**UNIVERZITET SINGIDUNUM**

**Fakultet za informatiku i računarstvo**

**Projektni zadatak**

**APLIKACIJA ZA HOTEL**

̶ Projekat iz Praktikum - Internet i veb tehnologije ̶

**Profesor: Kandidat:**

prof. dr  *Vladislav Miškovic Una Svetlana Tuba*

**Asistent:** *2017/204277*

Milan Tair

**Beograd, *2020*.**

Sadržaj

[Uvod 3](#_Toc44862690)

[Cilj razvoja 3](#_Toc44862691)

[Obim sistema 3](#_Toc44862692)

[Prikaz proizvoda 4](#_Toc44862693)

[Perspektiva proizvoda 4](#_Toc44862694)

[Funkcija proizvoda 4](#_Toc44862695)

[Karakteristike korisnika 4](#_Toc44862696)

[Ograničenja 4](#_Toc44862697)

[Reference 5](#_Toc44862698)

[Specifikacija zahteva 5](#_Toc44862699)

[Spoljašnji inerfejsi 5](#_Toc44862700)

[Funkcije 5](#_Toc44862701)

[Pogodnost za upotrebu 5](#_Toc44862702)

[Zahtevane performance 5](#_Toc44862703)

[Zahtevi baze podataka 6](#_Toc44862704)

[Projektna ograničenja 6](#_Toc44862705)

[Sistemske karakteristike softvera 6](#_Toc44862706)

[Dopunske informacije 6](#_Toc44862707)

[Verifikacija 6](#_Toc44862708)

# Uvod

## Cilj razvoja

Cilj razvoja sistema je omogućavanje administrativnom osoblju hotela da uređuje spisak soba i sala kojim jedan hotel raspolaže kao i da vrši rezervacije istih. Sistem prikazuje administrativnom osoblju termine kada su određene sobe ili sale zauzete.

## Obim sistema

Sistem realizuje sledeće funkcije iz realnog sistema:

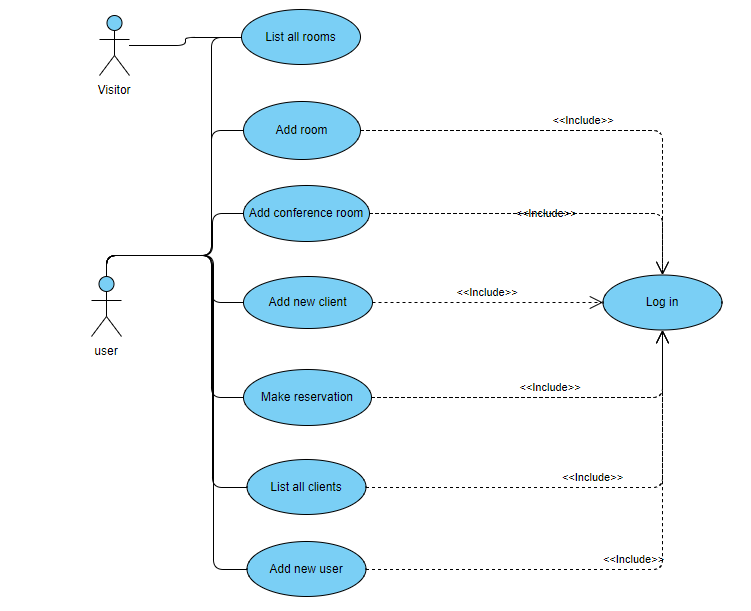
* unos podataka o novim sobama (za sobe broj kreveta, vrstu kreveta/struktura, izlaz na terasu, strana sveta ka kojoj su orijentisani prozori, sprat, dodatne pogodnosti kao što su besplatan Internet, garderober, prikaz iz odvojenog hodnika i slično)
* unos podataka o novim konferencijskim salama (broj mesta, površinu, posedovanje bine/govornice, projektora za prezentacije, ozvučenja i slično)
* unos podataka o klijentima na čije ime se vrši rezervacija
* rezervaciju soba i sala,
  + proveru da li je soba ili sala slobodna u zahtevanom terminu
  + rezervisanje sobe ili sale za klijenta iz sistema
* unos podataka o novim članovima administrativnog osoblja
* prikaz spiska dostupnih soba i sala za konferencije, sa podacima o ukupnom broju raspoloživih kreveta, mesta itd.
* prikazivanje spiska slobodnih i zautetih soba u određenom vremenskom period

## Prikaz proizvoda

### Perspektiva proizvoda

Sistem ne predviđa mogućnost povezivanje sa drugim sistemima iz okruženja.

### Funkcija proizvoda



### Karakteristike korisnika

Planirano je da sistem može biti korišćen od strane osoba koje imaju osnovno poznavanje rada na računaru. Potrebna je kratka obuka koja bi prikazala administrativnom osoblju šta sve postoji i na koji način se koristi.

### Ograničenja

* Potrebno je da se za svaku sobu i salu čuvaju svi predviđeni podaci, odnosno ne postoji mogućnost da se ne zna da li soba poseduje wifi, sto i slično.
* Svaka rezervacija se vodi na tačno jednog klijenta čiji su podaci zavedeni u sistemu. Niije moguće rezervisati istu prostoriju od strane više klijenata u istom vremenskom periodu (preklapajućem periodu).
* Sve prostorije (sobe i sale) se rezervišu na minimum jedan da i rezervacija važi od 00:00 jednog dana do 23:59 drugog dana.
* Nakon rezervacije, nije moguće menjati cenu prostorije.
* Podaci o rezervacijama i klijentima su isključivo dostupni administrativnom osoblju koji su identifikovani pomoću korisničkog imena i lozinke.

# Reference

Projektni zadatak sa zahtevima:

<http://zadatak.singidunum.ac.rs/app/piivt-biranje-tema/?action=topic&id=24>

# Specifikacija zahteva

## Spoljašnji inerfejsi

Nije predviđeno povezivanje sistema sa drugim sistemima iz okruženja.

## Funkcije

List all rooms: Za odabrani period izlistavaju se sve sobe u hotelu pri čemu sobe koje nisu slobodne u odabranom period se označavaju na poseban način i ne postoji opcija pravljenja rezervacije.

Add room: Upisuju se svi podaci o jednoj sobi i u bazi se kreira zapis sa tim podacima.

Add conference room: Upisuju se svi podaci o jednoj konferencijskoj sali i u bazi se kreira zapis sa tim podacima.

Add new client: Upisuju se svi podaci o jednom klijentu i u bazi se kreira zapis sa tim podacima. Onemogućeno je dupliranje klijenta pri čemu se dva klijenta smatraju istim ukoliko imaju istu email adresu.

Make reservation: Bira se od/do datum rezervacije, soba (odnosno sala) i klijent na čije ime se vrši rezervacija. Nakon odabir svih podataka, u bazi se kreira zapis sa ovom rezervacijom.

List all clients: Ispisuju se podaci o svim klijentima koji se trenutno nalaze u sistemu.

Add new user: Upisuju se svi podaci o jednom članu administrativnog osoblja i u bazi se kreira zapis sa tim podacima. Onemogućeno je dupliranje članova pri čemu se dva klijenta smatraju istim ukoliko imaju istu email adresu. Nije moguće imati dva klijenta sa istim korisničkim imenom.

Log in: Verifikacija administrativnog osoblja pomoću korisničkog imena i lozinke.

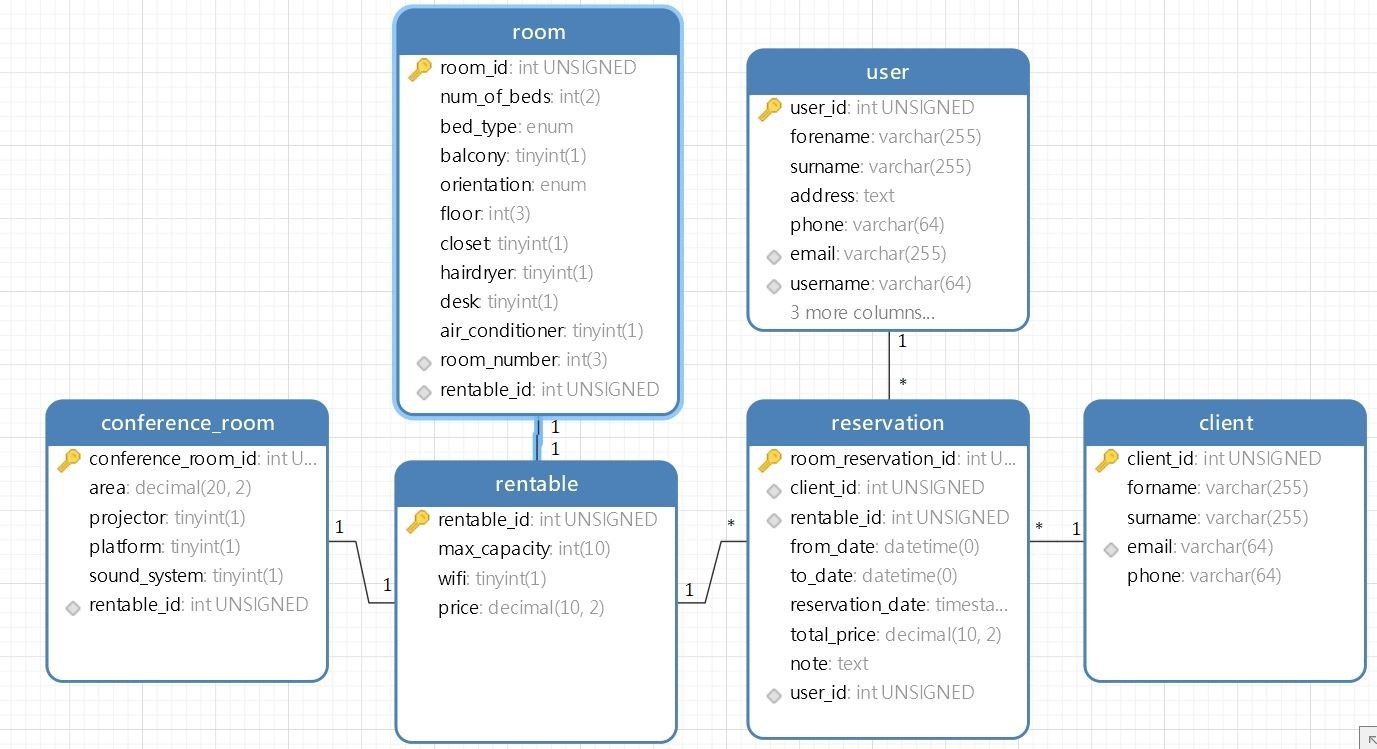
## Pogodnost za upotrebu

Potrebno je da sistem bude *user friendly* odnosno da sve forme za unos budu jasne i razumljive. Onemogućiti unošenje slobodnog teksta za podatke koji su tima ima/nema (bool). Smanjiti mogućnost pravljenja greške od strane administrativnog osoblje na minimum pri čemu smatrati da oni nemaju nikakva znanja i obrazovanja u računarstvu, već poseduju osnovno poznavanje rada na računaru.

## Zahtevane performance

Grafički interfejs veb sajta treba da bude realizovan sa responsive dizajnom.

## Zahtevi baze podataka



## Projektna ograničenja

Za izradu projekta nisu potrebna finansijska sredstva. Za razvoj veb aplikacija, potrebni su sledeći alati:

* Editor koda: Visual Studio Code Editor
* Server baze podataka: MySQL Server
* Klijent za bazu podataka: Navicat ili HeidiSQL
* Alat za verziranje koda: Github
* Platforma za izvršavanje programa: Node.js
* Internet pregledač: Google Chrom ili Firefox Mozilla

## Sistemske karakteristike softvera

## Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Nest.js razvojnog okvira i sav kôd aplikacije treba da bude organizovan prema pravilima MVC arhitekture. Baza podataka mora da bude relaciona i treba koristiti MySQL/MariaDB RDBMS. Sav generisani HTML kôd koji proizvodi aplikacija mora da bude 100% validan, tj. da generisani kôd prođe proveru W3C Validatorom (dopuštena su upozorenja, ali ne i greške). Aplikacija može grafički korisnički interfejs da generiše na strani servera, korišćenjem šablona za generisanje HTML koda (proizvoljan templating engine, integrisan sa Nest.js aplikacijom) ili da bude serviran statički deo stranice koji pomoću JavaScript-a dinamički formira komponente na front-end-u, a podatke doprema asinhrono kroz veb servis (API) metode obezbeđene u okviru same aplikacije.

## Potrebno je obezbediti određeni stepen provere podataka koji se od korisnika upućuju aplikaciji. Moguća su četiri sloja zaštite i to: (1) HTML pattern u poljima za unos podataka u formularima; (2) JavaScript validacija vrednosti unetih u polja za unos podataka u formularima na front-end-u; (3) Provera korišćenjem adekvatnih testova ili korišćenjem regularnih izraza na strani servera u Node.js aplikaciji (moguće je i korišćenjem izričitih šema - Schema) i (4) provera na nivou baze podataka korišćenjem okidača nad samim tabelama baze podataka.

## Neophodno je napisati prateću projektnu dokumentaciju o izradi aplikacije koja sadrži (1) model baze podataka sa detaljnim opisom svih tabela, njihovih polja i relacija; (2) dijagram organizacije elemenata aplikacije sa akcentom na isticanje MVC arhitekture na konkretnom primeru jednog odabranog zahteva, tj. rute/putanje koju Vaša aplikacija obrađuje; (3) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; (4) popis svih kontrolera i njihovih metoda koji obavljaju potrebnu programsku i poslovnu logiku sa ciljem izvršavanja svih predviđenih aktivnosti aplikacije, kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije objavljenom na stranici predmeta u sekciji sa predavanja.

## Izrada projekta mora da bude sprovođena korišćenjem alata za verziranje koda Git, a kompletan kôd aplikacije bude dostupan na javnom Git repozitorijumu, npr. na besplatnim GitHub ili Bitbucket servisima. Ne može ceo projekat da bude otpremljen u samo nekoliko masovnih Git commit-a, već mora da bude pokazano da je projekat realizovan u kontinuitetu, da su korišćene grane (branching), da je bilo paralelnog rada u više grana koje su spojene (merging) sa ili bez konflikata (conflict resolution).

## Dopunske informacije

Za prikazani sistem nema dopunskih informacija.

# Verifikacija

Za verifikaciju ispravnosti back end dela koristiti POSTMAN i rezultate priložiti uz dokumentaciju.