* MusterijaController
  + GET Zahtevi - Prikaz mušterijinih zahteva.
    - API poziv: https://<imeWebSajta>/Musterija/Zahtevi/{id}
    - Parametri: ID mušterije čiji se zahtevi prikazuju.
  + GET PrikaziUsluge - Prikaz usluga iz baze, za kreiranje cenovnika.
    - API poziv: https://<imeWebSajta>/Musterija/PrikaziUslugeParametri: ID mušterije čiji se zahtevi prikazuju.
    - Parametri: Nema
  + GET Profil - Prikaz ličnih podataka mušterije.
    - API poziv: https://<imeWebSajta>/Musterija/Profil/{id}
    - Parametri: ID mušterije čiji se profil prikazuje.
  + GET Obaveštenja - Prikaz svih mušterijinih obaveštenja.
    - API poziv: https://<imeWebSajta>/Musterija/Obavestenja/{id}
    - Parametri: ID musterije čija se obaveštenja prikazuju.
  + POST PostaviPitanje - Postavljanje pitanja u odeljku za pitanja.
    - API poziv: https://<imeWebSajta>/Musterija/PostaviPitanje
    - Parametri: Objekat tipa Pitanje koji se šalje kao parametar iz body-ja.
  + POST KreirajRecenziju - Kreiranje recenzije i davanje ocene o hotelu generalno.
    - API poziv: https://<imeWebSajta>/Musterija/KreirajRecenziju
    - Parametri: Objekat tipa Recenzija koji se šalje kao parametar iz body-ja.
  + POST KreirajZahtev: Kreiranje novog zahteva.
    - API poziv: https://<imeWebSajta>/Musterija/Kreirajzahtev
    - Parametri:Objekat tipa Zahtev koji se šalje kao parametar iz body-ja.
  + DELETE ObrisiZahtev - Brisanje izabranog zahteva.
    - API poziv: https://<imeWebSajta>/Musterija/ObrisiZahtev
    - Parametri:Objekat tipa Zahtev koji se šalje kao parametar iz body-ja.
  + PUT AzurirajZahtev - Izmena nekih podataka ukoliko zahtev nije krenuo sa obradom.
    - API poziv: https://<imeWebSajta>/Musterija/Azurirajzahtev
    - Parametri:Objekat tipa Zahtev koji se šalje kao parametar iz body-ja.
  + PUT AzurirajProfil - Izmena određenih podataka profila Mušterija
    - API poziv: https://<imeWebSajta>Musterija/AzurirajProfil
    - Parametri: Izmenjeni podaci u vidu objekta tipa Mušterija.Objekat Mušterija se šalje kao parametar iz body-ja.
  + PUT ProcitanoObavestenje - Oznaka da se procitano obavestenje vise ne prikazuje
    - API poziv: https://<imeWebSajta>Musterija/ProcitanoObavestenje/{idObavestenja}
    - Parametri: id obavestenja se salje kroz url

# 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]:

Sistem će da podrži do 1000 simultanih pristupa korisnika portalu.

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. Web aplikacija Hotel Petsylvania će biti dostupna 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.