Projektni zahtev

Aplikacija za rezervaciju bazena

Nikola Jović 2017202369

Fakultet za informatiku i računarstvo

Univerzitet Singidunum

Sadrzaj

[Uvod 3](#_Toc75710014)

[Cilj razvoja 3](#_Toc75710015)

[Obim sistema 3](#_Toc75710016)

[Prikaz proizvoda 3](#_Toc75710017)

[Perspektiva proizvoda 3](#_Toc75710018)

[Funkcije proizvoda 3](#_Toc75710019)

[Karakteristike korisnika 4](#_Toc75710020)

[Ograničenja 4](#_Toc75710021)

[Definicije 5](#_Toc75710022)

[Reference 5](#_Toc75710023)

[Specifikacija zahteva 6](#_Toc75710024)

[Spoljašnji interface 6](#_Toc75710025)

[Funkcije 9](#_Toc75710026)

[Pogodnosti za upotrebu 10](#_Toc75710027)

[Zahtevane performance 10](#_Toc75710028)

[Zahtevi baze podataka 10](#_Toc75710029)

[Projektna ograničenja 12](#_Toc75710030)

[Sistemske karakteristike softvera 12](#_Toc75710031)

[Dopunske informacije 12](#_Toc75710032)

# Uvod

Cilj razvijanja veb aplikacije je laksa rezervacija termina za treninge ili rekreaciju na bazenima u Srbiji. Dostupna je za korišćenje svim korisnicima koji imaju kreiran nalog, ukoliko to nije slučaj novi korisnici mogu napraviti sebi taj nalog i na taj način im se omogćava pristup sistemu. Dok će samo administrator moći da menja sadržaj stranica i njime upravlja.

## Cilj razvoja

Aplikacija će omogućiti svim korisnicima da besplatno rezervisu svoj termin kao i da pregledaju sve termine gde su rezervisali svoje mesto. Dok će vlasnicima bazena biti olaksana prodaja termina i mozda će neki korisnici čak i po prvi put za njih čuti.

## Obim sistema

Sistem aplikacije se sastoji od baza podataka, serverskog dela aplikacije i korisnickog dela aplikacije. Sistem zahteva konstantnu vezu sa internetom i ne može da funkcioniše bez njega. Korisnik i administrator aplikaciji pristupaju putem internet pregledača[1].

## Prikaz proizvoda

Funkcionalni zahtevi aplikacijesu:

* Prijava korisnika i administratora, koja i omogućava njihovu autentifikaciju
* Registracija korisnika
* Pregled slobodnih termina u kojima ima slobodnih mesta
* Korisnik moza da pregleda sve svoje termine koje je rezervisao a oni nisu prošli
* Korisnik može da režervise i otkaže termin i da menja podatke na svom profilu
* Administrator može da menja stranice

Nefunkcionalni zahtevi:

* Aplikacija treba da pruži nesmetan rad klijentima i da omogući što bržu rezervaciju klijentima
* U samoj aplikaciji će na više nivo aplikacije biti realizovana provera podataka pre samog upisa u bazu
* Aplikacija treba da bude otporna na pokušaj zlonamernog korisnika i da omogući što veći nivo bezbednosti podatak o korisnicima

Aplikacija treba da omogući brz i bezbedan sistem za korisnike, koji mogu da joj pristupe u bilo kom trennutku iz pomoć samo internet konekcije.

### Perspektiva proizvoda

Aplikacija je na raspolaganju korisnicima svakog trenutka i može joj se pristupiti uz pomoć internet konekcije i u samo par trenutaka rezervisati termin za trening ili rekraciju. Takodje, podaci korisnika su na najvišem nivou osigurani i bezbedni.

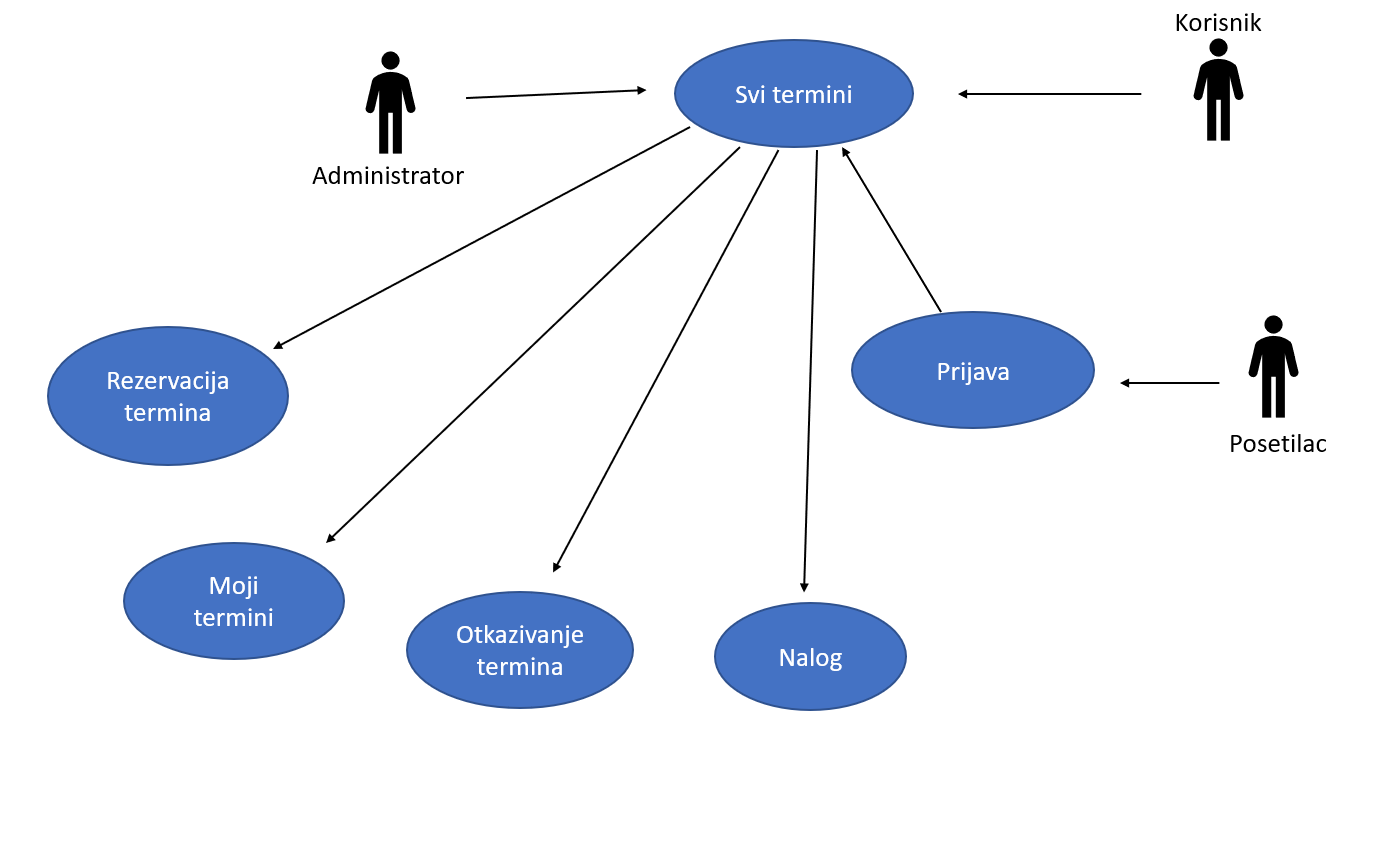
### Funkcije proizvoda

Na dijagramu slučajeva korišćenja[2](slika 1) prikazane su funkcije sistema namenjene krajnjem korisniku. U sistemu postoje dva tipa korisnika, administrator i korisnik. Oba tipa korisnika prvo moraju da se prijave(autorizuju) na sistem sa pristupnim parametrima, administratori se prijavljuju korisnickim imenom i lozinkom, dok se korisnici prijavljuju email-om koji su ostavili prilikom registracije na sistem.

Nakon uspesne prijave, oba tipa korisnika se dalje prosledjuju na spisak svih mogućih termina. Korisnici mogu samo da rezervišu termin, a administratori i da otkazu taj termin.

Nakon sto korsnik rezerviše termin, taj termin mu se izlistava u „Moji termini“, sve dok termin ne prodje ili korisnik otkaze registraciju.

Administrator jedini moze da dodaje nove termine i da menja postojeće.



Slika 1 Diagram slucajeva koriscenja 1

### Karakteristike korisnika

Korisnik mora da ima pristup internetu. Da ima iskustva u radu sa Internet pregledačem i osnovnim funkcijama računara. Očekuje se da je do sada na nekom od drugih sajtova već kreirao nalog i vršio neki vid rezervacije ili kupovine.

### Ograničenja

Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Express biblioteke. Aplikacija mora da bude podeljena u dve nezavisne celine: back-end veb servis (API) i front-end (GUI aplikacija). Sav kôd aplikacije treba da bude organizovan u dva Git spremišta u okviru istog korisničkog naloga za ovaj projekat.

Baza podataka mora da bude relaciona i treba koristiti MySQL ili MariaDB sistem za upravljanje bazama podataka (RDBMS) i u spremištu back-end dela aplikacije mora da bude dostupan SQL dump strukture baze podataka, eventualno sa inicijalnim podacima, potrebnim za demonstraciju rada projekta.

Back-end i front-end delovi projekta moraju da budi pisani na TypeScript jeziku, prevedeni TypeScript prevodiocem na adekvatan JavaScript. Back-end deo aplikacije, preveden na JavaScript iz izvornog TypeScript koda se pokreće kao Node.js aplikacija, a front-end deo se statički servira sa rute statičkih resursa back-end dela aplikacije i izvršava se na strani klijenta. Za postupak provere identiteta korisnika koji upućuje zahteve back-end delu aplikacije može da se koristi mehanizam sesija ili JWT (JSON Web Tokena), po slobodnom izboru.

Sav generisani HTML kôd koji proizvodi front-end deo aplikacije mora da bude 100% validan, tj. da prođe proveru W3C Validatorom (dopuštena su upozorenja - Warning, ali ne i greške - Error). Grafički korisnički interfejs se generiše na strani klijenta (client side rendering), korišćenjem React biblioteke, dok podatke doprema asinhrono iz back-end dela aplikacije (iz API-ja). Nije neophodno baviti se izradom posebnog dizajna grafičkog interfejsa aplikacije, već je moguće koristiti CSS biblioteke kao što je Bootstrap CSS biblioteka. Front-end deo aplikacije treba da bude realizovan tako da se prilagođava različitim veličinama ekrana (responsive design).

Potrebno je obezbediti proveru podataka koji se od korisnika iz front-end dela upućuju back-end delu aplikacije. Moguća su tri sloja zaštite i to: (1) JavaScript validacija vrednosti na front-end-u; (2) Provera korišćenjem adekvatnih testova ili regularnih izraza na strani servera u back-end-u (moguće je i korišćenjem izričitih šema - Schema za validaciju ili drugim pristupima) i (3) 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 baze podataka; (3) dijagram organizacije delova sistema, gde se vidi veza između baze, back-end, front-end i korisnika sa opisom smera kretanja informacija; (4) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije po ISO standardu.

Izrada oba dela aplikacije (projekata) i promene kodova datoteka tih projekata moraju da bude praćene korišćenjem alata za verziranje koda Git, a kompletan kôd aplikacije bude dostupan na javnom Git spremištu, npr. na besplatnim GitHub ili Bitbucket servisima, jedno spremište za back-end projekat i jedno za front-end projekat. 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).

### Definicije

[1]Internet pregledač - poseban program čija je osnovna funkcija da omogući korisniku da pregledava web stranice, kao i sve prateće sadržaje širom globalne mreže

[2]UML - predstavlja grafički prikaz skupa svih slučajeva korišćenja u sistemu ili delu sistema. Ukazuje na granice sistema i njegovu interakciju sa spoljašnjim svetom.

# Reference

Tomašević, Violeta Razvoj aplikativnog softvera, Univerzitet Singidunum, Beograd

Node - <https://nodejs.org/en/docs/>

React - <https://reactjs.org/docs/getting-started.html>

# Specifikacija zahteva

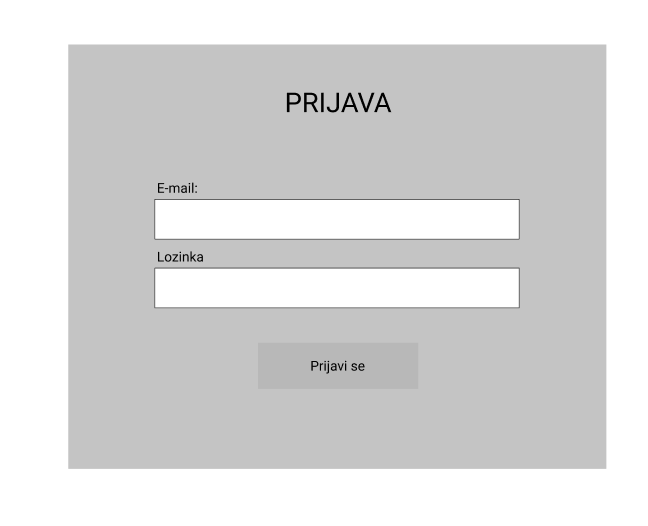
Aplikacija će pristupati bazi podataka za izvršavanje različitih upita. Softver će za komunikaciju koristiti HTTP protocol. U skladu sa tim neophodno je da korisnici poseduju web pregledač koji podržava HTTP protokol. Funkcionalnosti koje su na raspolaganju zaposlenima treba da budu implementirane tako da im omoguće što jednostavniji unos podataka, njihovu obradu, pretragu po raznim kriterijumima i opcije za automatizovane kreiranje polisa.

## Spoljašnji interface

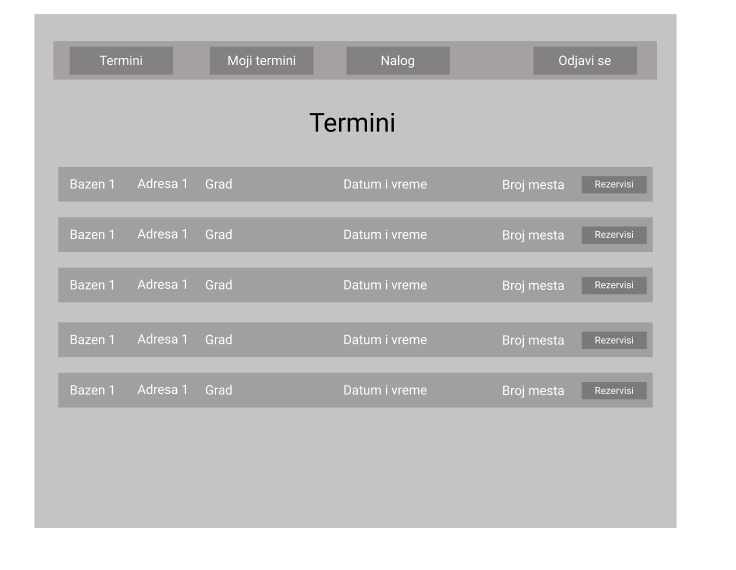
Nakon što korisnik pristupi aplikaciji otvoriće mu stranica za prijavu u aplikaciju. Ukoliko korisnik nema svoj nalog potrebno je da ga napravi.



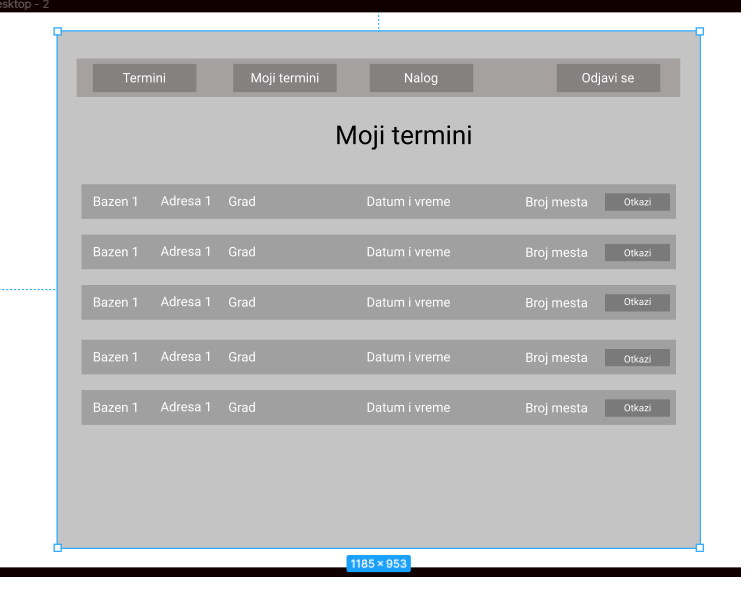
Nakon sto je napravio nalog korisniku se otvara stranica za prijavu u aplikaciju, I on mora da unese parameter kao sto su email I lozinka sa Kojima se registrovao na system.



Ukoliko korisnik unese nevažeće podatke, pristup će biti odbijem I korisniku će se prikazati poruka da prokuša ponovo. Nakon sto korisnik unese validne podatke biće usmeren na listu svih termina.



Korisnik moze da vidi sve slobodne termine u Kojima ima mesta I da rezervise termin za sebe klikom na dugme rezerviši.

Nakon što korisnik rezerviše Slobodan termin on moze da taj I ostale termine koje je rezervisao vidi kliko na dugme Moji termini. 

Nakon sto se korisniku izlistaju svi termini koje je on rezervisao a ti termini nisu prosli, korisnik moze da otkaze termin klikom na dugme otkazi. Nakon sto je korisnik otkazao termin on se brise iz liste Moji termini.

## Funkcije

U okviru Sistema postoje dve vrste korisnika korisnik I administrator. Aplikaciji ne mogu da pristupe ne prijavljeni korisnici, prilikom ulaska u aplikaciju neophodno se je prijaviti da bi aplikacija imala uvid u ulogu korisnika, da li je on administrator ili obican korisnik. Administrator moze da vidi sve termine kao I broj slobodnih mesta, kao I da ih otkazuje. Dok korisnik moze samo da rezervise odredjeni termin I da otkaze rezervaciju. Korisnik moze takodje da unosi nove termine, da brise naloge korisnika koji su neaktivni, kao I da uredjuje stranice sa pravilima.

## Pogodnosti za upotrebu

Zbog mogućnosti pristupa putem interneta aplikacija je jako prakticna za upotrebu. Ne zahteva nikakvo instaliranje, fleksibilnog je dizajna za sve uredjaje pa korisnici mogu pristupiti I putem telefona u trenutku kada njima to odgovora.

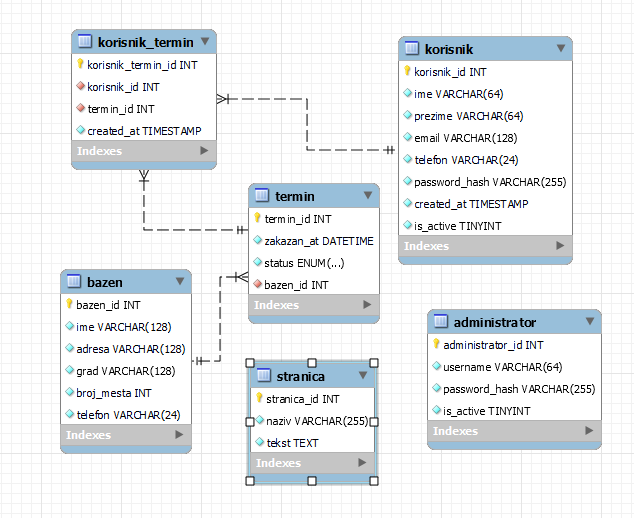
## Zahtevane performance

Performanse koje aplikacija treba da pruzi korisnicima je brz odziv, dostupnost, przo procesuiranje zahteva. Takodje aplikacija treba podjednako dobro da radi I sa velikim brojem korisnika istovremeno.

Slabost aplikacije je jedino ta što zavisi od interneta.

## Zahtevi baze podataka

Model baza podataka



Tokom rada aplikacije koristiće se strukturirani podaci. Strukturirani podaci su veoma organizovani podaci i lako razumljivi računaru. U relacionim bazama podataka se relativno brzo mogu unositi, pretraživati i manipulisati strukturirani podaci, što je njihova najatraktivnija karakteristika.

Količina podataka koja se nalazi u bazi ograničena je resursima servera baze podataka.

Sve funkcionalnosti sistema su usko povezane sa bazom. Svi prikazi podataka, kao i izmene, dodavanja i kreiranja se vrše direktno iz baze ili u bazu podataka. Baza mora biti neprestano raspoloživa za korišćenje.

Oznake: **PK** - označava primarni ključ tabele (jedinstveni identifikator zapisa). **NN** - ograničenje da podatak uvek mora imati definisnanu vrednost (NOT NULL). **UN** - Neoznačeni broj, ne može imati negativnu vrednost (UNSIGNED). **UQ** - jedinstvena vrednost, nijedan drugi zapis ne može imati istu takvu vrednost polja (UNIQUE). **AI** - Primarni ključ se automatski uvećava za 1 pri unosu novog podatka u tabelu (AUTO INCREMENT)

Tabela administrator:

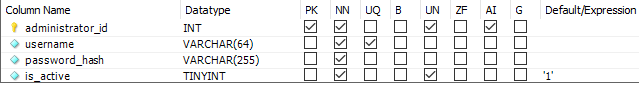


Tabela korisnik

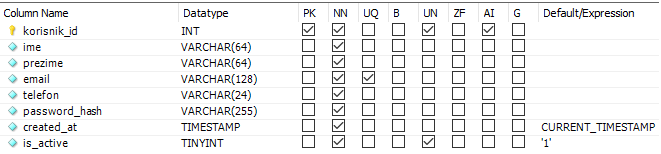


Tabela termin



Tabela bazen

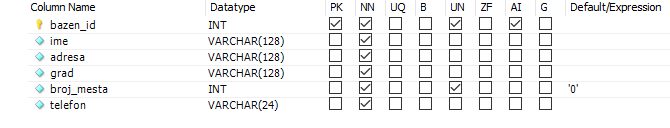


Tabela korisnik\_termin

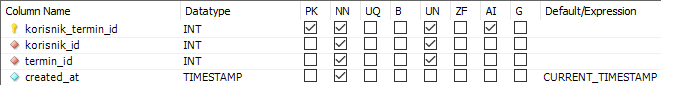
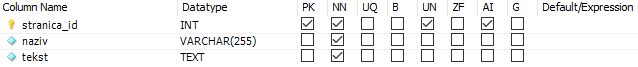


Tabela stranica



## Projektna ograničenja

Neophodni su dodatni uređaji i softveri koji obezbeđuju bezbedan prenos podataka. 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.

## Sistemske karakteristike softvera

Mora se obratiti pažnja na bezbednost podataka prilikom njihove razmene izmedju korisnika i veb servera, kao i izmedju baze podataka i veb servera. Baza podataka mora biti raspoloživa za korišćenje non stop kako se ne bi narušila interaktivnost aplikacije. Sve funkcionalnosti su usko vezane za bazu podataka. Celokupan sistem mora biti pouzdan kako se ne bi narušio integritet klijenata. Održavanje servera i backup podataka se mora vršiti redovno po unapred utvrdjenom rasporedu i učestanosti.

## Dopunske informacije

Sistem rešava problem rezervacije putem telefona Ili ličnim dolaženjem vec se to obavlja u samo par klikova.