Oblikovanje programske potpore

Ak. god. 2014./2015.

*Sustav restorana*

Dokumentacija, Rev. 2

Grupa: *Avengers*

Voditelj: *Vilim Stubičan*

Datum predaje: *15. Siječnja 2015.*

Nastavnik: *<Ime i prezime nastavnika>*

Popis članova grupe i zaduženja

*Vilim Stubičan* – razrada arhitekture sustava, osmišljanje i kreiranje baze podataka, opis razreda i dijagram razreda, dijagram objekata, implementacija programskog rješenja, korištene tehnologije i alati

*Oleg Jakovljev* – funkcionalni zahtjevi, implementacija programskog rješenja, upute za korištenje

*Domagoj Korman –* implementacija programskog rješenja, testiranje programskog rješenja, isječci programskog koda za opise temeljnih funkcionalnosti sustava

*Lana Murar –* dnevnik sastajanja, funkcionalni zahtjevi, testiranje programskog rješenja, upute za korištenje

*Nejra Muslić –* ostali zahtjevi, pojmovnik, testiranje programskog rješenja

*Marko Novina –* arhitektura sustava, pojmovnik

*Filip Popić –* ostali zahtjevi, implementacija programskog rješenja, testiranje programskog rješenja

Sadržaj

[1. Dnevnik promjena dokumentacije 4](#_Toc409037919)

[2. Opis projektnog zadatka 5](#_Toc409037920)

[3. Pojmovnik 10](#_Toc409037921)

[4. Funkcionalni zahtjevi 12](#_Toc409037922)

[4.1. Opis obrazaca uporabe 13](#_Toc409037923)

[4.2. Sekvencijski dijagrami 23](#_Toc409037924)

[5. Ostali zahtjevi 41](#_Toc409037925)

[6. Arhitektura i dizajn sustava 42](#_Toc409037926)

[6.1. Svrha, opći prioriteti i skica sustava 42](#_Toc409037927)

[6.2. Dijagram razreda s opisom 52](#_Toc409037928)

[6.3. Dijagram objekata 58](#_Toc409037929)

[6.4. Ostali UML dijagrami 59](#_Toc409037930)

[7. Implementacija i korisničko sučelje 61](#_Toc409037931)

[7.1. Dijagram razmještaja 61](#_Toc409037932)

[7.2. Korištene tehnologije i alati 62](#_Toc409037933)

[7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava 64](#_Toc409037934)

[7.4. Ispitivanje programskog rješenja 66](#_Toc409037935)

[7.5. Upute za instalaciju 72](#_Toc409037936)

[7.6. Korisničke upute 77](#_Toc409037937)

[Klijent 77](#_Toc409037938)

[8. Zaključak 80](#_Toc409037939)

[9. Reference 81](#_Toc409037940)

[Dodatak A: Dnevnik sastajanja 82](#_Toc409037941)

[Dodatak B: Instalacija XAMPP-a 85](#_Toc409037942)

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Opis promjene/dodatka** | **Autor(i)** | **Datum** |
| 0.1 | Napravljen predložak. | Stubičan | 29.10.2014. |
| 0.2 | Napravljen pojmovnik | Popić | 30.10.2014. |
| 0.3 | Uređen opis projektnog zadatka | Jakovljev | 30.10.2014. |
| 0.4. | Uređeni funkcionalni zahtjevi | Jakovljev, Muslić, Popić | 01.11.2014. |
| 0.5 | Baza podataka | Stubičan | 10.11.2014. |
| 0.6. | Funkcionalni zahtjevi | Murar, Muslić | 11.11.2014. |
| 0.7 | Ostali zahtjevi | Muslić | 11.11.2014. |
| 0.8 | Obrasci Uporabe i poravnanje | Popić,  Jakovljev | 12.11.2014. |
| 0.9 | Arhitektura sustava | Stubičan, Novina, Korman | 07.11.2014. |
| 1.0 | Ispravljanje sitnih pogrešaka te završno uređenje dokumentacije. | Avengers | 14.11.2014. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Opis projektnog zadatka

Cilj projekta koji obavlja ova grupa jest implementacija sustava rezervacija i naplate mjesta u restoranima koji omogućuje uvid u ponude restorana te poslovanje istih na dnevnoj bazi. Pet je glavnih dionika sustava prema čijim će se potrebama i načinima korištenja oblikovati sustav, a oni su: gosti, ugostitelji, dobavljači, posjetitelji web portala i konobari.

Implementirani sustav će koristiti svim korisnicima, prvenstveno gostima u smislu da će im omogućiti potrebne informacije za organiziranje ugodne večeri u jednom od restorana. Također, ugostiteljima će se olakšati poslovanje jer će sve podatke o svojim poslovnicama, njihovim zalihama, rezervacijama i prometom koji se odvija unutar restorana događa imati u jednom sustavu, a također će im omogućiti i pregled te izmjenu menija kao i praćenje stanja zaliha i dobave namirnica. Vlasnici restorana će moći i voditi evidenciju o financijskom stanju ugostiteljskog objekta i prihodima. Biti će moguće i pratiti narudžbe pojedinih menija ili obroka i namirnica potrebnih za pripremu istih. Dobavljačima će sustav pomoći u organizaciji poslovanja jer će na jednoj lokaciji moći pratiti sve nadolazeće narudžbe od ugostitelja. Podaci o aktualnim rezervacijama moraju biti dostupne konobarima kako bi mogao klijente rasporediti po slobodnim stolovima. Sustav ocjenjivanja pojedine lokacije omogućuje recenzentima da iste podatke iskoriste u svojim istraživanjima, ali ujedno i gostima da im omogući lakši odabir za svoju ponude. I registrirani i neregistrirani posjetitelji portala će moći ocijeniti pojedine restorane, no registriranim korisnicima se prati statistika i prikazuju se ocjene dodijeljene pojedinim restoranima. Ocjene restorana će se pokazivati na izborniku restorana uz njihove ponude i opcije za rezervacije mjesta ili menija. Trenutno sada postoji više sustava koji imaju samo neke od navedenih funkcionalnosti, ali niti jedan koji obuhvaća sve navedeno te sustav itekako ima svrhu i ima smisla krenuti u njegovu realizaciju. Ovakav će sustav objediniti sve potrebne funkcije za upravljanje restoranom i omogućiti potencijalnim klijentima lako i pristupačno upoznavanje s ponudama restorana i rezervaciju mjesta. Sustav će također omogućiti i jednostavnu izmjenu ponude restorana na dnevnoj bazi te olakšati upravljanje ugostiteljima i koordinaciju dobavljača.

Gosti, ugostitelji i dobavljači moraju biti registrirani korisnici, čime dobivaju prava na korištenje sustava. Prilikom registracije, korisnik popunjava sljedeće podatke kroz web obrazac:

* Tip računa (gost, ugostitelj, dobavljač)
* Korisničko ime
* Lozinka
* Email adresa

Opcionalno:

* Kontakt broj
* Kreditna kartica preko koje će se vršiti naplata

Još jedna od korisnih funkcionalnosti koje će sustav pružati ugostiteljima je grupiranje pojedinih jela ili jelovnika po vrstama, na primjer stvaranje pojedinih grupa za vegetarijanska jela, salate, alkoholna pića, egzotične koktele ili specijalitete. Osim grupiranja po vrsti, biti će moguće i stvoriti posebne kategorije jela poput specijaliteta, jela s popustom ili brzih jela. Uz pojedine stavke biti će priloženi i cjenici kako bi korisnici imali funkcionalan i privlačan pregled ponude. Fizičke osobe ili osobe koje u ime vlastite tvrtke nude navedene sadržaje kroz sustav nazvat ćemo jednim imenom Ugostiteljima. Ugostitelji mogu za svaku svoju poslovnicu unijeti ponudu jela i menija koristeći *Content Management System (dalje, CMS)* . Kada ugostitelj postavi svoju poslovnicu aktivnom, to znači da pruža uslugu rezervacije mjesta u istoj poslovnici. Korisnik koji se odluči za rezervaciju mora ispuniti sljedeće podatke:

* Datum i vrijeme rezervacije,
* Broj sjedećih mjesta,
* Način naručivanja hrane
  + Online : narudžba se kreira na stranici
  + U restoranu : narudžba se obavlja prilikom dolaska u restoran

Opcionalno:

* Napomena vezana uz rezervaciju

Ukoliko ugostitelj želi oglašavati svoju poslovnicu, dužan je putem CMS-a unijeti sljedeće podatke:

* Jedinstveno ime poslovnice,
* Adresa,
* Kontakt,
* Kapacitet
  + Web obrazac za više tipova stolova (2, 4, 6… sjedećih mjesta i broj istih)
* Radno vrijeme

Radi lakše organizacije menija, Ugostitelj ima mogućnost kreiranja kategorija jela za svaku poslovnicu pojedinačno. Sve kategorije zahtijevaju sljedeće podatke:

* Ime kategorije
* Stanje (aktivna ili neaktivna)

Za svaku poslovnicu Ugostitelj može unositi neograničen broj jela u ponudi. Svako dodavanje jela zahtjeva:

* Ime jela
* Cijena
* Skup namirnica
  + Namirnica
  + Normativ odabrane namirnice
* Kategorija
* Dostupno

Ugostitelj ima ugrađenu podršku za obradu rezervacija. Kada korisnik završi rezervaciju, dobije unikatni kod koji je identifikator njegove rezervacije. S tim kodom u restoranu potvrđuje rezervaciju. Ugostitelj ima mogućnost obrade rezervacije. Prilikom toga, mijenja sljedeće stavke rezervacije:

* Stanje rezervacije (rezervirano, plaćeno, otkazano)
* Način plaćanja

Za narudžbe koje se obavljaju u restoranu:

* Meni

Svaki ugostitelj ima uvid u trenutno stanje zaliha u svakoj poslovnici. U skladu s time, može kreirati narudžbu namirnica od dobavljača. Prilikom postavljanja narudžbe, Ugostitelj mora unijeti sljedeće podatke:

* Za svaku namirnicu
  + Količina
  + Dobavljač

Dobavljači imaju uvid u narudžbe koje su zaprimili od Ugostitelja. Za mogućnost primitka narudžbe, svaki dobavljač mora odabrati proizvođača određene namirnice.

Kako bi Ugostitelji znali s kime započinju poslovni odnos, imaju uvid u listu svih Dobavljača koja sadrži osnovne podatke o istima.

Svi korisnici imaju mogućnost postavljanja upita pojedinim Ugostiteljima preko poslovne jedinice koju Ugostitelj oglašava. Tim putem korisnici mogu saznati dodatne informacije o poslovnici. Pregled svih restorana će biti pretraživ po ključnim vrijednostima.

Sustav podržava i ocjenjivanje pojedinih poslovnica. Ocijeniti restoran može bilo koji registrirani korisnik te ima mogućnost ostavljanja pismenog komentara. Sustav ocjenjivanja redovito računa ocjenu restorana te prikazuje istu na pregledu poslovnice.

Kako bi sustav oblikovali i razvili što kvalitetnije, izabrana je objektno orijentirana arhitektura zbog odlika stabilnosti, modularnosti i pouzdanosti koje se vrlo dobro uklapaju u sve zahtjeve sustava. Sustav ima dvije osnovne zadaće: klijent postavlja upit, a poslužitelj omogućava izvršavanje upita. [MODEL = KLIJENT-POSLUŽITELJ]

Centralni dio sustava je svakako baza podataka koja će ujediniti funkcionalnosti raznih dionika i povezati ih u dinamičku mrežu u stvarnom vremenu. Velike količine podataka o normativima jela, rezervacijama, narudžbama, ponudama restorana, itd., potrebno je spremiti negdje. U tu svrhu, sustav koristi relacijsku bazu podataka čija je struktura organizirana na način da se poklapa s predodžbom akcija koje se događaju unutar poslovanja restorana. Tip baze podataka koju smo odabrali je MySql.

Web stranica će biti implementirana pomoću aktualnih web tehnologija – HTML, CSS, JavaScript, jQuery, PHP, MySQL. Ključ uspješne izrade funkcionalnog proizvoda je jednostavno i pristupačno sučelje koje će korisnicima omogućiti lako ostvarivanje željenih akcija i restoranskom osoblju jednostavan pregled potrebnih funkcija. PHP je jedan od jednostavnijih tehnologija kreirana upravo za razvoj web aplikacije, dok je AngularJS vrlo moćan alat za laganu implementaciju objektno orijentiranog pristupa u web aplikaciju. Bitno svojstvo kod kojeg će u implementaciji trebati posvetiti pažnju je sigurnost podataka i netransparentnost zaštićenih korisničkih pojedinosti. Ovako ostvaren sustav će uvelike pojednostaviti vođenje restorana i objediniti mnoge zadaće koje bi inače morao obavljati skup aplikacija. Time će se korisnicima sustava efektivno smanjiti troškovi poslovanja i kompleksnost korištenja programske potpore u poslovanju ugostiteljskih objekata a klijentima omogućiti jednostavan i bezbolan način pregleda ponuda, rezervacije mjesta kao i uvid u ocjene pojedinih restorana i trenutne ponude ili popuste.

Arhitektura sustava će se bazirati na već spomenutom objektno orijentiranom pristupu, koristeći *MVC (Model – View – Controller)* oblikovni obrazac kojim se odvaja prezentacijski, logički i podatkovni dio aplikacije.

AngularJS MVC i baza podataka će biti detaljnije opisani u nastavku.

Jedna od prvih ključnih točaka u implementaciji sustava su ostvarivanje osnovnih funkcija koje sustav pruža gostima restorana i dizajn korisničkog sučelja. Funkcionalnost baze podataka koja leži u centru sustava je još jedna ključna točka projekta pošto se na bazu nadovezuju sve funkcije koje dionici mogu obavljati. Ključne točke projekta će se također odnositi na dodavanje mogućnosti ugostitelja i sučelja za konobare.

# Pojmovnik

**AngularJS**: Strukturni radni okvir (eng. framework) za dinamičke web aplikacije. Izvrsno organizira JavaScript kod te omogućuje izradu brzih web stranica koji se u paru sa jQuery-ijem izvrsno nadopunjavaju. Jednostavan je za učenje te omogućava sintaksna proširenja HTML-a doprinoseći lakšoj čitljivosti koda i povećavajući samu učinkovitost rješenja.

**CSS**: (eng. CSS - Cascading Style Sheets) Stilski jezik koji se rabi za opis reprezentacije dokumenta napisanog pomoću HTML jezika. Oblikuje izgled i raspored HTML-a bez potrebe za prikazivanjem njegovog sadržaja.

**HTML**: (eng. HTML - HyperText Markup Language) Prezentacijski jezik za izradu web stranica. Svaki hipertekst dokument sadržava sadržaj te hiperveze na ostale sadržaje koje web preglednik jednostavno dobavlja. Osnovna građa svake HTML web stranice su znakovi (eng. tags) koji opisuju kako i gdje će se nešto prikazati u web pregledniku poštujući uređenu hijerarhijsku strukturu.

**JavaScript**: Najrašireniji skriptni programski jezik kojeg podržavaju svi poznati preglednici. Služi za razvoj klijent-server Internet aplikacija. JavaScript je jezik interpreter, što znači da se ne prevodi (eng. compiling), već se odmah slijedno izvršava. Omogućava izravno kodiranje unutar HTML stranica. Osim brzine i preglednijeg web sučelja, osvježavanje podataka bez potrebe za osvježavanjem cijele stranice (Ajax) te mogućnost trenutnog pretraživanja podataka su neke od značajnih prednosti JavaScript-a.

**jQuery**: Brza, jednostavna ali sadržajno bogata JavaScript biblioteka. Omogućava jednostavno korištenje i manipulaciju nad HTML dokumentima te pojednostavljuje mnoge usluge poput upravljanja događajima (eng. event handling), upravljanje animacijama te korištenje Ajax-a. Unaprijed pripremljenom programskom potporom značajno se olakšava implementacija različitih servisa koji rade neovisno o web pregledniku.

**MySQL**: Besplatan sustav za upravljanje bazom podataka. MySQL baze podataka su relacijske baze podataka, iznimno brze i pogodne za skladištenje velikih količina podataka, zbog čega se često u praksi i koriste. MySQL ima podršku za rad sa brojnim programskim jezicima poput PHP-a koji je isto javno dostupan, što ih čini široko dostupnima.

**PHP**: PHP je jedan od najpopularnijih i najmoćnijih skriptnih jezika javne upotrebe za dinamičko generiranje HTML koda, što znači da se pomoću PHP-a može kreirati HTML stranica na serveru prije nego što je ona poslana samom klijentu sa svim pripadajućim sadržajem. PHP je orijentiran na poslužitelja (eng. server) te kada poslužitelj primi zahtjev, pomoću PHP dokumenta generira HTML kod koji šalje korisniku. Svojom sintaksom sličan je mnogim drugim jezicima (na primjer C-u).

# Funkcionalni zahtjevi

**Dionici ovog sustava su :**

* Ugostitelj
* Dobavljač
* Gost
* Posjetitelj web portala
* Konobar

**Aktori i njihovi funkcionalni zahtjevi:**

* **Ugostitelj, inicijator**
* Oglašava svoju ponudu na web koja sadrži naziv restorana, kratki opis, jelovnik te kapacitet restorana.
* U svakom trenutku ima mogućnost uvida u trenutno stanje zaliha namirnica.
* Kod narudžbe i pripreme hrane motri stanje zaliha, te po potrebi kontaktira dobavljača.
* U ponudi su opisani jelo i piće grupirani po vrstama, te kompletan cjenik.
* Može definirati specifične kategorije jela (“gableci” po danima, posebne ponude, specijaliteti kuće i sl.).
* **Dobavljač, inicijator**
* Vodi evidenciju restorana i ugostiteljskih objekata te proizvođača hrane tako što ima uvid u stanje zaliha po dobavi i isporuci hrane.
* **Gost, inicijator**
* Rezervira mjesto, jelovnik i stol.
* **Konobar, sudionik**
* Pri dolasku gostiju na “pametnom uređaju” provjerava stanje rezervacija i narudžbi te ih smješta za slobodne stolove.
* Ako gost naručuje u restoranu, konobar to evidentira čitanjem bar koda.
* **Posjetitelj web portala, inicijator**
* Brojčano ocjenjuju i komentiraju restorane ( i registrirani i neregistrirani posjetitelji )
* Registrirani:
* Može samostalno napraviti rezervaciju mjesta.
* Mora dati svoje osobne podatke.
* Opcionalno može ostaviti trajno zabilježene i ostale kontakt podatke kao što je broj telefona i mobitela te podatke o svojim kreditnim karticama kojima se vrši naplata.
* **Nezavisna “gastro-inspekcija”, inicijator**
* Na svome web portalu recenzira restoran i njegovu ponudu, klasificira ga te posjetiteljima otvara prostor gdje oni mogu ocijeniti i komentirati ponudu restorana.
* **Poslužitelj, sudionik**
* Pristupa bazi podataka i obrađuje zahtjeve za pristup bazi podataka.
* **Baza podataka, sudionik**
  + Pohranjuje podatke o ponudi, zalihama hrane, rezervacijama, narudžbama, ocjenama, korisnicima, ugostiteljima.
  + Omogućava pristup pohranjenim podacima putem poslužitelja.

4.1. Opis obrazaca uporabe

* UC1 – Prijava
* **Glavni sudionik:** Korisnik
* **Cilj:** Uspostava korisničke sjednice
* **Sudionici:** Korisnik, baza podataka, poslužitelj, sučelje
* **Preduvjeti:** Registracija, pristup poslužitelju, dostupnost baze podataka
* **Željeni scenarij:** Uspostava korisničke sjednice nakon uspješne prijave
* **Mogući drugi scenariji:** Neuspjeh prijave zbog nepostojanja korisnika u bazi podataka, pogreška u unosu podataka, nedostupnost poslužitelja, nedostupnost baze podataka
* UC2 – Registracija
* **Glavni sudionik:** Posjetitelj web stranice
* **Cilj:** Registracija u sustav
* **Sudionici:** Korisnik, baza podataka, poslužitelj, sučelje
* **Preduvjeti:** Pristup poslužitelju, dostupnost baze podataka, dostupnost korisničkog imena
* **Željeni scenarij:** Slanje korisničkih detalja u bazu podataka i uspješna registracija korisnika
* **Mogući drugi scenariji:** Neuspjela prijava zbog postojanja korisničkog imena u bazi podataka, pogreška u unosu korisničkog imena i lozinke (nedozvoljeni znakovi ili nepotpuna prijava), nedostupnost poslužitelja, nedostupnost baze podataka
* UC3 – Oglašavanje i ponuda sastojaka
* **Glavni sudionik:** Dobavljač
* **Cilj:** Oglašavanje i ponuda namirnice određene cijenom i količinom
* **Sudionici:** Baza podataka, poslužitelj, sučelje
* **Preduvjeti:** Prijavljen dobavljač, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Dobavljačnakon od
* abira vrste, cijene i količine određene namirnice uspješno oglašavana ugostiteljima svoju ponudu.
* **Mogući drugi scenariji:** Greška u bazi podataka, nepotpun i/ili neispravan unos podataka, nedostupan poslužitelj
* UC4 – Rezervacija stola
* **Glavni sudionik:** Korisnik
* **Cilj:** Rezervacija stola i transakcija po želji korisnika
* **Sudionici:** Baza podataka, poslužitelj, korisnik, sučelje
* **Preduvjeti:** Prijavljen korisnik, dostupnost baze podataka, mogućnost pristupa poslužitelju, postojanje slobodnog stola, korisnik unio važeće podatke za naplatu
* **Željeni scenarij:** Uspješna rezervacija i naplata stola
* **Mogući drugi scenariji:** Greška u bazi podataka, neuspjela rezervacija, popunjeni svi stolovi, nedostupan poslužitelj, pogrešne korisničke informacije, pogrešni podaci za plaćanje, nedovoljna sredstva za provedbu transakcije
* UC5 – Rezervacija jelovnika
* **Glavni sudionik:** Korisnik
* **Cilj:** Rezervacija jelovnika i transakcija po želji korisnika
* **Sudionici:** Baza podataka, poslužitelj, korisnik, sučelje
* **Preduvjeti:** Prijavljen korisnik, dostupnost baze podataka, mogućnost pristupa poslužitelju, izvršena rezervacija mjesta ili stola, korisnik unio važeće podatke za naplatu
* **Željeni scenarij:** Uspješna rezervacija i naplata stola
* **Mogući drugi scenariji:** Greška u bazi podataka, neuspjela rezervacija, popunjeni svi stolovi, nedostupan poslužitelj, pogrešne korisničke informacije, pogrešni podaci za plaćanje, nedovoljna sredstva za provedbu transakcije
* UC6 – Poništavanje rezervacije
* **Glavni sudionik:** Korisnik
* **Cilj:** Odustajanje od rezervacije
* **Sudionici:** Baza podataka, poslužitelj, korisnik, sučelje
* **Preduvjeti:** Prijavljen korisnik, dostupnost baze podataka, mogućnost pristupa poslužitelju, postojanje rezervacije
* **Željeni scenarij:** Brisanje zapisa rezervacije u bazi podataka
* **Mogući drugi scenariji:** Greška u bazi podataka, nedostupan poslužitelj, pogrešni podaci za plaćanje, istek roka za izmjenu rezervacije
* UC7 – Odjava korisnika
* **Glavni sudionik:** Korisnik
* **Cilj:** Prekid korisničke sjednice
* **Sudionici:** Baza podataka, poslužitelj, korisnik, sučelje
* **Preduvjeti:** Prijavljen korisnik, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Uspješna odjava korisnika i preusmjeravanje korisnika na početnu stranicu
* **Mogući drugi scenariji:** Korisnik već odjavljen, nedostupan poslužitelj
* UC8 – Promjena korisničkih podataka
* **Glavni sudionik:** Korisnik
* **Cilj:** Izmjena korisničkih podataka u bazi
* **Sudionici:** Baza podataka, poslužitelj, korisnik, sučelje
* **Preduvjeti:** Prijavljen korisnik, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Izmjena korisničkog zapisa u bazi podataka
* **Mogući drugi scenariji:** Greška u bazi podataka, nedostupan poslužitelj, nedozvoljene korisničke informacije
* UC9 – Promjena podataka o restoranu
* **Glavni sudionik:** Ugostitelj
* **Cilj:** Izmjena podataka o restoranu u bazi
* **Sudionici:** Baza podataka, poslužitelj, ugostitelj, sučelje
* **Preduvjeti:** Prijavljen ugostitelj, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Izmjena korisničkog zapisa u bazi podataka
* **Mogući drugi scenariji:** Greška u bazi podataka, nedostupan poslužitelj, nedozvoljene korisničke informacije
* UC10 – Oglašavanje ponude
* **Glavni sudionik:** Ugostitelj
* **Cilj:** Unos ponude restorana
* **Sudionici:** Baza podataka, poslužitelj, ugostitelj, sučelje
* **Preduvjeti:** Prijavljen ugostitelj, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Ugostitelj unosi podatke o svojem restoranu: naziv restorana, kratki opis, detalje o kapacitetima ( broj i raspored stolova i stolaca), jelovnik., te se ti podaci prikazuju kao oglas na web stranici.
* **Mogući drugi scenariji:** Greška u bazi podataka, nedostupan poslužitelj, nedozvoljen ili višestruki unos
* UC11 – Recenziranje restorana na vlastitom web prostoru
* **Glavni sudionik:** Gastro-inspekcija
* **Cilj:** Gastro-inspekcija na stranici restorana dobiva kod koji joj omogućuje recenziranje restorana na vlastitoj web stranici
* **Sudionici:** Baza podataka, gastro-inspekcija, sučelje
* **Preduvjeti:** Dostupnost baze podataka, mogućnost pristupa poslužitelju, gastro-inspekcija mora imati svoju web stranicu
* **Željeni scenarij:** Nezavisna gastro-inspekcija od restorana dobije kod s kojim ona na svom web prostoru generira odjeljak s osnovnim podacima o restoranu i mogućnošću komentiranja
* **Mogući drugi scenariji:** Nedostupna baza podataka, neispravan unos pri čemu sustav dojavljuje pogrešku, gastro-inspekcija nema svoju web stranicu
* UC12 – Izmjena jelovnika
* **Glavni sudionik:** Ugostitelj
* **Cilj:** Izmjena jelovnika
* **Sudionici:** Baza podataka, poslužitelj, ugostitelj, sučelje
* **Preduvjeti:** Prijavljen ugostitelj, dostupnost baze podataka, mogućnost pristupa poslužitelju, oglašena ponuda
* **Željeni scenarij:** Ugostitelj mijenja određene dijelove jelovnika. Izmijenjena ponuda se automatski ažurira na web stranici
* **Mogući drugi scenariji:** Nedostupna baza podataka, neispravan unos ili nedopuštena izmjena pri čemu sustav dojavljuje pogrešku, višestruki unos
* UC13 – Stvaranje jelovnika
* **Glavni sudionik:** Ugostitelj
* **Cilj:** Stvaranje jelovnika
* **Sudionici:** Baza podataka, poslužitelj, ugostitelj, sučelje
* **Preduvjeti:** Prijavljen ugostitelj, dostupnost baze podataka, mogućnost pristupa poslužitelju, oglašena ponuda
* **Željeni scenarij:** Ugostitelj dodaje novi jelovnik sastavljen od jela koje grupira prema vrstama (meso, riba, vegetarijanska, salate, topla predjela, hladna predjela, slastice...). U ponudi navede i ponudu pića prema vrstama. Također može definirati specifične kategorije jela („gableci“ po danima, posebne ponude, specijaliteti...)
* **Mogući drugi scenariji:** Nedostupna baza podataka, neispravan unos pri čemu sustav dojavljuje pogrešku, višestruki unos.
* UC14 – Pregled zaliha
* **Glavni sudionik:** Ugostitelj, baza podataka
* **Cilj:** Pregled stanja zalihe restorana
* **Sudionici:** Baza podataka, poslužitelj, ugostitelj, sučelje
* **Preduvjeti:** Prijavljen ugostitelj, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Ugostitelj ima uvid u zalihe restorana preko sučelja. Baza dohvaća tražene podatke o zalihama namirnica te ih prikazuje u sučelju.
* **Mogući drugi scenariji:** Nedostupna baza podataka ili poslužitelj
* UC15 – Izmjena stanja zaliha
* **Glavni sudionik:** Ugostitelj, baza podataka
* **Cilj:** Izmjena i ažuriranje stanja zalihe restorana
* **Sudionici:** Baza podataka, poslužitelj, ugostitelj, sučelje
* **Preduvjeti:** Prijavljen ugostitelj, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Ugostitelj ima uvid u zalihe restorana koje može promijeniti sukladno s potrošenim namirnicama. Baza dohvaća ažurirane podatke o zalihama namirnica te ih prikazuje u sučelju.
* **Mogući drugi scenariji:** Nedostupna baza podataka ili poslužitelj
* UC16 – Ocjenjivanje restorana
* **Glavni sudionik:** Korisnik
* **Cilj:** Ocjenjivanje i komentiranje usluge restorana
* **Sudionici:** Poslužitelj, korisnik, sučelje, baza podataka
* **Preduvjeti:** Prijavljen korisnik, mogućnost pristupa poslužitelju, mogućnost ocjenjivanja i komentiranja restorana, postojanje rezervacije u restoranu koji se ocjenjuje
* **Željeni scenarij:** Korisnik nakon konzumacije u restoranu odluči ocijeniti uslugu restorana te mu daje ocjenu od 1 do 5. Ocjena restorana se zapisuje u bazu podataka te se ukupni prosjek ocjena prikazuje na izborniku restorana. Korisnik dodatno može komentirati uslugu restorana.
* **Mogući drugi scenariji:** Nedostupna baza podataka ili poslužitelj
* UC17 – Obavi transakciju
* **Glavni sudionik:** Korisnik
* **Cilj:** Potvrđivanje rezervacije i online naplata
* **Sudionici:** Korisnik, sučelje, poslužitelj
* **Preduvjeti:** Prijavljen korisnik, rezervacija odabrana, poslužitelj dostupan, dostupna su dovoljna sredstva na računu
* **Željeni scenarij:** Korisnik nakon rezervacije izabire izravno plaćanje preko sustava. Novčana transakcija se izvršava a korisnik dobiva potvrdu o uspješnosti transakcije.
* **Mogući drugi scenariji:** Nedostupna baza podataka ili poslužitelj, nedovoljna sredstva za plaćanje, neuspjela bankovna transakcija
* UC18 – Pregledaj stanje stolova
* **Glavni sudionik:** Konobar, baza podataka
* **Cilj:** Dostupan pregled rezerviranih stolova
* **Sudionici:** Baza podataka, poslužitelj, konobar, sučelje
* **Preduvjeti:** Prijavljen konobar, dostupnost baze podataka, mogućnost pristupa poslužitelju
* **Željeni scenarij:** Konobar pristupa sučelju preko kojega ima ažuran uvid u stanje rezervacija stolova. Ovo omogućava konobaru da korisnika s rezervacijom odvede na stol s njegovom rezervacijom.
* **Mogući drugi scenariji:** Nedostupna baza podataka ili poslužitelj



Slika 1 - Dijagram obrazaca uporabe sa tri aktora:  
korisnik, konobar, baza podataka



Slika 2 - Dijagram obrazaca uporabe sa četiri aktora:  
 ugostitelj, gastro-inspekcija, dobavljač i baza podataka

## 4.2. Sekvencijski dijagrami

**Obrazac uporabe UC1: Prijava**

Posjetitelj web stranice se prijavljuje u sustav koristeći podatke s kojima se registrirao. Na login formi upisuje svoje korisničko ime i lozinku – ti podaci se šalju poslužitelju koji provjerava bazu podataka i traži pripadajuće podatke. Ako dani podaci odgovaraju podacima u bazi, prijava je uspjela i korisnička sjednica započinje. Ako dani podaci ne odgovaraju niti jednom zapisu u bazi podataka, ispisuje se poruka o neuspjeloj prijavi korisnika. Na formi za prijavu postoji i poveznica za registraciju korisnika u slučaju da korisnik još nije registriran. Ovisno o ulozi prijavljenog korisnika, dodjeljuju mu se dozvoljene funkcije. Sekvencijski dijagram za prijavu korisnika prikazan je na slici 4-1.



Slika 4‑3 – Sekvencijski dijagram za prijavu korisnika

**Obrazac uporabe UC2: Registracija**

Neprijavljeni posjetitelj web stranice može se odlučiti registrirati na stranici u slučaju da već nema korisnički račun. Klikom na poveznicu otvara se stranica sa formom za registraciju. Korisnik unosi svoje podatke, uključujući:

* Korisničko ime
* Ime i prezime
* Lozinku te ponovljenu lozinku
* Adresu e-pošte

Dodatno, korisnik može unijeti i podatke za plaćanje, adresu i telefon. U slučaju unosa nedozvoljenog znaka ili greške pri unosu (npr. polja za lozinku i ponovljenu lozinku nisu jednaka) forma za registraciju asinkrono javlja grešku pripadajućim simbolom. Dok korisnik ne ispuni sva polja i razriješi sve pogreške u unosu, nije moguće obaviti registraciju. Kad korisnik pošalje zahtjev za registraciju poslužitelju, ažurira se baza podataka korisnika a korisnička sjednica započinje. Sekvencijski dijagram registracije prikazan je na slici 4-2.



Slika 4‑4 - Sekvencijski dijagram za registraciju novog korisnika

**Obrazac uporabe UC3 – Oglašavanje i ponuda sastojaka**

Dobavljač preko posebne vrste računa ima uvid u sve sastojke koji su potrebni ugostiteljima za nesmetani rad restorana. Preko prilagođenog sučelja može oglasiti svoje zalihe određenih namirnica koje su određene vrstom, cijenom i količinom koju je spreman u određenom trenutku dobaviti. Pretpostavljeno je da dobavljač ima neograničene količine zaliha te je uvijek spreman ubaciti nove ponude. Ukoliko uspješno oglasi svoju ponudu ugostiteljima će u sustavu biti vidljive nove zalihe koje mogu otkupiti od dobavljača. Sekvencijski dijagram oglašavanja i ponude sastojaka nalazi se na Slici 4-3.



Slika 4‑5 – Sekvencijski dijagram za oglašavanje ponude sastojaka

**Obrazac uporabe UC4: Rezervacija stola**

Gost restorana – prijavljeni korisnik može u po želji odabranom restoranu rezervirati stol. Da bi bila moguća rezervacija stola, treba izvršiti rezervaciju mjesta u restoranu. Odabirom restorana iz pregleda na početnoj stranici i pritiskom na gumb za rezervaciju otvara se obrazac za rezervaciju. Korisnik odabire stol i šalje svoj odabir poslužitelju. Poslužitelj tad u bazi podataka provjerava dostupnost stola. Ukoliko je odabrani stol dostupan on se rezervira te se korisniku šalje poruka da je rezervacija uspješno obavljena i pruža mu se mogućnost rezervacije jelovnika. Ako stol nije slobodan korisnik mora odabrati neki drugi stol. Sekvencijski dijagram rezervacije stola prikazan je na slici 4-4.



Slika 4‑6 – Sekvencijski dijagram za rezervaciju stola

**Obrazac uporabe UC5: Rezervacija jelovnika**

Korisnik bira jelovnik po svom ukusu te ga rezervira. Da bi korisnik imao mogućnost odabira jelovnika, on mora imati rezervirano mjesto i/ili stol u restoranu. Poslužitelj prima rezervaciju te provjerava u bazi podataka ima li dovoljno zaliha namirnica potrebnih da se napravi naručeni jelovnik. Ako ima dovoljno zaliha, korisnik dobije poruku u kojoj se potvrđuje rezervacija jelovnika. Ako zaliha nema, rezervacija se odbacuje i korisnik prima poruku o neuspješnoj rezervaciji jelovnika. Korisnik također može odabrati hoće li plaćanje obaviti unaprijed, neposredno preko sustava ili dolaskom u restoran. Sekvencijski dijagram rezervacije jelovnika prikazan je na slici 4-5.



Slika 4‑7 – Sekvencijski dijagram za rezervaciju jelovnika

**Obrazac uporabe UC6: Poništavanje rezervacije**

Korisnik želi poništiti svoju rezervaciju i šalje zahtjev za poništavanjem. Da bi korisnik mogao poništiti rezervaciju, on mora zahtjev za poništavanje poslati u vremenskom roku u kojem je moguće poništiti rezervaciju. Poslužitelj prima zahtjev za poništavanjem te u bazi podataka provjerava je li prošao rok za poništavanje rezervacije. Ukoliko rok nije prošao, rezervacija se briše iz baze podataka te se korisniku javlja da je poništavanje rezervacije uspješno obavljeno. Ako je rok prošao, obavještava se korisnik da nije moguće poništiti rezervaciju. Sekvencijski dijagram poništavanja rezervacije prikazan je na slici 4-6.



Slika 4‑8 – Sekvencijski dijagram za poništavanje rezervacije

**Obrazac uporabe UC7: Odjava**

Prijavljeni korisnik može završiti korisničku sjednicu odjavom iz sustava. Ako je posjetitelj stranice prijavljen, pritiskom na gumb „Log Out“ poslužitelju se šalje zahtjev za odjavu te sjednica završava. Sekvencijski dijagram odjave prikazan je na slici 4-7.



Slika 4‑9 – Sekvencijski dijagram za odjavu korisnika

**Obrazac uporabe UC8: Promjena korisničkih podataka**

Korisnik želi izmijeniti neke svoje podatke:

* Korisničko ime
* Ime i prezime
* Lozinku
* Adresu e-pošte

i dodatno, ako ih je prethodno odlučio unesti, podatke za plaćanje, adresu i telefon.  
Tada on ode na opciju izmjene podataka i unese novi podatak kojim želi zamijeniti postojeći. Poslužitelj tada izmjenjuje podatke u bazi podataka ukoliko je to moguće. Ako je zapis u bazi uspješno promijenjen, korisnik dobije potvrdu da je izmjena uspješno napravljena. Ako zapis u bazi nije mogao biti promijenjen, korisnik dobije obavijest da izmjena nije moguća te razlog zašto nije moguća zamjena podataka. Sekvencijski dijagram promjene korisničkih podataka prikazan je na slici 4-8.



Slika 4‑10 – Sekvencijski dijagram za promjenu korisničkih podataka

**Obrazac uporabe UC9: Promjena podataka o restoranu**

Ugostitelj želi izmijeniti neki od podataka o restoranu:

* naziv restorana
* kratki opis
* detalje o kapacitetima ( broj i raspored stolova i stolaca)
* jelovnik

Tada on klikne na opciju izmjene podataka i upiše nove podatke kojima želi zamijeniti stare.

Poslužitelj prima nove podatke te u bazi podataka izmjenjuje postojeće podatke novima ukoliko je to moguće. Ako je zapis u bazi podataka uspješno promijenjen, ugostitelju se prikaže ažurirana stranica. Ukoliko izmjena nije bila moguća, dobije obavijest da se izmjena nije mogla izvršiti te razlog zašto se nije mogla izvršiti. Sekvencijski dijagram promjena podataka o restoranu prikazan je na slici 4-9.



Slika 4‑11 – Sekvencijski dijagram za promjenu podataka o restoranu

**Obrazac uporabe UC10: Oglašavanje ponude**

Ugostitelj želi oglasiti svoj restoran. Da bi to bilo moguće, on mora biti prijavljen. Na mjestu za unos ponude restorana ugostitelj unosi podatke o svojem restoranu:

* naziv restorana
* kratki opis,
* detalje o kapacitetima ( broj i raspored stolova i stolaca)
* jelovnik

Poslužitelj prima ponudu te je dodaje u bazu podataka ukoliko je to moguće. Ako je ponuda uspješno dodana u bazu podataka, ugostitelj dobije obavijest da je ponuda uspješno napravljena te se pojavio oglas na web stranici koji sadrži ponudu restorana. Ukoliko nije bilo moguće ponudu dodati u bazu podataka, ugostitelj dobije poruku da se dogodila pogreška, da nije bilo moguće postaviti oglas te razlog zbog kojeg nije bilo moguće postaviti oglas. Sekvencijski dijagram oglašavanja ponude prikazan je na slici 4-10.



Slika 4‑12 – Sekvencijski dijagram za oglašavanje ponude

**Obrazac uporabe UC11: Recenziranje restorana na vlastitom web prostoru**

Gastro-inspekcija na svom web prostoru želi recenzirati restoran, klasificirati ga te otvoriti posjetiteljima prostor za ocjenjivanje i komentiranje. Ona klikom na gumb „Recenziraj nas na svom site-u“ šalje zahtjev za kodom restorana da bi joj se to omogućilo. Poslužitelj u bazi podataka dohvaća kod tog restorana. Ukoliko je moguće dohvatiti kod u bazi podataka, taj kod se prikaže gastro-inspekciji. Ako nije moguće dohvatiti kod gastro-inspekcija dobije obavijest da je došlo do pogreške i da se kod ne može dohvatiti. Sekvencijski dijagram recenziranja restorana na vlastitom web prostoru prikazan je na slici 4-11.



Slika 4‑13 – Sekvencijski dijagram za recenziranje restorana na vlastitom web prostoru

**Obrazac uporabe UC12: Izmjena jelovnika**

Ugostitelj želi izmijeniti jelovnik. Klikne na izmjenu te unese izmjene koje želi napraviti u jelovniku. Da bi ugostitelj mogao izmijeniti jelovnik, nužno je da je taj jelovnik već prije napravio. Poslužitelj tada unosi promjene u bazu podataka. Ako je unos izmijenjenih podataka moguć, ugostitelju se prikaže ažurirani jelovnik. Ukoliko nije moguće u bazi promijeniti podatke, ugostitelj dobije poruku da je nastala greška, da unos nije moguć te razlog zbog kojeg je greška nastala. Sekvencijski dijagram izmjene jelovnika prikazan je na slici 4-12.

****

Slika 4‑14 – Sekvencijski dijagram za izmjenu jelovnika

**Obrazac uporabe UC13: Stvaranje jelovnika**

Ugostitelj želi u oglasu za svoj restoran napraviti jelovnik. Tada on klikne na mjesto predviđeno za unos jelovnika u oglas te dodaje jelovnik sastavljen od jela koje grupira prema vrstama (meso, riba, vegetarijanska, salate, topla predjela, hladna predjela, slastice...). U ponudi navede i ponudu pića prema vrstama. Također može definirati specifične kategorije jela („gableci“ po danima, posebne ponude, specijaliteti...). Kada ugostitelj unese jelovnik, poslužitelj ga pokušava zapisati u bazu podataka. Ako je zapis bio moguć, ugostitelj dobije potvrdu da je jelovnik uspješno dodan te se prikaže uneseni jelovnik. Ukoliko nije bilo moguće zapisati jelovnik u bazu podataka, ugostitelj dobije obavijest da nije moguće dodati zapis te razlog zbog kojega to nije moguće. Sekvencijski dijagram stvaranja jelovnika prikazan je na slici 4-13.

****

Slika 4‑15 – Sekvencijski dijagram za stvaranje jelovnika

**Obrazac uporabe UC14: Pregled zaliha**

Ugostitelj u svakom trenutku može vidjeti stanje zaliha restorana. Ugostitelj šalje zahtjev za pregledom zaliha. Poslužitelj tada u bazi podataka dohvaća podatke o zalihama, te ako je dohvat podataka o zalihama bio moguć ugostitelju se prikazuje stanje zaliha. Ukoliko dohvat podataka nije bio moguć, pošiljatelj dobije poruku da se dogodila greška i da nije moguće prikazati podatke. Sekvencijski dijagram pregleda zaliha prikazan je na slici 4-14.



Slika 4‑16 – Sekvencijski dijagram za pregled zaliha

**Obrazac uporabe UC15: Izmjena stanja zaliha**

Ugostitelj, osim što ima uvid u zalihe restorana, može ih i promijeniti sukladno s potrošenim namirnicama. Kada se neka namirnica iskoristi, ugostitelj mijenja zalihe te namirnice. Poslužitelj tada u bazu podataka upisuje izmijenjene podatke. Ukoliko je upis izmijenjenih zaliha u bazu podataka moguć, ugostitelj dobije potvrdu da je izmjena zaliha uspješno obavljena. Ako izmjena nije moguća, ugostitelj dobije poruku da se dogodila greška, da izmjena podataka nije obavljena te dobije razlog zbog kojeg se izmjena nije izvršila. Sekvencijski dijagram izmjene stanja zaliha prikazan je na slici 4-15.



Slika 4‑17 – Sekvencijski dijagram za izmjenu stanja zaliha

**Obrazac uporabe UC16: Ocjenjivanje restorana**

Prijavljeni korisnik može ocijeniti restoran te komentirati uslugu u restoranu. Da bi on to mogao mora postojati njegova rezervacija u restoranu koji se ocjenjuje jer ne može ocijeniti niti komentirati restoran u kojem nije bio. Korisnik doda svoju ocjenu te poslužitelj ažurira ukupnu ocjenu restorana u bazi podataka na temelju dane ocjene. Ukoliko želi, korisnik može ostaviti komentar o restoranu koji poslužitelj tada upisuje u bazu podataka. Sekvencijski dijagram ocjenjivanja restorana prikazan je na slici 4-16.



Slika 4‑18 – Sekvencijski dijagram za ocjenjivanje restorana

**Obrazac uporabe UC17: Obavljanje transakcije**

Korisnik nakon uspješno izvršene rezervacije i odabranog jelovnika može izabrati izravno plaćanje. Da bi to bilo moguće, korisnik mora imati upisane podatke za plaćanje u svojim korisničkim podacima. Ako korisnik ima dovoljno sredstava na računu da se može izvršiti transakcija, dobije potvrdu da je transakcija uspješno izvršena. Ukoliko se dogodila greška pri transakciji, korisnik dobije obavijest da se dogodila pogreška i da transakcija nije bila uspješna. Sekvencijski dijagram obavljanja transakcije prikazan je na slici 4-17.



Slika 4‑19 – Sekvencijski dijagram za obavljanje transakcije

**Obrazac uporabe UC18: Pregled stanja stolova**

Konobar ima mogućnost pregleda stanja stolova, tj. tko je koji stol rezervirao kako bi bio u mogućnosti smjestiti goste za njihove stolove. Konobar šalje zahtjev za pregledom stanja stolova. Poslužitelj u bazi podataka dohvaća podatke o stanju stolova. Ako je bilo moguće dohvatiti podatke u bazi podataka, konobaru se prikaže stanje stolova. Ako nije bilo moguće dohvatiti podatke, konobar prima poruku da se dogodila pogreška, te da nije moguće prikazati stanje stolova. Sekvencijski dijagram pregleda stanja stolova prikazan je na slici 4-18.



Slika 4‑20 – Sekvencijski dijagram za pregled stanja stolova

# Ostali zahtjevi

* Odlučili smo se na korištenje PHP programskog jezika, te AngularJS za lakšu izradu web stranica. Sve nadogradnje bi trebale biti napravljene s istim alatima.
* Ukoliko korisnik napravi neku nepredviđenu ili krivu akciju, sustav ne smije biti narušen.
* Nadogradnja funkcionalnosti ne smije narušiti stabilnost sustava.
* Sustav treba moći podržati istovremeni rad više korisnika
* Sustav u svrhu ispitivanja mora imati barem sedam korisnika različitih vrsta, barem 3 restorana i barem 5 ponuđenih različitih jela za svaki od restorana.
* Poruke moraju imati jasno definirane formate koje razvijeni sustav mora valjano interpretirati te prema njima činiti odgovarajuće akcije u sustavu.
* Formati poruka trebaju biti razumljivi običnom korisniku.
* Korisnici trebaju imati mogućnost međusobne komunikacije u sustavu.
* Sustav treba biti jednostavan za korištenje.
* Odziv sustava ne bi trebao biti duži od 20 sekundi.
* Točnost informacija mora biti zajamčena.
* Sustav i dokumentacija trebaju biti izvedeni do veljače 2015. godine.

# Arhitektura i dizajn sustava

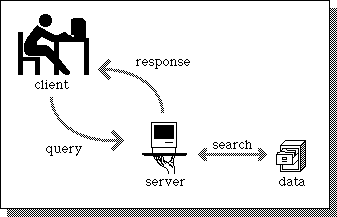
## Svrha, opći prioriteti i skica sustava

***Svrha i opći prioriteti sustava***

Izbor arhitekture sustava je bitan korak prilikom oblikovanja sustava. Različiti stilovi s njihovim prednostima i nedostacima trebaju biti razmatrani da bi se najbolje rješenje odabralo. Odabrana arhitektura mora biti dovoljno fleksibilna i omogućiti nadogradnju sustava.

Arhitektura našeg sustava je objektno usmjerena. Razlog odabira objektno usmjerene arhitekture jest njena stabilnost, pouzdanost, jednostavnog oblikovanja te mogućnost jednostavnijeg održavanja sustava i nadogradnje istoga. Sustav je podijeljen na razrede, što omogućava višu razinu apstrakcije koja je bliža ljudskom načinu razmišljanja. Točnije, sustav koji gradimo je raspodijeljeni sustav pa je povezivanje različitih zadaća sustava olakšano. Naš sustav ima dvije osnovne zadaće: traženje obavljanja zadaće (zahtjev klijenta) te samo obavljane tražene zadaće (obrada zahtjeva i vraćanje odgovora). Ovakav tip arhitekture je model klijent-poslužitelj arhitektura. Klijent-poslužitelj model susrećemo svakodnevno na svim tipovima web stranica, elektroničke pošte i mnogih drugih.

Vremenski tijek komunikacije klijent – poslužitelj se može podijeliti na faze (slika 6-1). Poslužitelj započinje s radom i čeka zahtjeve od klijenta. Klijent koristi sustav, obavlja neki rad i razne operacije te (po potrebi) započinje komunikaciju s poslužiteljem na način da šalje zahtjev za uslugom ( *request* ). Poslužitelj ( *server* ) je središnje računalo koje prima i obrađuje korisničke zahtjeve te vraća odgovor. Poslužitelj nije dostupan za obavljanje usluga samo jednom klijentu, on osigurava usluge većem broju klijenata. Pokušavanjem spajanja klijenta na poslužitelj, poslužitelj može odbiti ili dozvoliti vezu ovisno o dozvolama koje klijent ima. Sljedeća faza je obrada zahtjeva te nakon nje vraćanje rezultata klijentu. Navedeni proces se nastavlja sve do prestanka rada.



Slika 6‑1 – Vremenski tijek koncepta „klijent – poslužitelj“

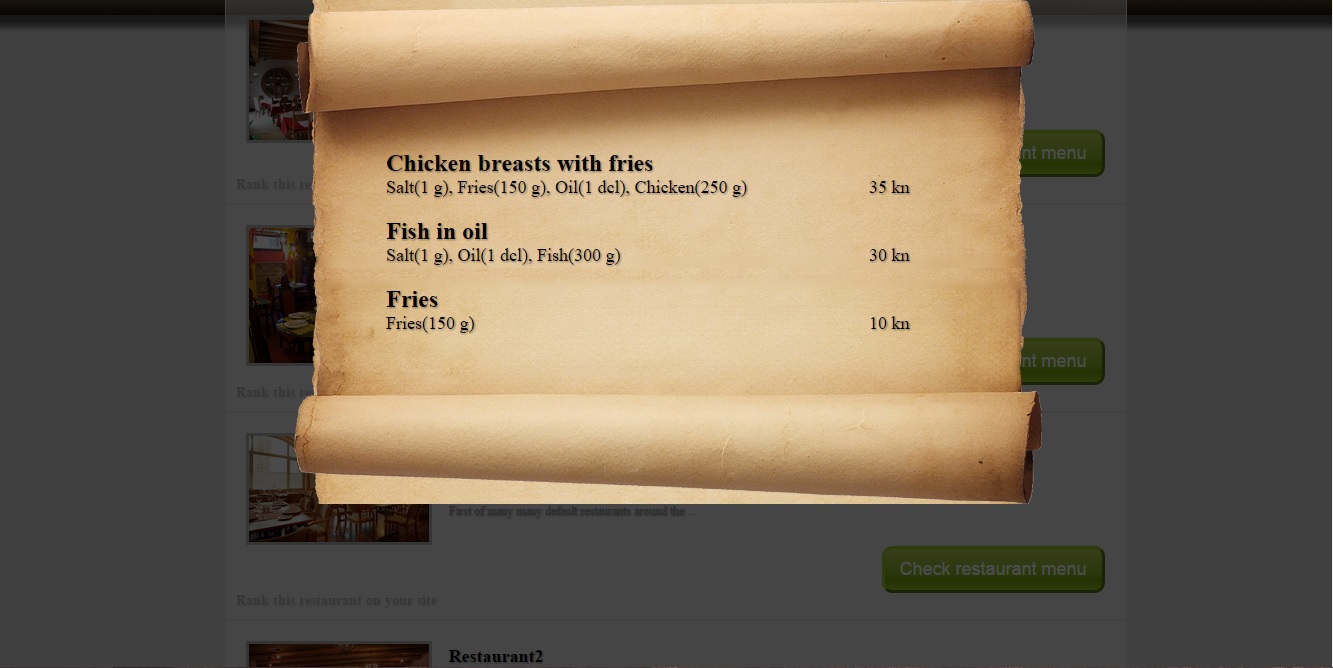
Primjer jedne klijent – poslužitelj zadaće je promjena korisničkih podataka. Ako registrirani korisnik želi promijeniti svoje podatke, mora pristupiti bazi podataka da bi prvo dobio uvid u svoj profil. Komunikacija se obavlja preko poslužitelja, koji pripremi tražene podatke i proslijedi ih klijentu. Ovaj model je odabran radi velike fleksibilnosti korisnicima. Administrator tako ima veće mogućnosti nadziranja sustava, a poslužitelj je zadužen za obradu svih zahtjeva od strane administratora i strane svih ostalih korisnika.

Arhitektura opisanog sustava sastoji se od sljedećih podsustava:

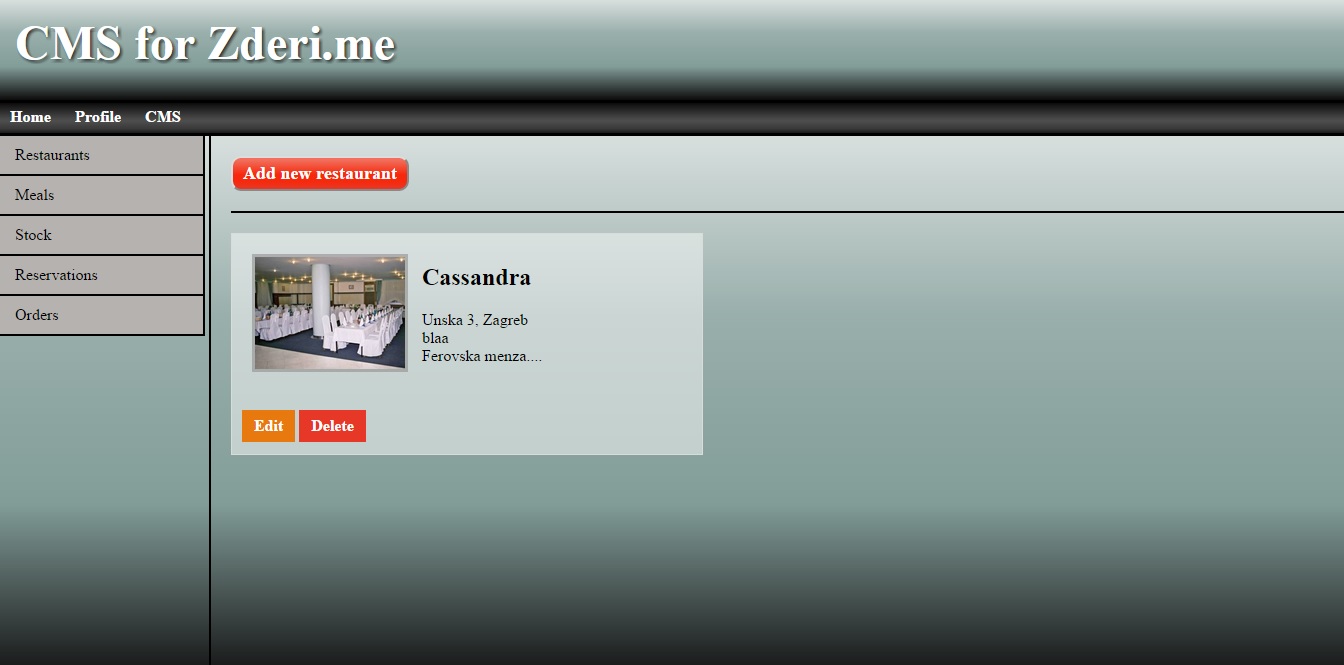
* Grafičko sučelje
* Baza podataka***Grafičko sučelje***

Grafičko korisničko sučelje (engl. Graphical User Interface – GUI) omogućuje korištenje svih funkcija sustava unutar vizualnog sučelja, što jako pojednostavljuje korištenje. Ono omogućuje svim korisnicima da intuitivno i bez ikakvog dodatnog znanja i vještina, na jednostavan i brz način, dođu do zacrtanog cilja.

Grafičko sučelje naše stranice je napisano od početka, bez korištenja dodatnih alata. Dizajn našeg grafičkog sučelja je vrlo moderan i jednostavan za korištenje. Na početnoj stranici (slika 1) se nalazi popis restorana, te se klikom na gumb Check restaurant menu dolazi do jelovnika pojedinog restorana.

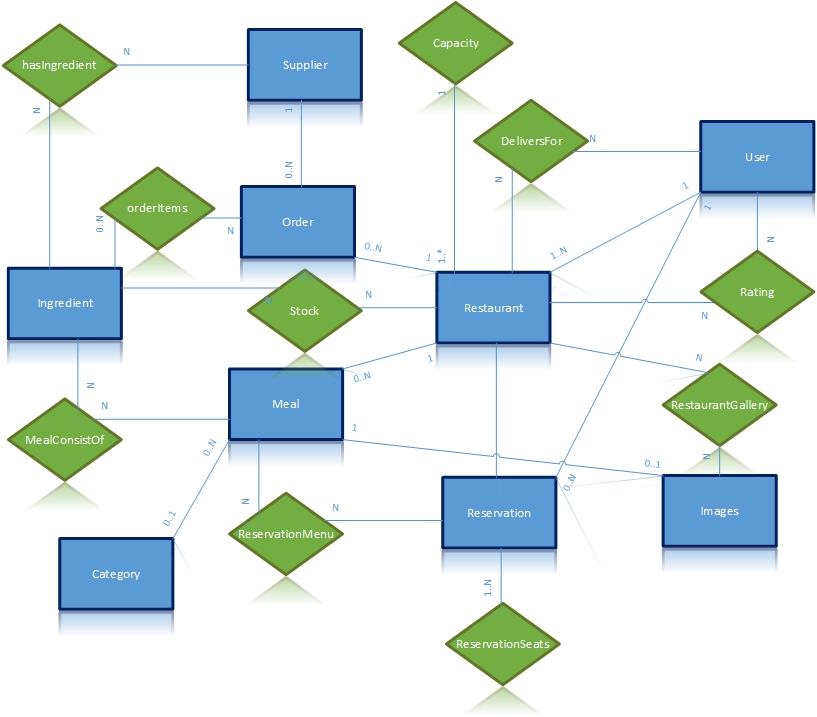


U našem sustavu postoji više vrsta korisnika koji se mogu registrirati. Korisnik, koji može rezervirati mjesta u restoranu i ocjenjivati ih. Dobavljač, koji dodatno može dobavljati sastojke vlasnicima restorana koji mogu dodavati svoje restorane i jela koja nude njihovi restorani. Na slici 3 je izgled CMS-a jednog vlasnika restorana.

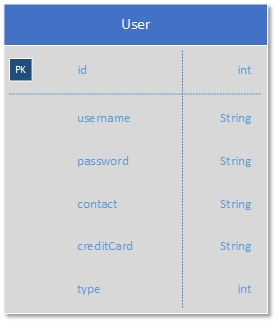


***Baza podataka***

Za potrebe web aplikacije odlučili smo koristiti relacijsku bazu podataka koja je u današnje vrijeme jedna od najkorištenijih modela baze podataka zbog relacijske strukture koja se podudara s predodžbom događaja realnog svijeta i podataka koje ti događaji generiraju. Relacijska baza podataka sastoji se od skupa relacija (tablica) pri čemu je svaka relacija opisana relacijskom shemom koja obuhvaća naziv relacije i skup pripadajućih atributa. Relacije su provedene kroz normalizacijske forme da bi se ostvario što bolji model sa što manje redundancije podataka. Izradu baze podataka ćemo ostvariti koristeći *MySQL*, a točniji pregled dizajna baze se može naći na slici Slika 6-2 gdje možemo vidjeti i veze među entitetima. Sve relacije generirane iz ER-modela su prikazane nakon ER-modela s odgovarajućim opisima atributa te primarnim i stranim ključevima.



Slika ‑ – Dizajn baze podataka

  
  
  
- **id**: šifra korisnika

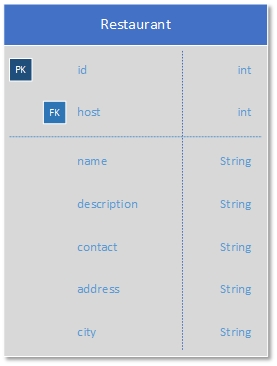
- **username** : korisničko ime za spajanje na sustav

- **password** : lozinka za spajanje na sustav

- **contact** : kontakt broj korisnika (mobilni telefon ili telefon)

- **creditCard** : broj kreditne kartice preko koje se vrši naplata

- **type** : zastavica za raspoznavanje tipa korisnika (gost, ugostitelj, dobavljač)



- **id** : šifra restorana

- **host** : šifra ugostitelja (vlasnika) restorana

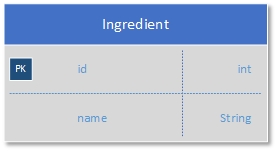
- **name** : naziv restorana

- **description** : kratki opis restorana

- **contact** : kontakt broj restorana

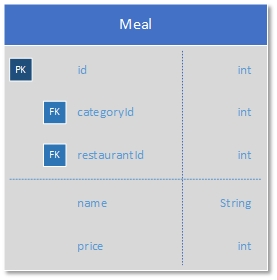
- **address** : fizička adresa restorana

- **city** : grad u kojem se nalazi restoran



- **id** : šifra namirnice

- **name** : naziv namirnice

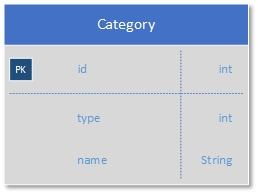
- **id** : šifra jela

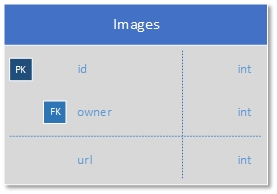
- **categoryId** : šifra pripadne kategorije jela

- **restaurantId** : šifra restorana

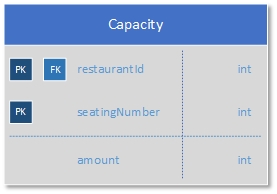
- **name** : naziv jela

- **price** : cijena jela u HRK

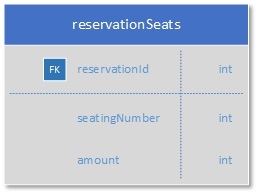
  
  
- **id** : šifra kategorije jela  
- **type** : tip kategorije (aktivna ili ne)  
- **name** : naziv kategorije

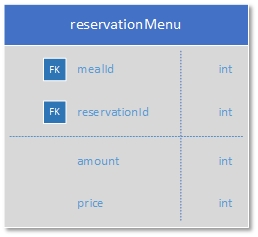


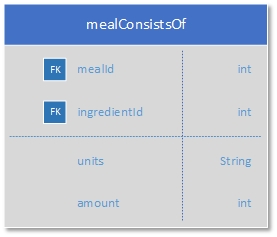
- **id** : šifra slike  
- **owner** : šifra korisnika koji je postavio sliku  
- **url** : fizički put do slike

  
  
  
- **restaurantId** : šifra restorana  
- **seatingNumber** : veličina sjedeće pozicije (1 sjedeće mjesto, 2 sjedeća mjesta, itd.)  
- **amount** : količina pripadajućih sjedećih pozicija

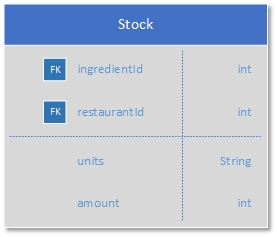
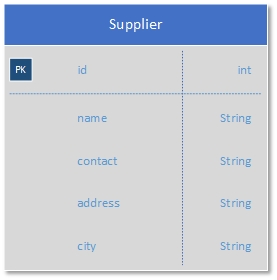
- **id** : šifra rezervacije  
- **userId** : šifra korisnika koji je postavio rezervaciju  
- **restaurantId** : šifra restorana u kojem je postavljena rezervacija  
- **barcode** : bar kod narudžbe  
- **timestamp** : vrijeme rezervacije  
- **status** : status rezervacije (u toku, plaćeno, otkazano)

  
  
- **reservationId** : šifra rezervacije  
- **seatingNumber** : veličina sjedeće pozicije  
- **amount** : količina sjedećih pozicija

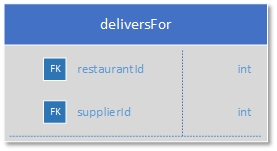
  
  
  
- **mealId** : šifra jela  
- **reservationId** : šifra rezervacije  
- **amount** : količina jela  
- **price** : cijena jela u trenutku postavljanja rezervacije

  
  
  
- **mealId** : šifra jela

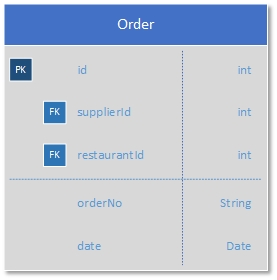
- **ingredientId** : šifra namirnice  
- **units** : mjerna jedinica količine namirnice (g, l)  
- **amount** : količina namirnice

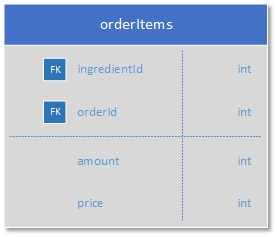
  
  
- **ingredientId** : šifra namirnice  
- **restaurantId** : šifra restorana  
- **units** : mjerna jedinica količine namirnice (g, l)  
- **amount** : količina namirnice

- **id** : šifra dobavljača  
- **name** : naziv dobavljača  
- **contact** : kontakt broj dobavljača (mobilni telefon ili telefon)  
- **address** : fizička adresa dobavljača  
- **city** : grad u kojem se nalazi dobavljača

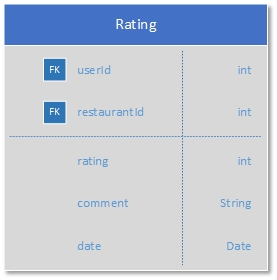
- **restaurantId** : šifra restorana  
- **supplierId** : šifra dobavljača

  
  
- **ingredientId** : šifra namirnice  
- **supplier** : šifra dobavljača  
- **price** : trenutna cijena namirnice

- **id** : interna šifra narudžbe  
- **supplierId** : šifra dobavljača  
- **restaurantId** : šifra restorana  
- **orderNo** : službena šifra narudžbe (od dobavljača do dobavljača drukčija)  
- **date** : datum narudžbe

  
  
- **ingredientId** : šifra namirnice  
- **orderId** : šifra narudžbe  
- **amount** : količina namirnice  
- **price** : cijena namirnice u trenutku postavljanja narudžbe

  
  
  
- **imageId** : šifra slike restorana  
- **restaurantId** : šifra restorana

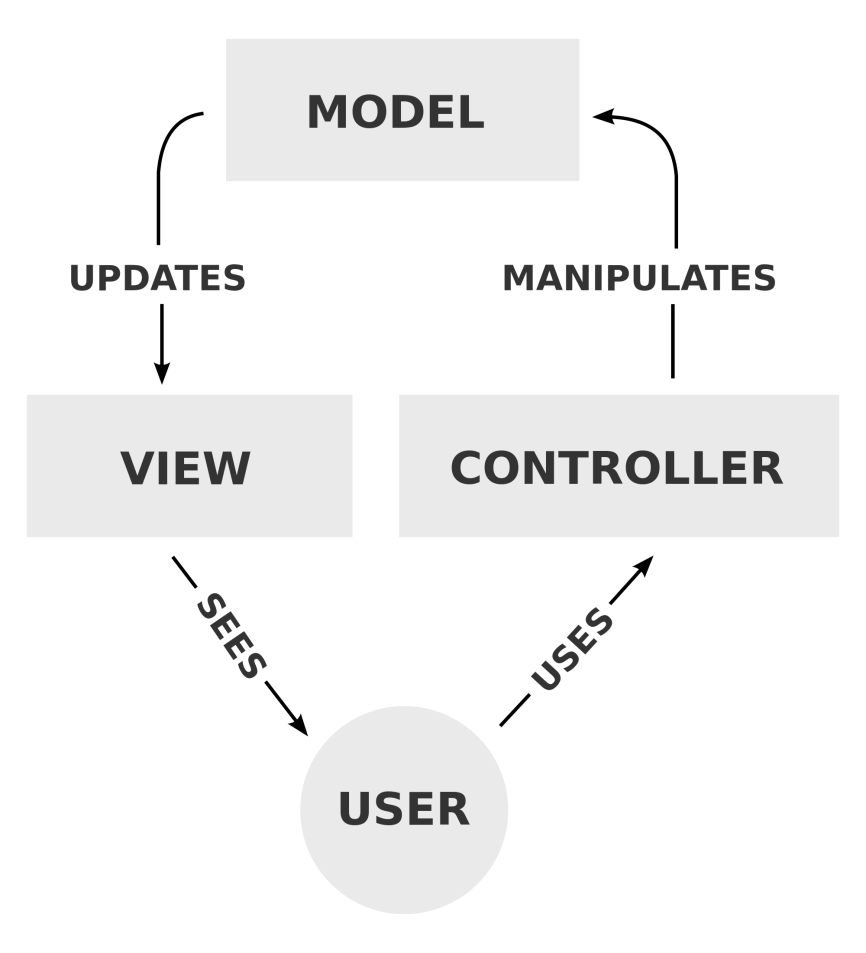
  
  
  
  
- **userId** : šifra korisnika  
- **restaurantId** : šifra restorana  
- **rating** : ocjena restorana  
- **comment** : dodatni komentar na ocjenu  
- **date** : datum postavljanja ocjene

## Dijagram razreda s opisom

Razredi su jedinice apstrakcije u objektno orijentiranoj paradigmi. Razred sadrži opis strukture instanci, tj. podatke koji implementiraju obilježja i metode koje implementiraju ponašanje razreda. Dijagram razreda možemo definirati kao cjelokupnu logiku sustava te veza i vidljivosti modela koji ga sačinjavaju. Simboli prikazani na dijagramima razreda su: razredi, pridruživanja (apstraktni prikaz veza koje postoje između razreda), atribute, operacije i generalizacije (grupiranje razreda u hijerarhiju nasljeđivanja).

AngularJS je Javascript framework koji uvelike olakšava kreiranje web aplikacija pa smo se mi odlučili na kombinaciju AngularJS-a i PHP-a kao backend. Arhitektura sustava koristi objektno orijentirani pristup, pa AngularJS s ugrađenim MVC (*Model-ViewController*) modelom daje odlično rješenje za nas. MVC je oblikovni obrazac koji odvaja prezentacijski, logički i podatkovni dio aplikacije, a u AngularJS-u je to izvedeno kroz direktive i *scope* te kontrolere. Koristeći PHP kao backend servis, povezujemo bazu podataka s kontrolerima i prikazom cijelog dizajna.

AngularJS MVC model se bazira na HTML stranici kao *View* sloju s definiranim klasama kontrolera unutar kojih svaki ima svoj model koji je definiran kroz varijablu *scope*. Koristeći Javascript i jQuery elemente uz postojeće kontrole nad objektima od AngularJS-a ćemo ostvariti potpunu funkcionalnost stranice.  
  
MVC je oblikovni obrazac čija se funkcionalnost temelji na tri međusobno povezane komponente. *Model* je komponenta koja služi za pohranu aplikacijske i podatkovne logike. Omogućava uvid u podatke i stanja aplikacije na unaprijed predviđene načine i promjenu podataka i stanja pomoću unaprijed predviđenih modela. *View* je komponenta zaslužna za prikaz podataka i interakciju s korisnicima. *Controller* je komponenta koja povezuje sve zajedno. Zaslužna je za kontrolu korisničkih događaja poput klika mišem ili unos tipkovnicom. Ujedno, omogućuje ponovnu uporabu dijela programa zaslužnog za logiku aplikacije.



Slika 6‑3 – Prikaz oblikovnog obrasca MVC

U nastavku je svaki dio MVC-a detaljnije opisan, te su navedeni i opisani razredi svakog sloja.

***Controller***

Ovaj sloj upravlja interakcijom s korisnikom na temelju njegovih akcija na stranici (klik, unos preko tipkovnice, pomak miša, itd.). Interpretira akcije te oblikuje podatke iz modela te ih prikazuje. Razredi ovog sloja su:

**RestaurantController:**

Kontroler zaslužan za prikazivanje svih restorana na početnoj stranici te povezivanja rezervacije s prikazom ponuda. Svaki restoran ima svoju ponudu gdje su navedeni najosnovniji podatci. Uz popis restorana, moguće je pogledati cjelokupni meni pojedinog restorana. Svaki restoran je moguće ocijeniti (ukoliko je korisnik registriran, odnosno, prijavljen).

**ReservationController:**

Omogućuje prijavljenim korisnicima da rezerviraju mjesto u jednom od ponuđenih restorana. Pruža metode za odabir vremena, sjedećih mjesta te menija za večeru.

**HostController:**

Omogućuje ugostiteljima rad u CMS-u. Pruža metode za dodavanje kategorija jela, unošenje novih jela, kreiranje narudžbi, obavljanje transakcija i kontrolu nad rezervacijama te uvid u zalihe restorana.

**SupplierController:**

Omogućuje dobavljačima rad u CMS-u. Pruža metode za potvrđivanje, izvršavanje i otkazivanje narudžbi te dodavanje dostupnosti određenih namirnica.

**UserController:**

Omogućuje korisnicima uvid u osobne podatke. Pruža metode za izlistavanje dosadašnjih rezervacija i promjenu osobnih podataka.

**AdminController:**

Omogućuje administratorskim računima potpunu kontrolu nad svih ostalim komponentama sustava. Također, može mijenjati postavke pojedinih namirnica kao i dodavanje novih.

**LoginController:**

Omogućava prijavu korisnika u sustav. Prilikom prijave se autorizira korisnik te mu se prikazuje dio za koji ima ovlasti.

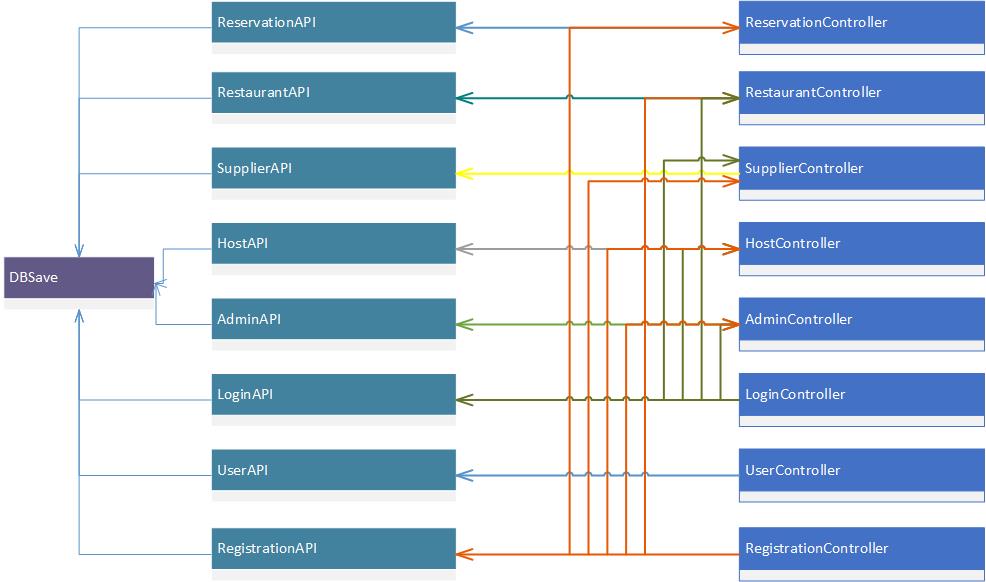
**RegistrationController:**

Omogućava registraciju korisnika u sustav. Prilikom unosa podataka automatski se isti validiraju te se korisnik opredjeljuje za tip računa.

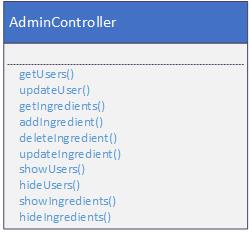
Uz sve navedene funkcionalnosti, Controlleri upravljaju i autorizacijom korisnika te omogućuju prikaz pojedinih dijela samo korisnicima koji imaju za to dodijeljene ovlasti.

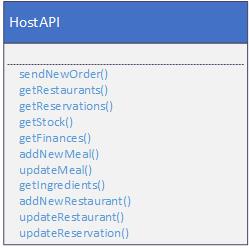
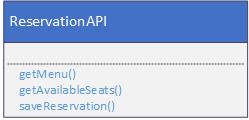
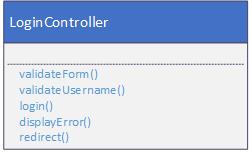
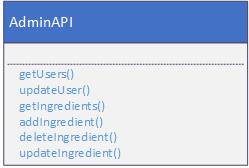
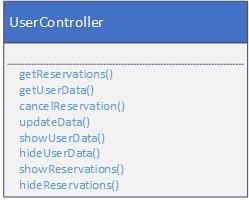
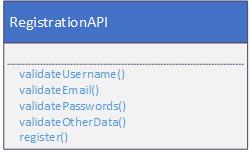
Da bi se Controlleru olakšalo upravljanje podacima, između njega i baze podataka postoji sloj takozvanih „Backend API“-a. Svaki razred ovog sloja omogućava validaciju, prikaz, unos, brisanje i uređivanje podataka. Svaki „Backend API“ je imenovan u skladu s imenom odgovarajućeg Controllera (primjer: *LoginController* koristi *LoginAPI* ). Veze između pojedinih razreda su prikazane na dijagramu razreda (slika 6-4).

Najvažnija prednost ovakve arhitekture je apstrakcija pristupa podacima – Controlleri ne znaju što se događa iza sučelja za pristup podacima i kako se njima upravlja, zna samo da će dobiti odgovor. Neke dodatne prednosti su lakša izgradnja sustava, veća fleksibilnost, čitljiviji kod…



Slika 6‑4 Dijagram razreda



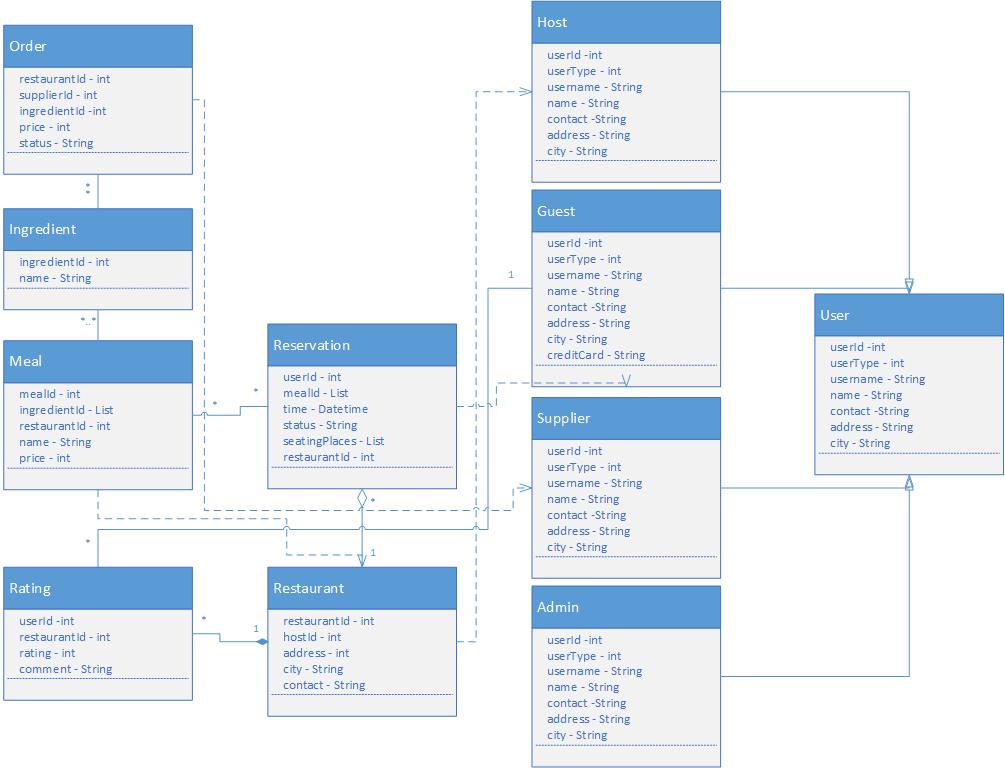


***Model***

Model je sloj u kojem se nalaze svi podaci. U AngularJS-u ovu ulogu ima *scope* koji je ulazna varijabla za Controller. Svaki Controller ima svoj *scope*. U kombinaciji s bazom podataka imamo vrlo moćan model. S obzirom da je model podataka detaljno opisan u opisu baze podataka, nema potrebe za detaljnijim razmatranjem ovdje. Prikaz ovisnosti podataka u modelu može se naći na slici .

***View***

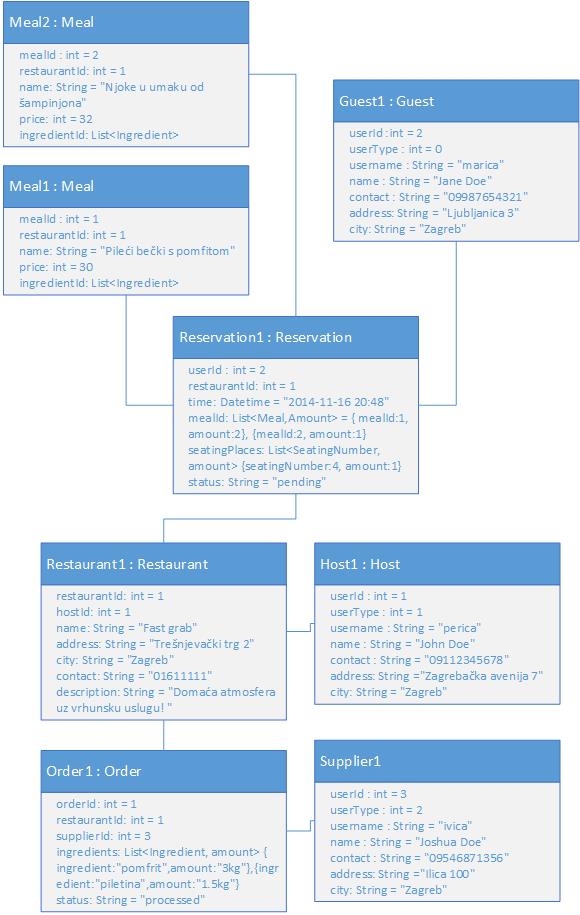
Sloj zaslužan za prikaz podataka. AngularJS definira *View* kao HTML kod na web stranici na koju povezuje podatke iz *scope*-a. Detaljnije razmatranje ovog sloja će biti obrađeno u 7. poglavlju.



Slika 6‑5 – Prikaz ovisnosti podataka o modelu (sloju)

## Dijagram objekata

Dijagram objekata na slici 6-6 prikazuje stanje sustava kada u sustavu postoji 1 restoran naziva „Fast Grab“ s njegovim vlasnikom „John Doe“ i njegova dva jela. U sustavu također postoji gost „Jane Doe“ s postavljenom rezervacijom određenog datuma. Također, postoji dobavljač „Joshua Doe“ i jedna obavljena narudžba između dobavljača i restorana.



Slika 6‑6 – Dijagram objekata

## Ostali UML dijagrami

***Komunikacijski dijagram***

***Dijagram stanja***

***Dijagram aktivnosti***

Dijagram aktivnosti prikazuje temeljnu funkcionalnost sustava, to jest postupak rezervacije jelovnika sa cjelokupnim prikazom radnog toka. Operacije se odvijaju slijedno te obuhvaćaju više obrazaca uporabe od početne prijave, same rezervacije mjesta i jelovnika pa sve do odustajanja od same rezervacije.



Slika 27 - Dijagram aktivnosti

***Dijagram komponenti***

Dijagram komponenti prikazuje međuovisnost komponenti sustava na implementacijskoj razini. Korišteni MVC model se oslanja na arhitekturu klijent-poslužitelj na način opisan dijagramom: korisnik u web-pregledniku vidi korisničko sučelje koje se sastoji od JavaScript funkcija koje obrađuju unose i izbore korisnika, PHP skripti koje služe za definiranje prikaza korisniku i CSS datoteka koje služe za formatiranje i stiliziranje prikaza. Na dijagramu su prikazane samo komponente ključne za funkcionalnost sustava te komunikaciju klijenta i poslužitelja (Controller).



6-9 Dijagram Komponenti

6-9

# Implementacija i korisničko sučelje

## Dijagram razmještaja

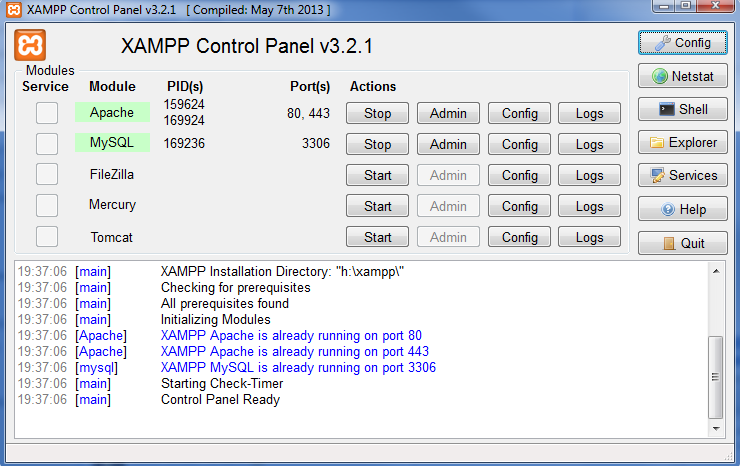
Dijagram 7-1 prikazuje osnovne komponente implementacije sustava – računalo poslužitelja na kojem je MySQL baza podataka. Na poslužiteljskom računalu se također izvršavaju skripte za provjere i dohvat podataka zahtijevanih od spojenih korisnika preko implementiranih API funkcija. Korisničko računalo je drugi bitan dio sustava, spaja se na poslužiteljsko računalo http protokolom. Iz korisničkog sučelja, kojeg čini funkcionalnost za formatiranje prikaza i za obradu korisničkog unosa, podaci se provjeravaju pri prijavi u sustav ili sinkrono pritiskom na pojedinu poveznicu ili tipku na sučelju.



7-1 Dijagram Razmještaja

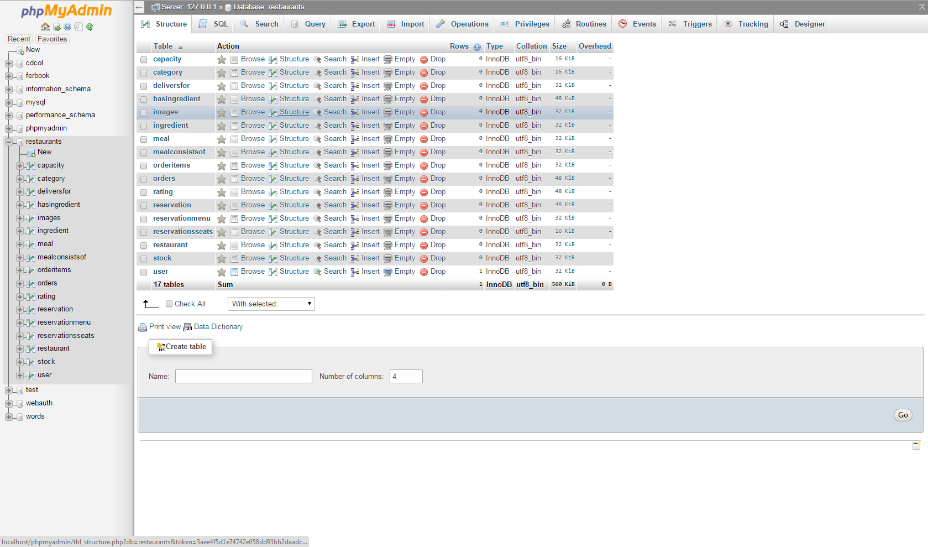
## Korištene tehnologije i alati

Pri početnoj izradi pa sve do konačne faze podizanja sustava na stvarni server, korištena je XAMPP platforma (verzija v3.2.1). XAMPP omogućava uspješno simuliranje i integriranje svih potrebnih servisa i usluga za rad virtualnog poslužitelja na običnom računalu. Za simuliranje rada na localhost-u korišten je Apache Server baziran na otvorenome kodu koji se pomoću XAMPP-a jednostavno pokreće u pozadinskom režimu rada.



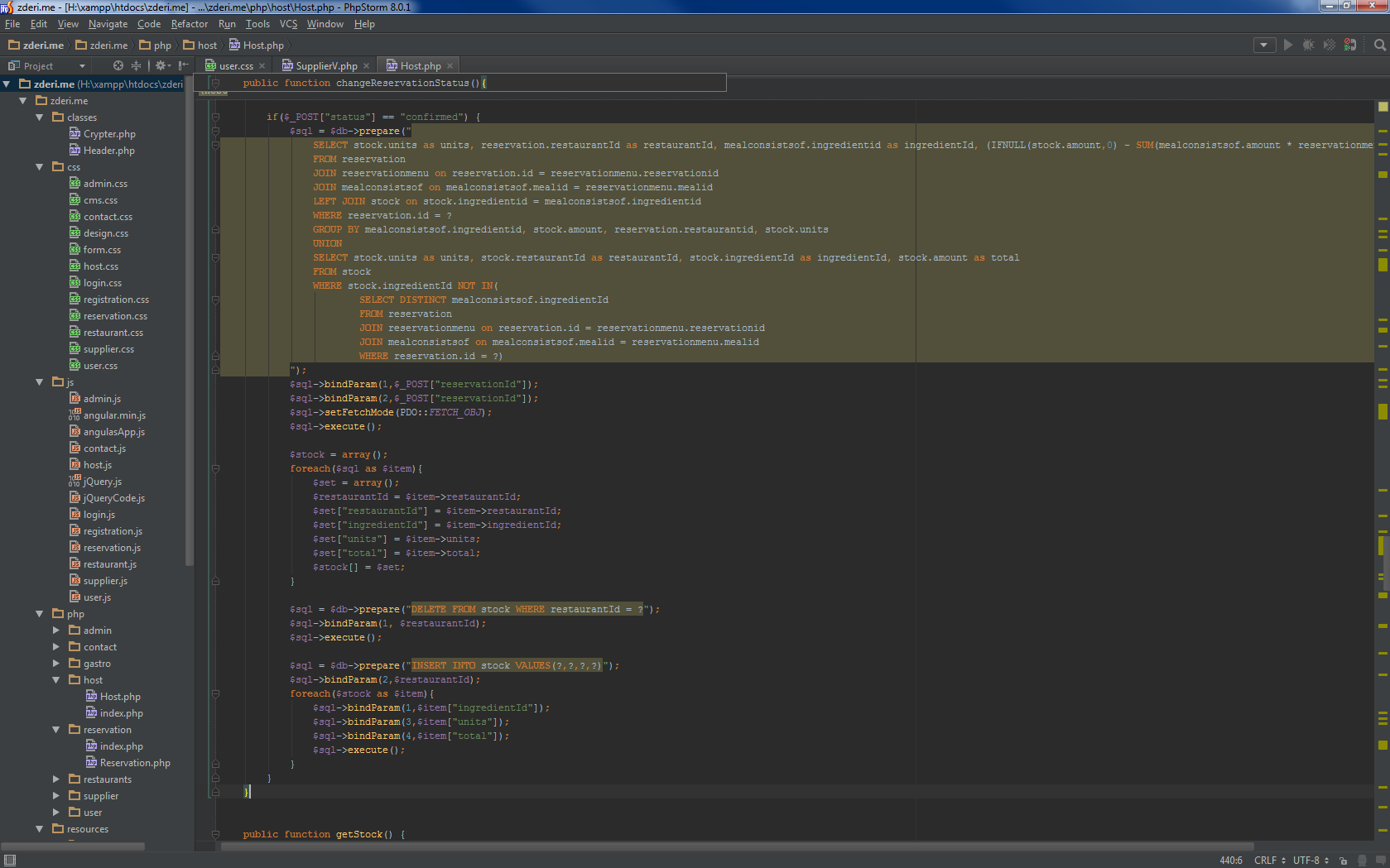
Slika 28 - XAMPP sa prikazom pokrenutih servisa

Za potrebe spremanja podataka korištena je MySQL relacijska baza podataka koja je dio XAMPP paketa te se u samo nekoliko koraka, uz pomoć phpMyAdmin sučelja za upravljanje bazom, jednostavno inicijalizira i pripremi za rad sa poslužiteljem.



Slika 29 - phpMyAdmin sustav za upravljanje MySQL bazom podataka

Kako bi se olakšala i ubrzala implementacija kao razvojno okruženje korišten je PhpStorm 8.0.1, koji uvelike pospješuje cjelokupnu izradu web projekata baziranih na PHP platformi. Osim intuitivnog i jednostavnog korištenja korisniku omogućava dinamičko ispravljanje i upozoravanje na potencijalne pogreške, te pomoću raznih dodataka integrira sve potrebne programske okvire za izradu web stranice.



Slika 30 - Razvojno okruženje (*IDE*) PhpStorm 8.0.1

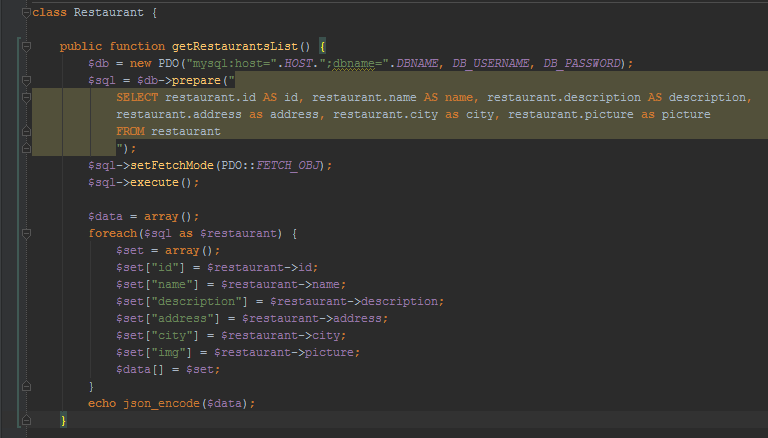
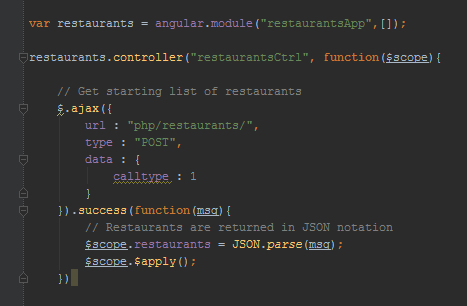
Sam sustav, kao što je prethodnim poglavljima opisano, izgrađen je u PHP platformi (verzija 5.3.28) uz korištenje AngularJS radnog okvira (verzija 1.2.28) koji olakšava implementaciju objektno usmjerene paradigme unutar aplikacije. Sučelje, raspored i oblikovanje entiteta organizirano je pomoću CSS3 stilskog jezika koji se izvrsno nadopunjuje sa HTML5 prezentacijskim jezikom, te omogućuje prikaz svih animacija, translacija i sličnih vizualnih prijelaza koji upotpunjavaju doživljaj i olakšavaju korištenje web stranice. Uz neizostavni JavaScript koji je postao nezamjenjiv posljednjih desetak godina, jQuery (verzija 1.11.1) izvrsno proširuje i povećava ponovnu uporabu programskih isječaka vezanih uz samo manipuliranje prikaza i sadržaja na stranici.

## Isječak programskog koda vezan za temeljnu funkcionalnost sustava

Programski kod na slikama 7.3.1, 7.3.2 i 7.3.3 pokazuje ostvarenje izlistavanja i prikaza liste restorana na početnoj stranici. Kod prikazuje komunikaciju između AngularJS-a kroz njegov *Controller* s PHP backend skriptom kroz AJAX poziv, koja je u ulozi *Modela* te HTML koda nadograđenim AngularJS-om da bi se ostvario *View*. Korišteni model razmjene se koristi u svim dijelovima projekta te je od temeljne važnosti za funkcionalnost sustava.



**Slika 7.3.1** *HTML- AngularJS kombinacija*



**Slika 7.3.3** *API za prikupljanje podataka iz baze*

**Slika 7.3.2** *Angular deklaracija i AJAX poziv*

## Ispitivanje programskog rješenja

**TEST 1: Prijava u sustav**

**Ulaz:**

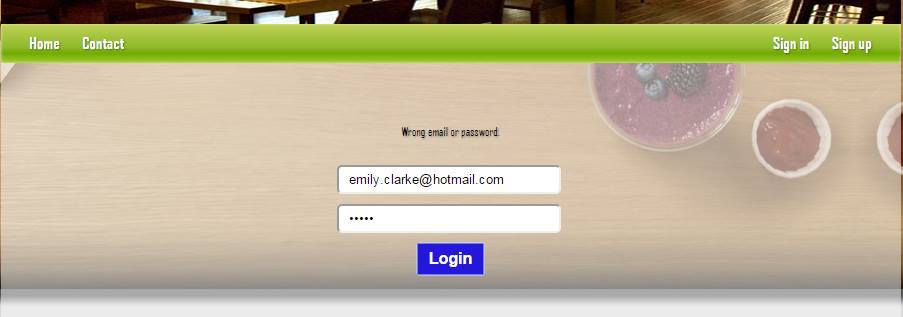
* Korisničko ime i lozinka za prijavu na sustav

**Očekivani izlaz:**

* Preusmjeravanje na početnu stranicu ako je prijava uspješna
* Poruka o pogrešnim korisničkim podacima u slučaju neuspjeha



Slika 31 - Test 1: Ispravna prijava, početna stranica



Slika 32 - Test 1: Neispravni korisnički podaci

**Izlaz**:

Izlaz odgovara očekivanom izlazu. Nakon svih provjera korisnik je uspješno prijavljen u sustav.

**TEST 2: Rezervacija**

**Ulaz:**

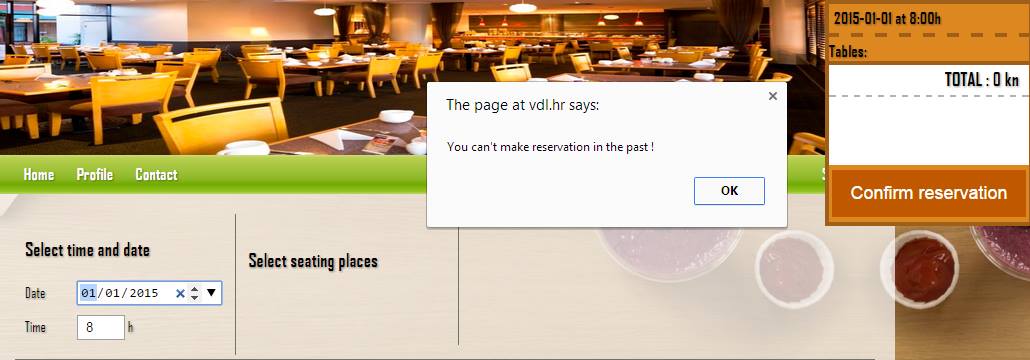
* Stavke narudžbe
* Vrijeme i mjesto rezervacije
* Odabir stola

**Očekivani izlaz:**

* Web obrazac pazi na to da je format datuma ispravno unesen te da su popunjena sva potrebna polja.
* Rezervacija je uspješno dodana u bazu i prikazan je broj rezervacije za korisnika.



Slika 33 - Test 2: Uspješna rezervacija



Slika 34 - Test 2: Neispravan unos datuma

**Izlaz**:

Izlaz odgovara očekivanom izlazu. Nakon svih provjera korisnik je uspješno obavio rezervaciju i dobio broj rezervacije.

**TEST 3: Dodavanje komentara i ocjena**

**Ulaz:**

* Tekst komentara i ocjena

**Očekivani izlaz:**

* Poruka o uspješnoj dodanoj ocjeni i eventualnom komentaru, ukoliko je korisnik prijavljen u sustav.



Slika 35 - Test 3: Korisnik je uspješno ocijenio i komentirao uslugu restorana

**Izlaz**:

Izlaz ne odgovara u potpunosti očekivanom izlazu. Nakon svih provjera korisnik dobije poruku „*Thank you for your opinion“* za svaki restoran , a ne samo za ocjenjeni.

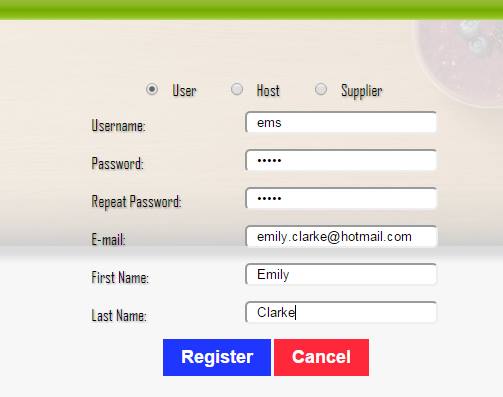
**TEST 4: Registracija**

**Ulaz:**

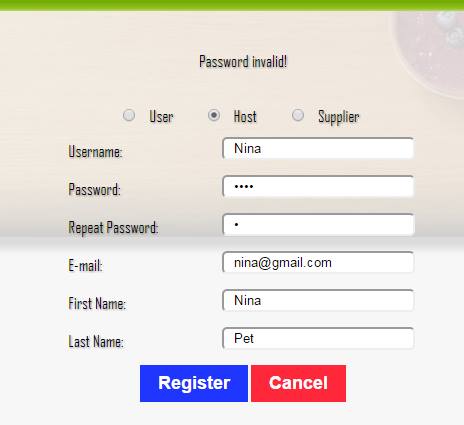
* Odabir vrste korisnika ( „*User*“ , „*Host*“ i „*Supplier*“)
* Korisničko ime
* Lozinka
* Provjera lozinke
* E-mail adresa
* Ime i prezime

**Očekivani izlaz:**

* Preusmjeravanje na početnu stranicu ukoliko je korisnik unio ispravne podatke
* Poruka o pogrešnom unosu podataka



Slika 36 - Test 4: Unos podataka za registraciju.



Slika 37 - Test 4 - Neispravan unos provjere lozinke

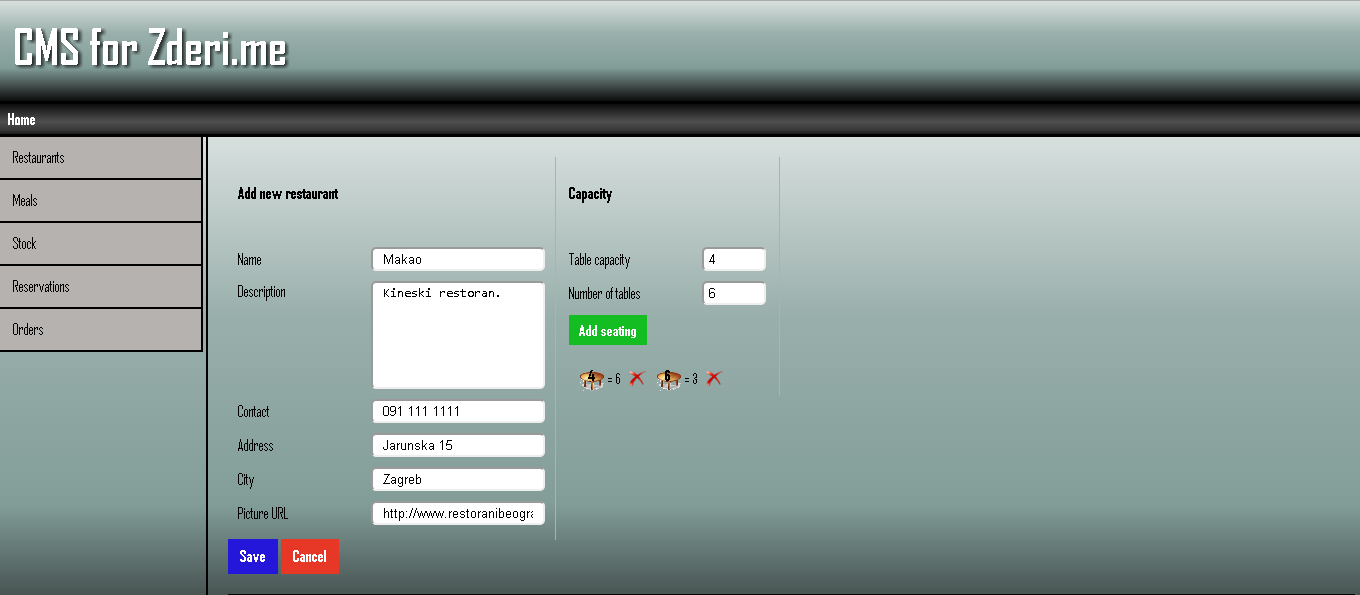
**Izlaz**:

Ukoliko su podaci ispravno uneseni korisnik je preusmjeren na početnu stranicu.

**TEST 5: Oglašavanje ponude**

**Ulaz:**

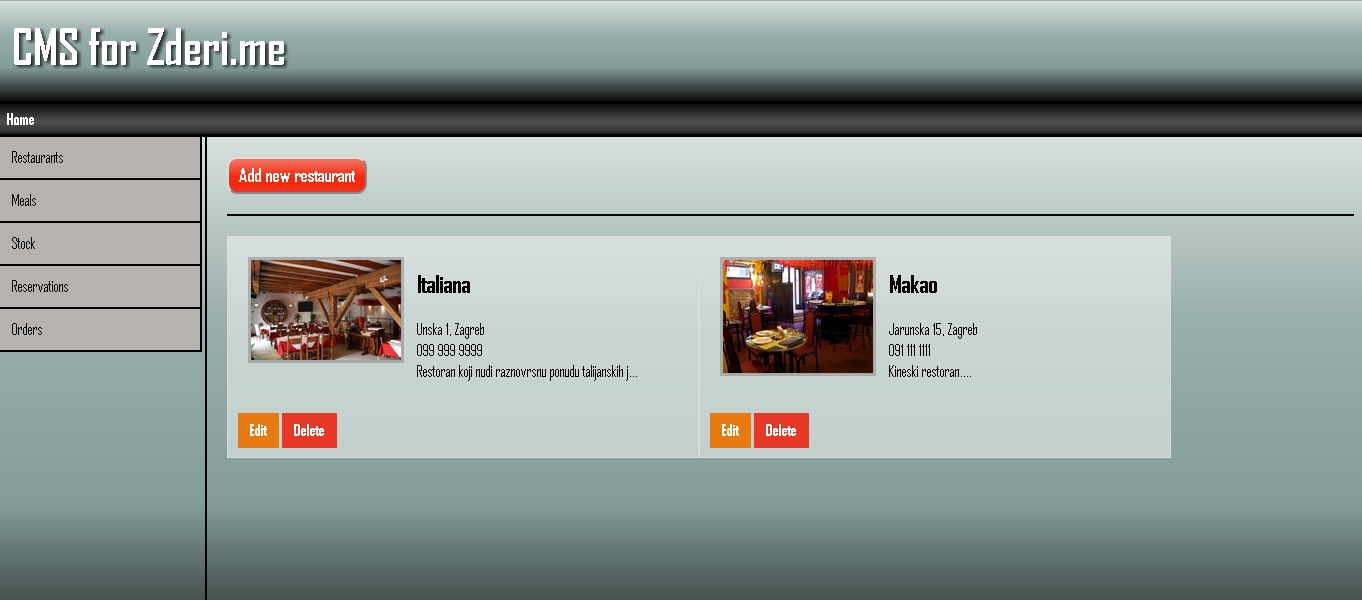
* Ime restorana
* Opis restorana
* Kontakt
* Adresa restorana
* Grad
* URL slike
* Broj stolova zajedno s njihovim kapacitetom



Slika 38 Test 5 - Unos podataka o restoranu

**Očekivani izlaz:**

* Restoran se sa svojom slikom i podacima pojavio na ekranu te na početnoj stranici.



Slika 39 Test 5 - Profil restorana na ekranu

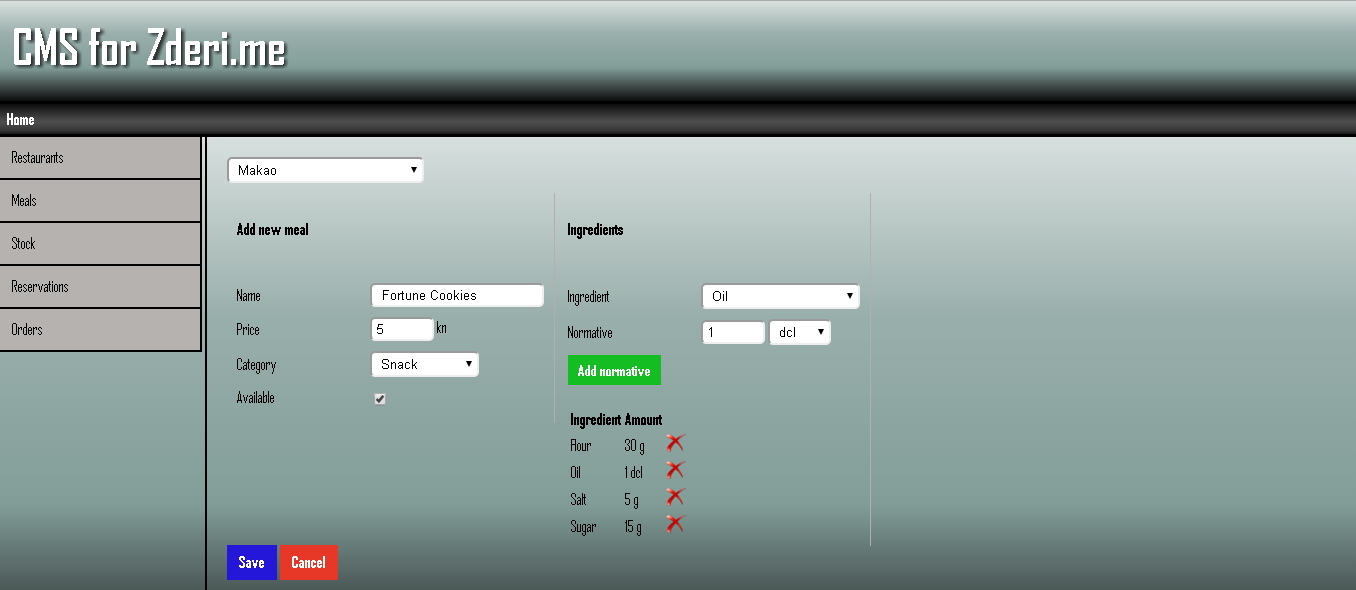
**Izlaz**:

Izlaz odgovara očekivanom izlazu. Pojavio se profil restorana na ekranu te se nalazi u listi svih restorana koji su oglašeni na početnoj stranici.

**TEST 6: Stvaranje jelovnika**

**Ulaz:**

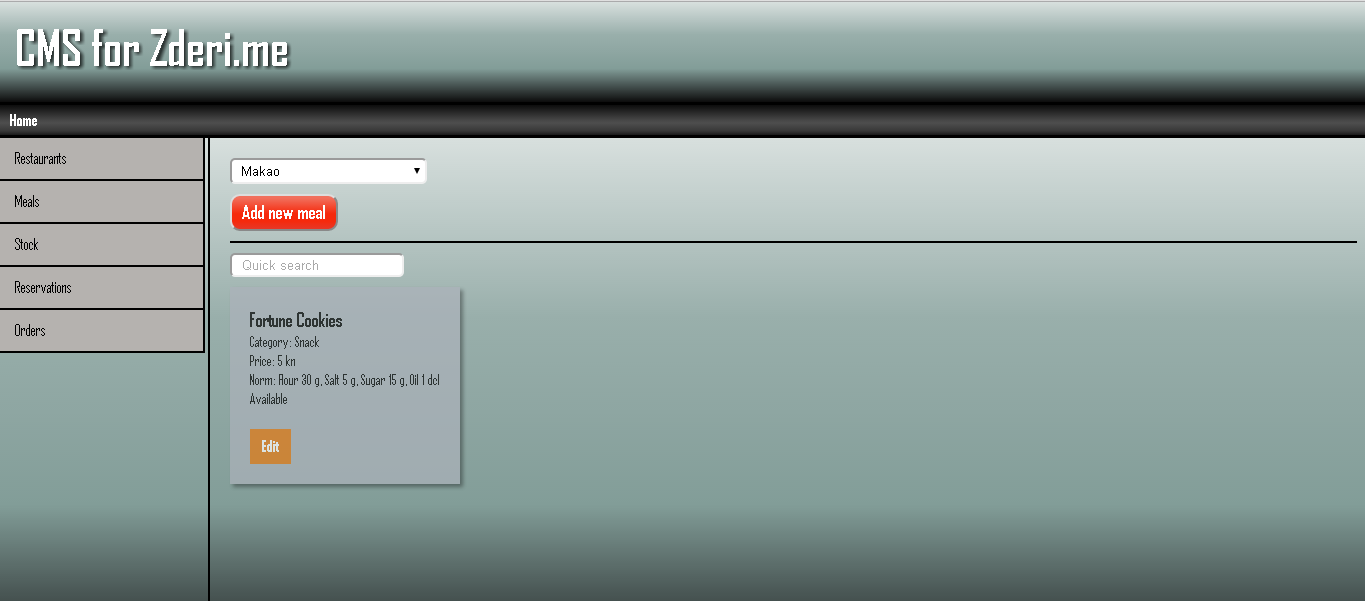
* Ime jelovnika
* Cijena
* Kategorija
* Dostupnost
* Sastojci
* Količina sastojaka



Slika 40 Test 6 -Unos podataka o jelovniku

**Očekivani izlaz:**

* Na ekranu se pojavi jelovnik sa sastojcima.



Slika 41 Test 6 - Prikaz jelovnika na ekranu

**Izlaz**:

* Izlaz odgovara očekivanom izlazu. Na ekranu se pojavi jelo sa svojim sastojcima i cijenom.

## Upute za instalaciju

7.5.1 Lokalna instalacija

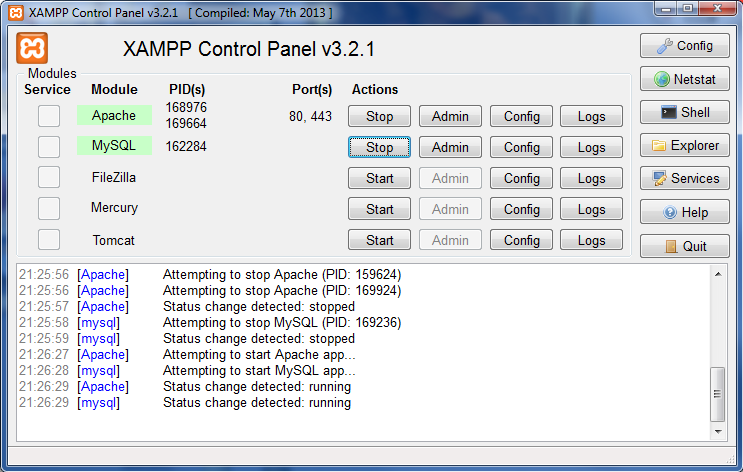
Za lokalno pokretanje aplikacije potreban je lokalni server. Preporučamo XAMPP ili WAMP. Zbog jednostavnosti priručnika, objasnit ćemo podešavanje XAMPP-a budući smo ga koristili prilikom razvijanja (više informacija o WAMP-u možete pronaći [ovdje](http://www.wampserver.com/en/)). Detaljniji opis o samoj instalaciji XAMPP-a možete pronaći u Dodatku B.

Nakon instalacije, XAMPP-a, potrebno je kreirati databazu i korisnika koji će imati ovlasti za upravljanje danom databazom.

Otvorite XAMPP Control Panel, te pokrenite Apache i MySQL module.

Ukoliko Vam se Apache modul ne želi pokrenuti, provjerite imate li aktivnu aplikaciju Skype. Ukoliko se ne odabere drukčije, Skype rezervira portove 80 i 443 za svoje procese, a isti su potrebni za pokretanje Apache servera na lokalnom računalu.

*Ove postavke promijenite ulaskom u Skype te* ***Tools -> Options -> Advanced -> Connection****, te odznačite* ***Use port 80 and 443 for additional incoming connections****, te ponovno pokrenite Skype i Apache modul.*

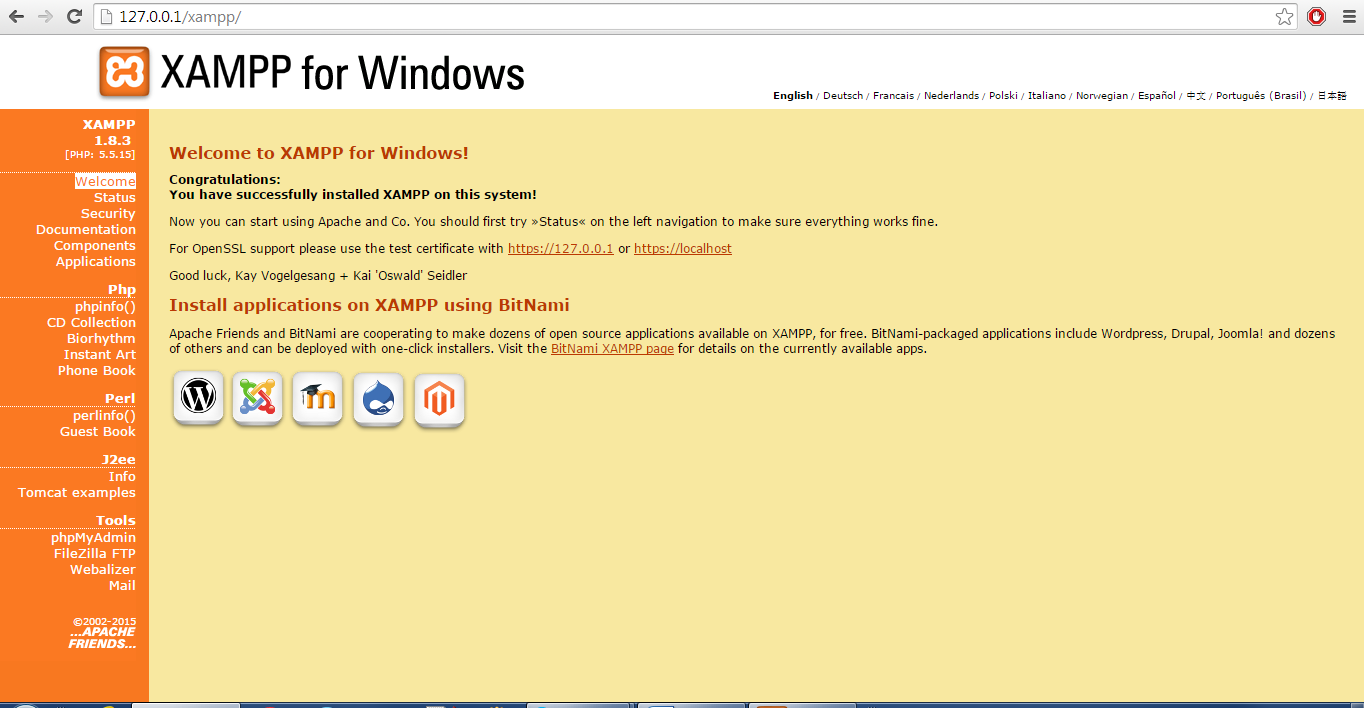


**Slika 7.5.1.1** *XAMPP Control Panel*

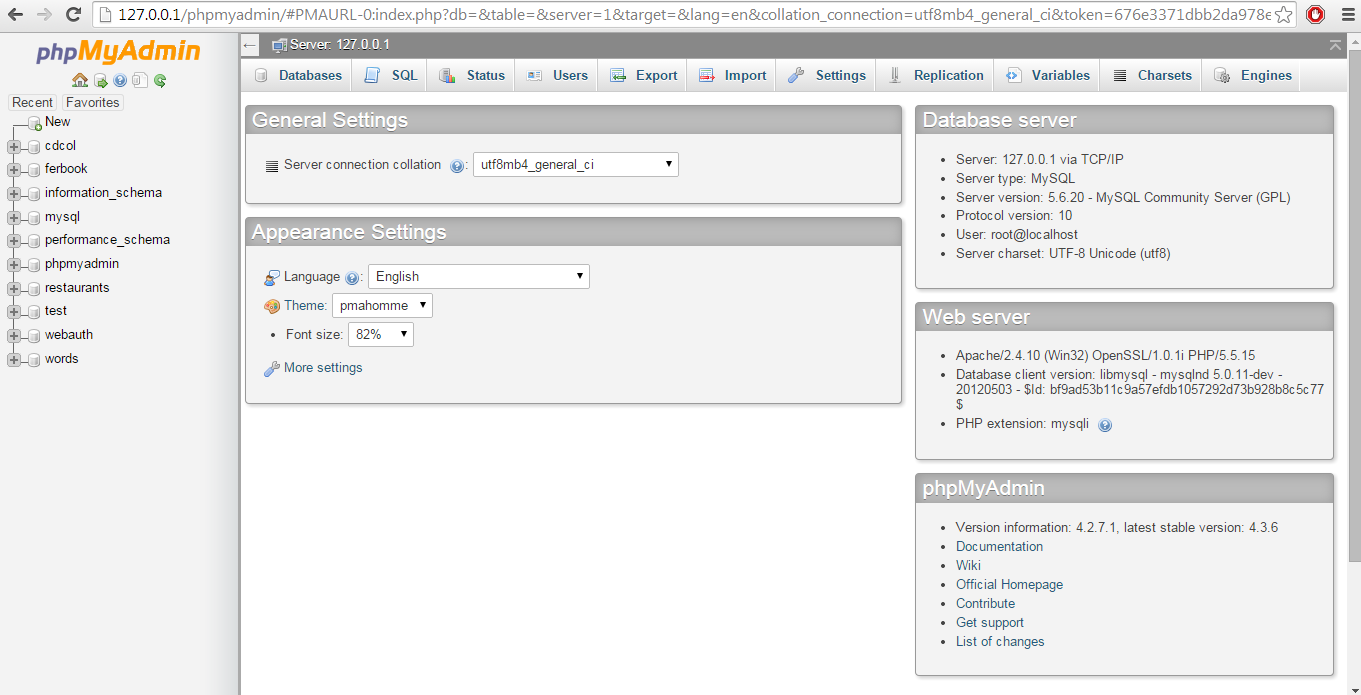
Kada imate pokrenut Apache modul, možete pristupiti XAMPP postavkama jednostavno otvarajući bilo koji Internet preglednik i odlazeći na jednu od dvije sljedeće adrese:

<http://localhost>

<http://127.0.0.1>



**Slika 7.5.1.2** *XAMPP Control Panel*

Klikom na *phpMyAdmin* dolazite do dijela u kojem možete upravljati Vašim databazama.

Odaberite tab *Databases*, te kroz sučelje kreirajte databazu *Restaurants* s *utf-8\_general\_ci* collation-om.

**Slika 7.5.1.3** *phpMyAdmin*

*Možete nazvati databazu i drukčije, ali će to zahtijevati par dodatnih promjena koje će biti objašnjene kasnije.*

Sljedeće, pod tabom *Users* kreirajte korisnika *root* s lozinkom *toor* i odaberite *Host -> local*. Također, odaberite sve privilegije.

*Ukoliko se odlučite za drugačije korisničko ime i lozinku, zahtijevat će par dodatnih promjena.*

Dalje, koristeći tab *Import,* pronađite datoteku *restaurants.sql* unutar dobivenog source koda i uvezite ju u databazu.

*Restaurants.sql* sadrži cjelokupni izvorni SQL kod potreban za kreiranje Vaše databaze. Databaza je trenutno prazna, izuzev korisnika [*admin@gmail.com*](mailto:admin@gmail.com) s lozinkom *admin*. Preporučamo Vam da prvim prijavljivanjem u sustav promijenite lozinku za administratorski račun.

Sljedeći korak je prekopirati cjelokupni source kod u */putanja/do/xampp/htdocs/ime\_datoteke*.

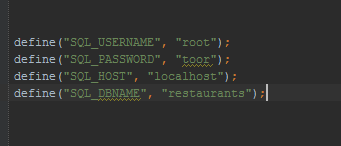
Ekvivalent *htdocs* datoteci u WAMP-u je *www* datoteka.

Ukoliko ste pratili sve korake i koristili postavke navedene kroz njih, sada možete pristupiti svojoj aplikaciji kroz bilo koji internetski preglednik preko URL-a:

<http://localhost/ime_datoteke>

<http://127.0.0.1/ime_datoteke>

Naravno, umjesto vrijednosti *ime\_datoteke* će biti naziv Vaše aplikacije.

Ukoliko ste odabrali svoje vrijednosti za ime databaze i pristupnih podataka, iste promijenite u source kodu u datoteci */constants.php* .

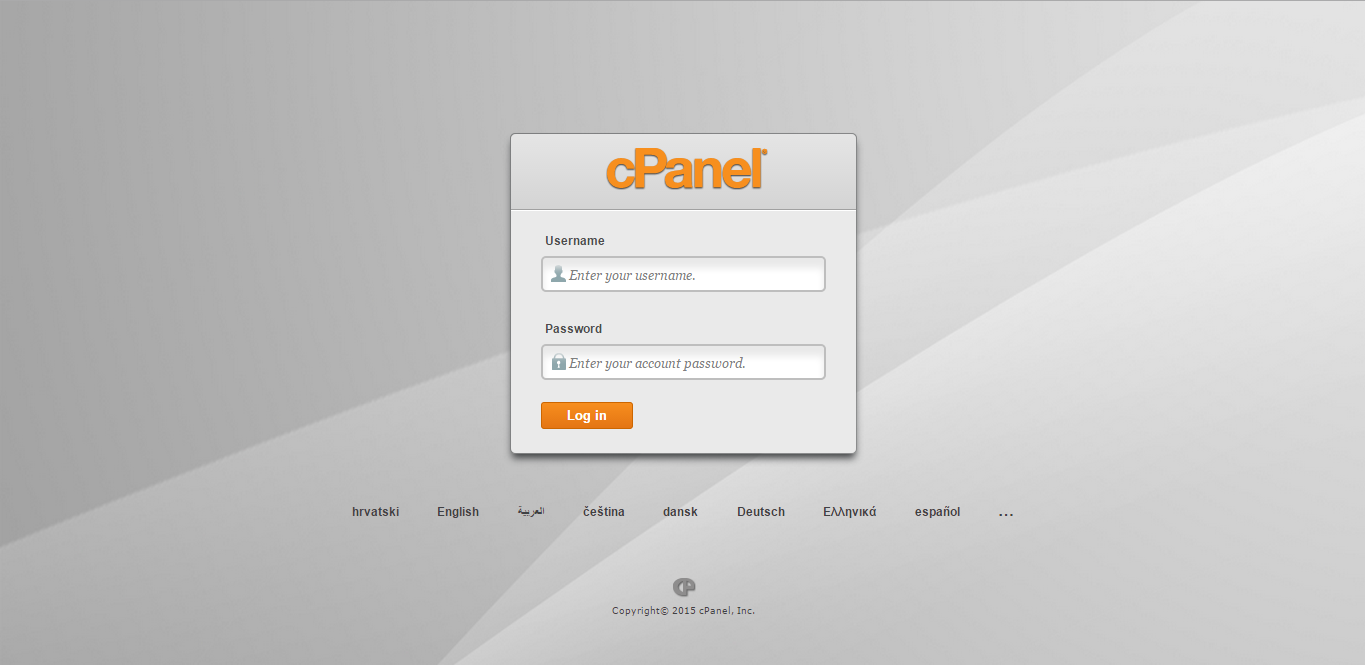
**Slika 7.5.1.4** *constants.php*

7.5.2 Serverska instalacija

Za publishanje stranice na Internet, potreban vam je vlastiti server.

Ovo objašnjenje se nastavlja pretpostavljajući da imate server i pristupne podatke za CPanel i FTP.

Prvo trebamo bazu kreirati. Postupak je identičan lokalnom postavljanju koji je opisan u poglavlju 7.5.1. CPanelu pristupate preko URL-a:

 <http://vas-server/cpanel>

**Slika 7.5.2.1** *CPanel prijava*

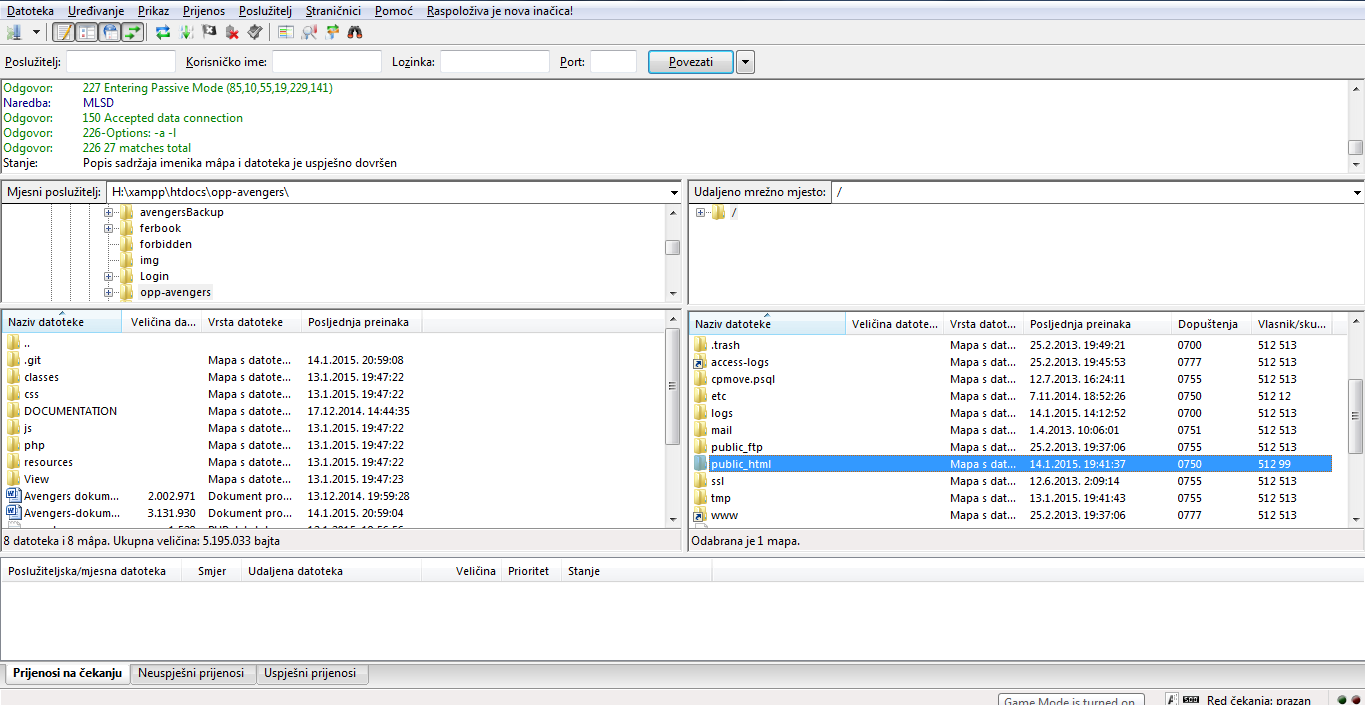


**Slika 7.5.2.2** *CPanel*

Koristeći *MySQL databases* kreirajte databazu i korisnika sa svim privilegijama, te dodijelite novo kreiranu databazu novo kreiranom korisniku.

Iste podatke unesite u */constants.php*, s jednom razlikom naspram lokalnih postavki: pod host ostavite prazno.

**Slika 7.5.2.3** *constants.php*

Koristeći bilo koji FTP klijent (preporučujemo *FilleZilla Client*) povežite se na server koristeći pristupne podatke. Kopirajte cjelokupni source kod u datoteku /*public\_html*.

**Slika 7.5.2.4** *FilleZilla Client*

Nakon prebacivanja datoteka na server, stranica će vam biti dostupna na:

<http://vas_server>

## Korisničke upute

Korisničke upute podijeljene su prema tipovima korisnika koji na raspolaganju imaju određene funkcionalnosti prilikom korištenja sustava. Tako korisnici mogu biti:

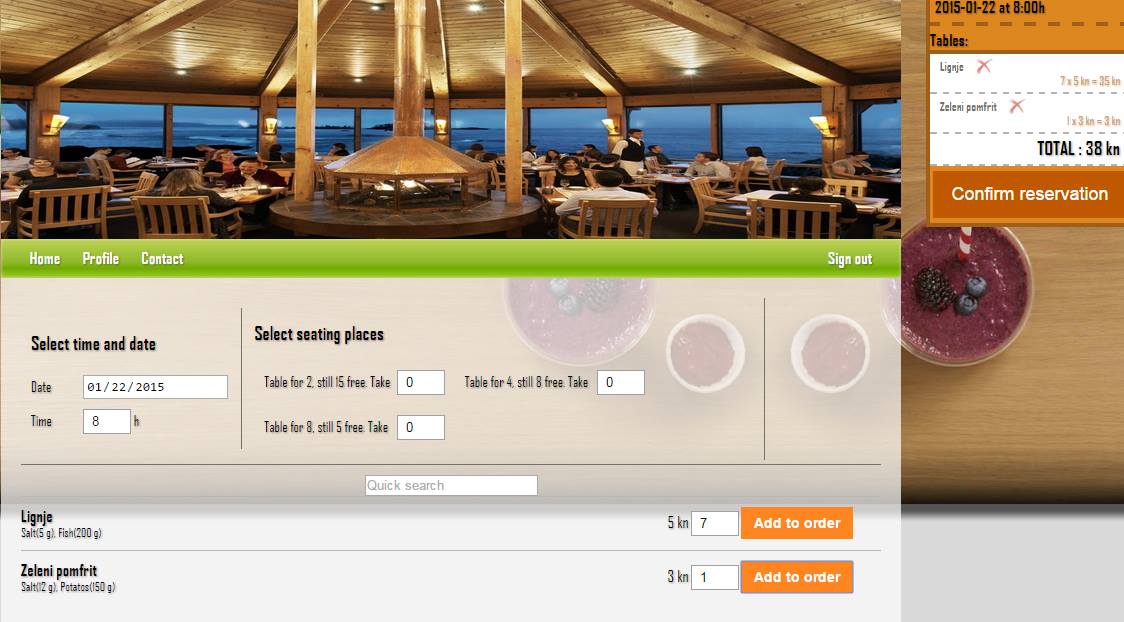
* Klijent
* Ugostitelj
* Dobavljač
* Administrator



Slika 42 - Početna stranica

### Klijent

Kada klijent posjeti početnu stranicu, ima na uvid popis restorana, adrese restorana opis restorana, te slike restorana. Navigacijski gumbi su: „*Check restaurant menu*“, „*Rank this restaurant on your site*“, „*Sign out*“, „*Home*“, „*Profile*“ i „*Contact*“. Klikom na gumb „*Check restaurant menu*“ prikazuje se ponuda svih jelovnika odabranog restorana, a klikom na „*Reserve here*“ korisnik rezervira dostupna jela i stol. Da bi korisnik rezervirao jelovnik mora odabrati jelovnik, upisati količinu i pritisnuti na gumb „*Add to order*“. Zatim korisnik treba odabrati vrijeme i datum rezervacije pritiskom na „*date*“ i „*time*“, te odabrati mjesto sjedenja upisivanjem količine stolova.



Slika 43 - Rezervacija jelovnika i stola

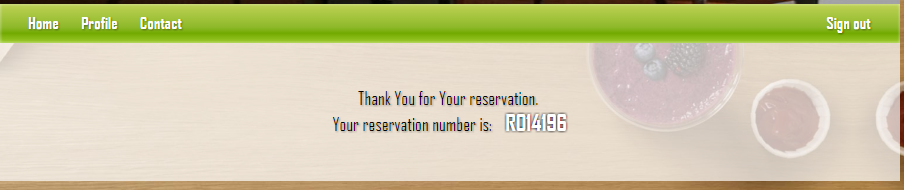
Da bi korisnik obrisao jelo iz rezervacije, potrebno je da pritisne „X“ koji se nalazi pokraj naziva jela. Nakon toga, jelovnik ili opcija bit će uklonjeni iz košarice.

Klijent može i povećati količinu jela tako što će pritisnuti „–„ ili „+“ ispod opcija jela.

Klikom na „*Confirm reservation*“ korisnik će biti proslijeđen na stranicu koja mu prikazuje broj njegove rezervacije.



Slika 44 - Košarica



Slika 7-18: Broj rezervacije

# Zaključak

Projekt je uspješno završen u usporedbi s početnim planom i vizijom. Svi postavljeni ciljevi su uspješno ostvareni i testirani.

Prilikom ostvarivanja projekta, najveća prepreka je bila prvotno susretanje članova tima s web tehnologijama i primjenom istih. Zahvaljujući iskustvu pojedinih članova, preostali članovi su napredovali i stekli nova znanja koja će im jednog dana koristiti. Primjerice, najveći problem je bio rad s AngularJS-om koji je Javascript framework i po tome se razlikuje od većine drugih framework-a.

Za brže i kvalitetnije ostvarenje projekta je, osim dobre organizacije posla, potrebno puno iskustva u radu s web tehnologijama da bi se usvojili standardi koji uvelike olakšavaju izradu i manjih i velikih projekata.

Na trenutnoj implementaciji sustava prvi korak u daljnjem razvoju bi bio implementacija višejezičnog i viševalutnog sustava, tako da projekt bude dostupan na većem rasponu. Daljnji razvoj sustava bi također implementirao Google Analytics sustav i okrenuo se nabavljanju što većeg broja klijenata i korisnika. Također, implementirati i mobilnu verziju projekta koja bi omogućila pristup još većem broju klijenata.   
Optimizacija cijelog sustava bi se mogla dodatno izvršiti prelaskom na NOSQL tip databaze (npr., MongoDB) i promjenom backend skriptnog jezika u NodeJS, s čime bi cijeli projekt prešao na Web Javascript projekt.

# Reference

*Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.*

1. Oblikovanje programske potpore, FER ZEMRIS, [http://www.fer.hr/predmet/opphttp://www.fer.hr/predmet/opp](http://www.fer.hr/predmet/opp)
2. Oblikovanje programske potpore, FER ZEMRIS, [http://www.zemris.fer.hr/predmeti/opphttp://www.zemris.fer.hr/predmeti/opp](http://www.zemris.fer.hr/predmeti/opp)
3. I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
4. T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
5. Tigris.org Open Source Software Engineering Tools, [http://readyset.tigris.org/http://readyset.tigris.org/](http://readyset.tigris.org/)
6. Software engineering ,Rutgers University, [http://www.ece.rutgers.edu/~marsic/Teaching/SEhttp://www.ece.rutgers.edu/~marsic/Teaching/SE](http://www.ece.rutgers.edu/~marsic/Teaching/SE)
7. I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, [http://www.ece.rutgers.edu/~marsic/books/SEhttp://www.ece.rutgers.edu/~marsic/books/SE](http://www.ece.rutgers.edu/~marsic/books/SE)
8. Concepts: Requirements, [http://www.upedu.org/upedu/process/gcncpt/co\_req.htmhttp://www.upedu.org/upedu/process/gcncpt/co\_req.htm](http://www.upedu.org/upedu/process/gcncpt/co_req.htm)
9. UML 2 Class Diagram Guidelines, [http://www.agilemodeling.com/style/classDiagram.htmhttp://www.agilemodeling.com/style/classDiagram.htm](http://www.agilemodeling.com/style/classDiagram.htm)
10. Domain Class Diagram Modeling Standards and Guidelines, [http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdfhttp://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf](http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf)
11. ArgoUML, [http://argouml.tigris.org/http://argouml.tigris.org/](http://argouml.tigris.org/)

# Dodatak A: Dnevnik sastajanja

**Sastanak br.1 : Srijeda 15.10.2014. 18.00-19.00h**

**Prisutni:**

Vilim Stubičan, Oleg Jakovljev, Filip Popić, Domagoj Korman, Nejra Muslić, Marko Novina, Lana Murar

**Sadržaj sastanka:**

Upoznavanje članova grupe, diskusija o zadatku koji trebamo napraviti. Svaki član je izrazio želju za određenim dijelom koji bi htio obraditi.

**Sastanak br.2 : Ponedjeljak 20.10.2014. 17.00-18.00h**

**Prisutni:**

Vilim Stubičan, Oleg Jakovljev, Filip Popić, Domagoj Korman, Nejra Muslić, Marko Novina, Lana Murar

**Sadržaj sastanka:**

Detaljno proučavanje projektnog zadatka te predloška dokumentacije nakon kojeg je napravljena inicijalna podjela zadataka:

*Vilim Stubičan* – razrada arhitekture sustava, osmišljanje i kreiranje baze podataka, opis razreda i dijagram razreda, dijagram objekata, implementacija programskog rješenja, korištene tehnologije i alati

*Oleg Jakovljev* – funkcionalni zahtjevi, implementacija programskog rješenja, upute za korištenje

*Domagoj Korman –* implementacija programskog rješenja, testiranje programskog rješenja, isječci programskog koda za opise temeljnih funkcionalnosti sustava

*Lana Murar –* dnevnik sastajanja, funkcionalni zahtjevi, testiranje programskog rješenja, upute za korištenje

*Nejra Muslić –* ostali zahtjevi, pojmovnik, testiranje programskog rješenja

*Marko Novina –* arhitektura sustava, pojmovnik

*Filip Popić –* ostali zahtjevi, implementacija programskog rješenja, testiranje programskog rješenja

**Sastanak br.3 : Srijeda 29.10.2014. 17.00-19.30h**

**Prisutni:**

Vilim Stubičan, Oleg Jakovljev, Filip Popić, Domagoj Korman, Nejra Muslić, Marko Novina, Lana Murar

**Sadržaj sastanka:**

Definiranje svih funkcionalnih zahtjeva.

**Sastanak br.4: Petak 7.11.2014. 12.00h-14.30h**

**Prisutni:**

Vilim Stubičan, Domagoj Korman, Marko Novina

**Sadržaj sastanka:**

Integracija arhitekture i dizajna sustava u dokument.

**Sastanak br.5: Petak 14.11.2014. 18.00-19.30h**

**Prisutni:**

Vilim Stubičan, Oleg Jakovljev, Filip Popić, Domagoj Korman, Nejra Muslić, Marko Novina, Lana Murar

**Sadržaj sastanka:**

Timski pregled dokumentacije. Ispravljanje sitnih pogrešaka te završno uređenje dokumentacije kako bi bila spremna za predaju prve verzije.

# Dodatak B: Instalacija XAMPP-a

Instalaciju XAMPP-a možete preuzeti s [Apache Friends](https://www.apachefriends.org/index.html).

Nakon završenog preuzimanja, pokrenite instalaciju.

Preporučamo da XAMPP instalirate u priručni Lokalni disk (ne u sistemski C: ). Sistemski disk traži ovlasti za pokretanje datoteka, pa je puno jednostavnije pokretati s drugog lokalnog diska koji ne ograničava postavke toliko puno.

Za podešavanje lokalnog SMTP-a, napravite sljedeće promjene.

*xampp/php/php.ini*

Pronađite [mail function] i postavite:

SMTP=smtp.gmail.com

smtp\_port=587

sendmail\_from = vas\_gmail@gmail.com

sendmail\_path = "\"D:\xampp\sendmail\sendmail.exe\" -t"

*xampp/sendmail/sendmail.ini*

Postavite sljedeće postavke:

[sendmail]

smtp\_server=smtp.gmail.com

smtp\_port=587

error\_logfile=error.log

debug\_logfile=debug.log

auth\_username=vas\_gmail@gmail.com

auth\_password=vasa\_gmail\_lozinka

force\_sender=vas\_gmail@gmail.com

Nakon toga samo ponovno pokrenite XAMPP i postavili ste svoj lokalni server.