Oblikovanje programske potpore

Ak. god. 2013./2014.

Narudžba pizze

Dokumentacija, Rev. 1

Grupa: SilkRoad

Voditelj: Luka Strižić

Datum predaje: 28. 11. 2013.

Nastavnik: Nikola Bogunović

Asistent: Alan Jović

Popis članova grupe i zaduženja

Luka Strižić - Voditelj projekta, upravljanje ljudskim i projektnim resursima, opis projektnog zadatka, arhitektura, ER model

Matej Brkašić - UC dijagrami, sekvencijski dijagrami

Belma Gutlić - Zahtjevi i ostali zahtjevi

Davor Lovrić - Dodavanje referenci, superskriptiranje ključnih riječi sa referencama

Matija Srbić - Arhitektura, lektoriranje opisa projektnog zadatka, dijagram razreda

Petar Šegina - Arhitektura, objektni dijagrami

Nika Tilić - Rječnik pojmova

Sadržaj

[1. Dnevnik promjena dokumentacije 4](#id.gjdgxs)

[2. Opis projektnog zadatka 5](#id.30j0zll)

[3. Rječnik pojmova 6](#id.1fob9te)

[4. Funkcionalni zahtjevi 7](#id.3znysh7)

[5. Ostali zahtjevi 8](#id.2et92p0)

[6. Arhitektura i dizajn sustava 9](#id.tyjcwt)

[**6.1. Svrha, opći prioriteti i skica sustava 10**](#id.3dy6vkm)

[**6.2. Dijagram razreda s opisom 11**](#id.1t3h5sf)

[**6.3. Dijagram objekata 12**](#id.4d34og8)

[**6.4. Ostali UML dijagrami 13**](#id.2s8eyo1)

[7. Implementacija i korisničko sučelje 14](#id.17dp8vu)

[**7.1. Dijagram razmještaja 14**](#id.3rdcrjn)

[**7.2. Korištene tehnologije i alati 15**](#id.26in1rg)

[**7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava 16**](#id.lnxbz9)

[**7.4. Ispitivanje programskog rješenja 17**](#id.35nkun2)

[**7.5. Upute za instalaciju 18**](#id.1ksv4uv)

[**7.6. Korisničke upute 19**](#id.44sinio)

[8. Zaključak 20](#id.2jxsxqh)

[9. Reference 21](#id.z337ya)

[Dodatak A: Dnevnik sastajanja 22](#id.3j2qqm3)

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Opis promjene/dodatka** | **Autor(i)** | **Datum** |
| 0.10 | Uređen predložak, inicijalna podjela zadataka | Strižić | 19. 10. 2013. |
| 0.11 | Opis zadatka | Strižić | 27. 10. 2013. |
| 0.20 | Opisani dionici i aktori u funkcionalnim zahtjevima | Gutlić | 29. 10. 2013. |
| 0.21 | Dodani svi opisi obrazaca uporabe | Gutlić | 03. 11. 2013. |
| 0.30 | Uređivanje rječnika pojmova | Tilić | 10. 11. 2013. |
| 0.40 | Opis arhitekture sustava | Strižić | 19. 11. 2013. |
| 0.41 | Dopuna opisa zadatka | Srbić | 23. 11. 2013. |
| 0.50 | Dijagram razreda | Srbić | 23. 11. 2013. |
| 0.51 | Izmjena dijagrama razreda | Srbić | 25. 11. 2013. |
| 0.60 | ER model | Strižić | 26. 11. 2013. |
| 0.70 | Izmjena i dopuna opisa obrazaca uporabe, dijagrami obrazaca uporabe, sekvencijski dijagrami, opisi sekvencijskih dijagrama | Brkašić | 26. 11. 2013. |
| 0.80 | Napisani ostali zahtjevi | Gutlić | 27. 11. 2013. |
| 0.81 | Popis referenci | Lovrić | 27. 11. 2013. |
| 0.82 | Ispravak ER modela, opisana arhitektura | Strižić | 27. 11. 2013. |
| 0.90 | Uređivanje rječnika pojmova | Tilić | 28. 11. 2013. |
| 0.95 | Dijagram objekata | Šegina | 28. 11. 2013. |
| **1.0** | **Fork GDocs -> SVN, poliranje** | **Strižić** | **28. 11. 2013.** |

# Opis projektnog zadatka

Cilj projektnog zadatka jest implementacija web stranice pizzerije *“7”*. Kontraintuitivno, navedena pizzerija u svojoj ponudi neće imati isključivo pizze. Slično dot-com mjehuriću krajem 90-tih godina, kada bi svakoj tvrtci porasla vrijednost dionica i promet čim bi otvorila .com web stranicu, i Hrvati su shvatili prednosti korištenja interneta u poslovne svrhe. Porast prometa, veća zarada, mnogobrojniji korisnici, lakši i brži način kontaktiranja te pristupačnost većem broju stanovništva su samo neke od prednosti omogućavanja poslovanja preko interneta. Također, u 21. stoljeću je naručivanje hrane telefonom stvar prošlosti. Socijalna anksioznost, urbaniji i brži način života koji sve više naginje “oblaku” te sveprisutnost mreže potiču ljude da naručuju hranu putem interneta. Svaki restoran koji drži do sebe i svoga ugleda će to omogućiti krajnjim korisnicima - korist je obostrana.

Upravo radi već navedenih razloga naručivanje hrane preko interneta mora biti veoma jednostavno, štoviše, jednostavnije nego poziv restoranu! Intuitivno, jednostavno i pametno dizajnirano sučelje i landing page uvelike pomažu na putu do tog cilja.

Na naslovnoj će stranici korisnik vidjeti općenite informacije o restoranu poput imena vlasnika, adrese, prosječne ocjene korisnika, radno vrijeme, očekivano vrijeme dostave, minimalni iznos online narudžbe, cijenu dostave, popuste i akcije te način plaćanja. Već s jednim klikom miša korisnik će imati pristup savršeno organiziranome meniju gdje će moći pregledati kategoriziran popis jela, istaknutu cijenu i dostupnost. Prilikom odabira jela će moći specificirati dodatne opcije, primjerice veličinu porcije, dodatne začine, umake, itd. Posebno će biti istaknuta popularna, često naručivana jela. Osim od menija, jedan ga klik mišem dijeli i od košarice u kojoj prije narudbže ima pregled odabranih jela, sumu cijena i mogućnost brisanja ili mijenjanja količine pojedine stavke. Dakako, tu se nalazi i tipka za naručivanje, pritiskom na koju potvrđuje trenutni odabir jela i biva odveden do stranice unosa kontakt podataka i odabira načina plaćanja. Nakon naručivanja korisnik dobije potvrdu narudžbe na unesenu e-mail adresu. Nipošto se ne smije izostaviti mogućnost pregledavanja i unosa komentara na restoran i dobivenu uslugu. Stranica na kojoj se to radi je dostupna sa jednim klikom miša. Korisnik uz uneseni komentar može dati ocjenu restoranu, ili samo ocjenu, bez komentara. Smisao komentara nije tu samo za djelatnike i vlasnika, nego svi korisnici mogu vidjeti sve neobriane komentare. Feedback je bitan za poboljšanje kvalitete usluge kao i korisnikovog dojma. Ako korisnik primjeti da je savjet poslušan i da je zapravo netko pročitao njegov komentar veći su izgledi da će opet kupovati na tom mjestu. Za razliku od korisnika koji naručuju hranu, djelatnici restorana i vlasnik će imati dodatne mogućnosti. Te su dodatne mogućnosti dostupne nakon uspješnog log ina u sustav sa naslovne stranice.

Djelatnici restorana će na naslovnoj stranici moći vidjeti ukupni broj narudžbi za taj mjesec. Meni će moći biti uređivan. Djelatnici imaju mogućnost dodavanja ili brisanja kategorija kao i pojedinih jela, promjene dostupnosti i cijene, te vide koliko je puta koje jelo naručeno. Osim pregleda menija, djelatnik može pregledati i sve narudžbe. Svaka narudžba se sastoji od preslike košarice i osobnih podataka korisnika koji je napravio narudžbu. Tada djelatnik ima opciju preuzimanja te narudžbe ako istu nije već preuzeo neki drugi djelatnik. Preuzimanjem narudžbe se zapisuje ime tog djelatnika u narudžbu. Radi izdavanja računa ili prenošenja narudžbe dostavljaću, djelatnik će moći prenijeti narudžbu u definirani tekstualni oblik, spreman za printanje ili spremanje na računalo.

Vlasnik restorana ima ulogu administratora cijele web stranice. Uz sve dodatne mogućnosti koje ima djelatnik, iste ima i vlasnik, uz još neke. Na naslovnoj stranici vlasnik ima mogućnost mijenjanja svih podataka osim prosječne ocjene korisnika. Iako može obrisati komentar, ako mu se taj ne sviđa, ocjene nikako ne može obrisati niti promijeniti. Isto tako mu je idljiv i promet po mjesecima sa podatcima o broju narudžbi, najnaručivanijim jelima i ukupnom prometu za taj mjesec. Uz administratorske ovlasti dolazi i pravo pregleda podataka ostalih korisnika, u ovom slučaju djelatnika. Vlasnik može dodati nove djelatnike ili brisati stare te pregledati njihove osobne podatke i broj preuzetih narudžbi. Također, vlasnik jedini ima uvid u promet restorana, kojim se može uvjeriti koliko je dobro dizajniran sustav i koliki je porast prihoda ostvario poslovanjem preko interneta, ako ikada bude sumnjao.

Postoji nekoliko web stranica za naručivanje hrane putem interneta i većina njih funkcionira na sličan način. Razlikuju se po geografskom opsegu i broju restorana iz kojih nude naručivanje. Jedni servisi obuhvaćaju puno raznolikih restorana, dok drugi samo jedan lanac ili skup restorana po tipu kuhinje. Bitan faktor za uzeti u obzir pri isplativosti web stranice jest smještaj restorana i demografija korisnika. U urbanijim naseljima se ulog višestruko isplati zbog načina života, dok u ruralnijim ne toliko.

Obzirom na opseg online usluga i zahtjevanih performansi u ovom projektnom zadatku, za implementaciju je odabran Django framework15, pisan u jeziku Python16. Django omogućuje izrađivanje elegantnih web aplikacija dobrih performansi uz minimalni vremenski ulog. Krivo je reći da je Django najfleksibilnije i najbolje rješenje po pitanju performansi, međutim naglasak je na jednostavnosti, kako korištenja tako i implementacije i administracije. Sa Django instalacijom se pakira i SQLite baza podataka18. Samo ime sugerira da je rječ o klasičnoj SQL bazi podataka. koja je i više nego dovoljna za ovakve potrebe.

# Rječnik pojmova

|  |  |
| --- | --- |
| .txt format | Format za pohranjivanje tekstualnih datoteka |
| Administrator | Osoba odgovorna za vođenje posla, organizaciju i sl. |
| Aktor | Korisnik |
| Browser | Web preglednik; sluzi za pretrazivanje sadrzaja na internetu |
| Controller | U MVC arhitekturi: komponenta koja posreduje između unosa i promjene stanja modela ili viewa |
| Dionik | Osoba uključena u projekt |
| Django framework15 | Python okruženje visoke razine koje služi za razvoj web aplikacija |
| Dot-com mjehurić | Abnormalan rast cijena dionica nastao kao posljedica popularizacije interneta krajem 90.-ih godina 20. stoljeća |
| Feedback | Povratna informacija korisnika |
| Funkcionalni zahtjevi | Izjave o uslugama koje sustav mora pružati, kako će sustav reagirati na određeni ulazni poticaj, te kako bi se sustav trebao ponašati u određenim situacijama3 |
| Hard disk | Tvrdi disk; uređanj za pohranu digitalnih informacija |
| Landing page | Web stranica koja se otvara klikom na link u tražilici |
| Login | Proces identifikacije računalu |
| Model | U MVC arhitekturi: komponenta arhitekture koja čuva podatke web aplikacije, logiku koja čini srž aplikacije i funkcije |
| MVC arhitekrura17 | Uzorak software arhitekture Model-view-controller koji odvaja prikaz informacija od interakcije korisnika sa tim informacijama |
| Nefunkcionalni zahtjevi | Ograničenja u uslugama i funkcijama, kao što su vremenska ograničenja, (ne)usvojeni standardi, zahtjevi kvalitete, platformski zahtjevi, ograničenja u procesu razvoja i oblikovanja itd.3 |
| Online | Spojen na mrežu |
| Python16 | Viši programski jezik |
| SQLite18 | Relacijska baza podataka |
| Sudionik | Dio sustava potreban za provođenje zahtjeva |
| Sučelje | Područje interakcije dva sustava, npr. čovjeka i računala |
| UTF-8 | Međunarodni standard kodiranja znakova |
| View | U MVC arhitekturi: prikaz podataka iz modela (slika, tekst, dijagram, graf itd.) |
| Web aplikacija | Aplikacija koja koristi web preglednike kao klijente |

# Funkcionalni zahtjevi3

# Dionici:

* Klijenti
* Djelatnici pizzerije
* Vlasnik pizzerije

**Aktori i njihovi funkcionalni zahtjevi:**

* Klijent, inicijator:
  + Može vidjeti informacije o pizzeriji
  + Može pregledati meni i vidjeti česte narudžbe
  + Može pročitati komentare
  + Može ostaviti komentar
  + Može naručiti sve dostupno u meniju i uređivati košaricu
  + Može ocijeniti pizzeriju
  + Može upisati svoje osobne podatke prilikom narudžbe
* Djelatnik pizzerije, inicijator:
  + Može pregledati popis narudžbi i preuzeti ne preuzete
  + Može snimiti narudžbu u izlaznu datoteku
  + Može uređivati kategorije i jela
  + Vidi ukupan broj narudžbi za trenutni mjesec
* Vlasnik pizzerije, inicijator:
  + Može mijenjati sve podatke o pizzeriji
  + Ima opciju brisanja komentara
  + Vidi popis svih djelatnika i njihovih preuzetih narudžbi
  + Može dodavati/brisati djelatnike
  + Ima pristup izvještaju o prometu pizzerije
  + Sve ostale mogućnosti iste kao i djelatnik
* Baza podataka, sudionik:
  + Sadrži arhivu podataka o kategorijama, jelima
  + Pohranjuje komentare, ocjene i narudžbe
  + Čuva podatke o prometu

**Opis obrazaca uporabe13:**

* UC1 - *PogledajGlavnuStranicu*:
  + **Glavni sudionik**: Korisnik
  + **Cilj**: Pogledati sadržaj glavne stranice i primiti informacije o pizzeriji
  + **Sudionici**: Baza podataka
  + **Rezultat**: Primljene informacije o pizzeriji i naručivanju
  + **Željeni scenarij**:

1. Korisnik šalje upit i pristupa glavnoj stranici.  
2. Sustav vraća opće informacije o pizzeriji, meniju i načinu naručivanja.  
3. Ako je prijavljen djelatnik, daje informaciju o trenutnim narudžbama koje može zaprimiti.

* UC2 - *PogledajMeni*:
  + **Glavni sudionik:** Korisnik
  + **Cilj:** Pregledati ponudu jela pizzerije i potaknuti na naručivanje
  + **Sudionici:** Baza podataka
  + **Rezultat:** Primljene informacije o trenutnim jelima s mogućnošću narudžbe
  + **Željeni scenarij:**   
    1. Korisnik otvara meni da pregleda jela koja su trenutno moguća za naručivanje  
    2. Sustav vraća potrebne informacije o jelima s cijenom, označava često naručivane i daje opciju naručivanja svakog jela.
* UC3 - *PregledajKošaru:*
  + **Glavni sudionik:** Klijent
  + **Cilj:** Pregledati sadržaj košarice po stavkama s istaknutim cijenama stavki
  + **Sudionici:** Baza podataka
  + **Rezultat:** Primljene informacije o sadržaju košarice s mogućnošću brisanja ili izmjene
  + **Željeni scenarij:**

1. Klijent otvara košaricu da pregleda njen sadržaj i cijene

2. Sustav vraća potrebne informacije o stavkama i cijenama te daje opcije brisanja i izmjene

* UC4 - *DodajUKošaricu*:
  + **Glavni sudionik:** Klijent
  + **Cilj:** Dodati u košaricu željena jela
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Odabir nekog od ponuđenih jela u meniju
  + **Rezultat:** Dodana jela u košarici
  + **Željeni scenarij:**   
    1. Klijent odabire jelo (ili više njih) i zatraži stavljanje u košaricu.  
    2. Sustav prima zahtjev, stavlja izabrana jela u košaricu i računa sveukupni iznos za platiti.
* UC5 - *Naruči*:
  + **Glavni sudionik:** Klijent
  + **Cilj:** Naručiti željena jela
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Jedno ili više jela u košarici
  + **Rezultat:** Narudžba poslana djelatnicima i spremljena u bazu podataka.
  + **Željeni scenarij:**   
    1. Nakon što je dodao jela u košaricu, klijent upisuje osobne podatke i potvrđuje svoju narudžbu.  
    2. Sustav prima zahtjev, sprema narudžbu u bazu podataka zajedno sa upisanim osobnim podacima klijenta i šalje kopiju na klijentov e-mail.  
    3. Klijent prima povratnu informaciju da je uspješno obavio narudžbu.
* UC6 - *OstaviKomentar*:
  + **Glavni sudionik:** Klijent
  + **Cilj:** Ostaviti komentar o pizzeriji
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Minimalni broj slova u komentaru
  + **Rezultat:** Ostavljeni komentar zapisuje se u bazu podataka
  + **Željeni scenarij:**   
    1. Klijent upisuje i pošalje komentar.  
    2. Sustav provjerava ispunjava li komentar uvjet o minimalnom broju slova te sprema komentar u bazu podataka.
* UC7 - *PregledajKomentare*
  + **Glavni sudionik:** Korisnik
  + **Cilj:** Dobiti ispis svih komentara i ocjena pizzerije
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Postojanje komentara
  + **Rezultat:** Primljeni sadržaji svih ostavljenih komentara
  + **Željeni scenarij:**   
    1. Korisnik pristupa stranici sa komentarima i time šalje zahtjev sustavu za pregled komentara.  
    2. Sustav pretražuje bazu podataka i vraća sadržaj svih postojećih komentara.
  + **Mogući drugi scenariji**:  
    2. Ne postoji trenutno niti jedan komentar u bazi podataka.  
     1. Sustav obavještava korisnika o nepostojanju komentara.
* UC8 - *OcijeniRestoran*:
  + **Glavni sudionik:** Klijent
  + **Cilj:** Ocijeniti restoran sa ocjenom 1-5
  + **Sudionici:** Baza podataka
  + **Rezultat:** Ostavljena izabrana ocjena korisnika
  + **Željeni scenarij:**   
    1. Klijent otvara stranicu komentara i bira koju ocjenu želi ostaviti za pizzeriju

2. Ocjena se zapisuje u bazi podataka.

* UC9 - *ZaprimiNarudžbu*:
  + **Glavni sudionik:** Djelatnik
  + **Cilj:** Odabrati i primiti narudžbu
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Postojanje nezaprimljene narudžbe i prijava u sustav
  + **Rezultat:** Dobiveni podaci ozaprimljenoj narudžbi
  + **Željeni scenarij:**   
    1. Djelatnik šalje zahtjev sustavu za prikaz liste narudžbi  
    2. Sustav vraća listu narudžbi

3. Djelatnik iz prikazane liste narudžbi odabire nezaprimljenu koju želi  
 preuzeti.

4. Sustav označava narudžbu sa imenom djelatnika i vraća presliku  
 košarice klijenta sa njegovim dodatnim podacima.

* + **Mogući drugi scenariji**:

2. Ne postoji trenutno niti jedna narudžba.

1. Sustav vraća informaciju o tome da trenutno ne postoji niti jedna narudžba u bazi podataka pa nema mogućnosti odabira.

* UC10 - *SnimiNarudžbu*:
  + **Glavni sudionik:** Djelatnik
  + **Cilj:** Preuzeti narudžbu u izlaznu datoteku
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Prijava u sustav
  + **Rezultat:** Spremljena narudžba u tekstualnu datoteku na fizički disk.
  + **Željeni scenarij:**   
    1. Djelatnik šalje zahtjev za snimanjem narudžbe.  
    2. Sustav prima zahtjev za snimanje narudžbe te šalje djelatniku izlaznu datoteku koju će spremiti na disk u predodređenom tekstualnom obliku (.pdf).
* UC11 - *PrijaviSe*:
  + **Glavni sudionik:** Djelatnik
  + **Cilj:** Prijaviti se u sustav
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Već nije prijavljen
  + **Rezultat:** Djelatnik je prijavljen u sustav i može koristiti sve funkcije sustava u svojoj ovlasti.
  + **Željeni scenarij:**   
    1. Djelatnik upisuje i šalje sustavu svoje podatke (korisničko ime i lozinku).  
    2. Sustav provjera je li upisani podaci odgovaraju podacima u bazi podataka i daje odgovarajuće ovlasti.
  + **Mogući drugi scenariji:**  
    2. Podaci ne odgovaraju onima u bazi podataka, nisu pronađeni.  
     1. Sustav dojavljuje djelatniku grešku o neispravnim podacima.
* UC12 - *UrediPopisJela*:
  + **Glavni sudionik:** Djelatnik
  + **Cilj:** Izvršiti promjene na kategorijama ili jelima
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Prijava u sustav
  + **Rezultat:** Izvršene izmjene na postojećim kategorijama/jelima ili dodavanje novih.
  + **Željeni scenarij:**   
    1. Prijavljeni djelatnik šalje sustavu zahtjev za promjenu određene kategorije/određenog jela ili da doda potpuno nove.  
    2. Sustav prima i obrađuje zahtjev tako što mijenja ili dodaje nove podatke u bazi podataka te obaviještava djelatnika o uspješnosti.
  + **Mogući drugi scenariji**:  
    2. Već postoji kategorija ili jelo u bazi koje je djelatnik htio dodati.

1. Sustav ne dodaje novu kategoriju ili novo jelo u bazu podataka.

2. Sustav obaviještava djelatnika o nemogućnosti dodavanja.

* UC13 - *UrediDjelatnika*:
  + **Glavni sudionik:** Vlasnik
  + **Cilj:** Dodati ili obrisati djelatnika
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Prijava u sustav
  + **Rezultat:** Dodavanje novog računa djelatnika ili brisanje postojećeg
  + **Željeni scenarij:**   
    1. Prijavljeni vlasnik upisuje osobne podatke o novom djelatniku i šalje zahtjev za dodavanjem.  
    2. Sustav dodaje novog djelatnika u bazu podataka i obavještava vlasnika o uspješnosti radnje.
  + **Drugi mogući scenariji**:  
    2. Djelatnik s navedenim korisničkim imenom već postoji u bazi.  
     1. Sustav ne dodaje novog djelatnika u bazu podataka.

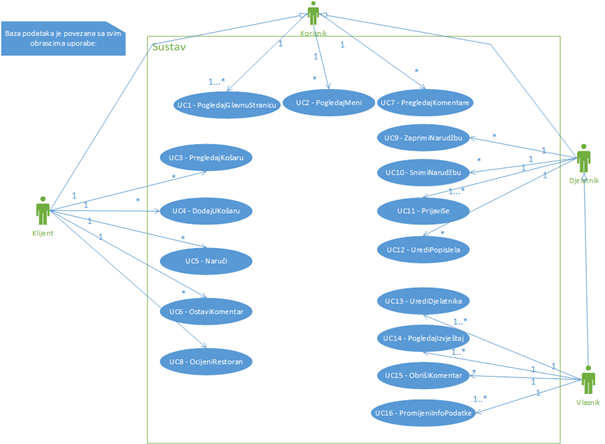
2. Sustav obavještava vlasnika o nemogućnosti dodavanja.

**ili:** 1. Prijavljeni vlasnik šalje zahtjev za brisanje postojećeg djelatnika te

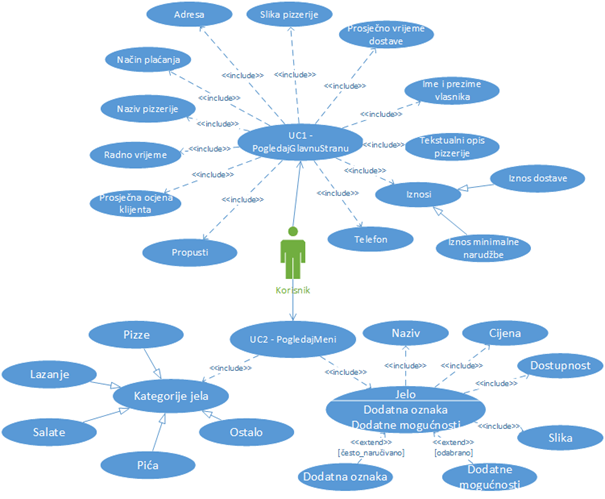
unosi njegovo korisničko ime.

2. Sustav briše odabranog djelatnika iz baze podataka i obavještava  
 vlasnika o uspješnosti radnje.

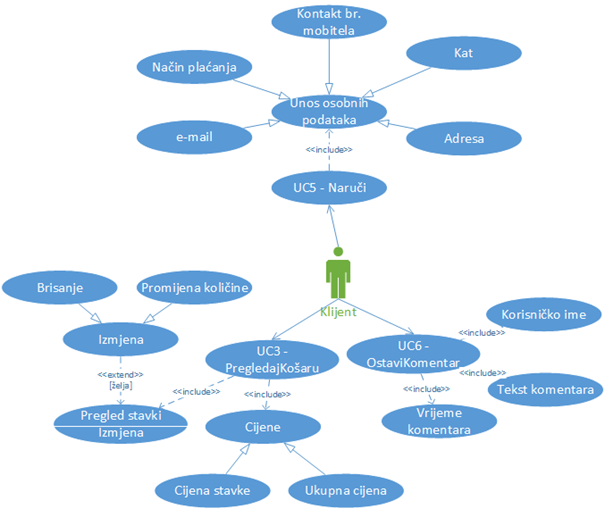
* UC14 - *PogledajIzvještaj*:
  + **Glavni sudionik:** Vlasnik
  + **Cilj:** Pogledati izvještaj o prometu pizzerije
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Prijava u sustav
  + **Rezultat:** Dobiven mjesečni izvještaj o prometu
  + **Željeni scenarij:**   
    1. Prijavljeni vlasnik šalje zahtjev za pregledom mjesečnog izvještaja.  
    2. Sustav prima zahtjev i vraća za svaki mjesec: broj narudžbi, proječnu cijenu narudžbe, tri najčešće naručivana jela i ukupni promet pizzerije.
* UC15 - *ObrišiKomentar*:
  + **Glavni sudionik:** Vlasnik
  + **Cilj:** Obrisati komentar o pizzeriji
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Prijava u sustav i postojanje komentara
  + **Rezultat:** Obrisan komentar koji se ne može vratiti
  + **Željeni scenarij:**   
    1. Prijavljeni vlasnik šalje zahtjev za brisanjem određenog komentara iz liste.  
    2. Sustav prima zahtjev i potpuno briše željeni komentar iz baze podataka.
* UC16 - *PromijeniInfoPodatke*:
  + **Glavni sudionik:** Vlasnik
  + **Cilj:** Promijeniti informacije o pizzeriji na glavnoj stranici
  + **Sudionici:** Baza podataka
  + **Preduvjeti:** Prijava u sustav
  + **Rezultat:** Promijenjeni podaci o pizzeriji
  + **Željeni scenarij:**   
    1. Prijavljeni vlasnik šalje zahtjev za promjenu podataka o pizzeriji na glavnoj stranici.  
    2. Sustav prima zahtjev o promjeni podataka i obrađuje ga.



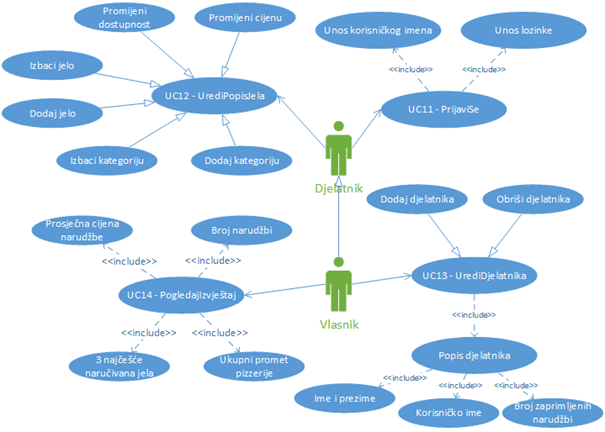
Slika 4.1. Dijagram obrazaca uporabe19 za UC1 do UC16



Slika 4.2. Detaljniji prikaz obrazaca UC1 i UC2



Slika 4.3. Detaljniji prikaz obrazaca UC3, UC5 i UC6

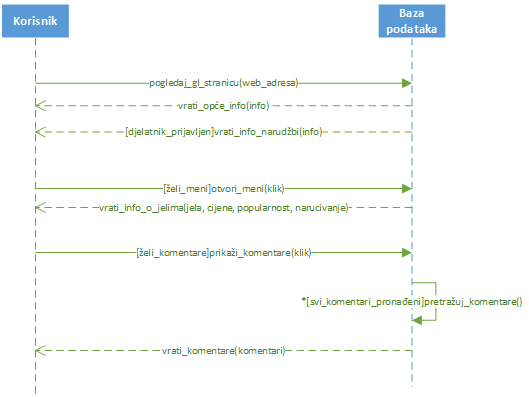


Slika 4.4. Detaljniji prikaz obrazaca UC11, UC12, UC13 i UC14

**Sekvencijski dijagrami14:**

Obrasci uporabe UC1, UC2 i UC7:

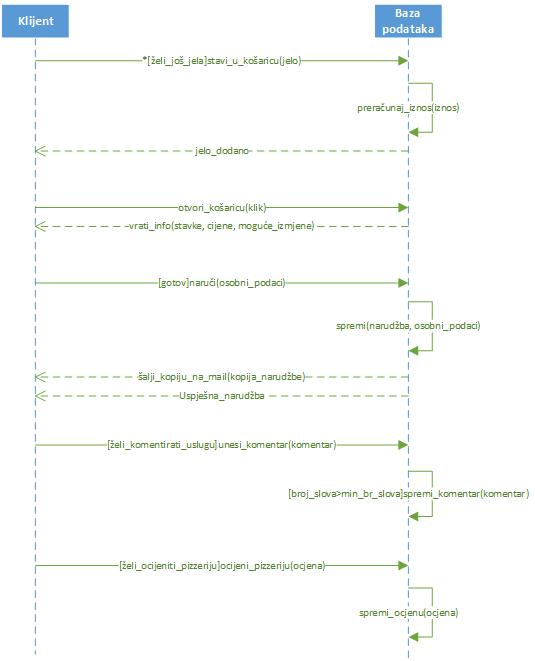
Korisnik započinje pregled glavne stranice unosom njene web adrese. Baza podataka vraća opće informacije glavne stranice, a ako je prijavljen djelatnik vraća i informacije o narudžbama. Ukoliko korisnik želi pogledati meni, jednim klikom ga otvara. Baza podataka vraća informacije o jelima te opcije naručivanja. Ukoliko korisnik želi pogledati komentare, opet jednim klikom zatražuje prikaz komentara. Baza podataka pretažuje komentare dok svi nisu pronađeni te vraća komentare korisniku.



Slika 4.5. Sekvencijski dijagram obrazaca uporabe19 UC1, UC2 i UC7

Obrasci uporabe UC3, UC4, UC5, UC6 i UC8:

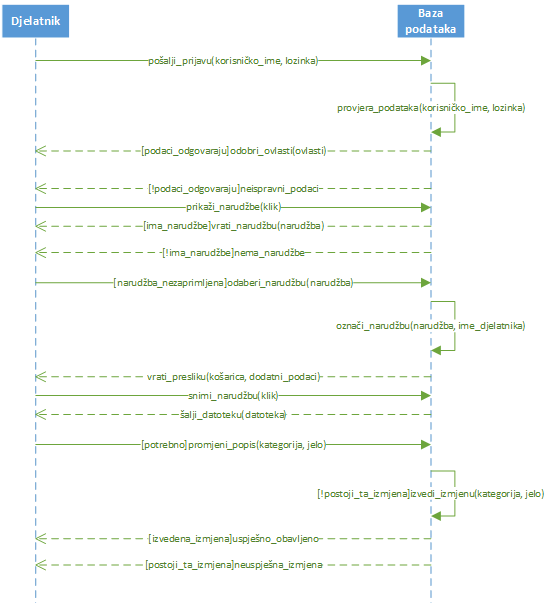
Klijent stavlja koliko god želi jela u svoju košaricu. Za svako jelo, baza podataka, preračunava ukupan iznos te obavještava klijenta da je jelo dodano. Kada klijent želi pregledati svoj odabir jela, otvara košaricu. Baza podataka vraća sve informacije o odabranim jelima s mogućnostima izmjene. Kada je klijent gotov s odabirom jela, odlučuje se za naručivanje te šalje osobne podatke. Baza podataka sprema klijentovu narudžbu i osobne podatke, šalje kopiju narudžbe na klijentovu e-mail adresu te ga obavještava o uspješnosti narudžbe. Nakon naručivanja, ako želi, klijent može komentirati uslugu unosom komentara. Baza podataka provjerava da li komentar zadovoljava uvjet minimalnog broja slova te, ako da, sprema komentar. Nakon toga klijent, ako želi, ocijenjuje pizzeriju unosom ocjene koja se sprema u bazu podataka.



Slika 4.6. Sekvencijski dijagram obrazaca uporabe UC3, UC4, UC5, UC6 i UC8

Obrasci uporabe UC9, UC10, UC11 i UC12:

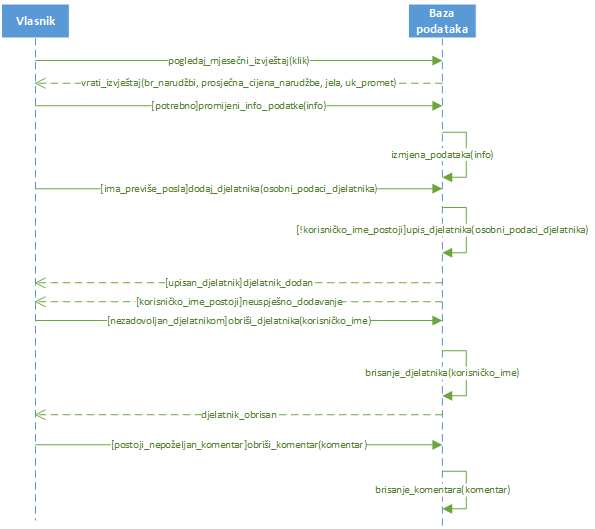
Djelatnik šalje prijavu u sustav unosom korisničkog imena i lozinke. Baza podataka provjerava te podatke te, ako podaci odgovaraju, odobrava ovlasti djelatniku. Ako podaci ne odgovaraju, javlja da su uneseni neispravni podaci. Nakon prijave, djelatnik zahtjeva prikaz narudžbi te mu baza podataka vraća narudžbe ako ih ima. Ako ih nema, javlja da nema narudžbi. Djelatnik odabire nezaprimljenu narudžbu. Baza podataka označava tu narudžbu imenom djelatnika te mu vraća presliku narudžbe sa sadržajem košarice i osobnim podacima. Djelatnik zahtjeva snimanje narudžbe te od baze dobiva datoteku narudžbe. Kada je završio narudžbe, ako je potrebno, djelatnik mijenja popis kategorija i jela. Baza izvršava izmjenu ako ona već ne postoji te javlja djelatnika da je uspješno obavljena. Ako je postojala ta izmjena, djelatniku se javlja neuspješnost.



Slika 4.7. Sekvencijski dijagram obrazaca uporabe UC9, UC10, UC11 i UC12

Obrasci uporabe UC13, UC14, UC15 i UC16:

Vlasnik odabire pregled mjesečnog izvještaja. Baza podataka mu vraća izvještaj sa svim podacima. Ako je potrebno vlasnik unosi drugačije informacije o pizzeriji. Baza podataka vrši izmjenu podataka. Ako je bilo previše posla vlasnik želi dodati novog djelatnika unosom njegovih osobnih podataka. Ako uneseno korisničko ime već ne postoji baza podataka upisuje djelatnika te vlasnik dobiva poruku da je djelatnik dodan. Ako je postojalo uneseno korisničko ime, vlasnik dobiva poruku da je dodavanje neuspješno. Ako je vlasnik nezadovoljan djelatnikom želi ga obrisati unosom njegovog korisničkog imena. U bazi podataka se briše djelatnik te vlasnik dobiva poruku da je djelatnik obrisan. Na kraju vlasnik pregledava komentare te, ako postoji nepoželjan komentar, zahtjeva njegovo brisanje. Baza podataka briše odabrani komentar.



Slika 4.8. Sekvencijski dijagram obrazaca uporabe UC13, UC14, UC15 i UC16

# Ostali zahtjevi3

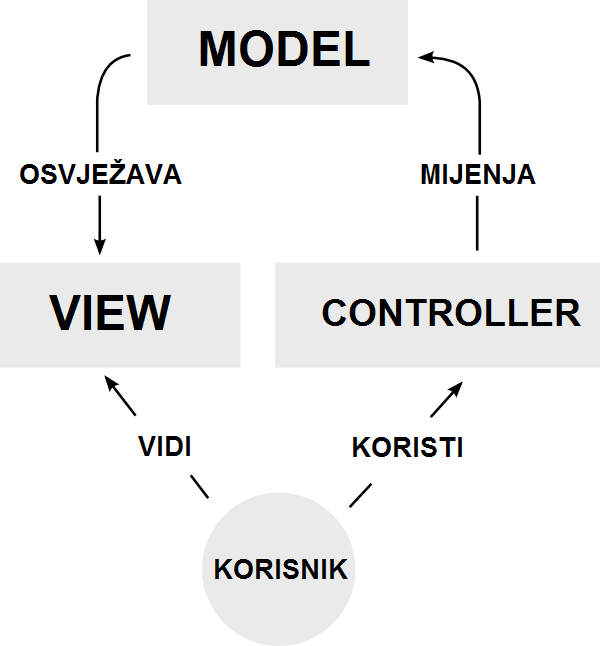
* Sustav mora podržavati istovremeni rad najmanje tri djelatnika, pet klijenata i jednog vlasnika.
* Sustav mora biti jednostavan za korištenje, bez potrebnih dodatnih uputa za korisnike.
* Sustav bi trebao davati pouzdane informacije o pizzerijii te ispravan izračun ukupnog iznosa.
* Tekstualni oblik izlazne datoteke i cijelo sučelje treba podržavati unos hrvatskih dijakritičkih znakova (UTF-8).
* Korisničko sučelje mora omogućiti izbor dodatnih mogućnosti koje dolaze sa dodijeljenim korisničkim dozvolama.
* Sustav mora svako jelo koje je naručeno barem dva puta češće od prosjeka naručivanja za sva jela označiti kao “često naručivano”.
* Sustav ne smije dozvoliti manju narudžbu od određene minimalne cijene.
* Sustav se ispituje sa najmanje četiri kategorije jela i najmanje 30 pojedinačnih jela.
* Sustav podržava nadogradnju izvornog koda uz korištenje Django 1.5 verzije.

# Arhitektura i dizajn sustava

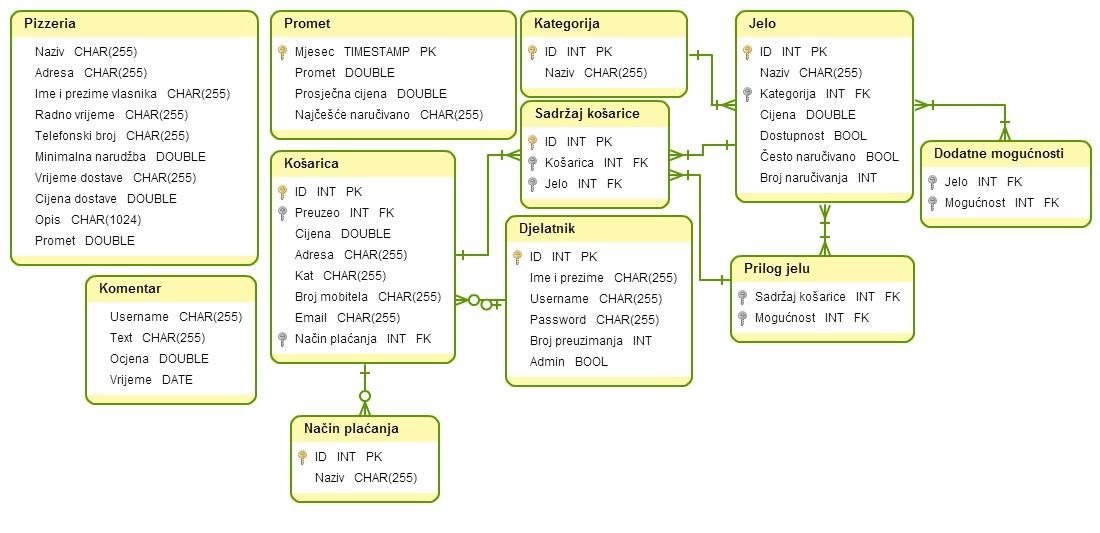
## Svrha, opći prioriteti i skica sustava

Sustav implementira online naručivanje hrane. Obzirom da vjerojatnost da korisnik uistinu nešto i naruči ovisi o lakoći korištenja sučelja, ono mora biti što jednostavnije i intuitivninje. Korisnik vrlo vjerojatno neće biti voljan instalirati dodatnu programsku podršku nego će se oslanjati na ono što već ima, a to je standardni browser. Zbog navedenih razloga je odluka pala na arhitekturu web aplikacije, konkretno često korištenu MVC arhitekturu17.

**MVC** arhitektura se sastoji od tri dijela: **model**, **view** i **controller**. Svaki dio aplikacije pripada jednom od tih dijelova, ovisno o njegovoj ulozi u sustavu. Zbog takve apstraktne podjele olakšava se implementacija i ideja korištenja, kao i održavanje sustava te dodavanja novih mogućnosti. To je razlog zašto baš ova arhitektura prevladava internet aplikacijama. **MVC** arhitektura uključuje i interakciju između komponenti. Korisnik koristi **controller** te vidi **view**. **Controller** mijenja model, a model osvježava **view**. Dakako, na slici nije ilustrirano da je korisnik zapravo iza preglednika.



Slika 6.1. MVC skica

**Model** je komponenta arhitekture koja čuva podatke web aplikacije (u obliku baze podataka), logiku koja čini srž aplikacije i funkcije. Prilikom promjene stanja **modela** svi asocirani **viewovi** i **controlleri** mogu biti obaviješteni o događaju. **View** tada osvježi svoj izlaz, a **controller** osvježi dostupne naredbe. **Model** će u našoj web aplikaciji sadržavati sučelje za SQLite bazu podataka, logiku za brojanje narudžbi, prometa te izračunavanja najčešće naručivanih jela. Baza podataka će držati informacije o jelima i kategorijama, narudžbama, kao i popis radnika i komentara, te konačno, osnovne informacije o samoj pizzeriji. 

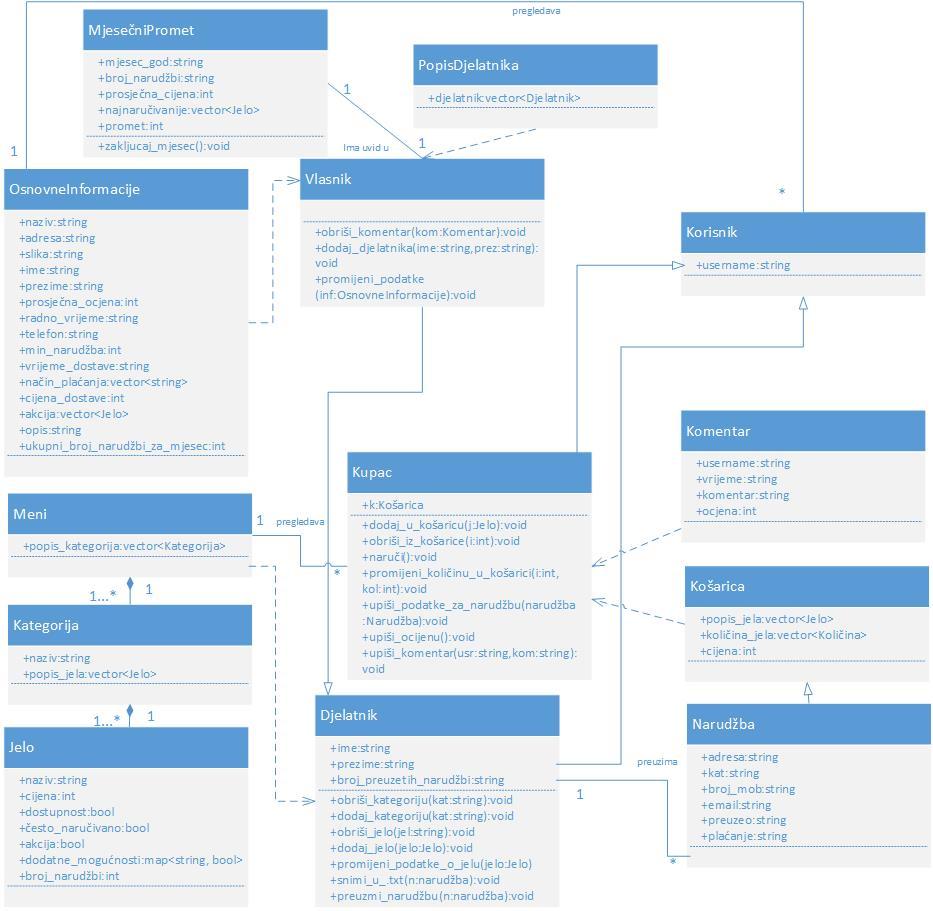
Slika 6.2. ER model baze podataka

Tablica **Pizzeria** sadrži osnovne informacija od kojih će se većina prikazivati na naslovnoj stranici svakom korisniku. Tablica **Promet** služi kao log iznosa prometa po mjesecima. Aktualni mjesec je u tablici **Pizzeria**, te se na kraju mjeseca automatski prebacuje u tablicu **Promet**. Tablica **Kategorija** čuva nazive kategorija. Tablica **Jelo** drži sve informacije o pojedinom jelu, uključujući kategoriju kojoj to jelo pripada. Bitno je reći da ona drži i popis dodatnih mogućnosti i priloga. Mogućnost odabira neke dodatne mogućnosti ovisi o jelu, pa se zato tablica **Jelo** refleksivno veže sama sa sobom kroz tablicu **Dodatne mogućnosti**. Tamo su povezana jela sa svim dodatnim mogućnostima koje se mogu birati za to pojedino jelo. **Košarica** je log svih podataka o svim dostavama. Očigledno, tablica **Sadržaj košarice** skupa sa tablicom **Prilog jelu** je veza između jela, košarica i odabranih priloga, odnosno utjelovljenje je popisa jela i pripadajućih dodatnih mogućnosti koji su unutar jedne košarice. Tablica **Komentar** služi kao zapisnik svih komentara i ocjena, dok tablica **Djelatnik** služi kao popis svih aktualnih djelatnika i administratora (vlasnika). Nakon što djelatnik preuzme neku košaricu, njegovo se ime zapiše u odgovarajući stupac tablice **Košarica**. U toj je tablici također i zapisan odabrani način plaćanja, sa popisa svih dostupnih naćina plaćanja iz tablice **Način plaćanja**.

**View** je bilo koji prikaz podataka iz **modela**, poput slike, teksta, dijagrama, grafa itd. Različiti **viewovi** mogu prikazivati iste podatke u različitim oblicima, dakle reprezentacija podataka ovisi isključivo o implementaciji **viewa**. U tom smislu je **view** pasivna komponenta **MVC** arhitekture - ne upravlja ničime, niti što mijenja. U našoj aplikaciji će **viewovi** biti pogledi na naslovnu stranicu, meni, košaricu, narudžbu, popis narudžbi za preuzimanje, popis komentara i popis radnika.

**Controller** je komponenta koja posreduje između unosa i promjene stanja **modela** ili **viewa**. **Controller** oblikuje korisnikov unos, pretvarajući ga u naredbe za interpretaciju **modelu** i/ili **viewu**. Na svakom **viewu** ćemo imati kontrole koje odgovaraju korisnikovoj razini prava. Administrator i djelatnik će moći uređivati neke podatke iz **modela**, koje će klijent samo moći gledati. Klijent će upravljati narudžbom koristeći određene kontrole, dok djelatnik neće.

## Dijagram razreda s opisom10



Slika 6.3. Dijagram razreda

Razred **OsnovneInformacije** za članove ima sve podatke koji će se prikazivati na naslovnoj stranici. Sukladno specifikacijama, postoje tri razine prava pristupa, te ovisno o njima neke od tih informacija će biti izostavljene iz prikaza.

Razred **Vlasnik** predstavlja vlasnika restorana koji jedini ima ulogu administratora. On jedini može uređivati i mijenjati **OsnovneInformacije** i jedini ima pristup objektima razreda **MjesečniPromet**.

Očigledno, **MjesečniPromet** se sastoji od podataka o prometu kroz mjesece i ostalih interesantnih podataka, poput najnaručivanijih jela i ukupnog prometa. Na kraju svakog mjeseca će se automatski pozvati funkcija *zakljucaj\_mjesec()* koja će spriječiti daljnje izmjene u proteklom mjesecu.

Osim što **Vlasnik** može mijenjati **OsnovneInformacije**, također može mijenjati i **PopisDjelatnika**. Vlasnik je osoba koja odlučuje tko se zapošljava u restoranu, stoga je prirodno da razred **Vlasnik** jedini može dodavati nove zaposlenike.

Razred **Meni** je hijerarhijski iznad razreda **Kategorija** i **Jelo**, te će kao takav sadržavati popis **Kategorija**. Prateći isti princip, svaki razred **Kategorija**, osim svog imena, sadrži popis jela koje pripadaju toj kategoriji. Objekti razreda **Jelo** će predstavljati pojedino jelo koje se može ubaciti u košaricu i naručiti.

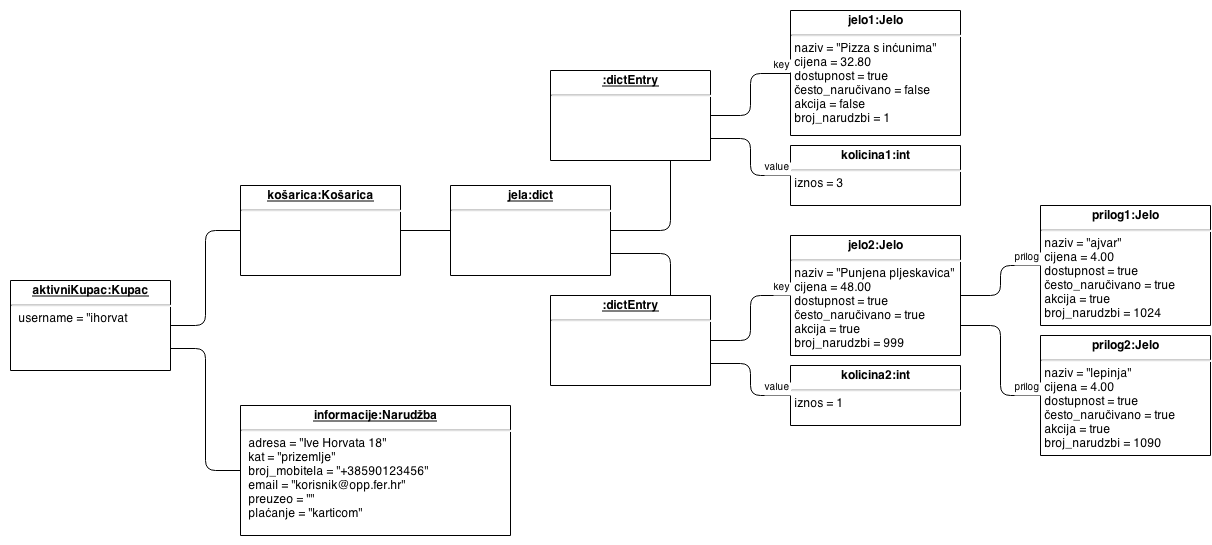
Jedini razred koji može uređivati **Meni** je razred **Djelatnik**, međutim, prema specifikacijama vlasnik ima sva prava kao i djelatnici i neka dodatna, stoga razred **Vlasnik** nasljeđuje razred **Djelatnik**. Djelatnik nad menijem vrši operacije dodavanja, brisanja i uređivanja jela i kategorija. To mu je pravo eksplicitno dodjeljeno samo nad **Menijem**, no zato jer **Meni** u sebi kao članove ima **Kategorije** i **Jela**, ista prava vrijedi i nad njima.

Osim što može uređivati **Meni**, djelatnik ima i opciju preuzimanja **Narudžbe**. Razred **Narudžba** nasljeđuje razred **Košarica**, zato jer sadrži sve iste članove kao i **Košarica**,pa se na to nadoveže sa osobnim podatcima kupca koji radi narudžbu. Preuzimanjem narudžbe djelatnik vidi jela koja su naručena i adresu na koju treba napraviti dostavu. Kao dodatnu mogućnosti tu se ističe spremanje te narudžbe u .txt format na hard disk.

Razredi **Djelatnik** i **Kupac** oboje nasljeđuju apstraktni razred **Korisnik**, zato jer su ultimativno i **Djelatnik** i **Vlasnik** i **Kupac** korisnici sustava. Ono što im je svima zajedničko je to da imaju korisničko ime, kao što stoji u **Korisniku**.

**Kupac** je razred koji stvara **Košarice** i bira što će sa **Menija** ići u nju. Kada je zadovoljan **Košaricom** koju je napravio, stavara **Narudžbu**. Osim naručivanja hrane, **Kupac** može ostaviti **Komentar** na uslugu, kao i ocjenu. I dok je stina da **Vlasnik** može brisati **Komentare** koji mu se ne sviđaju, ocjene ne može obrisati te su one trajno zapamćene u sustavu.

## Dijagram objekata20



Slika 6.4. Objektni dijagram svih aktivnih objekata u trenutku narudžbe

# Reference

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

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. Oblikovanje programske potpore, FER ZEMRIS, <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/>
6. Software engineering ,Rutgers University, <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/SE>
8. Concepts: Requirements, <http://www.upedu.org/upedu/process/gcncpt/co_req.htm>
9. UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
10. Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
11. ArgoUML, <http://argouml.tigris.org/>
12. UML, http://en.wikipedia.org/wiki/Unified\_Modeling\_Language
13. Use case Diagrams, http://en.wikipedia.org/wiki/Use\_case
14. Sequence Diagrams, http://en.wikipedia.org/wiki/Sequence\_diagram
15. Django, https://www.djangoproject.com/
16. Python, https://wiki.python.org/moin/
17. MVC, http://en.wikipedia.org/wiki/Model-view-controller
18. SQLite, http://en.wikipedia.org/wiki/SQLite
19. Creately - Diagram Software, http://creately.com/
20. Object Diagrams, http://en.wikipedia.org/wiki/Object\_diagram

# Dodatak A: Dnevnik sastajanja

+---------------------------------------------+

| Dnevnik sastajanja projektne grupe SilkRoad |

+---------------------------------------------+

-------------------------------------

S1: 06. 10. 2013., 20h, FER, Zgrada E

-------------------------------------

Prisutni:

Luka Strižić, Belma Gutlić, Petar Šegina, Nika Tilić

Sažetak:

Upoznavanje prisutnih članova tima i razgovor o mogućnostima implementacije (jezici, paketi, biblioteke, frameworkci).

Luka Strižić (voditelj), Belma Gutlić, Matija Srbić, Petar Šegina:

- fokus na implementaciju

Matej Brkašić, Davor Lovrić, Nika Tilić:

- fokus na dokumentiranje

Zaključci:

Luka Strižić deklariran voditeljem. Podjela zadataka na najvišoj razini po vještinama.

---------------------------------

S2: 18. 10. 2013., 13h, FER, aula

---------------------------------

Prisutni:

Luka Strižić, Matej Brkašić, Belma Gutlić, Davor Lovrić, Matija Srbić, Petar Šegina, Nika Tilić

Sažetak:

Konkretan dogovor o korištenim tehnologijama. Razrada zadataka.

Luka Strižić (voditelj):

- kooridniranje

- opis projektnog zadatka

- arhitektura

Matej Brkašić:

- zahtjevi

- reference

Belma Gutlić:

- zahtjevi

- reference

Davor Lovrić:

- zahtjevi

- rječnik pojmova

Matija Srbić:

- opis projektnog zadatka

- arhitektura

Petar Šegina:

- upoznavanje Django frameworka

- arhitektura

Nika Tilić:

- zahtjevi

- rječnik pojmova

Zaključci:

Započet rad na dokumentaciji. Podjeljeni konkretni zadatci nekim članovim. Inicijalne ideje o arhitekturi sustava i baze podataka.

-----------------------------

S3: 17. 11. 2013., 18h, Skype

-----------------------------

Prisutni:

Luka Strižić, Matej Brkašić, Belma Gutlić, Davor Lovrić, Matija Srbić, Petar Šegina, Nika Tilić

Sažetak:

Rekapitulacija napravljenog i nenapravljenog. Konačna podjela zadataka. Usklađivanje dijagrama i ideja.

Luka Strižić (voditelj):

- kooridniranje

- opis projektnog zadatka

- arhitektura

- ER dijagram

Matej Brkašić:

- use case dijagrami

- sekvencijski dijagrami

Belma Gutlić:

- zahtjevi

- ostali zahtjevi

Davor Lovrić:

- dodavanje referenci

- superskriptiranje riječi sa referencama

Matija Srbić:

- arhitektura

- lektoriranje opisa projektnog zadatka

- dijagram razreda

Petar Šegina

- arhitektura

- objektni dijagrami

Nika Tilić

- rječnik pojmova

- lektoriranje

Zaključci:

Definiranje zadnjih zadataka i završavanje dokumentacije. Ispravljanje typoa, lektoriranje, uređivanje, čišćenje. Sinkronizacija eventualnih neslaganja između pojedinih dijelova.