|  |  |  |
| --- | --- | --- |
| znak_boja_FIN | UNIVERZITET U KRAGUJEVCU  **Fakultet inženjerskih nauka**  Predmet: **Projektovanje informacionih sistema i baza podataka** |  |
|  | | |

**Projektni zadatak**

Veljko Jelenković 609/2019

Jovan Živadinović 608/2019

Lazar Joksimović 625/2019

Veljko Stefanović 613/2019

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

*Tema:*

**Turistička agencija**

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| U Kragujevcu  12. 02. 2023. |  | Prof. dr Milan Erić  Doc. dr Aleksandar Djordjević  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Sadržaj

[1. Koncizno opisati poslove i informacione potrebe posmatranog realnog sistema 2](#_Toc127312825)

[2. Definisati dijagram konteksta, stablo aktivnosti i dijagrame dekompozicije posmatranog realnog sistema 3](#_Toc127312826)

[2.1 Dijagram konteksta 3](#_Toc127312827)

[2.2 Dijagrami dekompozicije 3](#_Toc127312828)

[2.3. Dijagrami sekvenca i klasi 9](#_Toc127312829)

[3. Dizajnirati model podataka, definisati logičku šemu relacione baze podataka i međurelaciona ograničenja 11](#_Toc127312830)

[3.1 Entiteti 11](#_Toc127312831)

[3.1.1 Entitet Radnik 11](#_Toc127312832)

[3.1.2 Entitet Ponuda 11](#_Toc127312833)

[3.1.3 Entitet Rezervacija 12](#_Toc127312834)

[3.2 Veze 12](#_Toc127312835)

[3.2.1 Veza Radnik - Ponuda 13](#_Toc127312836)

[3.2.2 Veza Radnik - Rezervacija 13](#_Toc127312837)

[3.2.3 Veza Ponuda - Rezervacija 13](#_Toc127312838)

[3.3 Kompletan ER dijagram 14](#_Toc127312839)

[3.4 Logička šema relacione baze podataka 14](#_Toc127312840)

[3.4.1 Prevođenje entiteta 14](#_Toc127312841)

[3.4.2 Prevođenje veza 15](#_Toc127312842)

[3.5 Međurelaciona ograničenja 15](#_Toc127312843)

[3.6 Kompletna logička šema 15](#_Toc127312844)

[4. Razviti aplikaciju (softver) za posmatrani realni sistem 16](#_Toc127312845)

[4.1 Korišćenje aplikacije 16](#_Toc127312846)

[5. Zaključak 21](#_Toc127312847)

[Literatura 22](#_Toc127312848)

# Koncizno opisati poslove i informacione potrebe posmatranog realnog sistema

Cilj projektnog zadatka je razviti informacioni sistem koji simulira rad i funkcionisanje turističke agencije. Baza podataka ovog sistema sadrži podatke o ponudama koje pruža turistička agencija, rezervacijama koje su izvedene za konkretne ponude, kao i radnicima same agencije. Proces razvoja ovakvog informacionog sistema je važan, jer svako danas koristi internet, i korisnicima aplikacije koji žele da pregledaju ponude, omogućava se njihov pregled na “jedan klik”, kao i rezervacija mesta za određena putovanja. Takođe, radnicima agencije pruža se jednostavan interfejs za upravljanje ponudama i podacima o istim, vođenje sopstvenog profila i dodavanje novih članova agencije. Korisnicima aplikacije je poželjno predstaviti ponude sa svim neophodnim detaljima: program putovanja, hotel, smeštaj itd. Ovakav sistem i ovakva aplikacija upravo to omogućavaju.

Aplikacija odrađena za projektni zadatak jeste web aplikacija. Rađena je u Laravelu (PHP Framework) i Reactu (JavaScript library), a za stilizovanje je korišćen Tailwind CSS. Arhitektura aplikacije zasnovana je na React komponentama koje obezbeđuju interfejs za spoljnu komunikaciju, dok Laravel predstavlja backend aplikacije, koji obezbeđuje autorizaciju u aplikaciji i sadrži dobru sigurnosnu zaštitu. Pomoću Laravela je takođe moguće lako obezbediti test podatke, tako što se koristi faker biblioteka za generisanje vrednosti u svim kolonama različitih tabela (User, Offer, Reservation). Faker biblioteka je ugrađena u Laravel, a takođe je moguće napraviti i svoj custom FakerProvider, što je kroz aplikaciju iskorišćeno u više navrata (LocationProvider, AccomodationProvider...). Definisanjem Factory i Seeder fajlova za ponude i korisnike, omogućeno je predefinisanje korisnika i ponuda na početku rada aplikacije, pokretanjem komande 'php artisan migrate --seed'. Da bi se pokrenula aplikacija, neophodno je startovati server i pomoću artisana i pomoću npm-a, pa se tom prilikom pokreću dve komande u okviru foldera aplikacije: 'php artisan serve' i 'npm run dev'.  
Sama aplikacija sadrži nekoliko prikaza:  
1) početna stranica, na kojoj je moguće prijaviti se na aplikaciju ako ste STAFF ili admin (opcija registracije ne postoji, osim kada admin registruje nove STAFF korisnike), ili da nastavite dalje u aplikaciju kao gost.  
2) home stranica, odnosno stranica na kojoj se prikazuju sve dostupne ponude, pri čemu postoji paginacija, kao i mogućnost detaljne pretrage ponuda. Klikom na određenu ponudu, otvara se modal sa svim detaljnim informacijama o toj ponudi. Guest korisnicima je dozvoljeno samo da izvedu rezervaciju, a ukucavanje podataka za rezervaciju se takođe obavlja u istom modalu. Admin i Staff korisnici imaju mogućnost da u modalu obave izmenu ili brisanje izabrane ponude, a van modala im je dostupno i dugme za dodavanje nove ponude.  
3) profil stranica, koja je dostupna samo adminima i STAFF članovima, i na kojoj se prikazuju različiti interfejsi u zavisnosti od role; admin sadrži spisak svih ostalih registrovanih članova agencije osim njega, ima mogućnost da doda novog STAFF korisnika, promeni neke svoje podatke i potvrđuje rezervacije, dok STAFF može samo da promeni svoje podatke.  
Prednost korišćenja React biblioteke je što je maksimalno ispoštovan SPA princip (Single Page Application), što omogućava dobar UI/UX, koji se može primetiti kroz aplikaciju.  
Što se tiče sigurnosti, aplikacija i baza podataka su dobro zaštićene od napada na različite načine: zaštita ruta od neželjenih pristupa, hash-ovanje lozinki prilikom dodavanja novog korisnika u bazu, različiti interfejsi u zavisnosti od autorizacije i role itd. Jedan od potencijalnih sigurnosnih rizika je nepostojanje tokena od strane Laravela, koji u čistim Laravel aplikacijama postoji, ali je u ovom slučaju uklonjen jer je pravio probleme pri pozivima unutar aplikacije. Aplikacija je i dalje autorizovana, i to sa React strane, ali bi definitivno bilo bolje da odgovarajuća autorizacija postoji i na backendu. Ponovna implementacija ili vraćanje tog tokena u stanje korišćenja je jedan od predloga da se aplikacija unapredi i reši taj problem.

.

# Definisati dijagram konteksta, stablo aktivnosti i dijagrame dekompozicije posmatranog realnog sistema

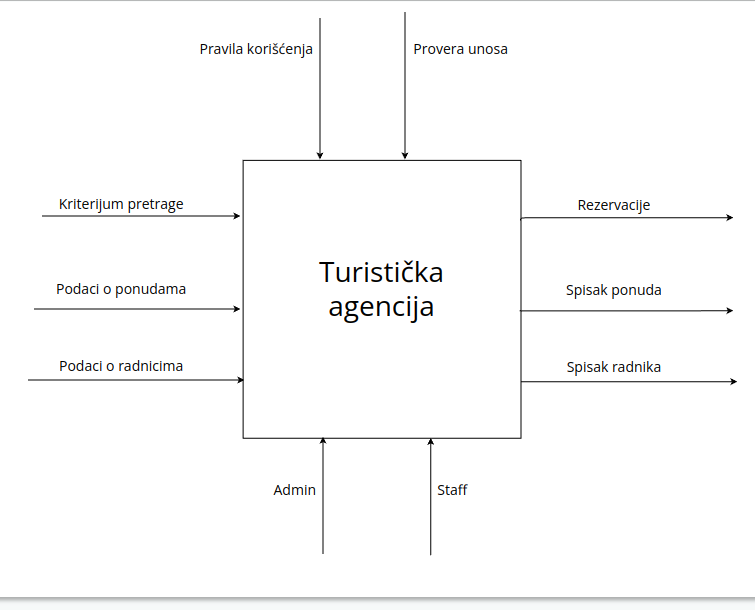
## 2.1 Dijagram konteksta

Dijagram konteksta je definisan jednim pravougaonikom koji predstavlja granicu modela koji se proučava. U tom sistemu i van njega teku informacije preko strelica. Kontekstni dijagram je najviši nivo apstrakcije koji se dekompozicionim dijagramima prevodi u niži nivo apstrakcije.

Odnos između aktivnosti i informacija je određen pomoću pravougaonika i strelica. Strelice sa leve strane su ulazi, strelice koje dolaze sa gornje strane su kontrole, strelice koje su sa desne strane, tj. izlaze iz pravougaonika su izlazi, a strelice sa donje strane dijagrama su mehanizmi.

U razmatranom primeru glavna aktivnost je Aerodrom, tačnije prodaja karata za postojeće letove. Što se tiče ulaza, postoje podaci o korisniku, podaci o letovima i kriterijum za pretragu putnika (broj pasoša). Kontrolu predstavljaju pravila korišćenja aplikacije i provera unosa, dok je admin jedini mehanizam u ovom sistemu. Izlazi iz ovog sistema su : spisak putnika, spisak letova i spisak prodatih karata.

Dijagram konteksta prikazan je na sledećoj slici:



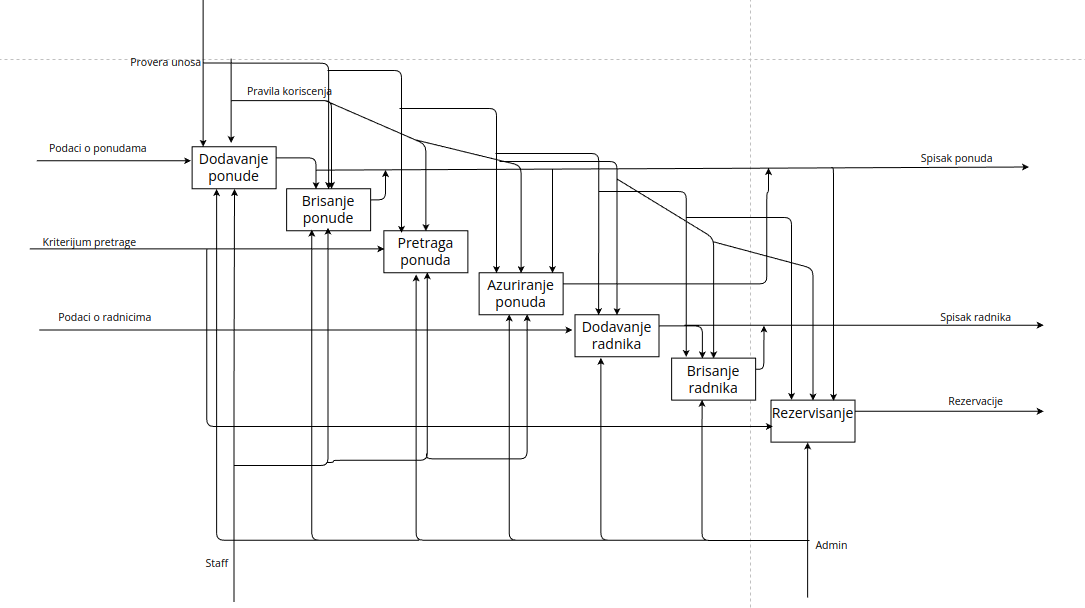
Slika 1. *dijagram konteksta*

## 2.2 Dijagrami dekompozicije

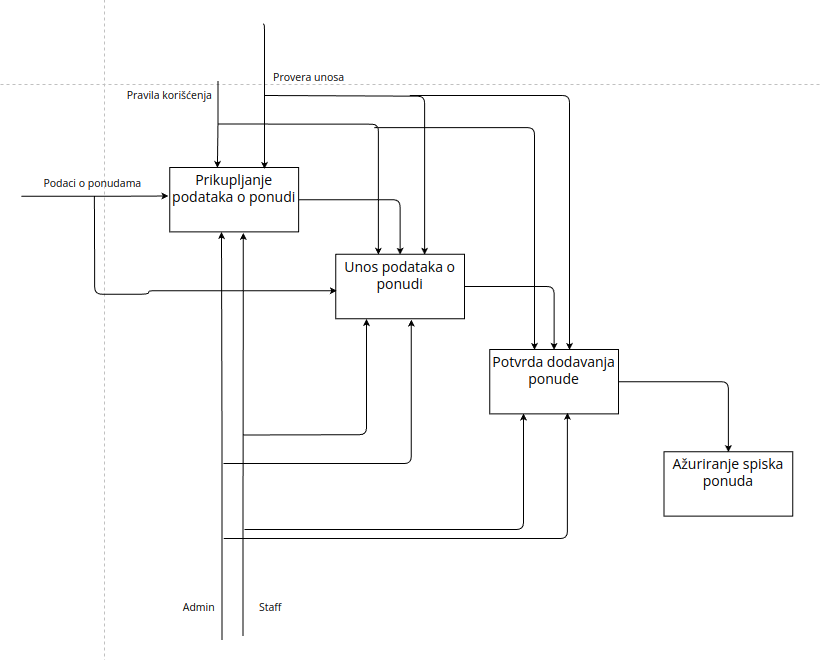
Definisanjem stabla aktivnosti uspostavile su se vertikalne veze između poslova, dok se izradom dekompozicionog dijagrama uspostavljaju horizontalne veze između poslova istog nivoa. Funkcije su, kao što je već rečeno, smeštene u pravougaonike koji se crtaju u dijagonalnom smeru, od gornjeg levog ugla strane ka donjem desnom uglu. Svakoj funkciji mora se dodeliti naziv u obliku glagolske fraze, te mora imati najmanje jednu kontrolnu i jednu izlaznu strelicu.

Strelice u okviru dekompozicionog dijagrama omogućuju tzv. horizontalno povezivanje definisanih poslova. Te strelice se nazivaju eksplicitne ili interne. U ovom primeru se nalaze pet dijagrama dekompozicije i to za aktivnost Turistička agencija za sedam podaktivnosti:

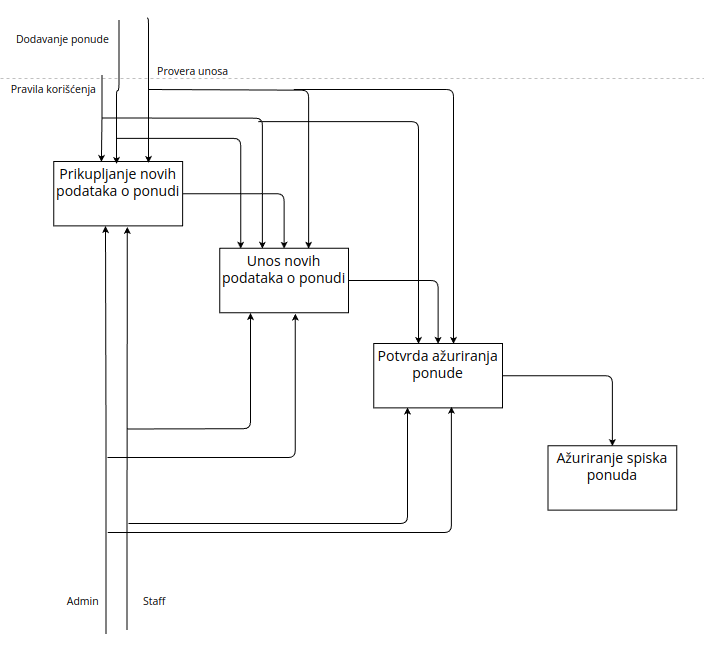
* Dodavanje ponude
* Pretraga ponude
* Brisanje ponude
* Ažuriranje ponuda
* Dodavanje radnika
* Brisanje radnika
* Rezervisanje



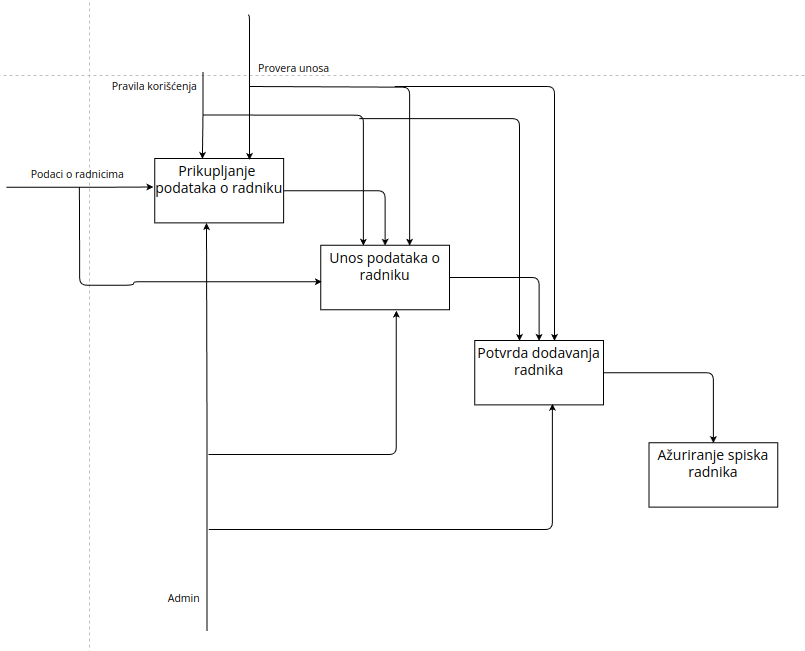
Slika 2. *Dijagram dekompozicije aktivnosti Turistička agencija*



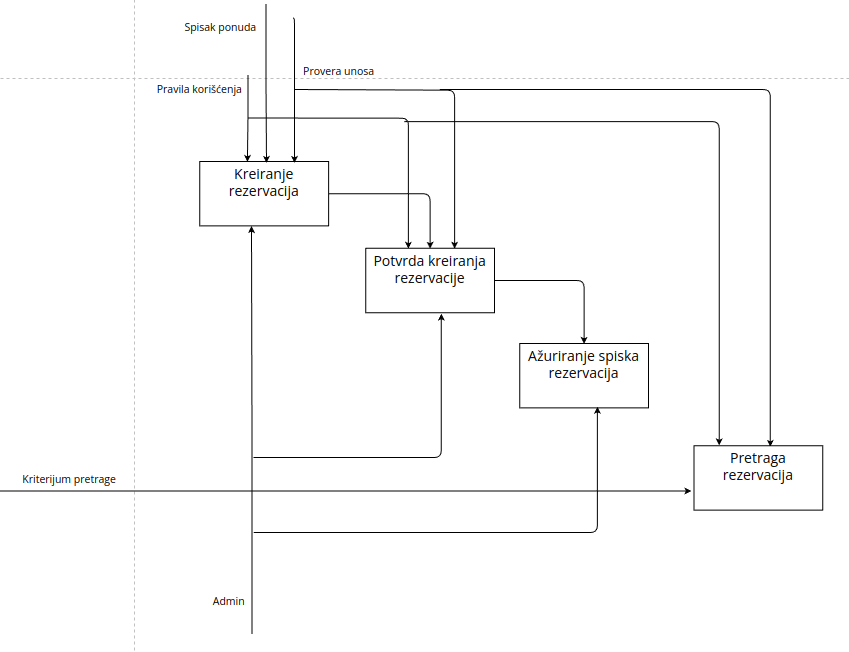
Slika 3. *Dijagram dekompozicije podaktivnosti dodavanje ponude*



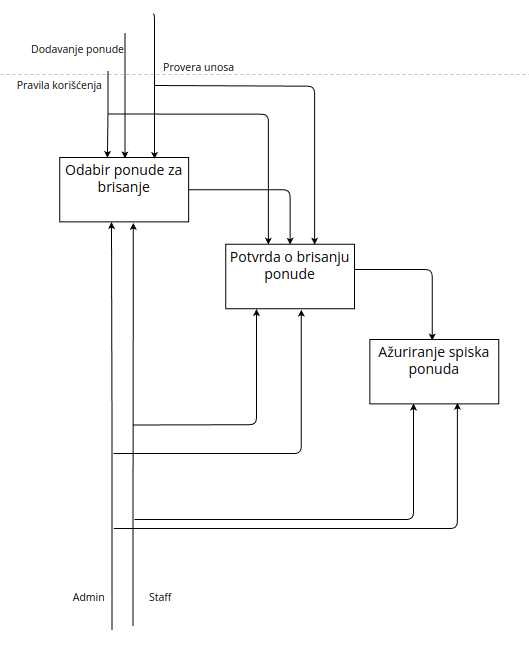
Slika 4. *Dijagram dekompozicije za podativnost ažuriranje ponuda*



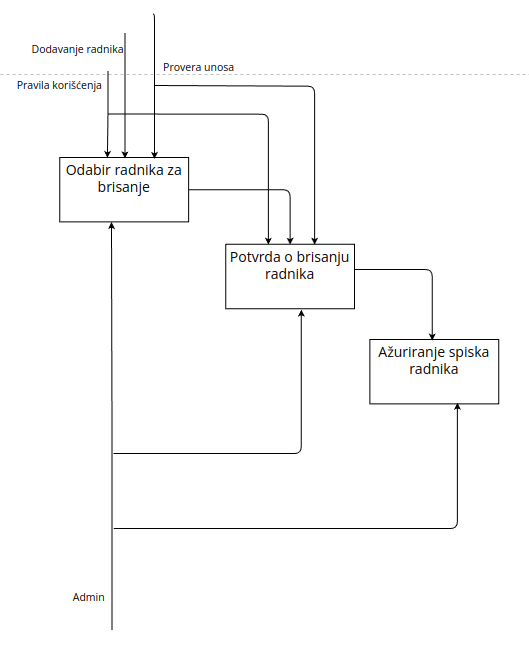
Slika 5. *Dijagram dekompozicije podaktivnosti dodavanje radnika*



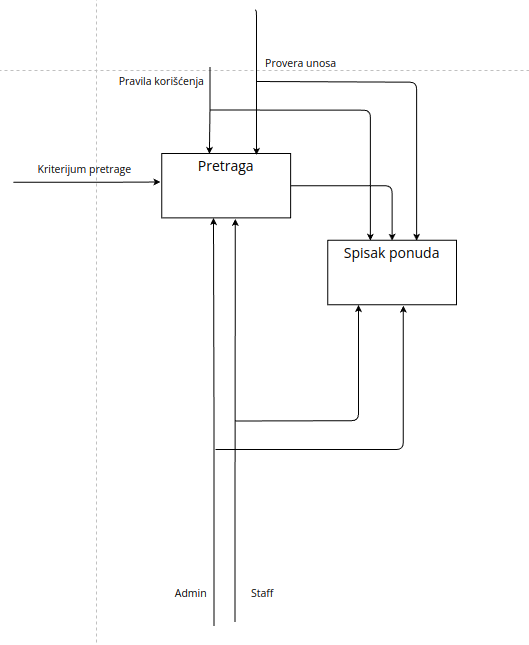
Slika 6. *Dijagram dekompozije podativnosti rezervisanje*



Slika 7. *Dijagram dekompozicije podaktivnosti brisanje ponude*



Slika 8. *Dijagram dekompozicije podaktivnosti brisanje ponude*

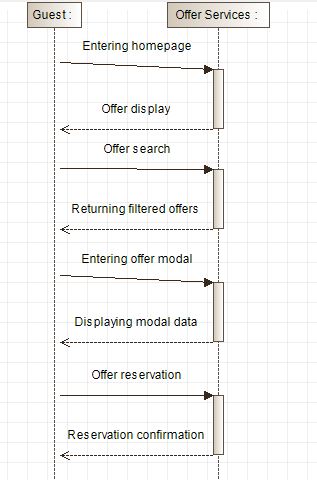
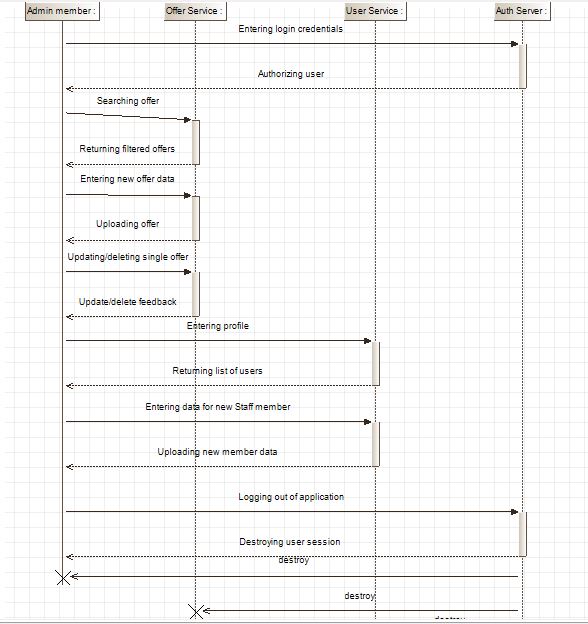


Slika 9. *Dijagram dekompozicije podaktivnosti pretraga ponuda*

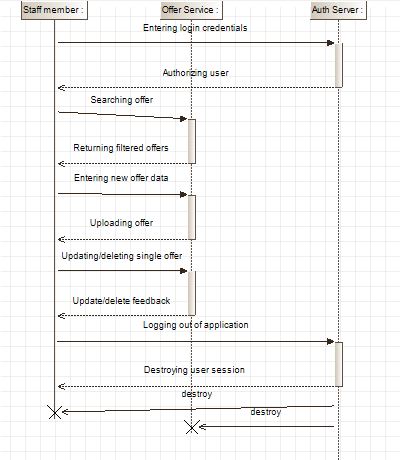
## 2.3. Dijagrami sekvenca i klasi

Dijagram sekvenci (engl. sequence diagram) prikazuje komunikaciju između skupa objekata, koja se ostvaruje porukama koje objekti međusobno razmenjuju u cilju ostvarivanja očekivanog ponašanja. Dijagram sekvenci može da sadrži aktere, objekte i poruke.

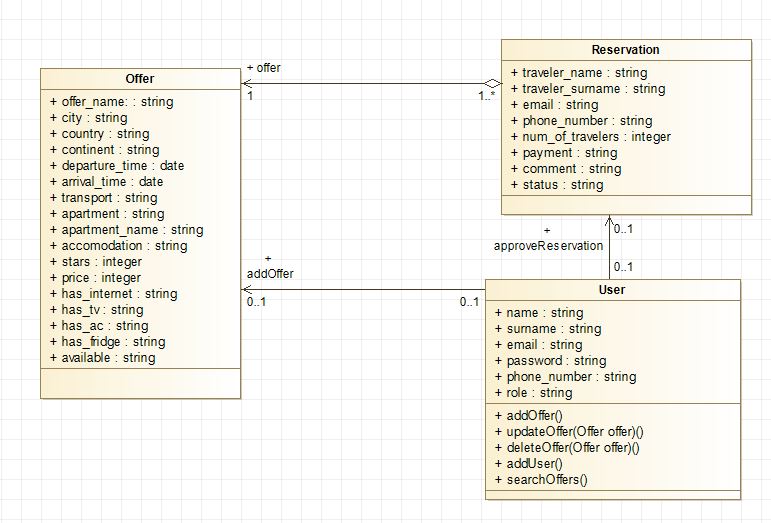
Dijagram klasa (engl. class diagram) je vrsta strukturnog dijagrama u softverskom inžinjeringu, koji opisuje strukturu sastava objašnjavajući klase unutar sastava, njihove atribute i odnose.



Slika 10. *Dijagram sekvenci za admina* Slika 11. *Dijagram sekvenci za guesta*



Slika 12. *Dijagram sekvenci za staff membera*

****

Slika 13. *Dijagram klasa*

# Dizajnirati model podataka, definisati logičku šemu relacione baze podataka i međurelaciona ograničenja

Modelovanje svakog realnog sistema počinje ER (Entity Relationship) dijagramom tako što se definišu entiteti i veze između njih. Za crtanje ER dijagrama korišćen je onlajn program ERDPlus. Iz ER dijagrama se izvodi ekvivalentna šema relacione baze podataka (logička šema).

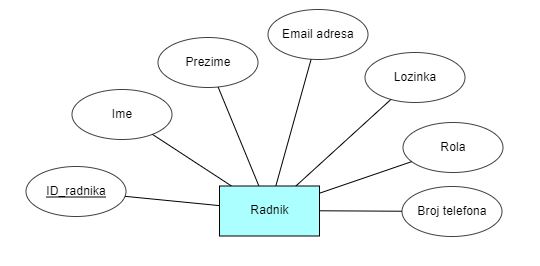
## 3.1 Entiteti

U sistemu su definisani entiteti Radnik, Ponuda i Rezervacija, i svaki od njih će biti objašnjen po navedenom redosledu. Davaće se entitet i atributi entiteta, među kojima je i primarni ključ, I svi zajedno biće grafički prikazani na slikama.

### 3.1.1 Entitet Radnik

Entitet Radnik predstavlja osobu koja radi u turističkoj agenciji. Radnici imaju mogućnost da potvrđuju rezervacije, unose, ažuriraju i brišu aranžmane, s tim što valja napomenuti da postoje dve role: ADMIN i STAFF. Rola predstavlja jedan od atributa entiteta. Svaki radnik turističke agencije ima svoj ID, koji predstavlja njegov primarni ključ, kao i ime, prezime, mejl i šifru kojim se registruje na sajt i broj telefona. Svaki radnik ima svoj jedinstveni nalog.

Entitet Radnik sa svojim atributima je prikazan na sledećoj slici:

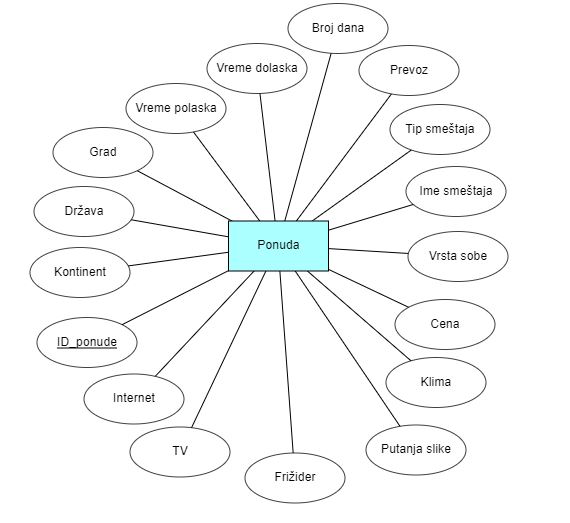


Slika 14. *Entitet Radnik sa relevantnim atributima*

### 3.1.2 Entitet Ponuda

Entitet Ponuda predstavlja konkretnu ponudu tj. aranžman za određeno putovanje, koje organizuje agencija. Ovo je entitet sa ubedljivo najviše atributa, obzirom da ponude treba da budu predstavljene detaljno. Svaka ponuda ima svoj jedinstveni ID, koji je jednoznačno identifikuje. Lokacija na koju se putuje je podeljena u tri atributa: kontinent, država i grad. Atributi su takođe i vreme dolaska i vreme polaska, broj dana koji se provodi na putovanju, tip prevoza, objekat smeštaja i njegov naziv, vrsta sobe i broj zvezdica smeštaja. Cena je veoma važan atribut, a takođe su atributi i detalji poput toga da li u sobi ima interneta, televizora, klime i frižidera. Kao dodatan atribut, postoji i putanja do slike konkretne destinacije.

Entitet Ponuda sa svojim atributima je prikazan na sledećoj slici:



Slika 15. *Entitet Ponuda sa relevenatim atributima*

### 3.1.3 Entitet Rezervacija

Entitet Rezervacija predstavlja rezervaciju neke osobe za konkretnu ponudu u okviru agencije. Rezervacija ima svoj ID (primarni ključ), i da bi ona bila kreirana, neophodno je da osoba rezerviše putovanje, time što će ostaviti podatke, poput imena, prezimena, mejl adrese, broja telefona, broja putnika, načina plaćanja. Takođe, moguće je postaviti i određenu napomenu u vezi putovanja, pa se i ona posmatra kao zaseban atribut. Rezervacija ima svoj status: ‘na čekanju’ ili ‘potvrđeno’.

Entitet Rezervacija sa svojim atributima je prikazan na sledećoj slici:



Slika 16. *Entitet Rezervacija sa relevantnim atributima*

## 3.2 Veze

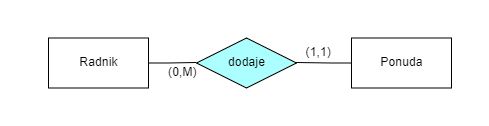
U prethodnom poglavlju smo opisali odgovarajuće entitete, dok će u ovom poglavlju biti opisane veze između entiteta. Biće opisane veze između entiteta:

* Radnik - Ponuda
* Radnik - Rezervacija
* Ponuda - Rezervacija

### 3.2.1 Veza Radnik - Ponuda

Radnik može da ne doda nijednu, a može i više ponuda (0, M), dok ponuda može biti dodata od jednog i samo jednog admina. Ova veza je označena sa Dodaje.

Veza je prikazana na sledećoj slici:

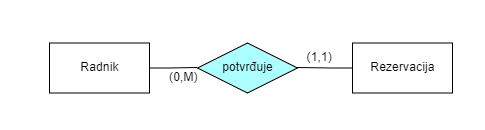


Slika 17. *Veza između entita Radnik i Ponuda*

### 3.2.2 Veza Radnik - Rezervacija

Radnik može da ne potvrdi nijednu, ili da potvrdi više rezervacija, dok rezervacija može biti potvrđena od strane jednog i samo jednog radnika.

Veza, koja je nazvana Potvrđuje, prikazana je na sledećoj slici:

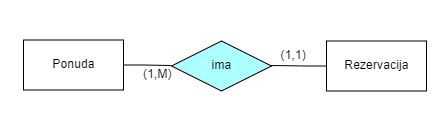


Slika 14. *Veza između entiteta Radnik i Rezervacija*

### 3.2.3 Veza Ponuda - Rezervacija

Jedna ponuda mora da ima minimum jednu rezervaciju, da bi se to putovanje uopšte ostvarilo, a najčešće ih ima više. Rezervacija važi za jednu i samo jednu ponudu.

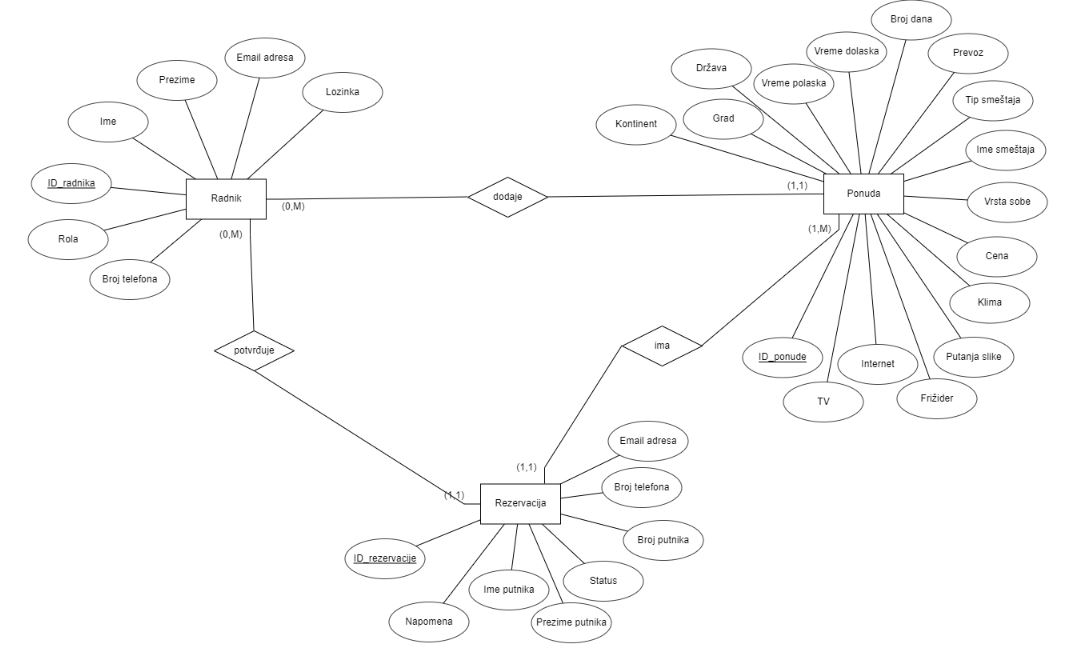
Ova veza je nazvana Ima i prikazana je na sledećoj slici:



Slika 18. *Veza između entiteta Ponuda i Rezervacija*

## 3.3 Kompletan ER dijagram

Nakon definisanih entiteta i veza između njih, sledi kompletan ER dijagram koji predstavlja celokupnu sliku realnog sistema i prikazan je na sledećoj slici:



Slika 19. *Kompletan ER dijagram*

## 3.4 Logička šema relacione baze podataka

Za dobijanje logičke šeme relacione baze podataka potrebno je prevesti sve entitete i veze ER dijagrama u njihov ekvivalentan oblik u relacionom dijagramu – relacije i međurelaciona ograničenja.

### 3.4.1 Prevođenje entiteta

Za prevođenje entiteta u šemu relacije koristi se sledeće pravilo: Svi tipovi entiteta postaju nezavisne šeme relacija. Ime tipa entiteta postaje ime šeme relacije. Obeležja objekta su obeležja šeme relacije. Za osnovne objekte identifikator entiteta postaje primarni ključ šeme relacije.

Korišćenjem ovog pravila dobijaju se sledeće relacije:

* Radnik (ID\_radnika, ime, prezime, email\_adresa, lozinka, rola, broj\_telefona)
* Ponuda (ID\_ponude, naziv, kontinent, država, grad, vreme\_polaska, vreme\_dolaska, broj\_dana, prevoz, tip\_smeštaja, ime\_smeštaja, vrsta\_sobe, cena, klima, internet, tv, frižider, putanja\_slike)
* Rezervacija (ID\_rezervacije, ime\_putnika, prezime\_putnika, email\_adresa, broj\_telefona, broj\_putnika, napomena, status)

### 3.4.2 Prevođenje veza

Prevođenje veza zavisi od tipa kardinalnosti. Uzimajući u obzir modelovanu bazu podataka za dati sistem potrebno je primeniti pravila za veze sa kardinalnošću (1,1):(0,M) i (1,1):(1,M).

Pravila za prevođenje ovih veza su:

Veze sa kardinalnošću (1,1) : (0,M) i (1,1) : (1,M) ne postaju posebne šeme relacija. Identifikator objekta sa strane za koju je gornja granica kardinaliteta preslikavanja GG=M postaje obeležje šeme relacije koja odgovara objektu sa strane za koju je GG = 1.

Uzimajući u obzir ova pravila, može se pristupiti prevođenju veza u međurelaciona ograničenja.

1.Veza Dodaje

U pitanju je veza između Radnika i Ponude, sa kardinalnošću (1,1):(0,M), tako da veza ne postaje posebna šema relacije, već se ključ entiteta Radnik dodaje entitetu Ponuda kao atribut, tj. spoljni ključ:

* Ponuda (ID\_ponude, naziv, kontinent, država, grad, vreme\_polaska, vreme\_dolaska, broj\_dana, prevoz, tip\_smeštaja, ime\_smeštaja, vrsta\_sobe, cena, klima, internet, tv, frižider, putanja\_slike, ID\_radnika)

2.Veza Potvrđuje

Ovo je veza između Radnika i Rezervacije, sa kardinalnošću (1,1):(0,M), što znači da veza ne postaje posebna šema relacije, već se ključ entiteta Radnik dodaje entitetu Rezervacija kao spoljni ključ:

* Rezervacija (ID\_rezervacije, ime\_putnika, prezime\_putnika, email\_adresa, broj\_telefona, broj\_putnika, napomena, status, ID\_radnika)

3.Veza Ima

Uzimajući u obzir da je u pitanju veza između Ponude i Rezervacije sa kardinalonošću (1,1):(1,M), veza ne postaje posebna šema relacije i entitet Rezervacija će sadržati ključ entiteta Ponuda:

* Rezervacija (ID\_rezervacije, ime\_putnika, prezime\_putnika, email\_adresa, broj\_telefona, broj\_putnika, napomena, status, ID\_radnika, ID\_ponude)

## 3.5 Međurelaciona ograničenja

Međurelaciona ograničenja se definišu za sve nasleđene atribute neke relacije – moraju biti podskup skupa vrednosti nasleđenog atrbuta u originalnoj relaciji od koje je taj atribut nasleđen. To uključuje sve strane atribute svake relacije. Na ovaj način su dobijena sledeća međurelaciona ograničenja:

* Rezervacija[ID\_radnika] ⊆ Radnik[ID\_radnika]
* Ponuda[ID\_radnika] ⊆ Radnik[ID\_radnika]
* Rezervacija[ID\_ponude] ⊆ Ponuda[ID\_ponude]

## 3.6 Kompletna logička šema

Na osnovu dosadašnjih izlaganja dolazi se do kompletnog relacionog sistema:

S={

Radnik (ID\_radnika, ime, prezime, email\_adresa, lozinka, rola, broj\_telefona)

Ponuda (ID\_ponude, naziv, kontinent, država, grad, vreme\_polaska, vreme\_dolaska, broj\_dana, prevoz, tip\_smeštaja, ime\_smeštaja, vrsta\_sobe, cena, klima, internet, tv, frižider, putanja\_slike, ID\_radnika)

Rezervacija (ID\_rezervacije, ime\_putnika, prezime\_putnika, email\_adresa, broj\_telefona, broj\_putnika, napomena, status, ID\_radnika, ID\_ponude)

}

I={

Rezervacija[ID\_radnika] ⊆ Radnik[ID\_radnika]

Ponuda[ID\_radnika] ⊆ Radnik[ID\_radnika]

Rezervacija[ID\_ponude] ⊆ Ponuda[ID\_ponude]

}

# Razviti aplikaciju (softver) za posmatrani realni sistem

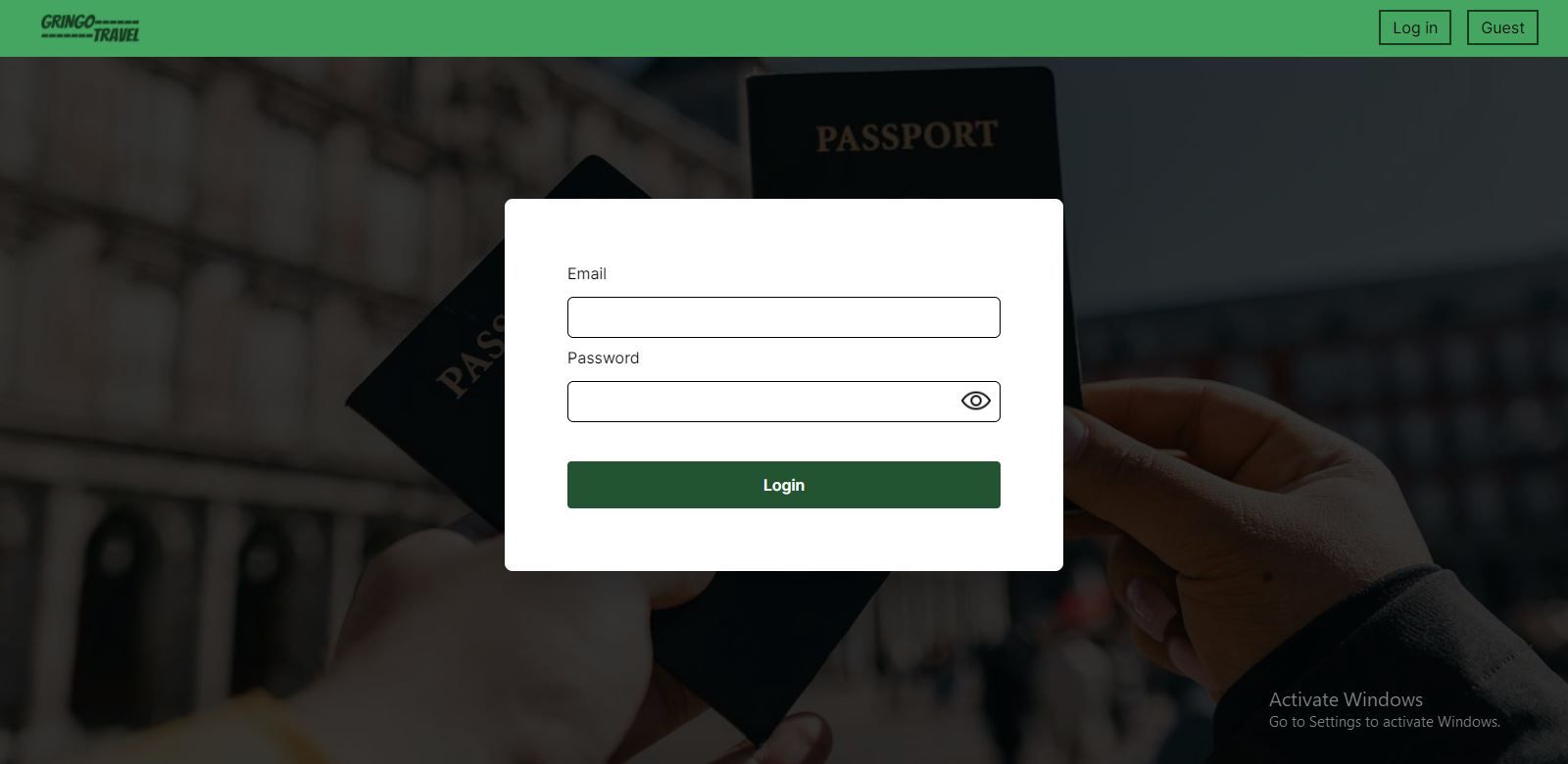
Softver koji je razvijen predstavlja web aplikaciju, i implementiran je pomoću Laravela sa backend strane i Reacta sa frontend strane. Sledi sekvencijalni prikaz svih funkcionalnosti aplikacije.

## 4.1 Korišćenje aplikacije

Aplikacija je dizajnirana za pregled i rezervacije ponuda za putovanja. Korisnici mogu koristiti aplikaciju kao *Guest* i oni imaju mogucnost da pregledaju sve ponude i posalju zahtev za rezervaciju koji kasnije odobrava admin.

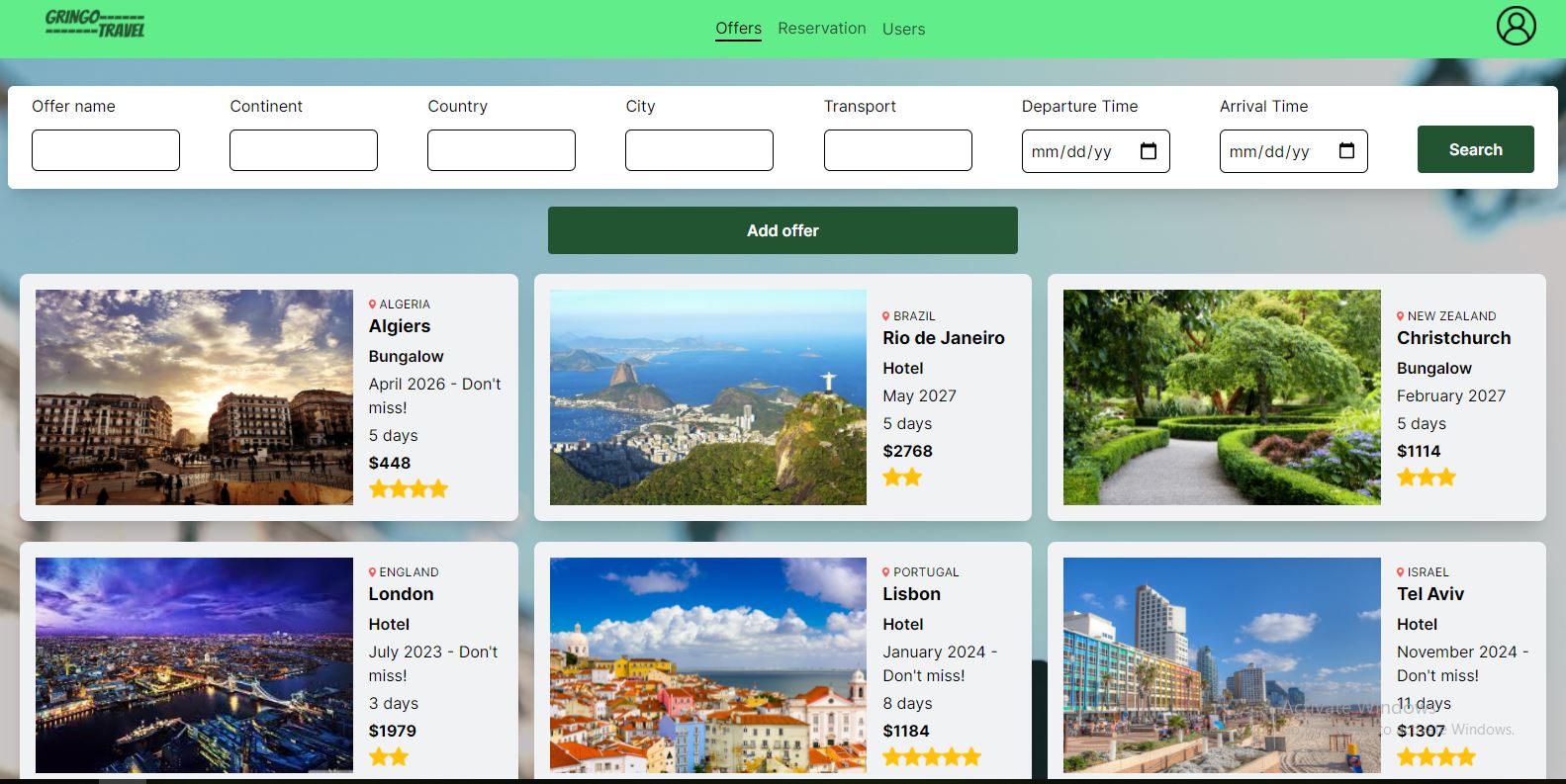


Slika 20. *Landing page aplikacije*

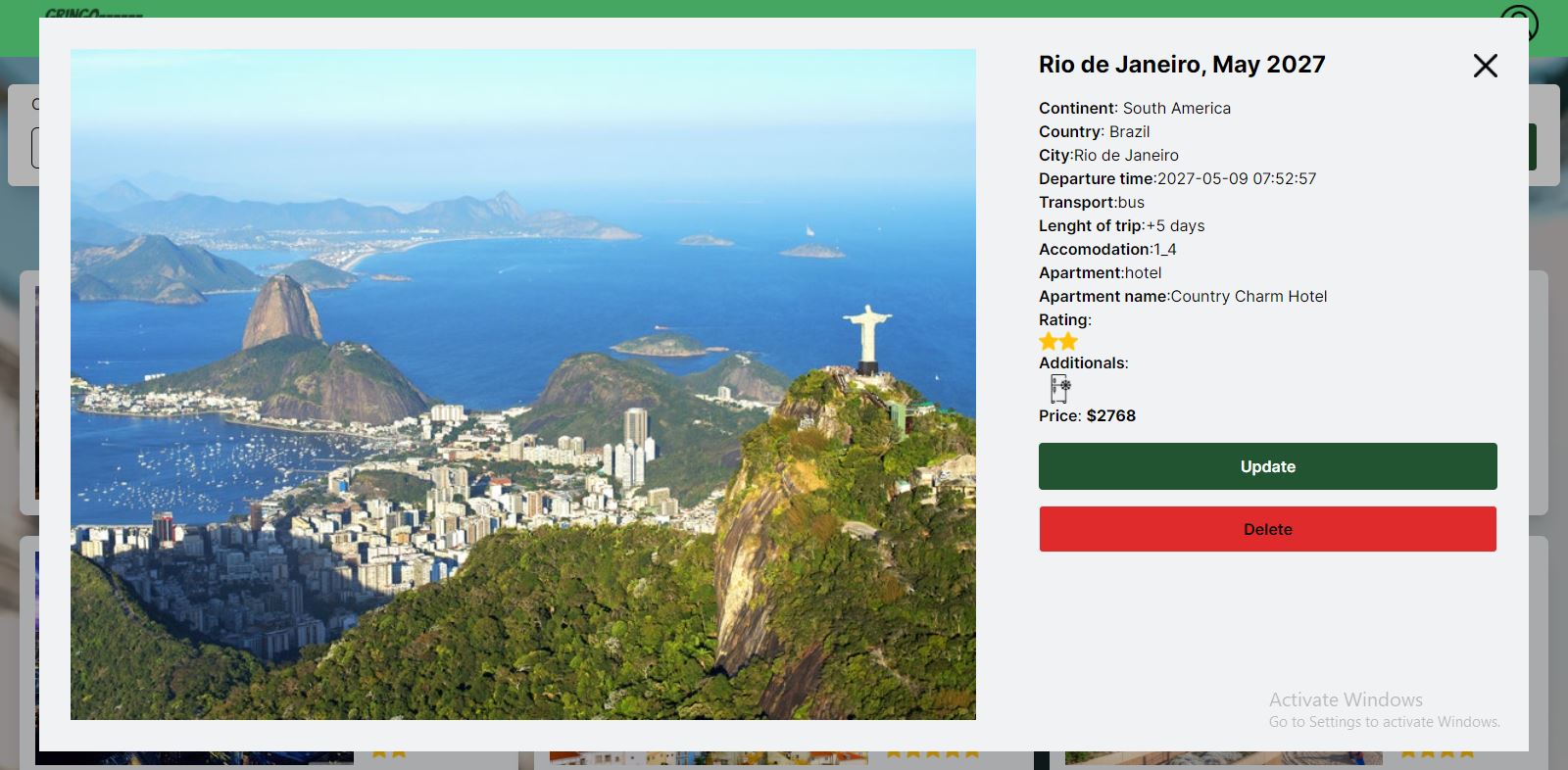


Slika 21. *Prozor za prijavu admina aplikacije*

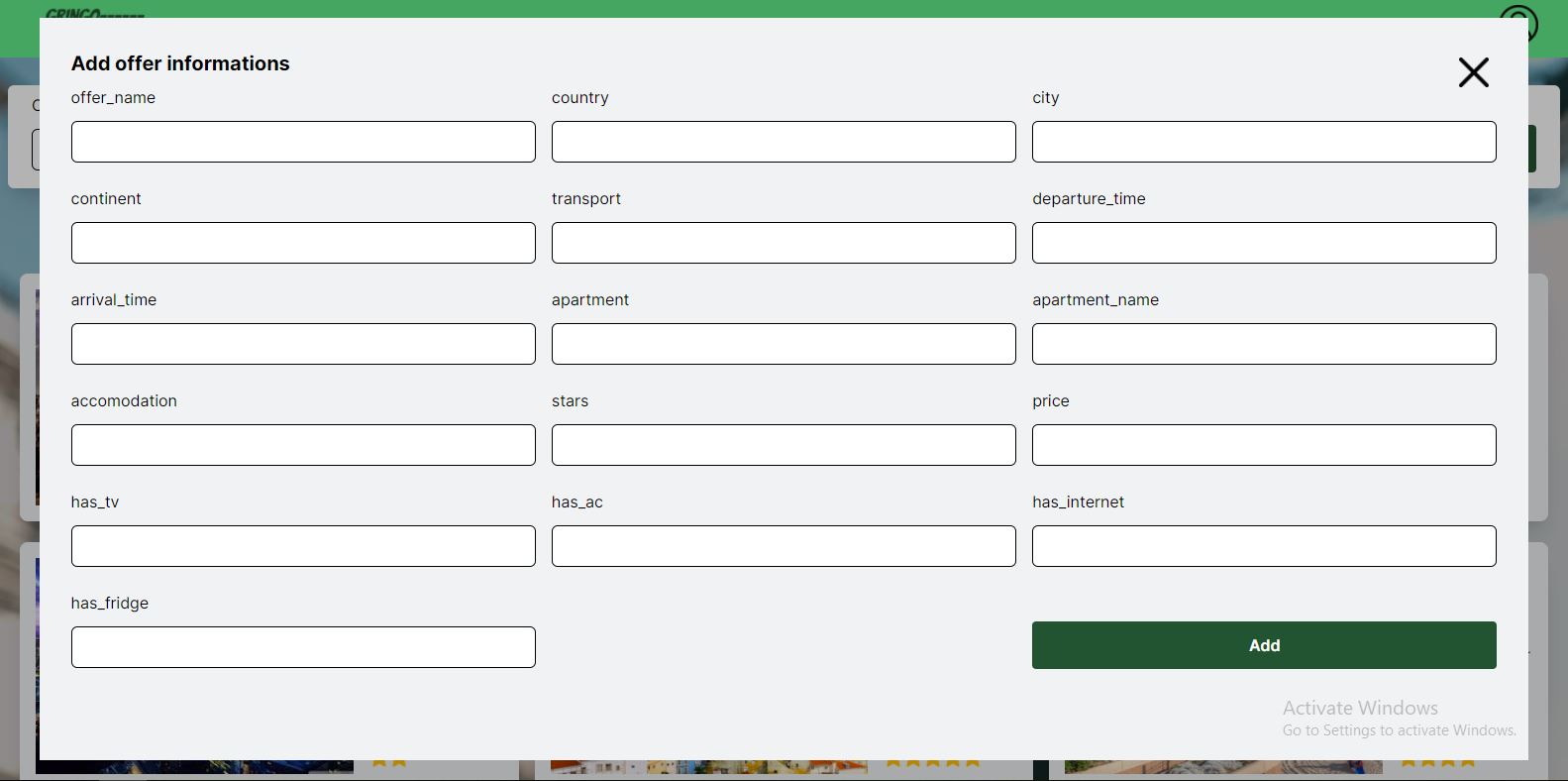
Nakon unetih ispravnih podataka, adminu se prikazuje stranica sa svim ponudama na kojoj on moze da pretrazuje ponude po raznim parametrima, pregleda detalje ponuda i izmeni ih, kao i da obrise ponude.



Slika 22. *Početna strana aplikacije*

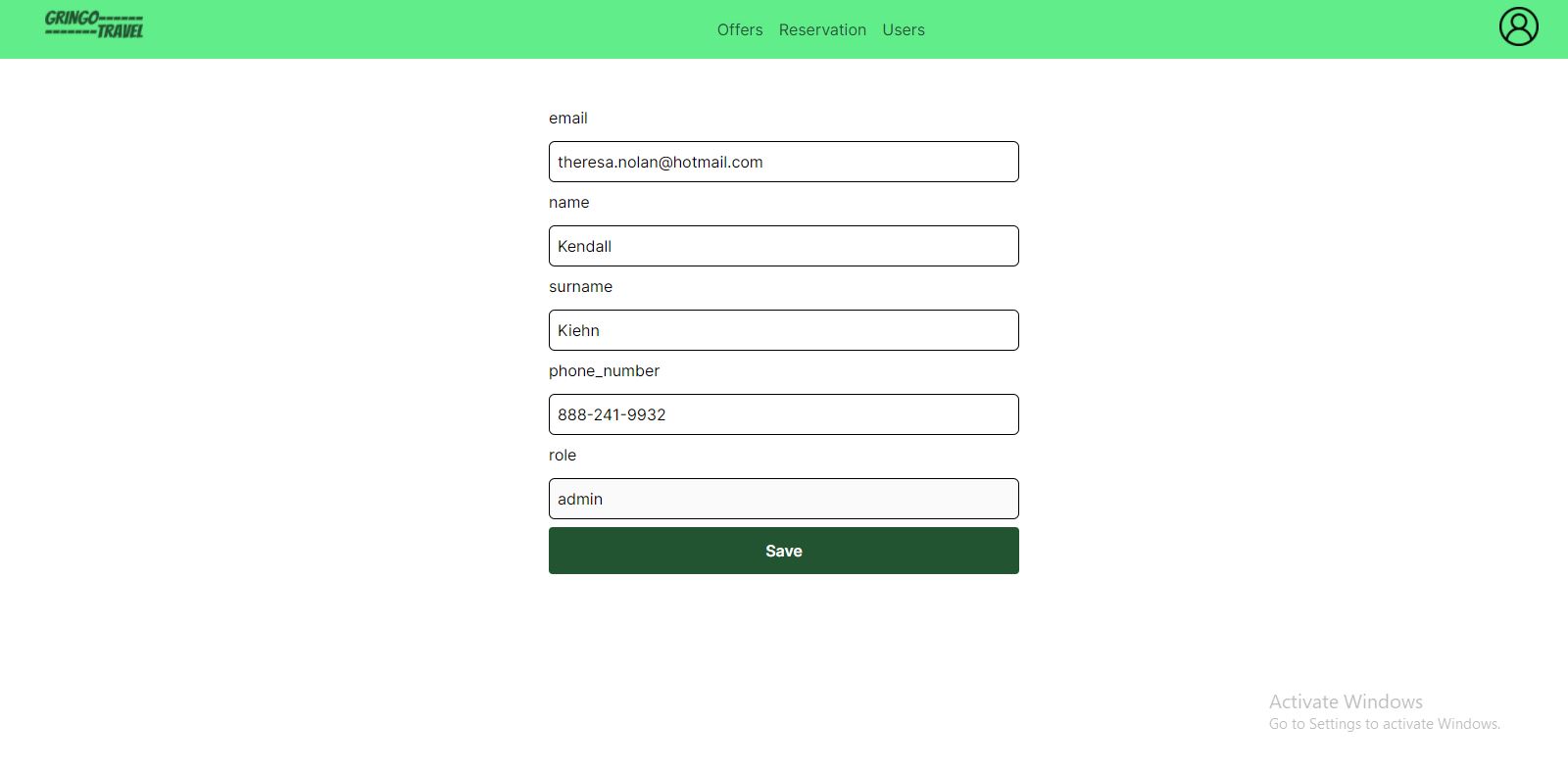


Slika 23. *Prozor za prikaz ponude*

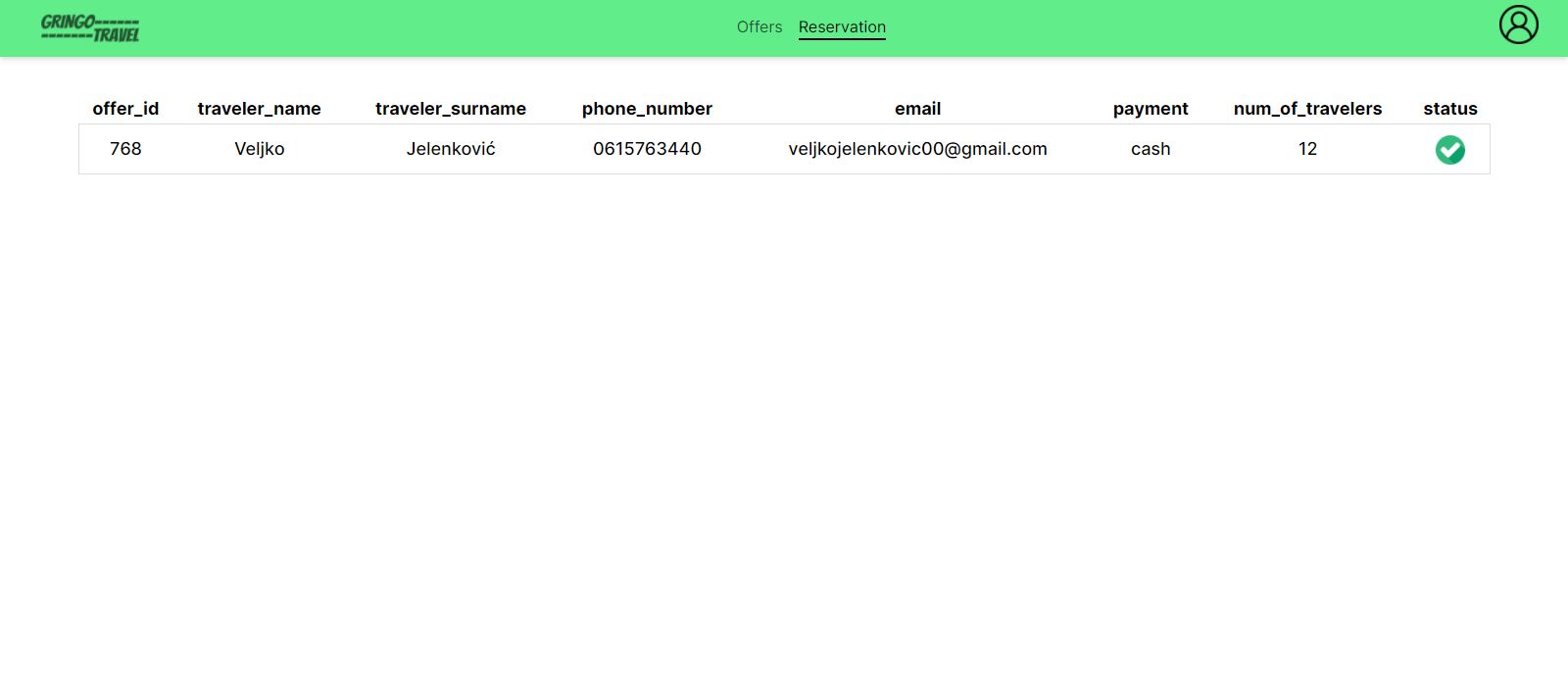


Slika 24. *Prikaz za dodavanje ponude*

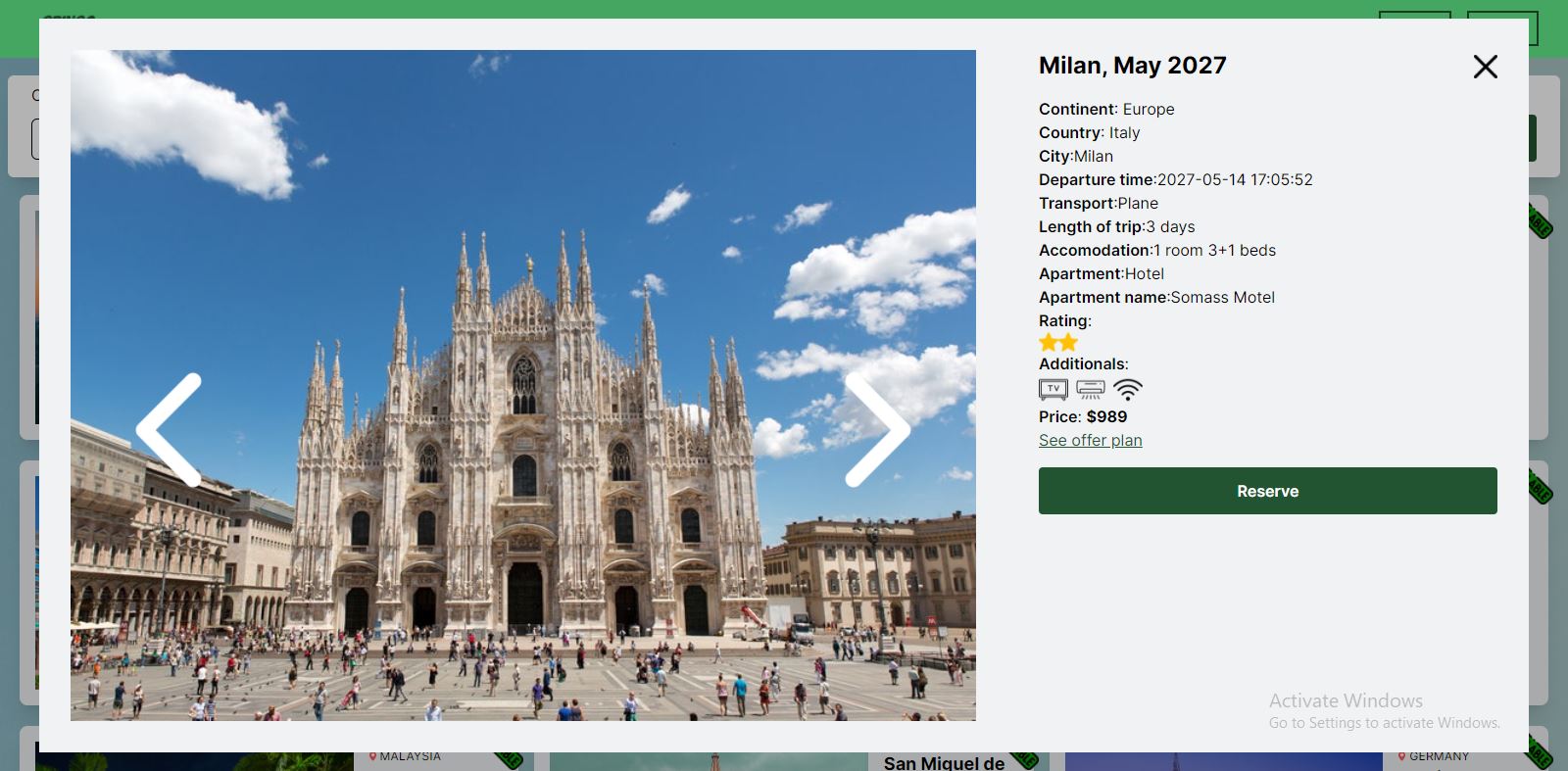
Admin ima i opciju da doda nove radnike u agenciji i to radi klikom na *Dodaj novog korisnika* gde unosi njegove podatke. Shodno tome moze videti i spisak svih radnika.



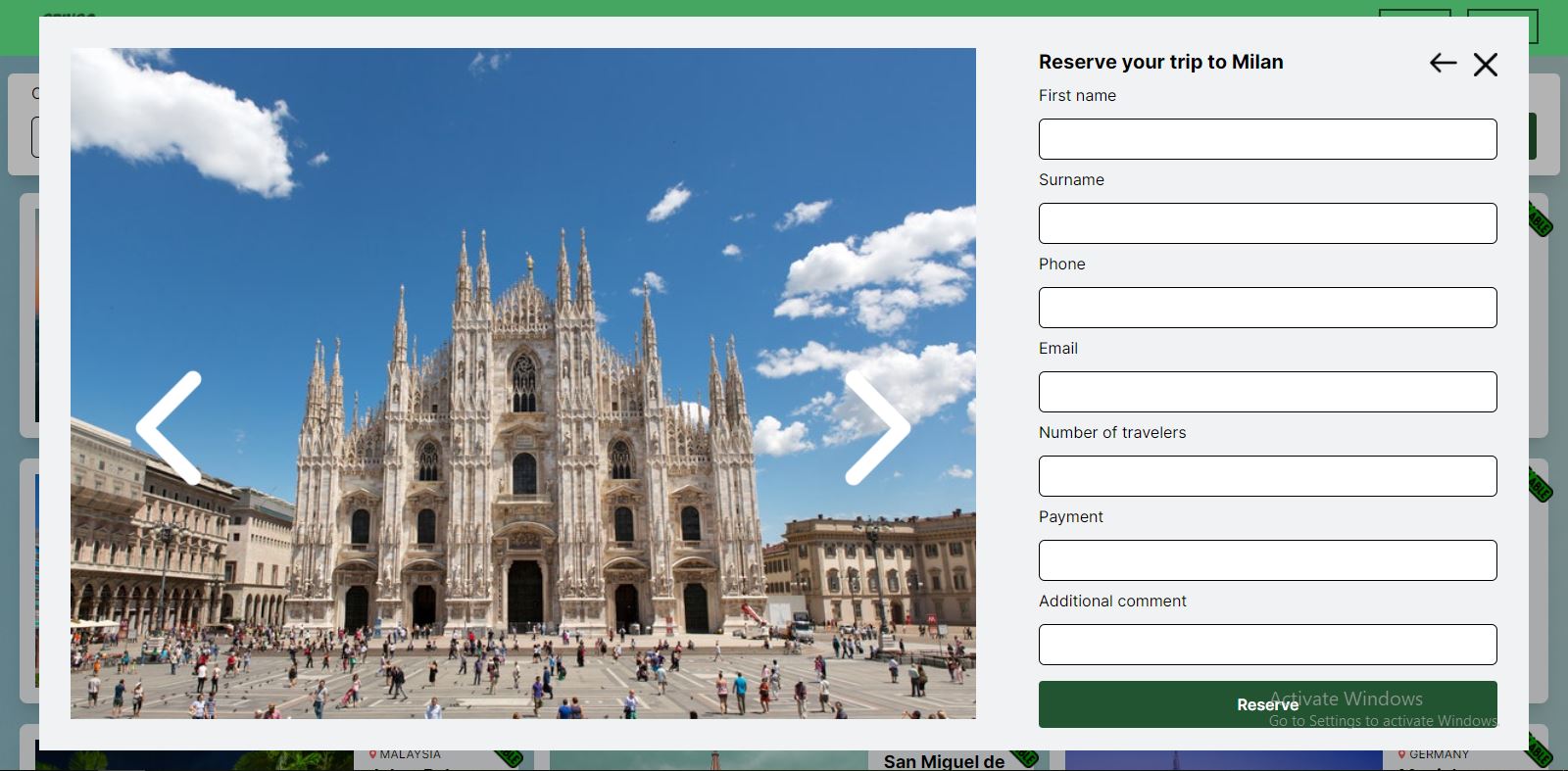
Slika 25. *Izmena podataka svog profila*



Slika 26. *Prikaz tabele rezervacija*

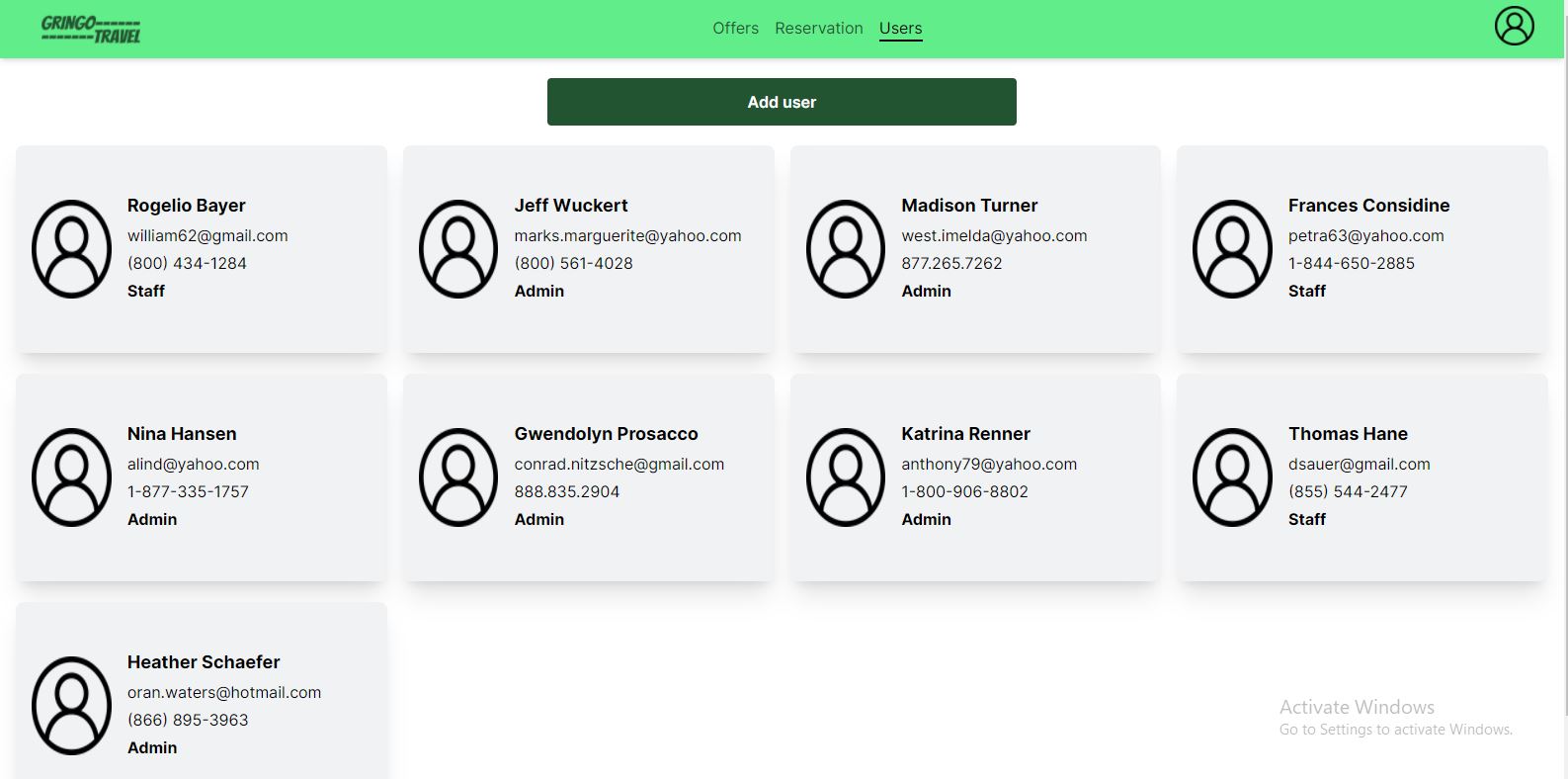


Slika 27. *Prikaz ponude za guesta*



Slika 28. *Forma za rezervaciju*

Radnik agencije ima iste mogucnosti kao admin osim dodavanja novih radnika i pregleda svih radnika.



Slika 29. *Lista radnika agencije*

# Zaključak

Kroz seminarski rad je prikazan kompletan proces razvoja informacionog sistema i baze podataka za turstičku agenciju, zajedno sa dijagramima koji prate sistem. Razvijena je web aplikacija sa autorizacijom, povezana sa bazom podataka tako da precizno obavlja CRUD operacije onda kada se to zahteva od radnika agencije, tačnije Admin i Staff korisnika.

Aplikacija je položna izmenama i dopunama, ali je u potpunosti funkcionalna u osnovnim aspektima koje turistička agencija zahteva.

# Literatura

[1] Branislav Lazarević, Baze podataka, Fakultet organizacionih nauka, Beograd 2003.,

[2] Dr Alempije V. Veljović, Praktikum iz analite informacionih sistema, Beograd, 2004.,

[3] http://moodle.fink.rs, kurs Projektovanje informacionih sistema i baza podataka

[4] https://stackoverflow.com/

[5] https://reactjs.org/docs/getting-started.html

[6] https://laravel.com/docs/9.x/installation

[7] https://v2.tailwindcss.com/docs

[8] <https://www.npmjs.com/package/react-toastify>

[9] https://docs.docker.com/