Funkcionalni zahtjevi ( ujedno i zadaci za napravit) :

* Login
* Registracija
* Pregledavanje restorana i menu-a
* Postavljanje rezervacije i prikaz rezervacija
* Unos/izmjene podataka (za vlasnike-> unos novih jedinica, unos novih menu-a, promjena podataka na istima; za sve korisnike-> profile update)
* Skladište restorana s automatskom nabavkom namirnica
* Ocjenjivanje restorana

Dodatne funkcionalnosti (treba raspraviti s asistentom):

* Admin panel (i općenita administracija)\*
  + Kontrola nad namirnicama, dobavljačima (dodavanje, brisanje, updateanje)
  + Kontrola nad korisnicima
  + Statistike i ocjene

**[***ime člana koji obavlja zadatak***] - Login**Treba kreirati login formu i validaju podataka koje korisnici daju. Podatke treba provjeravati i na klijentskoj strani i na serverskoj, buduci da je korisnik taj koji unosi podatke (jedan od nacina je napraviti button up event listener i dozvoliti da unosi samo znakove koje mi dopustamo [a-zA-Z0-9\_-] ili drugi nacin, prije submitanja forme iskoristiti jQuery *match()* funkciju koja provjerava validnost inputa koristeci regularne izraze [svakako naucite, jako korisna stvar za sve validacije]). Za validaciju podataka raditi u dogovoru s clanom koji radi registraciju (iste funkcije se mogu iskoristiti).  
Na serverskoj strani takoder napraviti provjeru za slucaj da je korisnik pronasao nacin da zaobide JS validaciju (a ima nacina, bez brige :P ). Ovako raditi cisto iz razloga sto za unos krivog formatiranog korisnickog imena ne moramo korisnika tjerati da ceka reload stranice, u realnom vremenu dobije informaciju da je pogrijesio pri unosu (nece dobiti informaciju da je krivi login, nego samo filtriramo losa korisnicka imena tipa m@rko -> radi „@“ znaka i slicni primjeri). Nakon uspjesnog logina postaviti $\_SESSION['active\_user'] varijablu na vrijednost username-a korisnika i redirectati na home page.

**[***ime člana koji obavlja zadatak***] -Registracija**Treba kreirati registracijsku formu koja uzima podatke o korisniku. Obavezni podaci su username , mail i password, dok se mogu opcionalno mogu unijeti i drugi podaci (odluciti sami, ali prijedlozi su ime, prezime, kontakt, kreditna kartica, adresa itd.). Dostupnost username-a i mail adrese se preporuca da se provjere AJAX-om na ranije kreirani backend API(proci cemo sto su API-i, tko ne zna).   
Validaciju podataka raditi i na klijentskoj i na serverskoj strani (isti razlog kao i kod logina). Takoder, raditi u suradnji s clanom koji radi login.  
Nakon validacije podataka, spremiti u bazu podatke o korisniku, postaviti $\_SESSION['active\_user'] na username novog korisnika, poslati mail s obavijesti da je registracija obavljena i porukom dobrodoslice, te redirectati na message page na kojem ce mu se takoder prikazati poruka dobrodoslice.

**[***ime člana koji obavlja zadatak***] -Pregledavanje restorana i menu-a**Ovo ce ujedno biti i home page. Izlistati restorane, a klikom na tab restorana otvoriti jelovnik s opcijom rezervacije na dnu i vrhu jelovnika koji redirecta na formu za postavljanje rezervacije. Ukolike korisnik nije logiran, umjesto gumba za rezervaciju, prikazati link do login-a i registracije.

**[***ime člana koji obavlja zadatak***] -Postavljanje rezervacije i prikaz rezervacija**Provjeriti ima li korisnik vec rezervaciju u tom restoranu. Ukoliko ima, ispisati poruku s mogucnoscu otkazivanja narudzbe. Ukoliko nema, otvoriti formu i omoguciti odabir vremena, stolova i menu-a na nacin da se odabirom vremena AJAX-om ucitaju dostupni stolovi 5 h unaprijed (ukoliko se odluci da svaka rezervacija vrijedi samo 5h ! ). Ne dopustiti zavrsetak rezervacije bez odabranog vremena i broja sjedecih mjesta. Nakon spremanja rezervacije, prikazati istu na stranici te poslati mail s detaljima o narudzbi.  
Isti clan bi trebao napraviti formu za unos narudzbe preko barkoda za korisnike koji odluce da ce naruciti sto ce jesti tek kada dodu u restoran (u tom slucaju, odabiru prazan menu prilikom rezervacije -> ispisati vidljivo poruku za tu mogucnost !).

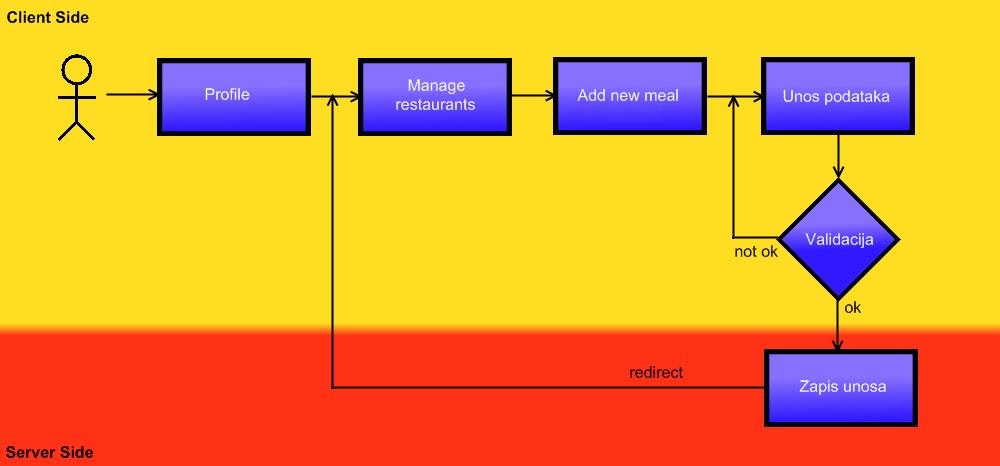
**[***ime člana koji obavlja zadatak***] -Unos/izmjene podataka**  
Omoguciti korisnicima da unose nova jela, kategorije, odabiru dobavljace te mijenjaju vlastite podatke. Svaki korisnik ima mogucnost „otvaranja“ restorana.  
Na vlastitu ruku odluciti hoce li korisnik moci kreirati jela i bez restorana ili ce moci kreirati jela samo za restoran. Prva opcija je malo slozenija, ali omogucava lakse dodavanje jela u menu preko posebne forme za tu akciju. Takoder omogucava korisniku da isto jelo postavi na menu-e od vise restorana bez ponovnog unosa.  
Forma za jelo treba omoguciti korisniku neogranicen broj namirnica prilikom slaganja recepture (najlakse za procitati podatke krecuci se po DOM-u forme) kao i provjeriti ime jela (ime jela se moze zauzeti samo jednom po korisniku/restoranu, ovisno o nacinu implementiranja dodavanja jela).  
Forma za kategoriju mora paziti da korisnik ne dodaje vec unaprijed definiranu kategoriju. Od pocetka ce biti dostupne neke default kategorije (s posebnom identifikacijom; kategorije poput Meso, Roštilj, Riba, Prilozi,itd.) tako da korisnik moze rasporediti svoja jela u smislene kategorije.  
Forma za dobavljace omogucuje korisniku da za odredene namirnice odabere pojedinacne dobavljace. U trenutku kada odabire dobavljaca, korisnik mora imati uvid u cijenu namirnice te ujedno vidjeti i popis ostalih namirnica kod dobavljaca. Najbolje je automatski postaviti dobavljaca za sve namirnice koje dobavljac ima, ukoliko korisnik za istu namirnicu nije vec odredio dobavljaca.  
Forma za mijenjanje podataka korisnika omogucava korisniku da promijeni sve opcionalne podatke koje je/nije unijeo prilikom registracije.

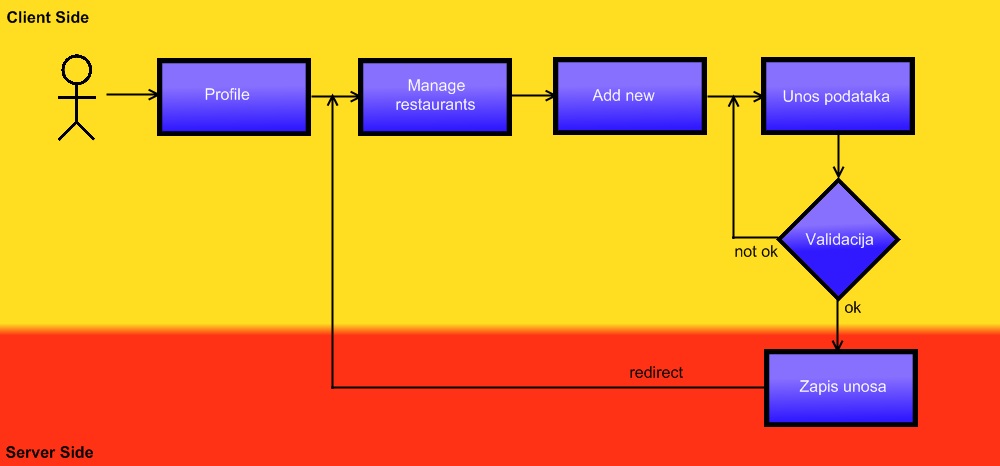
**[***ime člana koji obavlja zadatak***] -Skladište restorana s automatskom nabavkom**Clan koji ce raditi ovaj zadatak treba raditi u suradnji s clanom koji postavlja rezervacije koje imaju narudzbu postavljenu. Najbolji pristup jest takav da nakon zavrsetka postavljanja rezervacije, clan koji je taj dio programirao kreira instancu klase koja ce osvjeziti skladiste po narudzbi.  
Kada se dobije id narudzbe, proci po svim stavkama narudzbe i sa skladista oduzeti kolicinu namirnica za svako jelo. Nakon toga, proci po zalihama restorana, te za svaku namirnicu ispod 0 naruciti odredenu kolicinu namirnice od odredenog dobavljaca.

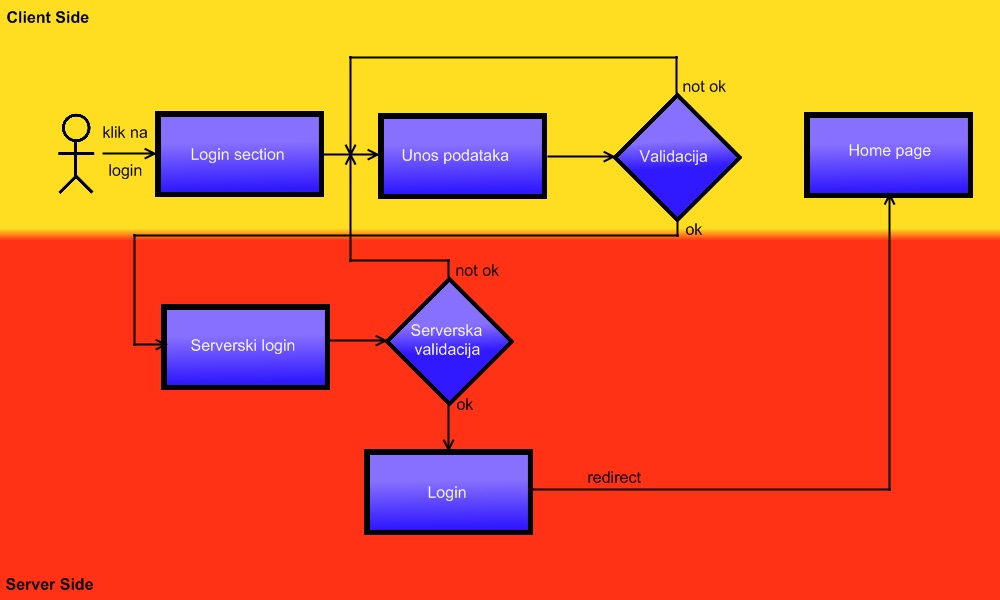
**[***ime člana koji obavlja zadatak***] -Ocjenjivanje restorana**Kreirati formu za ocjenjivanje restorana koja je vidljiva samo registriranim korisnicima. Forma se smije pojaviti jedino u trenutku ako je neki korisnik logiran, te omoguciti jednu ocjenu korisnika prema restoranu. Dozvoliti korisniku da mijenja ocjenu restorana te uzeti komentar pri svakoj izmjeni.  
Takoder napraviti prikaz ocjena za restoran koji ce se implementirati u glavni prikaz restorana (aritmetička sredina svih ocjena za restoran).

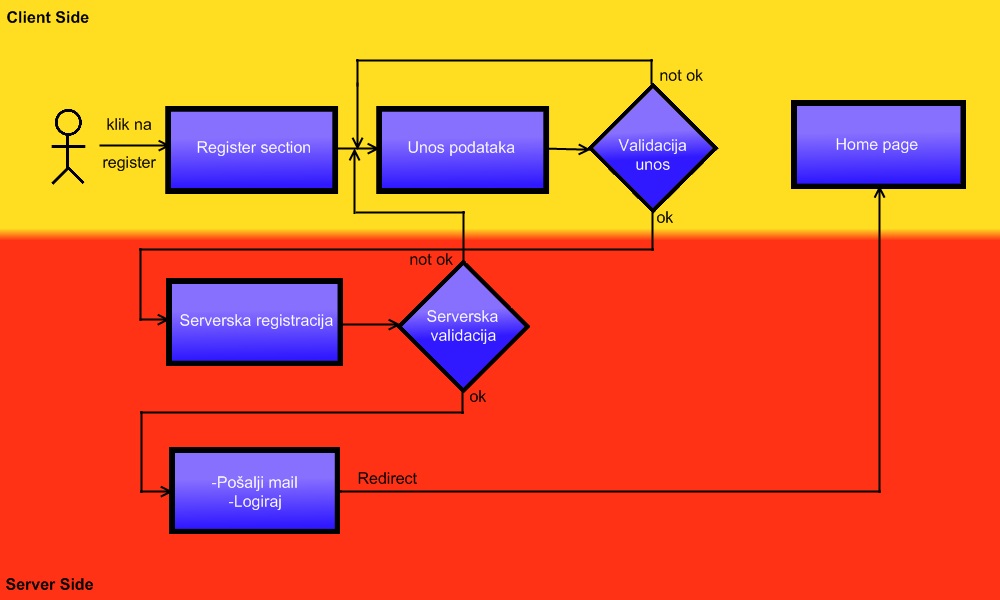
**[***ime člana koji obavlja zadatak***] -Administracija**  
Administracija ce biti dostupna barem na nivou administratora koji ce moci dodavati namirnice i popunjavati rezervacije onih korisnika koji narucuju u restoranu, te mijenjati status rezervacija (kada korisnik plati, ukoliko treba otkazati rezervaciju).  
-> svakog korisnika cemo smatrati administratorom za njegov restoran pa ce svaki vlasnik restorana moci mijenjati rezervacije, no samo globalni administrator moze dodavati nove sastojke (buduci da su ovo postavke koje svi korisnici vide)

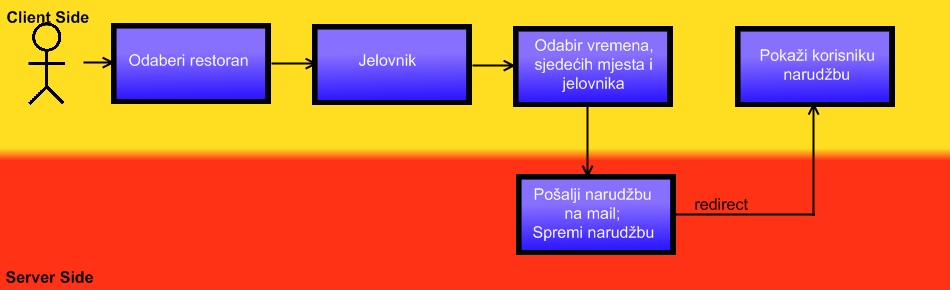
Tijek korištenja nekih od funkcija stranica

  
**Flowchart 1.** *Dodavanje novog jela*

  
**Flowchart 2.** *Dodavanje nove jedinice restorana*

  
**Flowchart 3.** *Login*

  
**Flowchart 4.** *Registracija*

  
**Flowchart 5.** *Obavljanje rezervacije*