**Univerzitet Singidunum**

**Fakultet za Informatiku I računarstvo**

**Projektovanje softvera**

**Veb aplikacija za hotel**

Profesor

Prof.Dr. Violeta Tomašević Studenti

Asistent Dušan Ilić 2018201006

Petar Jakić Lazar Janićijević 2016201911

Sadržaj

[1 Kratak opis 3](#_Toc106450897)

[2 Postupak razvoja softvera 3](#_Toc106450898)

[2.1 Kritične tačke: 3](#_Toc106450899)

[I 4](#_Toc106450900)

[II 4](#_Toc106450901)

[III 5](#_Toc106450902)

[IV 5](#_Toc106450903)

[3 Analiza zahteva 5](#_Toc106450904)

[3.1 Prikupljanje zahteva 5](#_Toc106450905)

[3.2 Funkcionalni zahtevi 6](#_Toc106450906)

[3.3 Zahtevi po pitanju performansi 6](#_Toc106450907)

[4 Projektovanje sistema 8](#_Toc106450908)

[4.1 Stil projektovanja 8](#_Toc106450909)

[4.2 Dijagram aktivnosti (*login*) 9](#_Toc106450910)

[4.3 Projektovanje programskog koda 11](#_Toc106450911)

[4.4 Završno projektovanje 11](#_Toc106450912)

[5 Implementacija softvera 11](#_Toc106450913)

[5.1 Lokalizacija podataka (ulaz-izlaz) 11](#_Toc106450914)

[5.2 Struktura podataka (baza podataka) 12](#_Toc106450915)

[5.3 Unutrašnja dokumentacija 13](#_Toc106450916)

[6 Testiranje softvera 14](#_Toc106450917)

[6.1 Jedinično testiranje komponenata 14](#_Toc106450918)

[Konverzija podataka 15](#_Toc106450919)

[Greška uklanjanja rezervacije 16](#_Toc106450920)

[6.2 Integraciono testiranje 17](#_Toc106450921)

[6.3 Završetak testiranja 17](#_Toc106450922)

[7 Isporuka softvera I obuka korisnika 18](#_Toc106450923)

[Korisničko uputstvo 18](#_Toc106450924)

# 1 Kratak opis

U ovom projektu kreirali smo aplikaciju na zahtev klijenta koja je namenjena korisnicima usluge hotela, gde će korisnici moći da se prijave, da rezervišu I otkazuju rezervacije, da pregledaju sve ponude hotela. Zatim kreiranje naloga za korisnike koji će imati privilegije da menjaju informacije na aplikaciji, da menjaju I dodaju sobe, konferencijske sale kao I rezervacije korisnika . Vodili smo računa I ostavili mogućnost da se isporučeni softver dalje razvija I unapređuje sa strane načina njegovog hostovanja tačnije prelazak na OpenShift platformu koja bi automatski skalirala resurse servera, te su tehnologije korišćene u skladu sa tim.

# 2 Postupak razvoja softvera

U ovom projektu izabrali smo model inkrementalnog faznog razvoja.Ovaj model smo izabrali da bismo mogli da prikazujemo korisniku napredak na radu razoja aplikacije, a i da ukoliko je nečim nezadovoljan, možemo blagovremeno da promenimo. Zatim smo u procesu razvoja na osnovu odziva klijenta mogli da menjamo raspored komponenti i način izgleda. Dok smo dodatne funkcionalnosti mogli lako da ubacujemo.

## 2.1 Kritične tačke:

* **I** – (22.5.2022)

Detaljna analiza, plan projektovanja i modelovanja za projekat

* **II** –(29.5.2022)

Pristupanje pravljenja osnovnog modela projekta sa ključnim funkcionalnostima.

Programiranje.

* **III** –(4.6.2022)

Ispravljanje grešaka, uvođenje naprednijih i složenijih funkcionalnosti, nastavak kodiranja kao i unapređivanje postojećeg koda radi veće efikasnosti i jednostavnosti,testiranje.

* **IV** – (11.6.2022) pregled dokumentacije, ponovna provera sintakse koda, promatranje mogućih unapređenja sa strane hostovanja same aplikacije kao i moguća buduća unapređenja ukoliko klijent bude zahtevao dodatne bezbednosne mere, finalizacija celog projekta.

### I

22.5.2022

* Projektovanje baze podataka.
* Razmatrali smo kojim ćemo se alatima koristiti i kako je najefikasnije, za nas koji pravimo aplikaciju, a potom i kako će biti administratorima aplikacije na strani klijenta biti najlakše administriranje aplikacije. U smislu programskog jezika i samih tehnologija kojima vladaju, sve u prevenciji toga da ne dođe do upotrebe tehnologija i programskih jezika koje administratori klijentske strane neće poznavati ili će im biti potrebna obuka.
* Pošto smo znali da će se aplikacija sastojati iz tri funkcionalne celine(*frontend*,*backend*,baza). Počeli smo da razmatramo u kojim smo tehnologijama najstručniji i koje možemo najbrže implemetirati. U dogovoru sa klijentom zaključili smo da ćemo za frontend koristiti JavaScript-ov Vue.js frejmvork, za backend Node.js i da ćemo za bazu koristiti ,ne standardn, relacionu bazu, već NoSQL MongoDB.
* Ciljevi ove nedelje jesu pravljenje skeleta samog projekta u vizuelnoj formi i detaljna analiza zahteva klijenta. Cilj je takođe bio da u sledeću fazu uđemo što je više moguće spremni i da minimizujemo propuste i šansu na vraćanje na prvi deo, pokušali smo da smanjimo što je moguće više propust nekog dela, koji bi nam kasnije oduzimao dodatno vreme za ispravku.

### II

29.5.2022

* Pošto smo definisali kako će sama aplikacija funkcionisati i koje ćemo alate koristiti, pristupili smo kodiranju same aplikacije.
* Radi boljeg odziva i uključivanja klijenta u projekat kreiranje same veb stranice, odnosno kako će izgledati stilistički.
* Komunikacija sa klijentom nakon kreiranja veb stranice, da bi se susreli sa zahtevima kljijenta. Ukoliko je klijent nečim nezadovoljan, da to i promenimo.
* Kreiranje baze sa svim njenim atributima i poljima, povezivanje frontend-a sa backend-om. Programiranje backend programa u Node.js, kreiranje osnovnih metoda funkcionalnosti. Postizanje funkcionalnosti backenda i servera baze podataka.
* Kodiranje u paru.
* Sugestije oko samog koda i šta je potrebno dodati ispraviti ili izbaciti.

### III

4.6.2022

* Finalizacija backenda i baze podataka.
* Inspekcija koda. Primena automatizovanih procesa i generički pristup kreiranju metoda u zamenu za osnovne funkcionalnosti prvobitno napisane, primena REST frejmvorka u zamenu za http metode.
* Povećanje efikasnosti koda i čitljivosti.
* Testiranje svih funkcionalnosti i delova koda.
* Testiranje otpornosti apikacije na otkaze kao i povezanosti komponenata.
* Ispravljanje svih grešaka.
* Nakon implementacije potrebnih promena, pregledu koda i kratak osvrt na sve što je urađeno.

### IV

11.6.2022

* Kreiranje više opcija kao strategije hostovanja sajta i mogućih budućih unapređenja sa strane otpornosti sajta kao i bezbodnosnih unapređenja.
* Komunikacija sa klijentom u vidu izbora strategije hostovanja sajta.
* Provera dokumentacije.
* Provera sintakse, i strukture koda.
* Finalizacija projekta.

# 3 Analiza zahteva

## 3.1 Prikupljanje zahteva

Klijent je definisao da želi 3 vrste korisnika veb aplikacije. Administratora koji će imati pristup svim komponentama aplikacije tom prilikom misleći na izvorni kod i celokupno okruženje aplikacije. Radnike koji će imati mogućnost da menjaju dodaju i brišu sobe, sale i korisnike. Krajnje korisnike koji će biti direktni korisnici ovih usluga. Krajnji korisnici bi trebalo da imaju mogućnost rezervacija soba i sala kao i informacije o tome koje su sobe-sale zauzete ili slobodne i u kojim terminima, takođe da se krajnjim korisnicima onemoguće rezervacije soba-sala u terminima koje je već neko rezervisao. Kupac je zahtevao i da je nemoguć pristup samom sadržaju aplikacije ako korisnik pre toga nije autentifikovan. Dakle ukoliko korisnik nema nalog ne može da pristupi sajtu. Ukoliko korisnik želi da pristupa sadržaju aplikacije mora postojati i mogućnost da se registruje. Za same sobe i sale definisao je skup zahteva koji trebaju biti ispunjeni za sobe naglasio je da je neophodno da imaju sledeće atribute koji će biti podložni izmenama od strane radnika (cena,slika,status,datum od-do kad je rezervisana soba, broj kreveta, tip kreveta, da li ima terasu, orijentacija sobe, sprat, dodatne opcije kao što su internet konekcija, garderober, koliko ima zvezdica), za konferencijske sale ( ime ,cena,slike,status,kada je rezervisana, površina, broj sedišta, podijum, projektor, ozvučenje, broj zvezdica ).

## 3.2 Funkcionalni zahtevi

1. Administrator koji će imati direktan pristup izvornom kodu veb aplikacije.
2. Zaposleni koji će morati da se autentifikuje pre pristupa veb aplikaciji.
3. Zaposleni će moći da menja,dodaje i briše sobe i konferencijske sale.
4. Zaposleni ima pristup svim informacijama o sobama i konferencijskim salama.
5. Zaposleni će moći da ima pristup podacima regularnih korisnika koji imaju nalog.
6. Krajnji korisnik ukoliko želi da ima pristup pristup sajtu mora da kreira nalog.
7. Krajnji korisnik može da rezerviše sobe i sale.
8. Krajnji korisnik ima informacije o svim sobama i salama.
9. Krajnji korisnik ne može da rezerviše sobe i sale koje su već rezervisane.
10. Krajnji korisnik treba da ima informaciju o tome od kad do kad je soba ili sala rezervisana.
11. Krajnji korisnik na osnovu odabranog termina za koji želi da rezerviše sobu ili salu dobija automatski proračunatu cenu rezervacije.
12. Krajnji korisnik može da izabere dodatne usluge za sobe i sale koje se dodatno naplaćuju.

## 3.3 Zahtevi po pitanju performansi

Kupac je zahtevao da aplikacija bude pre svega pouzdana i laka za održavanje. Nije želeo ništa neprovereno od tehnologija koje će se primeniti.

Kupac je želeo kratko vreme odziva aplikacije, i želeo je da se minimalizuje šansa zloupotrebe podataka kao i informacija o klijentima.

Kupac je želeo aplikaciju koja ukoliko je potrebno može lako da se modifikuje i unapredi sve u zavisnosti od budućeg poslovnog uspeha kupčevog biznisa. Ostavio je mogućnost ponovnog angažmana radi poboljšanja performansi aplikacije i dodatnih funkcionalnosti ukoliko zahtevi korisnika porastu.

Slučaj korišćenja **–** Korisnika i Zaposlenog koji vrši operacije nad podacima u aplikaciji.

Osnovni scenario

1. Korisnik dolazi na početnu stranicu veb sajta.
2. Korisnik ima nalog, unosi svoje parametre za pristup.
3. Sistem proverava da li su parametri za pristup ispravni.
4. Sistem autentifikuje korisnika i prosleđuje ga na stranicu veb sajta.
5. Korisnik gleda dostupne sobe i konferencijske sale.
6. Korisnik bira datum za koji želi da rezerviše sobu-konferencijsku salu.
7. Sistem proverava unete parametre.
8. Sistem potvrđuje rezervaciju i označava da je soba-sala zauzeta u izabranom terminu.

Proširenja

2a. Korisnik nema nalog, odlazi na stranicu za registraciju.

3a. Korisnik je uneo pogrešne parametre, sistem ga vraća na stranicu za logovanje.

6a. Korisnik je izabrao pogrešan datum vraća se na korak 6.

6b. Korisnik je izabrao pogrešan datum, vraća se na rezervaciju i bira novi datum ako je slobodan.

Diagram

Description automatically generated

Ilustracija 1 Slučaj korišćenja korisnika i zaposleno

# 4 Projektovanje sistema

## 4.1 Stil projektovanja

Za ovu aplikaciju izabrana je klijent-server arhitektura, jer je aplikacija orijentisana ka korisnicima koje će aplikacija kupca softvera opsluživati. Na osnovu analize zahteva, prepoznate su sledeće komponente sistema: Korisnik,Zaposleni,Soba,Konferencijska sala i Rezervacija. Na osnovu ovih zahteva bilo je jasnije kako će sama funkcionalnost programa otprilike izgledati uz dodatne analize i popravke.

Na slici ispod (il. 2) je prikaz arhitekture baze podataka, i funkcionalnosti svake od tabela i njihovih relacija.

A picture containing timeline

Description automatically generated

Ilustracija 2 Nacrt baze podataka

## 4.2 Dijagram aktivnosti (*login*)

Prikaz processa prilikom logovanja na svoj nalog zaposlenog ili korisnika.Zaposlenom je onemogućeno kreiranje naloga jer je to postupak koji mora da odobri samo administrator sajta zbog dodatnih privilegija koje zaposleni ima.

Diagram

Description automatically generated

Ilustracija 3 dijagram aktivnosti

## 4.3 Projektovanje programskog koda

Za projektovanje ove aplikacije izabrana je NoSQL (MongoDB) ne-relaciona baza podataka zbog svojih performasi, jednostavnijeg upravljanja I modifikovanja. *Backend* ove aplikacije se izvršava u Node.js , JavaScript programskom jeziku. Za Node.js je napravljen izbor zbog kompatibilnosti sa *frontend* delom koji je rađen u JavaScript jeziku, na frejmvorku Vue.js. Za Node.js takođe postoje pouzdani *image*-I koji se lako mogu iskoristiti , uz male promene, za prebacivanje aplikacije iz monolitne u mikroservisnu, pošto je klijent naglasio da bi možda u budućnosti želeo da prebaci svoju aplikaciju u mikroservisnu. Takođe ovaj programski jezik I frejmvorci korišćeni su kompatibilni sa postojećim *image*-ima koji postoje na RedHat OpenShift I Docker proverenim repozitorijumima.

## 4.4 Završno projektovanje

Kao I kod svake klijent-server arhitekture performanse I odziv aplikacije će u velikoj meri zavisiti od servera koji će isporučivati podatke. Sama bezbednost podataka koje šalje kao I bezbednost podataka zavisi u velikoj meri od same implementacije I fajervola koji će se primeniti na serverskoj strani, stoga je kao preporuka predočena mogućnost instalacije WAF I HAProxy programa otvorenog koda koji će dodatno ojačati bezbednost.

# 5 Implementacija softvera

Pošto su prethodne faze uspešno završene prelazi se na implementaciju softvera, odnosno samo pisanje programskog koda.

## 5.1 Lokalizacija podataka (ulaz-izlaz)

Neophodno je bilo voditi računa o koverziji podataka, u kakvom će formatu *backend* da prima podatke kao I u kom će formatu slati podatke, pošto je to najčešći uzrok curenja podataka I ugrožavanja bezbednosti podataka aplikacije.

Pošto se za svaki unos podataka vrsi u obliku forme u html-u, postoji nekoliko slučajeva o kojima treba voditi računa:

1. Kada se korisnik registruje, potrebno je da unese ispravne podatke
2. Kada se korisnik loguje.
3. Svi ulazni podaci korisnika su tekstualnog tipa za prethodna dva koraka.
4. Kada se zaposleni loguje poštuje istu proceduru kao i korisnik, razlika je ta što se njegov profil kreira pozadinski, administrator ga ubacuje, dobija kredencijale na osnovu kojih se loguje.
5. Minimizovan je tekstualni unos podataka korisnika prilikom rezervacije, on samo bira polja i datume rezervacije. Program na *frontendu,* na osnovu izabranih opcija šalje odgovarajući format *backend-*u u JSON formatu.
6. Svi podaci od strane *backend-*a ka *frontend-*u se automatski konvertuju u svoja polja.

## 5.2 Struktura podataka (baza podataka)

Na slici ispod je celokupna struktura baze podataka sa svim poljima. Sastoji se od 4 tabele.

Text

Description automatically generated

Ilustracija specifikacija baze podataka

## 5.3 Unutrašnja dokumentacija

Jedna funkcionalnost na strani frontenda je prikazana na slici. Poštovanje običaja dobre prakse sa komentarima i samim opisom funkcije.

Text

Description automatically generated

Figure 1 Prikaz implementacije koda

Text

Description automatically generated

Figure 2 Prikaz implementacije koda

# 6 Testiranje softvera

Celokupno testiranje izvršili smo pregledom pojedinačnih komponenti, zatim testiranjem celokupnog sistema.

## 6.1 Jedinično testiranje komponenata

Svaku funkciju u kodu su pregledala oba člana. Koristili smo alate *Postman*(desktop aplikacija) i terminalne programe *curl* i *httpie* za testiranje funkcionisanja backend logike servera.

Testovi su rađeni metodom crne kutije.

Pregled prijavljenih korisnika

Prvi problem je nastao u samoj logici koda. *Frontend* deo je bio napisan tako da pretražuje prijavljene korisnike koristeći niz, dok je backend slao objekat, a ne niz. Tu je došlo do greške.

Text

Description automatically generated

Ilustracija Greška 1

Problem je rešen tako što smo dodali deo koda koji je vraćao umesto objekta niz.

Text

Description automatically generated

Ilustracija Ispravka 1

### Konverzija podataka

Prilikom poziva funkcije koja treba da upiše datum za koji je konferencijska sala rezervisana došlo je do greške koja je sprečavala upis podataka u backend.Greška se nije bila vidljiva sve je radilo normalno. Problem je bio u konverziji tipa podataka, priman je tip String, a neophodna je bila konverzija u tip Number. Problem je otkriven iz Node.js server logova , gde je otkrivena greška undefined element. Inspekcijom smo otkrili gde je greška I da se javlja prilikom poziva ove funkcije na *frontend*-u. Na slici ispod je kod gde je nastala greška, a zatim I slika gde je kod ispravljen tako da je podatak konvertovan u odgovarajući tip.

Text

Description automatically generated

Figure Greška konverzije

Text

Description automatically generated

Ilustracija Ispravka konverzije tipa

### Greška uklanjanja rezervacije

Problem je nastao prilikom brisanja rezervacije. Kada je klijent zeleo da pretraži slobodne sobe za određeni datum prikazivali bi mu se pogrešni rezultati.Problem je nastao u samoj logici koda jer je došlo do pogrešnog indeksiranja prilikom dohvatanja objekta rezervacije iz liste.(Ako je željena rezervacija za uklanjanje bila na mestu [1] brisana je rezervacija na mestu [0], a sledeća rezervacija je dolazila na njeno mesto tako da je svako sledeće brisanje bilo pogrešno) Na slici ispod je kako izgleda na *frontend-*u opcija pregleda svojih rezervacija kao I opcije za menjanje rezervacija.

Legalna klasa : Brisanje izabrane rezervacije

Nelegalna klasa : Brisanje drugih rezervacija

Graphical user interface, application

Description automatically generated

Figure Greška prikaza

Ilustracija Greška brisanja

Text

Description automatically generated

Figure 5 Greška u indeksiranju

Text

Description automatically generated

Figure 6 Ispravka indeksiranja

Nakon ispravljenih grešaka sve je funkcionisalo kako je I zamišljeno.

## 6.2 Integraciono testiranje

Nastavak faze testiranja primenom integracionog testiranja. Korišćen je model koji najbolje odgovara načinu rada na aplikaciji , a to je od vrha ka dnu. Počeli smo od testiranja svih funkcionalnosti komponenti, počevši od glavne na dole. Prikaz dijagrama integracionog testiranja na aplikaciji.

Diagram

Description automatically generated

Figure 7 Integraciono testiranje od vrha ka dnu

Odlučili smo se za strategiju testiranja po dubini, za svaku komponentu su po dubini istestirane sve funkcionalnosti.

## 6.3 Završetak testiranja

Testiranje je rađeno do postizanja stanja u kome smo pronalazili 3 greške po nedelji. U nastavku je raspored pronađenih grešaka od početka testiranja u svakoj nedelji.

I nedelja : 6 grešaka

II nedelja : 8 grešaka

III nedelja : 4 grešaka

IV nedelja : 1 grešaka

# 7 Isporuka softvera I obuka korisnika

Pošto su sve prethodne faze uspešno realizovane, I nakon što je utvrđeno da ispunjava sve zadate projektne zahteve sledi finalna faza isporuke softvera I obuke korisnika.

Sledeća sekcija sadrži jedan deo celokupnog korisničkog uputstva prilikom inicijalnog pristupa korisnika veb aplikaciji.

## Korisničko uputstvo

Veb aplikacija ne dozvoljava pristup korisnicima koji nisu registrovani, te je uputsvo za registraciju I prijavu u nastavku.

1. Korisnik putem linka dolazi na početnu stranicu(pretpostavka je da nije registrovan)
2. Prikazuje mu se forma za pristup(logovanje), ukoliko korisnik nije registrovan neophodno je da se registruje klikom na dugme ‘don’t have an account?’.

Graphical user interface, application

Description automatically generated

Ilustracija 9 Login

1. Nakon klika na dugme prikazaće mu se kartica sa sledećim poljima: Username,Password,Forename I Surname, koja stoje za korisničko ime,šifru,ime i prezime respektivno, kao I polje ‘Do you agree with terms and conditions?’ koje treba da štiklira.

Graphical user interface, application

Description automatically generated

Ilustracija 10 Registracija

1. Korisničko ime treba da bude dugačko najmanje 10 karaktera.
2. Kada je uspešno popunio sva polja odgovarajućim podacima korisnik se klikom na dugme ‘Submit‘ registruje kao korisnik.
   1. Ukoliko je neko od polja zaboravljeno Ili popunjeno neispravnim podacima korisnik će dobiti informaciju u dnu kartice ispod dugmeta za registroanje, u koje poljen nije unet ispravan podatak.

Graphical user interface

Description automatically generated

Ilustracija 11 Pogrešni parametri

* 1. Ukoliko korisnik želi da obriše unite podatke unete u polja to čini klikom na dugme ‘Clear‘ u dnu kartice za registrovanje.

1. Nakon uspešnog registrovanja korisnik se automatski prosleđuje na stranicu za prijavu. Gde unosi svoje korisničko ime i šifru kreiranu u prethodnom koraku.

Klikom na dugme Login unosom ispravnih parametara uspesno biva prosleđen na početnu stranicu veb aplikacije.

Graphical user interface, application

Description automatically generated

Figure 8 Login korisnika

A picture containing text, sky, blue

Description automatically generated

Figure 9 Pošetna sstranica

Korisniku se odmah prikazuju osnovne funkcionalnosti Home,Rooms,Auditoriums,About,Profile I Logout. Za svako sledeće logovanje na aplikaciju neophodni su mu parametri Username i Password sa kojima se registrovao. Time je ovo korisničko uputstvo završeno.