E-hotel

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 24.04.2020. | 1.0 | Inicijalna verzija | Stefan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Pregled osnovnih podataka o laboratoriji 8

6.2.3 Pregled spiska članova 8

6.2.4 Pregled podataka o određenom članu laboratorije 8

6.2.5 Pregled publikacija po autoru 8

6.2.6 Pregled publikacija po tipu 8

6.2.7 Pregled publikacija po godini 8

6.2.8 Pregled spiska projekata 8

6.2.9 Pregled podataka o određenom projektu 8

6.2.10 Prijavljivanje 9

6.2.11 Ažuriranje podataka o članu 9

6.2.12 Dodavanje nove publikacije 9

6.2.13 Brisanje postojeće publikacije 9

6.2.14 Ažuriranje osnovnih podataka o laboratoriji 9

6.2.15 Kreiranje novog člana 9

6.2.16 Brisanje postojećeg člana 9

6.2.17 Arhiviranje postojećeg člana 9

6.2.18 Kreiranje projekta i postavljanje vođe 9

6.2.19 Ažuriranje podataka o projektu 9

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10

7.1.1 Korisnički interfejs 10

7.1.2 Aplikaciona logika 10

7.1.3 Pristup podacima 10

7.1.4 HTML 10

7.1.5 PHP 11

7.1.6 MySQL 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 PHP-CGI 11

8.1.4 MySQL Server 12

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

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 14

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 E-hotel softverskog sistema.

# Opseg dokumenta

Dokument se odnosi na E-hotel softverski sistem koji će biti razvijen od strane tima 016. Namena sistema je omogućavanje online rezervacija, preuzimanje ključa do sobe i korišćenja hotelskih usluga.

# Reference

Spisak korišćene literature:

1. E-hotel – Predlog projekta, V1.0, 2020, 016.
2. E-hotel – Planirani raspored aktivnosti na projektu, V1.0, 2020, 016.
3. E-hotel – Plan realizacije projekta, V1.0, 2020, 016.
4. E-hotel – Vizija sistema, V1.0, 2020, 016.
5. E-hotel – Specifikacija zahteva, V1.0, 2020, 016.

# 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. E-hotel softverski sistem će biti implementiran kao Web aplikacija zasnovana na Node.js serveru, React-u i MongoDB kao bazi podataka [4].
2. Klijentski deo PeNcIL portala će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, Google chrome, kao i Firefox (Mozilla) [4].
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].

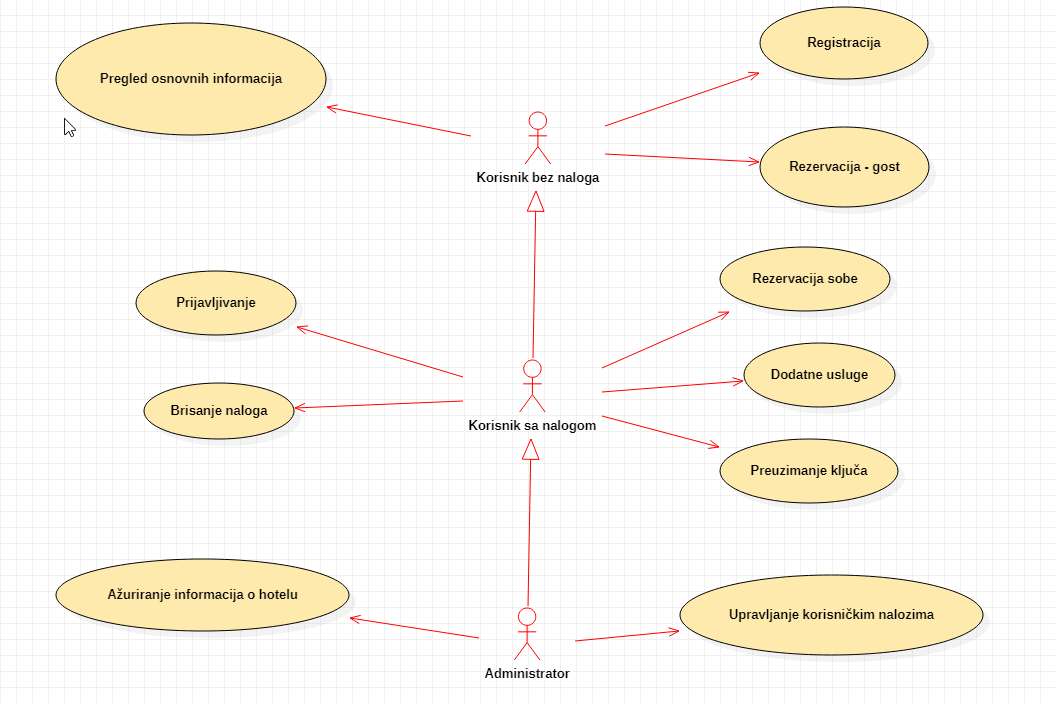
Slučajevi korišćenja E-Hotel web aplikacije su:

* *Pregled osnovnih informacija o hotelu*
* *Registracija*
* *Prijavljivanje*
* *Rezervacija sobe*
* *Korišćenje hotelskih usluga*
* *Brisanje naloga*
* *Kreiranje,izmena i brisanje informacija o hotelu*
* *Preuzimanje ključa*
* *Rezervacija sobe u hotelu*
* *Ažuriranje informacija o sobama*
* *Ažuriranje informacija o dodatnim hotelskim uslugama*
* *Brisanje naloga korisnika*

Ove slučajevi korišćenja mogu da iniciraju posetilac portala, korisnik sa nalogom ili administrator.

## Dijagrami slučajeva korišćenja

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



## Kratak opis slučajeva korišćenja

### Pregled osnovnih informacija o hotelu

Kratak opis: Prikaz stranice sa osnovnim informacijama o hotelu.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala(gost), Korisnik sa nalogom, Administrator.

### Registracija

Kratak opis: Prikaz stranice sa formom za registrovanje novih korisnika.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala(gost), Korisnik sa nalogom, Administrator.

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju. Akteri koji iniciraju slučaj korišćenja: Posetilac portala(gost), Administrator.

### Rezervacija sobe

Kratak opis: Prikaz opštih informacija o mogućnosti za rezervaciju.

Akteri koji iniciraju slučaj korišćenja: Korisnik sa nalogom, Administrator.

### Korišćenje hotelskih usluga

Kratak opis: Prikaz stranice sa uslugama mogućim za korišćenje.

Akteri koji iniciraju slučaj korišćenja: Korisnik sa nalogom, Administrator.

### Brisanje naloga

Kratak opis: Brisanje podataka iz baze.

Akteri koji iniciraju slučaj korišćenja: Korisnik sa nalogom, Administrator.

### Kreiranje, izmena i brisanje informacija o hotelu

Kratak opis: Ažuriranje informacija o hotelu.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Preuzimanje ključa

Kratak opis: Preuzimanje koda koji otvara sobu.

Akteri koji iniciraju slučaj korišćenja: Korisnik sa nalogom

### Rezervacija sobe u hotelu

Kratak opis: Rezervacija sobe u hotelu.

Akteri koji iniciraju slučaj korišćenja: Posetilac(gost).

### Ažuriranje informacija o sobama

Kratak opis: Ažuriranje informacija u bazi o dostupnosti odrađenih soba sa mogućnošću ubacivanja nove sobe u bazu podataka.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje informacija o dodatnim hotelskim uslugama

Kratak opis: Ažuriranje informacija u bazi o dostupnosti pojedinih hotelskih usluga sa mogućnošću ubacivanja novog tipa hotelske usluge u bazu podataka.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje naloga korisnika

Kratak opis: Brisanje naloga određenog korisnika 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.

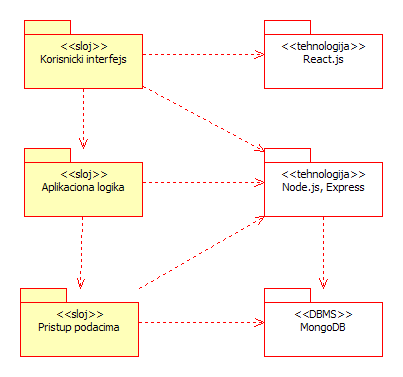
Logički pogled na E-hotel softverski sistem obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice 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 Node.js API zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži mongoose paket koji 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 sistema. U njemu je sadržan sav HTML, multimedijalni sadržaj i Javascript koji generiše HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i Node.js.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi E-hotel softverskog sistema. Sadrži Node.js API koji realizuje funkcionalnost karakterističnu za domen primene sistema i uspostavlja vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i Node.js paketa.

### Pristup podacima

sloj

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

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa Node.js i MongoDB baza podataka.

### React.js

tehnologija

Tehnologija React-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.

### Node.js

tehnologija

Tehnologija Node.js-a obezbeđuje mehanizam za pisanje i izvršavanje koda na strani servera. Ovaj kod opslužuje zahteve koji pristižu od klijenata i pristupa bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### MongoDB

DBMS

MongoDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju E-hotel softverskog sistema.

### Express

tehnologija

Tehnologija Express služi za rutiranje u okviru Node.js API.

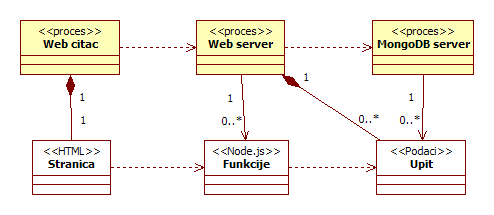
# Pogled na procese

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.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje E-hotel softverskog sistema kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju E-hotel sistema. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na Node.js i MongoDB bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za kreiranje i prikaz HTML stranica od podataka dobijenih od nekog Web servera u vidu JSON objekta. 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 odgovarajuće poddatke na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Web server rutira zahteve i prosledjuje ih odgovarajućim funkcijama koje će klijentu poslati tražene podatke potrebne za izradu web stranice ili proslediti funkcionalnost ka bazi podataka.

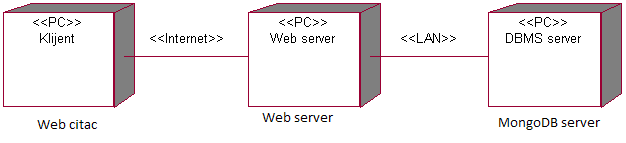
### MongoDB Server

MongoDB Server je proces koji izvršava funkcionalnost MongoDB 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 E-hotel softverskog sistema.



## Klijent

Pristup E-hotel softverskom sistemu 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). Moguća je i konfiguracija u kojoj je server na udaljenoj mašini.

## DBMS server

DBMS server je računar na kome se izvršava MongoDB Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka.

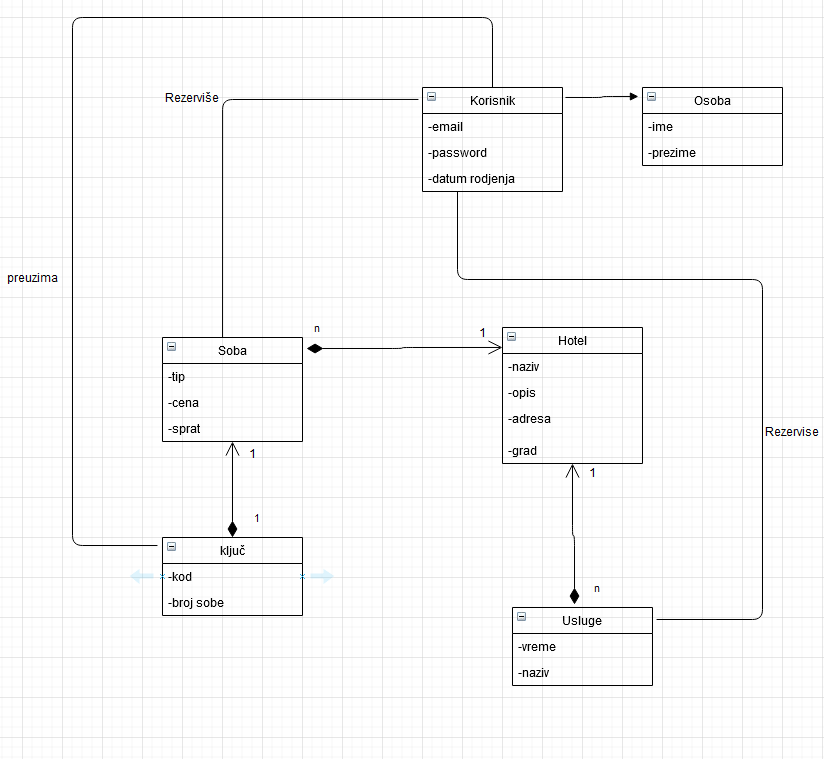
# Pogled na implementaciju sistema

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

## Model domena

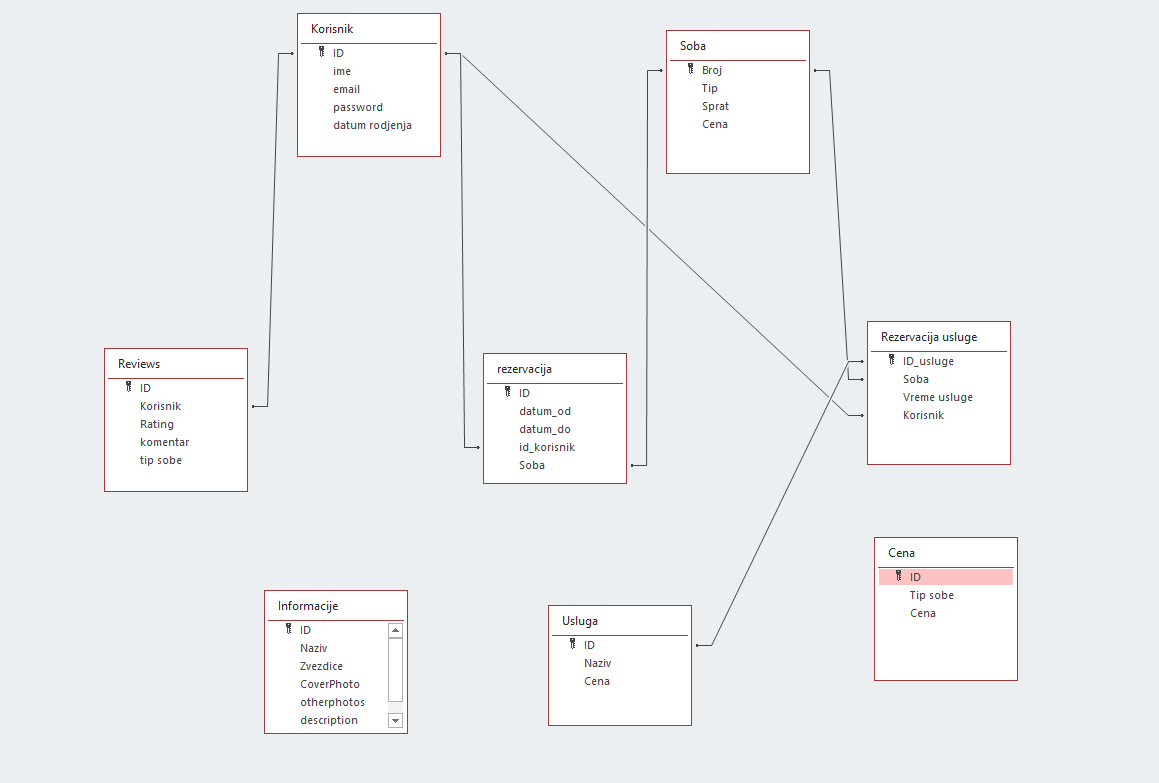
Model domena za koji se E-hotel portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *MS Access*-a, dok je migracija na MySQL obavljena pomoću *MySQL Migration Toolkit*-a.



## Komponente sistema

*NAPOMENA: E-hotel projekat nije kompletno implementiran, tj. implementiran je na nivou arhitekturnog prototipa. Zbog ovoga sledeću sekciju trebate shvatiti kao ilustraciju kako bi dokument trebao da izgleda. Opis komponenti sistema za vaše projekte mora biti kompletan! Procena složenosti vašeg rešenja će zavisiti od opisa sistema koji ovde date.*

Komponente sistema E-hotel portala su NODEJS skripti čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je NODEJS 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

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

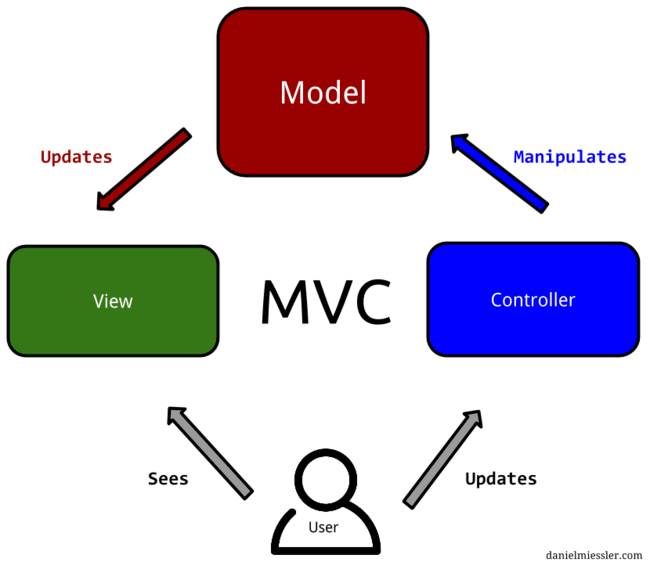


Komponenta **index.js** je implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu. Za izgled se koristi built-in blade template sistem koji je ubačen u NodeJS Framework.

Komponenta **main.js** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Index.NodeJS kreira Singleton instancu koja sadrži sve potrebne parametre sa filtriranje zahteva, rutiranje i aplikacionu logiku. Na sledećoj slici je ilustrovana aplikaciona logika Framework-a koji se koristi u E-hotel aplikaciji:



### Komponente za pristup podacima

Bazi podataka se pristupa preko Modela kreiranih u MVC arhitekturi (minimum svaka roditeljska tabela treba da ima svoj model). Koristi se *Object Relational Model.* Pomenuti skript se uključuje na početku index -a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.

Svaki model nasledjuje klasu *Model* koja sadrži sve metode koji u potpunosti zamenjuju *RAW* izvršenje SQL naredbi ka bazi podataka.

# 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 10000 simultanih pristupa korisnika portalu.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 3 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. E-hotel softverski sistem će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 5%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.