

# Web programiranje – Specifikacija projektnog zadatka

Računarstvo i automatika 2021/2022

## Projektni zadatak

Upotrebom tehnologija i tehnika primenjivanih na predavanjima i vežbama projektovati veb aplikaciju za naručivanje hrane prema sledećoj specifikaciji.

## Specifikacija zahteva

Potrebno je realizovati veb aplikaciju za sistem koji podržava naručivanje hrane (kao donesi.com ili wolt aplikacije). Aplikaciju koriste 4 grupe (uloge) korisnika: Kupac, Menadžer, Administrator i Dostavljač. Aplikacija rukuje sledećim entitetima:

### Korisnik

- Korisničko ime (jedinstveno)
- Lozinka
- Ime
- Prezime
- Pol
- Datum rođenja
- Uloga (Administrator, Menadžer, Dostavljač, Kupac)
- Sve porudžbine (ako je korisnik Kupac)
- Korpa (ako je korisnik Kupac)
- Restoran (ako je korisnik Menadžer)
- Porudžbine koje treba da dostavi (ako je Dostavljač)
- Broj sakupljenih bodova (ako je korisnik Kupac)
- Tip kupca

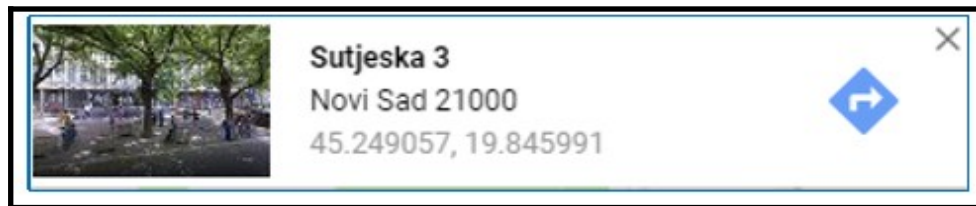
### Tip kupca

- Ime tipa (npr. Zlatni, Srebrni, Bronzani)
- Popust (procenat koji se koristi za obračunavanje cene porudžbine prilikom kupovine)
- Traženi broj bodova (potreban broj bodova kako bi korisnik postao npr. Zlatni kupac)

### Lokacija

- Geografska dužina
- Geografska širina

- Adresa u formatu: ulica i broj, mesto/grad, poštanski broj



Slika 1.

#### Restoran

- Naziv
- Tip restorana (italijanski, kineski, roštilj, i slično)
- Artikli koje restoran ima u ponudi
- Status (Radi ili Ne radi)
- Lokacija
- Logo restorana (slika)

#### Artikal

- Naziv
- Cena
- Tip (jelo, piće)
- Restoran kom pripada
- Količina (u gramima ili mililitrima)
- Opis
- Slika

#### Korpa

- Artikli koji se nalaze u korpi (pojedinačan artikal i količina)
- Korisnik čija je korpa
- Cena

#### Porudžbina

- Jedinstveni identifikator narudžbine (10 karaktera)
- Artikli koji su poručeni
- Restoran iz kog je naručeno
- Datum i vreme porudžbine
- Cena
- Kupac (ime i prezime)
- Status (Obrada, U pripremi, Čeka dostavljača, U transportu, Dostavljena, Otkazana)

#### Komentar

- Kupac porudžbine koji je ostavio komentar
- Restoran na koji se komentar odnosi

- Tekst komentara
- Ocena (na skali od 1 do 5)

Implementirati sledeće funkcionalnosti:

- Registracija - neregistrovan korisnik se registruje na aplikaciju popunjavajući polja koja su za to predviđena i nakon toga postaje Kupac
- Administratori se programski učitavaju iz tekstualnog fajla i ne mogu se naknadno dodati. **Menadžere i Dostavljače mogu kreirati samo administratori. Kupac ne može da postane Menadžer ni Dostavljač.**
- Prijavljivanje na sistem - neprijavljeni korisnik loguje se na sistem tako što unosi korisničko ime i lozinku korisnika za koji je registrovan. Nakon toga, korisnik je prijavljen i može da izvršava aktivnosti predviđene njegovom ulogom.
- Svi ulogovani korisnici mogu da vide svoje profile i da menjaju svoje lične podatke.
- Administratori imaju pregled svih korisnika registrovanih na sistemu.
- Menadžer ima pregled svog restorana, porudžbina koje su vezane za taj restoran i kupaca koji su poručili iz tog restorana.

Restoran

- **Administrator** može da kreira nov restoran. Prilikom kreiranja mora da unese informacije o nazivu, tipu, lokaciji i da izabere logo restorana. Pored toga, mora da izabere **menadžera** koji će biti zadužen za restoran. Prilikom odabira menadžera neophodno je omogućiti odabir samo menadžera koji nisu zaduženi ni za jedan restoran. Ukoliko ne postoji takav menadžer, omogućiti administratoru da registruje novog menadžera koji će automatski biti povezan sa novododatim restoranom.
- Prikaz svih restorana u okviru početne strane koja je vidljiva **svim tipovima korisnika** (uključujući i neregistrovane korisnike). Ovaj prikaz u okviru početne stranice treba da prikaže sve restorane i to tako da su pri vrhu (na početku) oni koji su otvoreni.
- Omogućiti pretragu **svih** restorana po sledećim kriterijumima:
  - Naziv restorana
  - Tip
  - Lokaciji (korisnik zadaje naziv grada ili države)
  - Prosečnoj oceni

Prilikom prikaza rezultata pretrage voditi računa o tome da se prikazu sledeći podaci:

- Naziv restorana
- Tip restorana
- Lokacija
- Logo restorana
- Prosečna ocena restorana

U okviru pretrage pružiti mogućnost sortiranja i filtriranja rezultata. Sortiranje je potrebno implementirati po rastućem ili opadajućem kriterijumu prema sledećim parametrima:

- Naziv restorana
- Lokacija
- Prosečna ocena

dok je filtriranje potrebno omogućiti po sledećim kriterijumima:

- Tip restorana
- Prikaz samo otvorenih restorana.

Izborom neke od prikazanih restorana, korisnik se prebacuje na prikaz tog konkretnog restorana.

- Prikaz jednog restorana koji treba da prikaže osnovne informacije o svakom restoranu:
  - Naziv
  - Tip
  - Status (Radi ili Ne radi)
  - Lokacija
  - Logo restorana (slika)
  - Ocena restorana (ukoliko ona postoji)
  - Komentare o restoranu (ukoliko oni postoje)
  - Lista artikala koje restoran nudi zajedno sa slikom, opisom i cenom

Prikaz adrese pored prikaza teksta lokacije u formatu definisanom unutar entiteta *Lokacija*, prikazati mapu ukoliko bude implementiran *dodatni zadatak 1*.

- **Menadžerima** je dostupna funkcionalnost dodavanje novih artikala u restoran za koji je zadužen, pri čemu je potrebno voditi računa da ne postoji artikal sa istim imenom. Prilikom kreiranja artikla neophodno je uneti naziv, cenu, tip i sliku, dok su opis i količina opcioni.
- Pored dodavanja novih artikala, omogućiti menadžerima funkcionalnost izmene podataka o postojećim artiklima.

Porudžbine

- **Kupci** imaju opciju pregleda svih svojih porudžbina u okviru svog korisničkog profila. Kod pregleda ovih porudžbina omogućiti pregled *nedostavljenih* porudžbina.
- **Dostavljači** imaju opciju pregleda svih porudžbina koje su u statusu *Čeka dostavljača* i one za koje su oni zaduženi u okviru svog korisničkog profila. Kod pregleda porudžbina za koje su oni zaduženi omogućiti pregled *nedostavljenih* porudžbina.
- **Menadžer** ima opciju pregleda svih porudžbina koje su vezane za restoran za koji je on zadužen.
- Porudžbine je moguće pretražiti, filtrirati i sortirati. Pretraga se vrši po sledećim kriterijumima:
  - Restoran iz kog je poručeno (samo za **Kupca** i **Dostavljača**)
  - Ceni porudžbine (cena u opsegu od-do)
  - Datumu porudžbine (od-do),
 dok se sortiranje vrši (rastuće i opadajuće) po:

- Imenu restorana iz kog je poručeno (samo za **Kupca** i **Dostavljača**)
- Ceni porudžbine
- Datumu porudžbine,

a filtriranje po:

- Tipu restorana (samo za **Kupca** i **Dostavljača**)
- Statusu porudžbine
- Porudžbine mogu da kreiraju samo **Kupci**. Kupac može da poruči iz bilo kog restorana koji je otvoren (status je **Radi**).

#### Koraci prilikom kreiranja porudžbine:

1. Kupac iz liste artikala koje restoran nudi dodaje željene artikle u korpu (U svakom trenutku kupac može da izbaciti artikal iz korpe. Kupac može da izabere količinu artikala koju želi da doda u korpu. Količina ne može biti manja od 1)
2. Nakon što je završio sa odabirom artikala, kupac ide na pregled korpe u kom mu se izlistaju informacije o artiklu (naziv, količina, cena, slika), kao i ukupna cena porudžbine. Kupac može da izbaciti artikal iz korpe ili da izmeni količinu artikla u korpi (količina ne može biti manja od 1). Ukupna cena korpe se ažurira sa svakom izmenom.
3. Kupac klikne na dugme poruči nakon čega se kreira porudžbina. Kreirana porudžbina je u statusu *Obrada* i dodaje se u listu porudžbina vezane za restoran iz kog je poručeno. Menadžer koji je zadužen za restoran može da promeni status porudžbine u status *U pripremi*.

Kada korisnik kreira porudžbinu dobija određeni broj bodova. Ovaj broj bodova se računa po sledećoj formuli:

$$\text{broj\_bodova} = \text{ukupna\_cena\_porudžbine} / 1000 * 133$$

- **Kupac** ima opciju da otkáže narudžbinu dok je ona u statusu *Obrada*. Ako se odustane od narudžbine gubi se broj bodova sledećoj formuli:

$$\text{broj\_izgubljenih\_bodova} = \text{ukupna\_cena\_porudžbine} / 1000 * 133 * 4$$

- **Menadžer** može da promeni status porudžbine iz *U pripremi* u *Čeka dostavljača*.
- **Dostavljač** ima opciju da iz liste porudžbina koje su u statusu *Čeka dostavljača* zatraži porudžbinu koju želi da dostavi. Ovaj zahtev se prosleđuje **Menadžeru** restorana koji može da odobri ili da odbije. Ukoliko odobri zahtev, status porudžbine se menja u *U transportu* i ta porudžbina se dodaje u listu porudžbina za koje je zadužen dostavljač koji je kreirao zahtev.
- **Dostavljač** ima opciju da porudžbine koje su u statusu *U transportu* prebaci u status *Dostavljena*.

#### Komentari:

- Nakon što porudžbina bude dostavljena (status narudžbine je **Dostavljena**), korisnik koji je kreirao porudžbinu može da ostavi komentar na restoran iz kog je poručio i isti ujedno i oceni.
- Nakon što je komentar kreiran, on se ne prikazuje na stranici restorana dok god ga **Menadžer** ne odobri (može i da ga odbije).
- **Kupci** mogu da vide **samo** komentare koje je **menadžer** odobrio.

- **Menadžeri i Administratori** mogu da vide **sve** komentare (prihvaćene i odbijene).

Korisnici:

- **Administrator** ima mogućnost prikaza svih registrovanih korisnika sistema. Može da ih pretražuje, filtrira i sortira. Pretraga se vrši po:

- Imenu
- Prezimenu
- Korisničkom imenu

Sortiranje je potrebno implementirati u rastućem i opadajućem režimu, a moguće ga je izvršiti prema sledećim parametrima:

- Imenu
- Prezimenu
- Korisničkom imenu
- Broju sakupljenih bodova.

Filtriranje se vrši prema sledećim parametrima:

- Ulozi
- Tipu korisnika.

Realizovati kombinovanu pretragu restorana po prethodno navedenim kriterijumima u pretrazi restorana.

Kombinovana ili višestruka pretraga znači da korisnik može da odabere više opcija po kojim će da vrši pretragu. Npr. korisnik može da odabere da pretražuje po lokaciji, tipu, nazivu i sl. (Slika 2.)



Slika 2.

### Dodatne napomene:

1. Brisanje svih entiteta u sistemu je **logičko**. Samo administratori imaju prava da brišu entitete. Svi entiteti koji se mogu dodati, mogu se i obrisati.
2. U zavisnosti od konkretne implementacije, **studenti mogu proizvoljno da prošire date entitete ili dodavati druge.**
3. Potrebno je da student samostalno definiše skalu bodova za Tipove korisnika (npr. da bi korisnik postao Srebrni potrebno je da sakupi 3000 bodova (i tom prilikom ima npr. popust od 3% prilikom svake kupovine), a da bi postao Zlatan potrebno je da sakupi 4000 bodova (i tom prilikom ima npr. popust od 5% prilikom svake kupovine)).

## Dodatni zadaci za više ocene

1. Prilikom odabira lokacije koristiti OpenLayers mape (<https://openlayers.org/>) ili neku alternativu mape za pretragu lokacije restorana na osnovu lokacije ili zadavanja lokacije prilikom kreiranja restorana.
2. Omogućiti administratorima prikaz svih "sumnjivih" korisnika koji često otkazuju svoje porudžbine. Ukoliko korisnik (**Kupac**) izvrši više od 5 otkazivanja porudžbina u periodu od mesec dana on se smatra sumnjivim i izdvaja se u okviru spiska takvih korisnika. Omogućiti **Administratoru** da može da blokira ovakve korisnike.
3. Administratori imaju mogućnost blokiranja registrovanih korisnika (ako nisu administrator). Blokirani korisnik ne može da se uloguje na svoj nalog i nema mogućnost da izvršava bilo koju akciju u zavisnosti od svoje uloge.

## Ocenjivanje projektnog zadatka

- Za verzionisanje projekta je potrebno koristiti git. Repozitorijum treba biti dostupan na <https://github.com> ili <https://gitlab.com>.
- Nakon formiranja tima i projekta, dostavite Vašem asistentu sledeće informacije:
  - Spisak članova tima
  - Link ka javnom repozitorijumu
  - U slučaju privatnog repozitorijuma, potražite korisničko ime asistenta i dodajte ga u contributors listu
- U git istoriji se mora jasno videti tok razvoja u koji moraju biti uključeni svi članovi tima. Ne prihvata se projekat sa malim brojem ogromnih commitova.
- Dizajn korisničkog interfejsa kao i stilovi (CSS) su prepušteni studentu i obavezni su. Dozvoljeno je korišćenje CSS biblioteka kao što su Bootstrap, Foundation i Materialize
- Aplikacija treba da trajno čuva podatke u tekstualnim datotekama, bilo u CSV ili JSON formatu
- Nije dozvoljena upotreba bilo koje baze podataka ili Java serijalizacije objekata
- Za maksimalnu ocenu 8 projekat se može implementirati uz pomoć JSP i Servlet tehnologija (MVC model 2)
- Koristi se Apache Tomcat 8
- Za ocenu 9 i 10 su sledeće opcije
  - JQuery + AJAX ili Vue.js 1.x ili 2.x za frontend
  - JAX-RS REST (Jersey) ili Spark Java backend
- Za ocenu 9 i 10 komunikacija između servera i klijenta isključivo pomoću JSON objekata prateći REST specifikaciju
- Za dozvolu upotrebe dodatnih biblioteka za navedene tehnologije se treba obratiti asistentu
- Nije dozvoljeno korišćenje tehnologija koje nisu navedene (npr. Spring, programski jezik osim Java i JavaScript itd)

Za ocenu 6 aplikacija ne treba da podrži funkcionalnost kreiranja porudžbine. Treba zadati "zabetonirane" vrednosti za karte i prilikom pokretanja aplikacija ih učitati.

Za ocenu 8 aplikacija ne treba implementirati dodatne zadatke.

Za ocenu 9 i 10 aplikacija mora da podržava sve navedene funkcionalnosti kao i dodatne zadatke.