

UML ZADACI

PRIMJERI MODELIRANJA UML DIJAGRAMIMA

Zagreb, 2021.

Sadržaj

Sadržaj.....	1
1. Dijagrami obrazaca uporabe.....	4
➤ Zadaci	4
1.1. Upravljanje hotelom	4
1.2. Prodaja autobusnih karata.....	4
1.3. Grozd računala.....	4
1.4. Rent-a-car kompanija	5
1.5. Portal za oglašavanje.....	5
➤ Rješenja	7
1.1. Upravljanje hotelom	7
1.2. Prodaja autobusnih karata.....	8
1.3. Grozd računala.....	9
1.4. Rent-a-car kompanija	10
1.5. Portal za oglašavanje.....	11
2. Dijagrami razreda	12
➤ Zadaci	12
2.1. Web trgovina.....	12
2.2. Tvrtka za popravak informatičke opreme.....	12
2.3. Program za crtanje	13
2.4. Programska potpora za licenciranje	13
2.5. Portal za oglašavanje.....	14
➤ Rješenja	15
2.1. Web trgovina.....	15
2.2. Tvrtka za popravak informatičke opreme.....	16
2.3. Program za crtanje	17
2.4. Programska potpora za licenciranje	18
2.5. Portal za oglašavanje.....	19

3.	Dijagrami stanja.....	20
➤	Zadaci	20
3.1.	Oglas na portalu za oglašavanje	20
3.2.	Tablični kalkulator	20
3.3.	Izvođenje poslova na procesoru.....	20
3.4.	Korisničko sučelje web trgovine	21
➤	Rješenja	22
3.1.	Oglas na portalu za oglašavanje	22
3.2.	Tablični kalkulator	23
3.3.	Izvođenje poslova na procesoru.....	24
3.4.	Korisničko sučelje web trgovine	26
4.	Dijagrami komponenti.....	27
➤	Zadaci	27
4.1.	MVC	27
4.2.	Portal za oglašavanje.....	27
➤	Rješenja	28
4.1.	MVC	28
4.2.	Portal za oglašavanje.....	28
5.	Dijagrami razmještaja.....	29
➤	Zadaci	29
5.1.	Klijent – poslužitelj.....	29
5.2.	Trorazinska arhitektura	29
5.3.	Vremenska prognoza	29
5.4.	Web trgovina.....	30
➤	Rješenja	31
5.1.	Klijent – poslužitelj.....	31
5.2.	Trorazinska arhitektura	33
5.3.	Vremenska prognoza	34
5.4.	Web trgovina.....	35

1. *Dijagrami obrazaca uporabe*

➤ *Zadaci*

1.1. **Upravljanje hotelom**

Modelirajte pomoću dijagrama obrazaca uporabe sustav upravljanja hotelom zadan sljedećim opisom.

Gosti hotela mogu putem interneta pretražiti hotelsku ponudu i napraviti rezervaciju sobe. Prilikom rezervacije sobe gosti mogu opcionalno rezervirati parkirno mjesto te dokupiti neke od sljedećih opcija: doručak, polupansion ili puni pansion. Recepcionar hotela može izdati sobu i napraviti naplatu. Kod izdavanja sobe recepcionar obavezno provjerava rezervaciju i programira ključ (karticu) za otvaranje sobe. Prilikom naplate recepcionar izdaje račun. Administrator sustava upravlja raspoloživim sobama, unosi parkirna mjesta te uređuje cjenik hotela. Upravljanje raspoloživim sobama podrazumijeva dodavanje novih soba, brisanje postojećih te unos podataka o sobi (broj ležaja, dodatni sadržaj itd.).

1.2. **Prodaja autobusnih karata**

Modelirajte pomoću dijagrama obrazaca uporabe aplikaciju za prodaju karata na autobusnom kolodvoru zadanu sljedećim opisom.

Aplikaciju koriste blagajnici i putnici. Blagajnik može na blagajni rezervirati mjesto ili prodati kartu putniku. Prodaja karte uključuje odabir relacije i naplatu. Naplata se može izvršiti u gotovini ili putem kartice. U slučaju da se naplata vrši putem kartice, u naplati sudjeluje putnik utipkavanjem PIN-a. Opcionalno kod prodaje karte blagajnik može izdati račun R1 ako putnik to zatraži. Osim na blagajni, rezervaciju mjesta može napraviti i putnik samostalno preko telefonskog automata.

1.3. **Grozđ računala**

Modelirajte pomoću dijagrama obrazaca uporabe načine korištenja grozda računala akademske zajednice.

Korisnici grozda su članovi akademske zajednice. Korisnici na grozdu mogu dodavati nove poslove za obradu, brisati postojeće poslove te mijenjati parametre postojećih poslova. Izmjena parametra postojećih poslova uključuje brisanje postojećeg posla i stvaranje novog posla. Također korisnici mogu slati podatke na grozđ računala, kopirati podatke i učitavati podatke s grozda računala.

Administratori grozda zaduženi su za dodavanje novih korisnika, promjenu podataka korisnika, brisanje postojećih korisnika te za konfiguraciju sustava. Konfiguracija sustava podrazumijeva podešavanje postavki raspoređivača poslova i izrade sigurnosne kopije podataka. Izrada sigurnosne kopije podataka je posebna vrsta kopiranja podataka. Administratori također mogu upravljati poslovima i podacima na grozdu.

1.4. Rent-a-car kompanija

Potrebno je izraditi dijagram obrazaca uporabe sustava koji služi kao podrška radu rent-a-car kompanije, a u žarištu sustava je kontrola flote rent-a-car vozila i rad s kupcima.

Korisnici koji iznajmljuju automobile mogu preko interneta napraviti i otkazati rezervaciju vozila. Kod stvaranja rezervacije korisnici moraju unijeti broj kreditne kartice. Korisnici mogu napraviti rezervaciju vozila samo u slučaju da u sustavu za zatražen vremenski period postoje slobodna vozila. Kod stvaranja rezervacije korisnici mogu prema želji i raspoloživim resursima napraviti i rezervaciju za dodatak: GPS uređaj, dječju sjedalicu, krovne nosače za bicikle ili skije i prikolicu.

Kada korisnici dođu u poslovnicu rent-a-car kompanije u kojoj trebaju preuzeti vozilo, zaposlenik provjerava postoji li rezervacija vozila za određenog korisnika. Ako rezervacija ne postoji, zaposlenik rent-a-car kompanije izrađuje rezervaciju za korisnika. Nakon što je rezervacija napravljena (ili je postojala od ranije), može se pristupiti izdavanju vozila. Pri izdavanju vozila, zaposlenik rent-a-car kompanije ispisuje obrazac o preuzimanju vozila koji korisnik potpisuje. Potpisani obrazac se skenira i unosi u računalo. Nakon što je obrazac skeniran, zaposlenik rent-a-car kompanije odvodi klijenta do automobila i predaje mu ključeve, prometnu dozvolu i obrazac o preuzimanju vozila. Kod povratka automobila, korisnik dolazi u poslovnicu rent-a-car kompanije gdje predaje obrazac o preuzimanju vozila, prometnu dozvolu i ključeve automobila. Zaposlenik odlazi s korisnikom do automobila, provjerava stanje automobila i preostalu količinu goriva. Nakon povratka u poslovnicu zaposlenik generira račun i provodi terećenje kreditne kartice korisnika.

Sustav također treba omogućiti zaposleniku nabavku i rashodovanje vozila te pratiti njihovo korištenje kroz vrijeme.

1.5. Portal za oglašavanje

Modelirajte pomoću dijagrama obrazaca uporabe portal za oglašavanje.

Oglašavati se mogu pokretnine, nekretnine i usluge. Oglasi su podijeljeni u kategorije: Nekretnine, Auto-moto, Roba i Usluge. Svaki oglas sadržava naslov, opis, cijenu usluge, naziv oglašivača i kontakt te može sadržavati slike i kratke video materijale, dok ostali podaci mogu

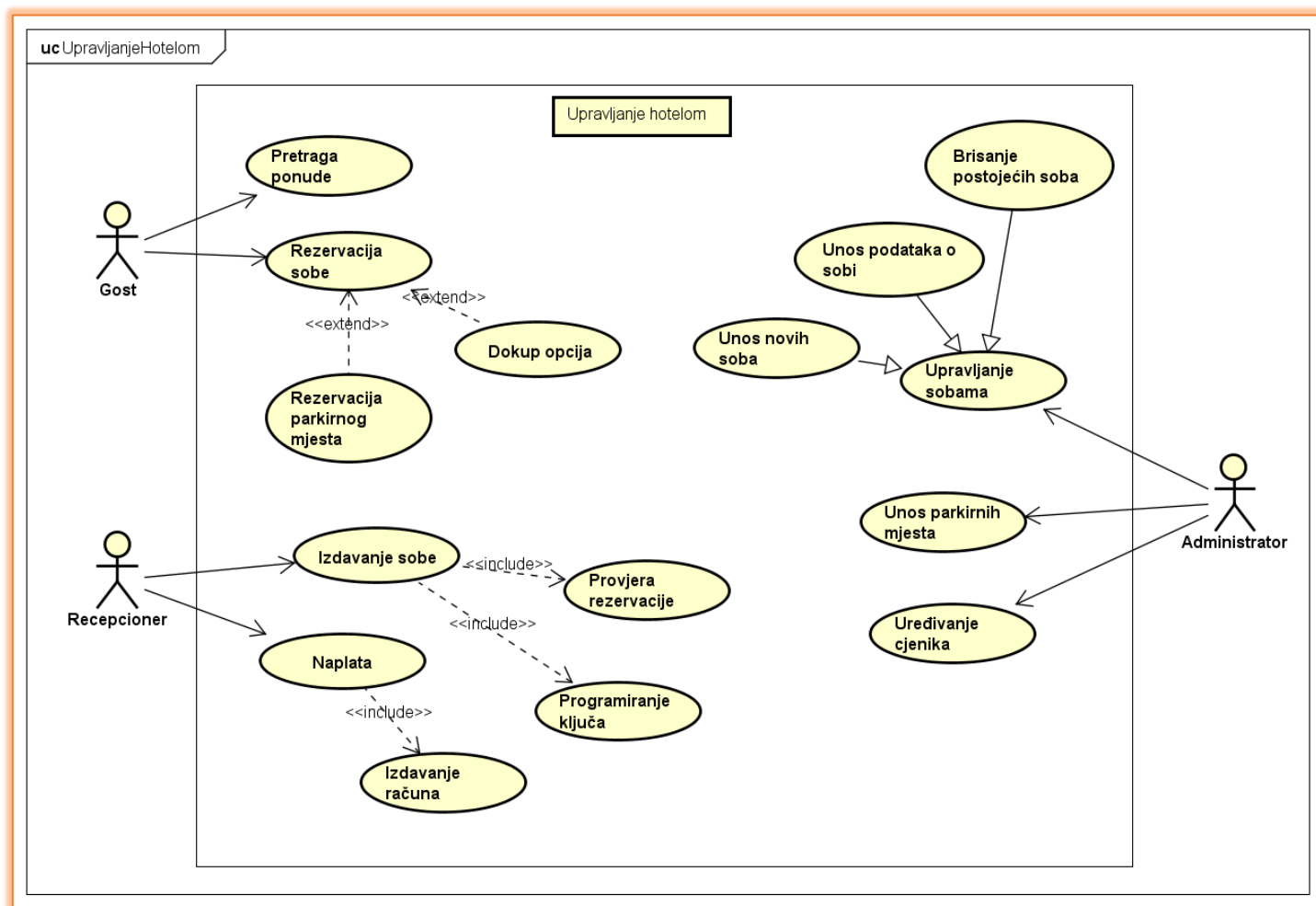
varirati od kategorije do kategorije. Svi oglasi su dostupni javno bez potrebe za registracijom. Oglasi u pojedinim kategorijama se sortiraju prema datumu objave. Portal također ima i tražilicu koja mora omogućiti pretraživanje oglasa minimalno prema kategoriji oglasa i riječima iz naslova. Također, prilikom pregledavanja oglasa treba omogućiti opciju pregleda svih oglasa od nekog oglašivača.

Kako bi objavili svoj oglas, potencijalni oglašivači se moraju registrirati na portalu koristeći online formu. Registrirati se mogu sve pravne i punoljetne fizičke osobe. Potrebni podaci za fizičku osobu su: OIB, ime, prezime, adresa, e-mail i kontakt telefon. Poslovni subjekt također mora navesti svoj OIB te još naziv tvrtke, sjedište tvrtke te e-mail i kontakt telefon. Sama registracija je besplatna no moguće je i plaćati mjesečnu članarinu ukoliko se želi ostvariti dodatne usluge poput bolje pozicije oglasa. Predviđene su tri kategorije članarine: Standard, Optimum i Premium s raznim pogodnostima. Oglašavanje u svim kategorijama osim Auto-moto i Nekretnine je besplatno. Za oglašavanje u prethodno spomenutim kategorijama korisnik mora mjesečno plaćati iznos minimalne članarine Standard. Uplata članarine se obavlja putem aplikacije, a za provedbu transakcije aplikacija se spaja na vanjski servis za plaćanje. Uplatu članarine korisnik može obaviti na svom profilu ili prilikom objave oglasa ako se oglašava u kategoriji za koju je nužno imati plaćenu članarinu.

Registrirani korisnici i administratori mogu se prijaviti u sustav pomoću korisničkog imena i lozinke. Korisnici imaju pravo upravljati svojim profilom što podrazumijeva uređivanje i brisanje profila. Također imaju pravo upravljati svojim oglasima: objavljivati nove oglase te uređivati i brisati postojeće. Prilikom objave oglasa u kategoriji koje se plaća, korisnik treba imati mogućnost uplate članarine ako je već ranije nije uplatio. Administratori mogu pregledati sve korisnike i pritom ukloniti korisnika te ukloniti oglas.

➤ Rješenja

1.1. Upravljanje hotelom



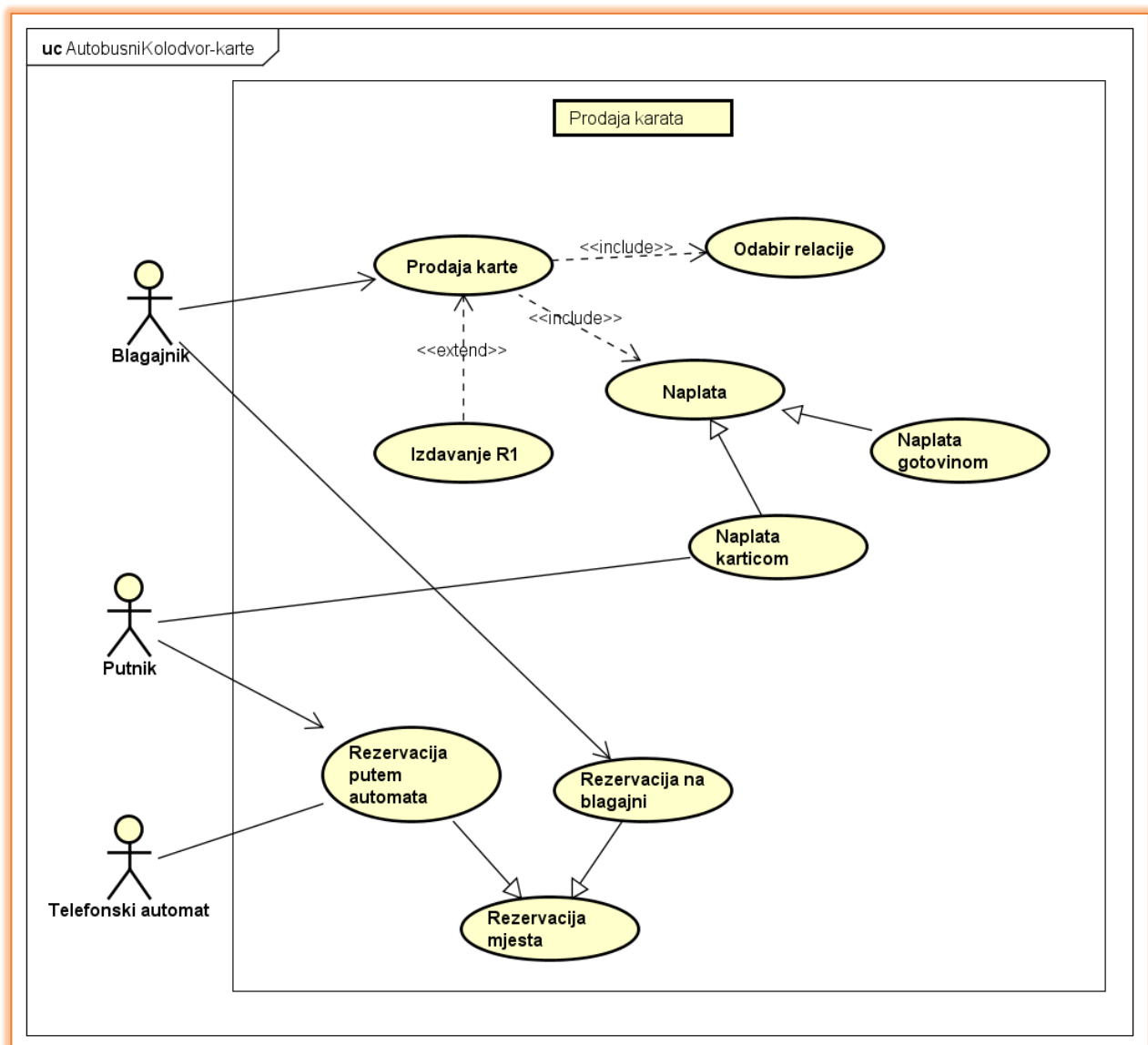
Komentar:

Rezervacija parkirnog mjesta i dokup opcija se ne moraju nužno izvršiti prilikom rezervacije sobe i stoga su ti obrasci povezani vezom „extend“ s glavnim obrascem. Nasuprot tome, prilikom izdavanja sobe nužno je provjeriti rezervaciju pa je s toga odabrana veza „include“.

Dokup opcija doručka, polupansiona i punog pansiona su objedinjene u jedan obrazac što znači da bi prilikom implementacije trebale biti ponuđene kao popis opcija iz kojeg korisnik odabire (naznačava) opcije koje želi.

Ovaj dijagram je moguće razlomiti u 3 manja dijagrama pri čemu svaki manji dijagram prikazuje funkcionalnosti jednog od aktora.

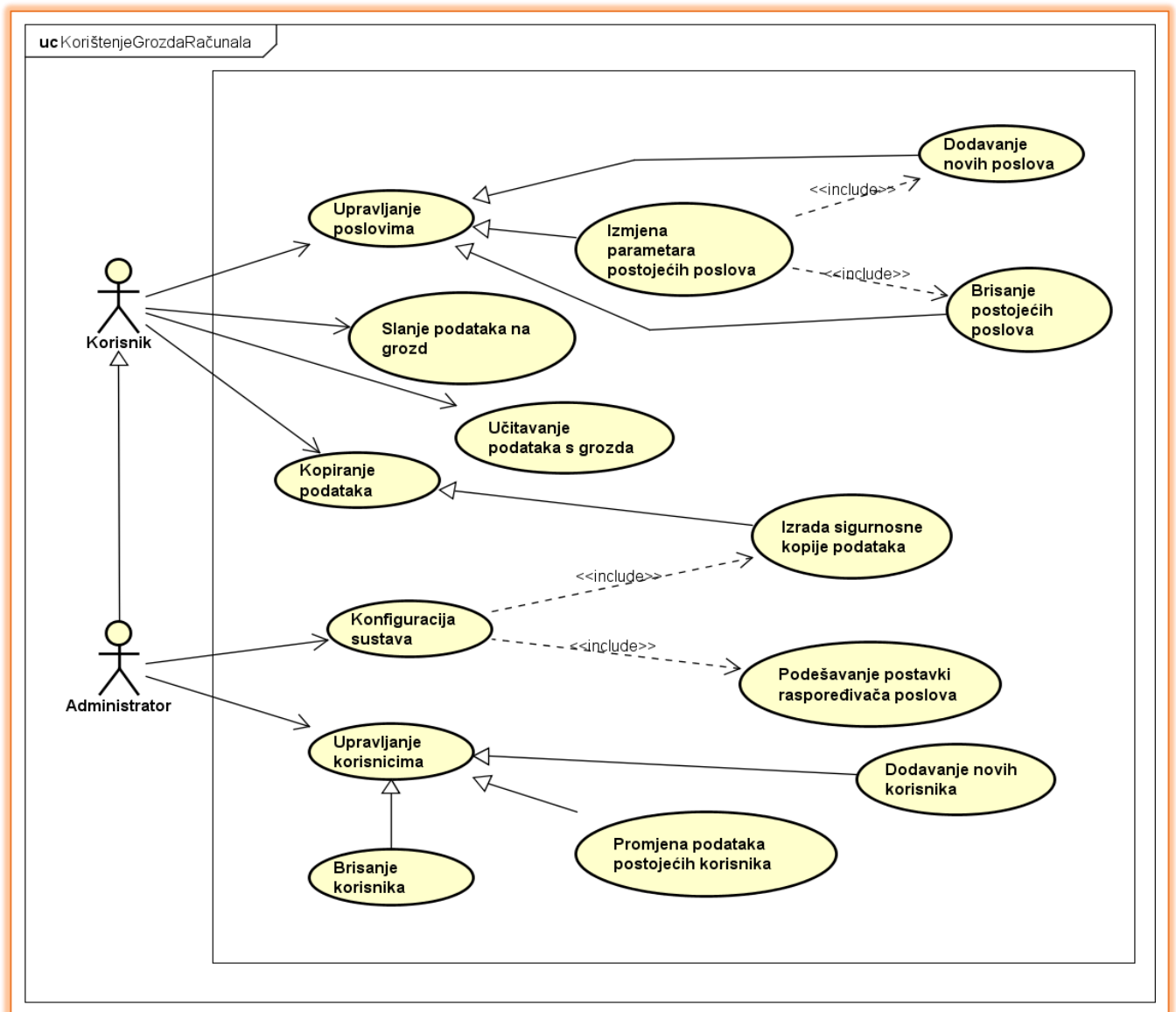
1.2. Prodaja autobusnih karata



Komentar:

„Unos PIN-a“ nije prikazan kao obrazac uporabe jer obuhvaća premali dio funkcionalnosti. Ta funkcionalnost je modelirana kroz naznaku sudjelovanja putnika u obrascu uporabe „Naplata karticom“ (dvosmjerna asocijacija). Ako bi se htjelo prikazati „Unos PIN-a“ kao zasebni obrazac uporabe, on bio onda bio povezan s obrascem „Naplata karticom“ vezom „include“.

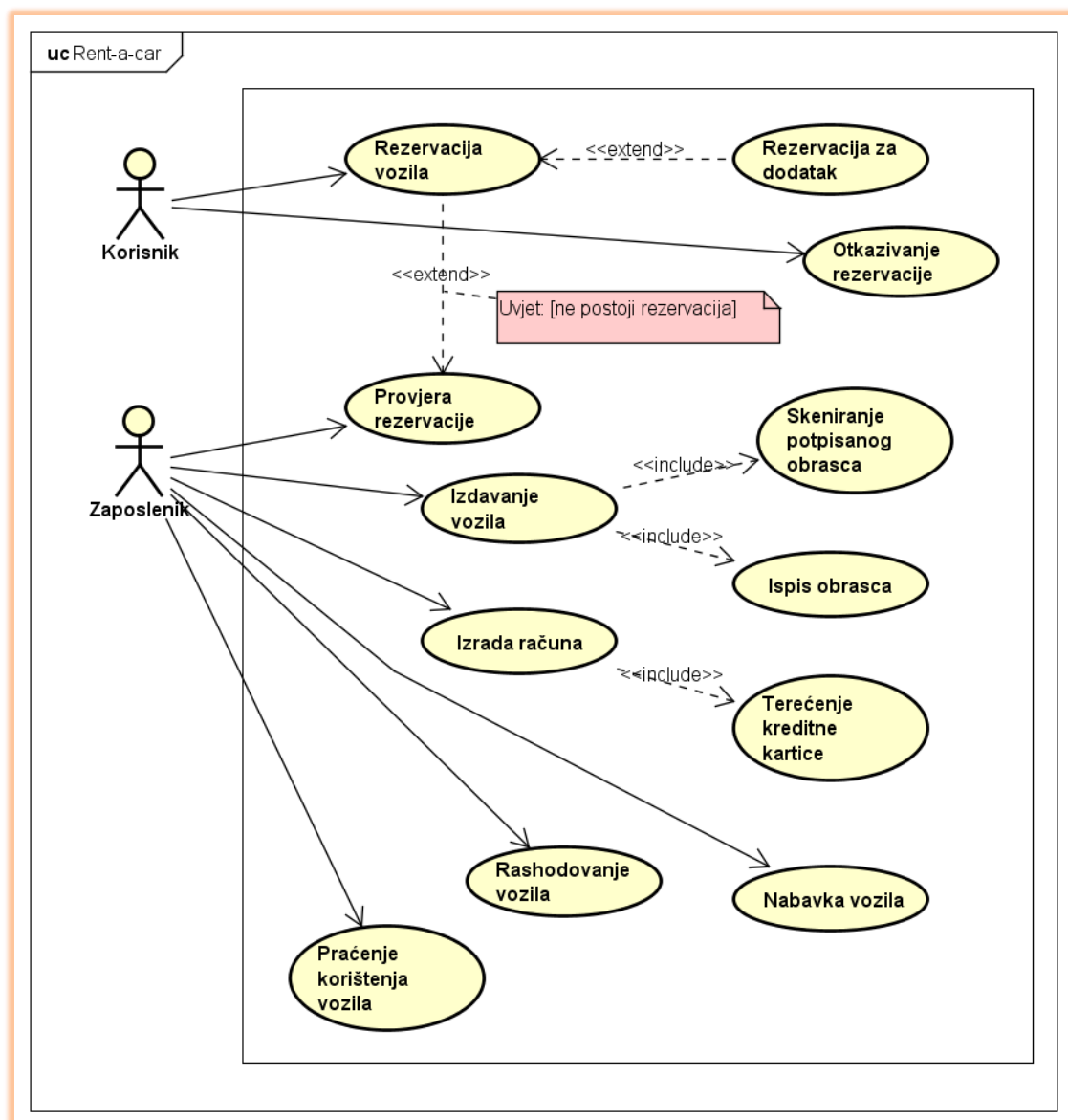
1.3. Grozd računala



Komentar:

Iako se obrazac uporabe „Upravljanje poslovima“ izričito ne spominje u tekstu, obrasci vezani uz stvaranje, uređivanje i brisanje (tzv. CRUD operacije) objekta se najčešće prikazuju kao specijalizacija obrasca Upravljanje.

1.4. Rent-a-car kompanija



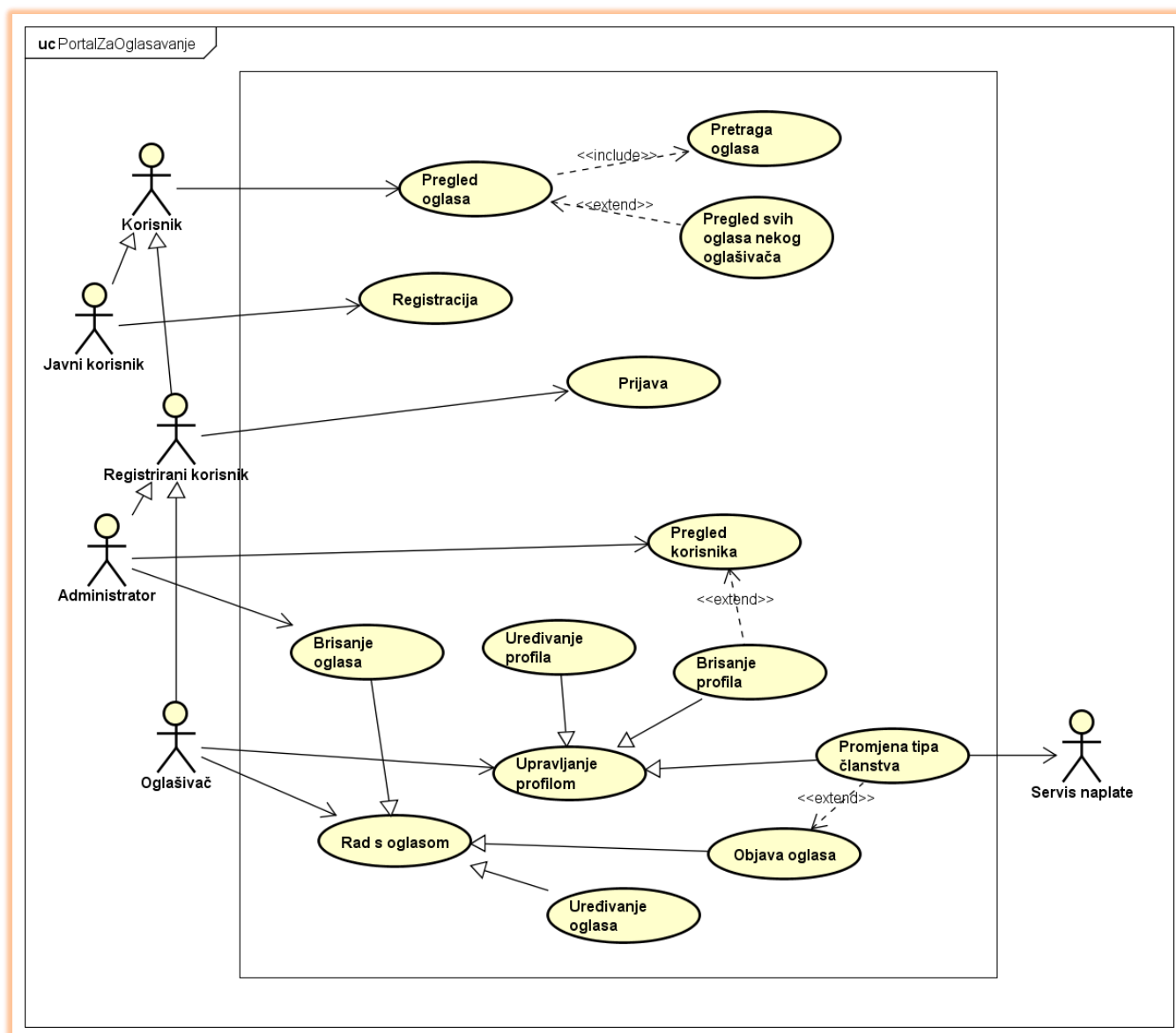
Komentar:

Unos broja kreditne kartice pri rezervaciji vozila nije potrebno izričito crtati kao uključeni obrazac uporabe budući da se radi o premaloj količini funkcionalnosti (jedno polje u web formi).

Napomena da korisnici mogu napraviti rezervaciju „samo u slučaju da u sustavu za zatražen vremenski period postoje slobodna vozila“ je ograničenje na obrazac Rezervacija vozila i na dijagramu se ne prikazuje.

Primopredaja automobila (ključevi, vozačka itd.) se ne prikazuje na dijagramu jer nije funkcionalnost programskog sustava, a dijagram u ovom slučaju opisuje programski sustav.

1.5. Portal za oglašavanje



Komentar:

Obrazac „Pretraga oglasa“ je povezan vezom „include“ s obrascem „Pregled oglasa“ budući da osnovni obrazac nikad ne prikazuje sve oglase nego uvijek radi pretragu (filtriranje) oglasa minimalno prema odabranoj kategoriji.

Obrazac „Promjena tipa članstva“ je vrsta upravljanja profilom (veza generalizacije prema uređivanju profila). Ova funkcionalnost također mora biti dostupna i prilikom objave oglasa (veza „extend“ na obrazac „Objava oglasa“) tako da korisnik može uplatiti članarinu prilikom objave oglasa ukoliko želi objaviti u kategoriji koja se plaća, a nema već uplaćenu članarinu.

2. Dijagrami razreda

➤ Zadaci

2.1. Web trgovina

Modelirajte pomoću dijagrama razreda sustav narudžbe i plaćanja u web trgovini zadan sljedećim opisom.

U web trgovini mogu kupovati registrirani i neregistrirani korisnici. Korisnici najprije ubacuju željene artikle u košaricu tako što odabiru artikl i količinu, čime u košarici nastaje stavka. Za svaki artikl naveden je naziv, opis i cijena po jediničnoj količini. Moguće je ubačeni artikl izbaciti ili mu promijeniti količinu. U svakom trenutku moguće je vidjeti ukupnu cijenu svih artikala u košarici. Kada je korisnik gotov s ubacivanjem svih artikala stvara narudžbu. Pri izradi narudžbe izračunava se ukupan iznos za plaćanje, te zapisuje datum i vrijeme, adresa isporuke, kontakt telefon i adresa e-pošte. Također prilikom stvaranja narudžbe odabire se način plaćanja: bankovnom karticom ili gotovinom (pouzećem). Kod plaćanja karticom potrebno je upisati podatke o kartici, a plaćanje se nakon potvrde narudžbe izvršava putem vanjskog servisa za plaćanje. Za narudžbu se prati njen status koji može biti: „u tijeku“ – prije potvrde narudžbe, „poslana“ – nakon potvrde do isporučenja, „isporučena“ – nakon što je narudžba uspješno isporučena. Za registrirane korisnike u sustavu već postoje podaci o kontakt telefonu, adresi e-pošte i adresi isporuke te je prilikom stvaranja narudžbe potrebno te podatke unaprijed dohvatiti i ostaviti korisniku mogućnost da ih ispravi po želji. Također, za registrirane korisnike se pamte sve ostvarene narudžbe i oni ih po želji mogu ponoviti. Narudžba se prije nego što je potvrđena može i izbrisati, tj. korisnik može odustati od narudžbe no artikli i dalje ostaju u košarici.

2.2. Tvrtka za popravak informatičke opreme

Modelirajte pomoću dijagrama razreda aplikaciju koja pomaže tvrtki koja se bavi popravkom informatičke opreme praćenje popravaka.

U tvrtki je zaposleno više serviseri i jedan vođa. Oni su djelatnici tvrtke čiji se podaci brišu njezinim zatvaranjem. Za svakog djelatnika zapisani su sljedeći podaci: ime, prezime, OIB. Vođa otvara radne naloge, a na svakom radnom nalogu radi najmanje jedan serviser. Nakon otvaranja radnog naloga vođa inicijalno procjenjuje trajanje radnog naloga i upisuje taj podatak u radni nalog. Jednom radnom nalogu pridružena je jedna stranka u čije ime je radni nalog otvoren. Za svaku stranku evidentira se: ime, prezime i broj telefona.

Jedan radni nalog može se sastojati od jednog ili više dodatnih radnih naloga ovisno o složenosti popravaka. Status radnog naloga može biti: „*zaprimljen*“, „*u obradi*“, „*završen – uspješno*“ i „*završen – neuspješno*“. Svaki radni nalog sadržava barem jednu servisnu akciju. Za svaku servisnu akciju evidentira se broj utrošenih čovjek-sati i cijena čovjek-sata i prema tome se izračunava cijena popravka za radni nalog. Akcije se dijele na: dijagnostiku, podešavanje i ugradnju. Ugradnja sadržava barem jednu računalnu komponentu koja se ugrađuje tijekom akcije. Svaka komponenta ima svoju cijenu.

2.3. Program za crtanje

Modelirajte elemente jednostavnog programa za crtanje prema sljedećem opisu.

Osnovni element programa za crtanje je radna površina. Na radnoj površini mogu se crtati različiti oblici: linije, poligoni i elipsoidi. Za svaki oblik je definirana RGB boja (boja koja se sastoji od R, G i B komponente). Linija je definirana svojom početnom i završnom točkom. Poligon je definiran skupom linija (barem tri) koje predstavljaju njegove stranice. Elipsoid je definiran skupom točaka koje omeđuju njegovu površinu. Točke su definirane koordinatama x i y . Za sve oblike moguće su operacije transformacije: rotacija, translacija, smanjenje i povećanje. Elipsoide i poligone je moguće animirati. Animacija se definira dinamički, a moguće je odabrati između dvije vrste: *FadeIn* i *FadeOut*. Obje vrste animacije rade na isti način i za elipsoide i za poligone.

2.4. Programska potpora za licenciranje

Modelirajte dijagramom razreda programsku potporu za licenciranje koja je opisana u nastavku.

Programska potpora za licenciranje omogućava prodavaču licenciranje pojedinačnog programskog proizvoda, dijelova proizvoda i cijele linije proizvoda. Programski proizvod se sastoji od jednog ili više dijelova (npr. prevoditelji za programske jezike, programski moduli za neku funkcionalnost i sl). Linija proizvoda obuhvaća skup proizvoda srodnih funkcionalnosti i sve njihove dijelove. Proizvodi mogu pripadati samo jednoj liniji, a dijelovi proizvoda mogu istovremeno biti dio više linija. Proizvodi iz iste linije mogu imati neke iste dijelove. Za svaku liniju proizvoda definiran je osnovni proizvod za koji kupac obavezno mora imati licencu ako želi koristiti bilo što iz te linije. Kupac može kupiti licencu za proizvode i dijelove proizvoda, a istu licencu ima samo jedan kupac. Licenca sadrži ključ koji dobiva kupac i koji može biti alfanumerički kod ili fizički USB ključ. Na osnovu vrste ključa prilikom kupnje licence generira se alfanumerički kod ili vrši programiranje USB ključa.

Kupci se dijele na privatne i poslovne te svaki kupac ima svoj identifikacijski broj. Za privatne osobe se pamte se ime, prezime, adresa e-pošte i kontakt telefon. Za poslovne osobe pamte se

naziv tvrtke, adresa e-pošte i kontakt telefon. Za svakog kupca se definira adresa isporuke robe i adresa isporuke računa koje sadrže naziv ulice i kućni broj, poštanski broj, grad i državu.

2.5. Portal za oglašavanje

Modelirajte pomoću dijagrama razreda portal za oglašavanje.

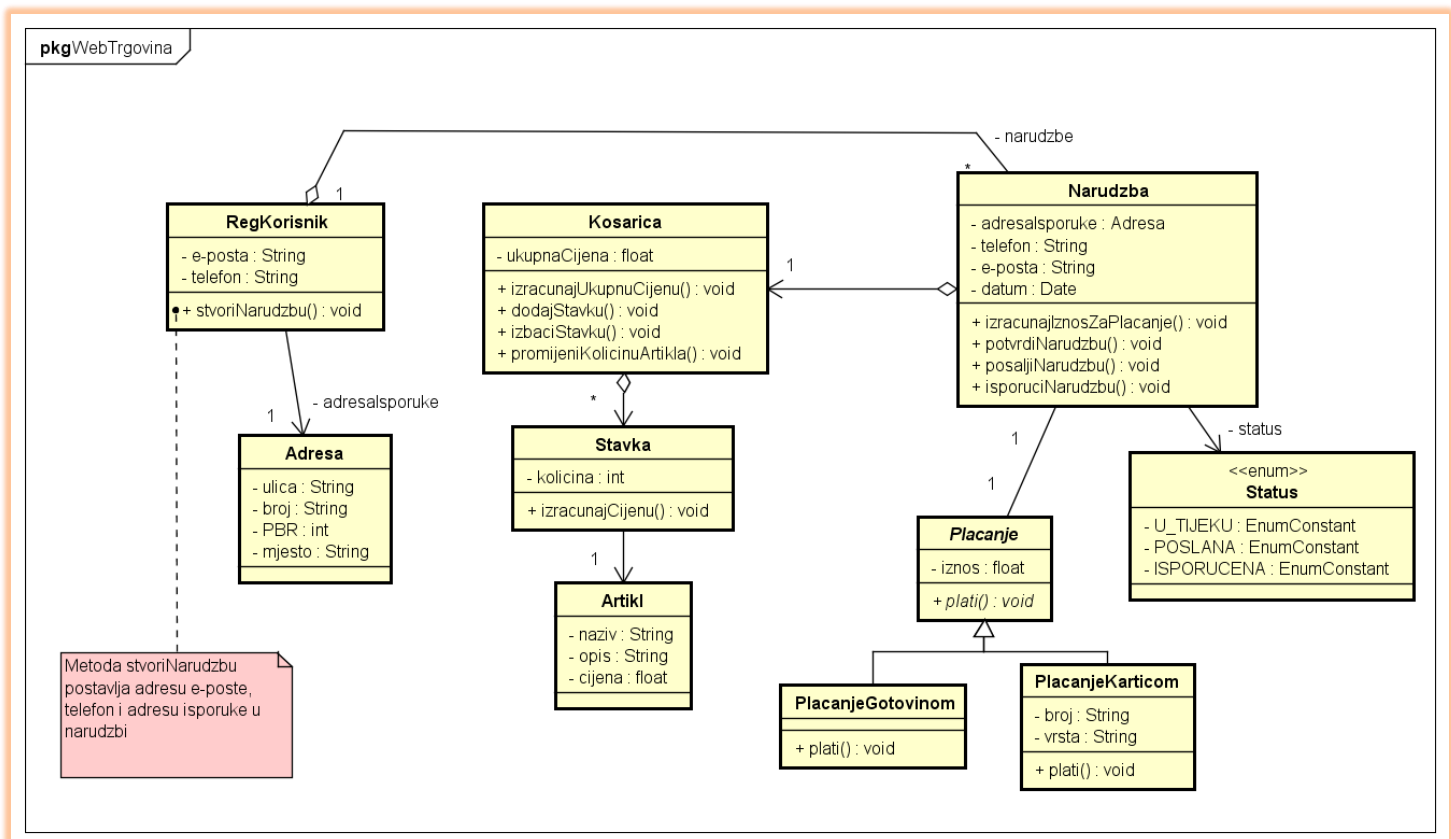
Oglašavati se mogu pokretne, nekretnine i usluge. Oglasi su podijeljeni u kategorije: Nekretnine, Auto-moto, Roba i Usluge. Svaki oglas sadržava naslov, opis, cijenu usluge, naziv oglašivača i kontakt te može sadržavati slike i kratke video materijale, dok ostali podaci mogu varirati od kategorije do kategorije, npr. za nekretnine treba navesti površinu nekretnine. Svi oglasi su dostupni javno bez potrebe za registracijom. Oglasi u pojedinim kategorijama se sortiraju prema datumu objave. Portal također ima i tražilicu koja mora omogućiti pretraživanje oglasa minimalno prema kategoriji oglasa i riječima iz naslova. Također, prilikom pregledavanja oglasa treba omogućiti opciju pregleda svih oglasa od nekog oglašivača.

Kako bi objavili svoj oglas, potencijalni oglašivači se moraju registrirati na portalu koristeći online formu. Registrirati se mogu sve pravne i punoljetne fizičke osobe. Potrebni podaci za fizičku osobu su: OIB, ime, prezime, adresa, e-mail i kontakt telefon. Poslovni subjekt također mora navesti svoj OIB te još naziv tvrtke, sjedište tvrtke te e-mail i kontakt telefon. Sama registracija je besplatna no moguće je i plaćati mjesečnu članarinu ukoliko se želi ostvariti dodatne usluge poput bolje pozicije oglasa. Predviđene su tri kategorije članarine: Standard, Optimum i Premium s raznim pogodnostima. Oglašavanje u svim kategorijama osim Auto-moto i Nekretnine je besplatno. Za oglašavanje u prethodno spomenutim kategorijama korisnik mora mjesečno plaćati iznos minimalne članarine Standard. Uplata članarine se obavlja putem aplikacije, a za provedbu transakcije aplikacija se spaja na vanjski servis za plaćanje. Uplatu članarine korisnik može obaviti na svom profilu ili prilikom objave oglasa ako se oglašava u kategoriji za koju je nužno imati plaćenu članarinu.

Registrirani korisnici i administratori mogu se prijaviti u sustav pomoću korisničkog imena i lozinke. Korisnici imaju pravo upravljati svojim profilom što podrazumijeva uređivanje i brisanje profila. Također imaju pravo upravljati svojim oglasima: objavljivati nove oglase te uređivati i brisati postojeće. Prilikom objave oglasa u kategoriji koje se plaća, korisnik treba imati mogućnost uplate članarine ako je već ranije nije uplatio. Administratori mogu pregledati sve korisnike i pritom ukloniti korisnika te ukloniti oglas.

➤ Rješenja

2.1. Web trgovina



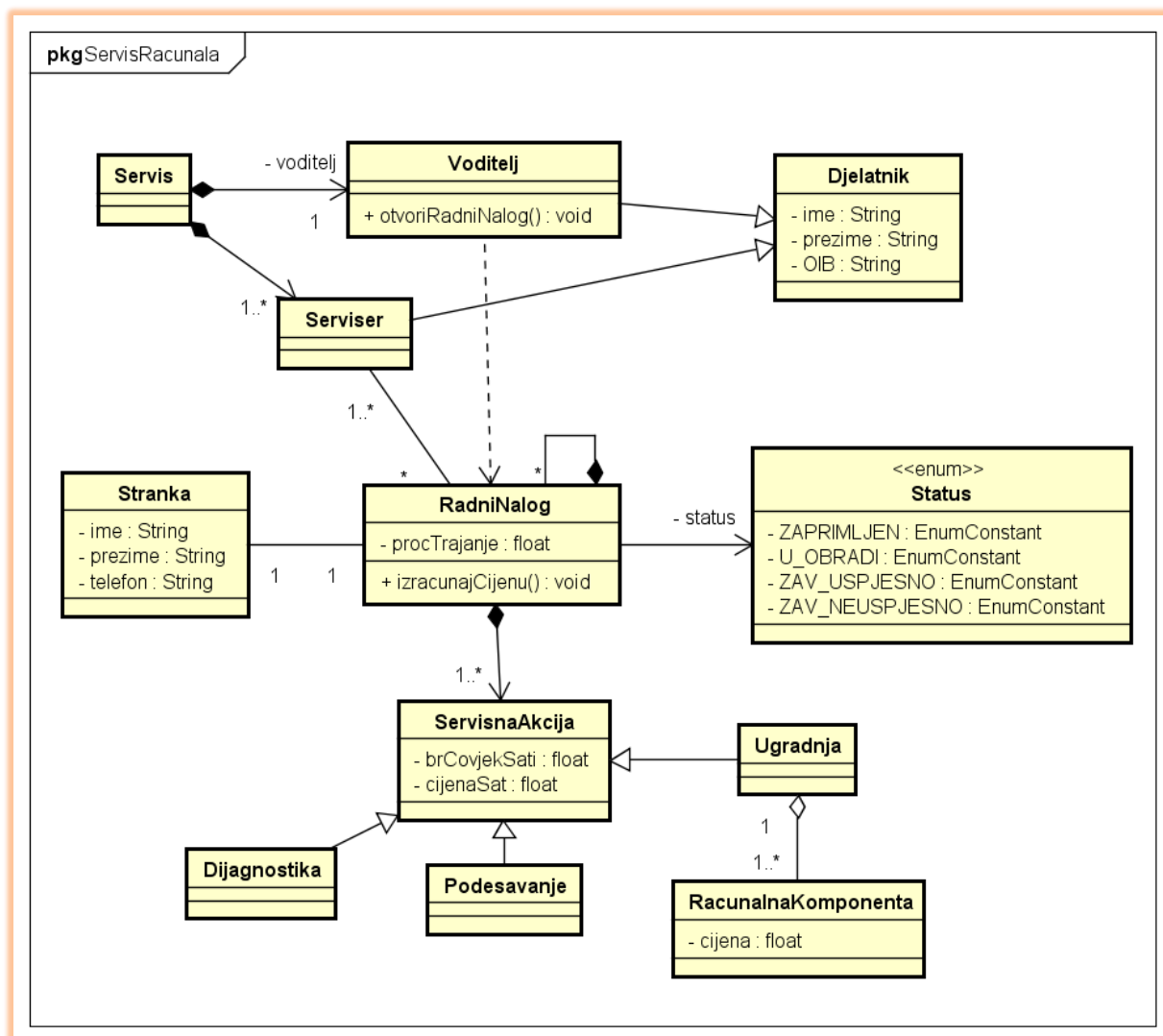
Komentar:

Atributi razreda mogu se navesti unutar samog razreda ili na vezi asocijacije (npr. `adresaIsporuke : Adresa`). Metoda `potvrdiNarudzbu` pokreće plaćanje, a metode `posaljiNarudzbu` i `isporuciNarudzbu` mijenjaju status Narudžbe. Razred `Placanje` je apstraktan te je nužno odabrati jednu od konkretnih implementacija plaćanja.

Metoda `stvoriNarudzbu()` razreda `Kupac` predstavlja odgovornost u domenskom modelu, međutim u konkretnoj implementaciji (npr. prema arhitekturi MVC), ova odgovornost može prijeći u upravljački sloj (Controller).

Posebno odustajanje od narudžbe nije potrebno modelirati jer narudžba jednostavno u tom slučaju neće biti pohranjena.

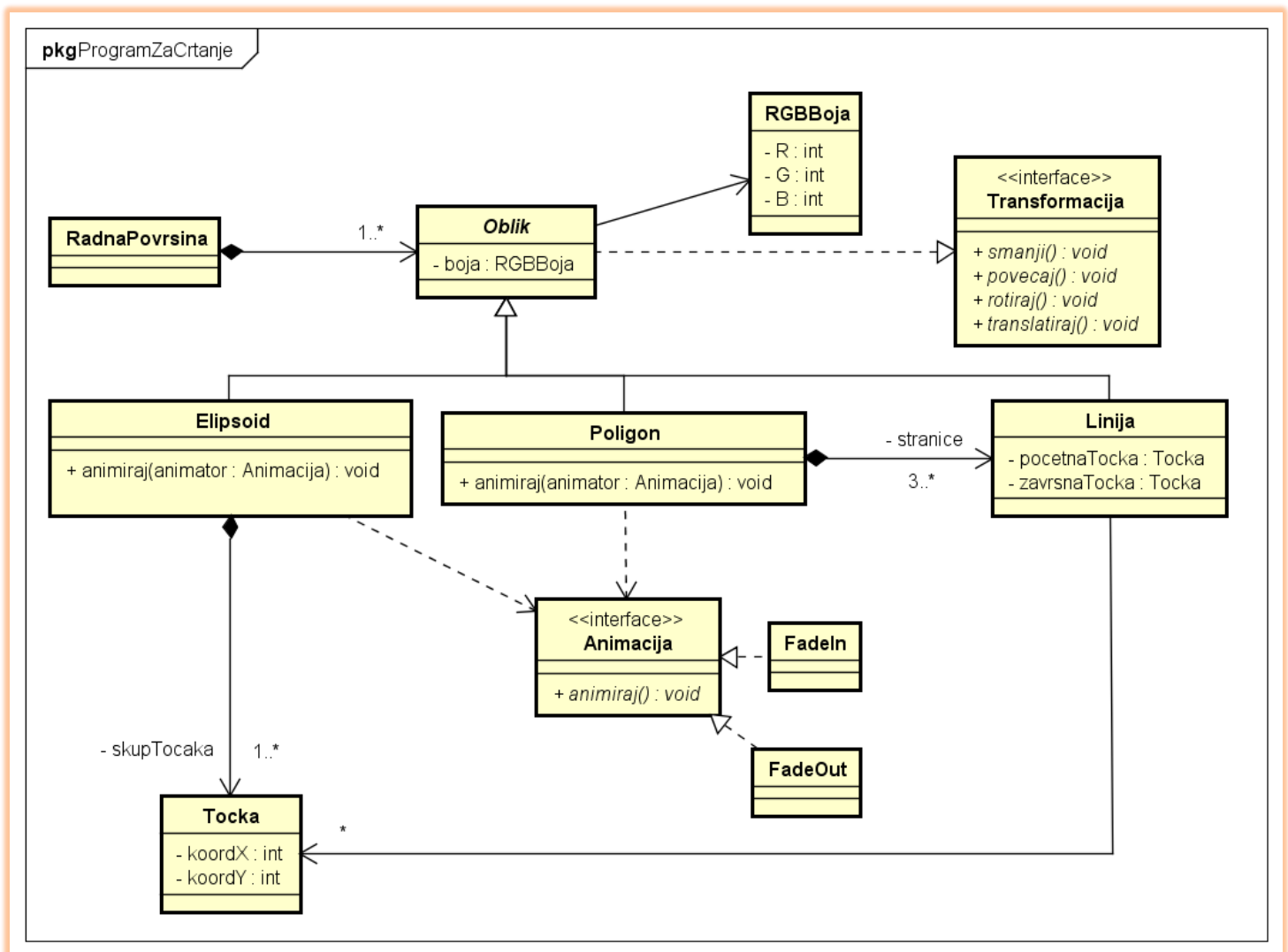
2.2. Tvrtka za popravak informatičke opreme



Komentar:

Voditelj stvara radni nalog putem metode `otvoriRadniNalog()` i stoga je razred `Voditelj` povezan s razredom `RadniNalog` vezom ovisnosti.

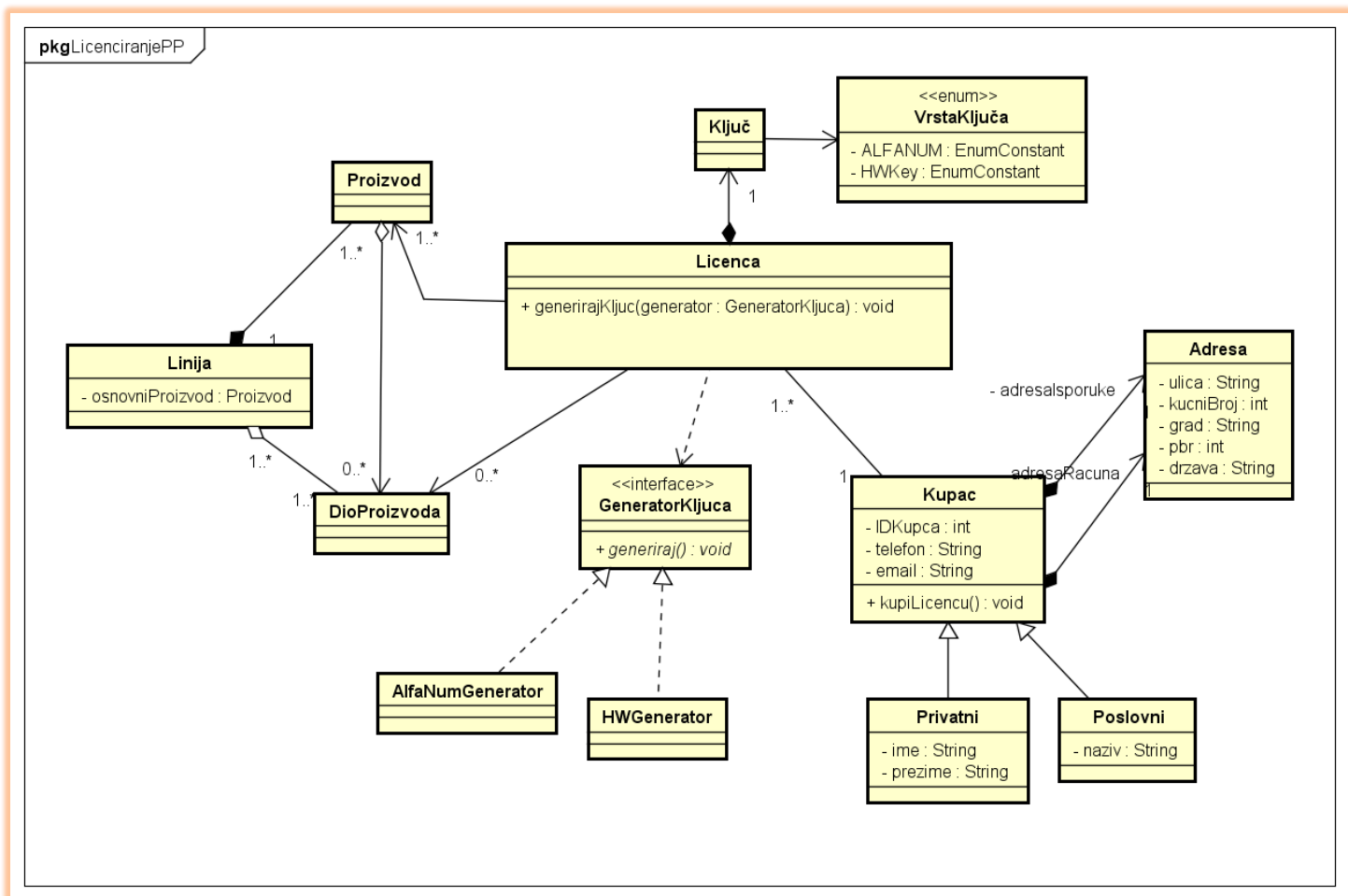
2.3. Program za crtanje



Komentar:

Budući da je za sve oblike zadano da se nad njima moraju moći izvesti su operacije transformacije (rotacija, translacija, smanjenje i povećanje) te operacije su izdvojene u sučelje *Transformacija*, a svaki konkretni razred koji nasljeđuje razred *Oblik* će implementirati način primjene tih operacija nad sobom. U slučaju animacije, zadane su dvije vrste animacije koje rade na isti način i za elipsoide i za poligone te je stoga konkretna animacija implementirana u razredima *FadeIn* i *FadeOut*, a razredi *Elipsoid* i *Poligon* koriste tu implementaciju pozivajući metodu *animiraj* iz sučelja *Animacija* te su zato povezani vezom ovisnosti sa sučeljem *Animacija*.

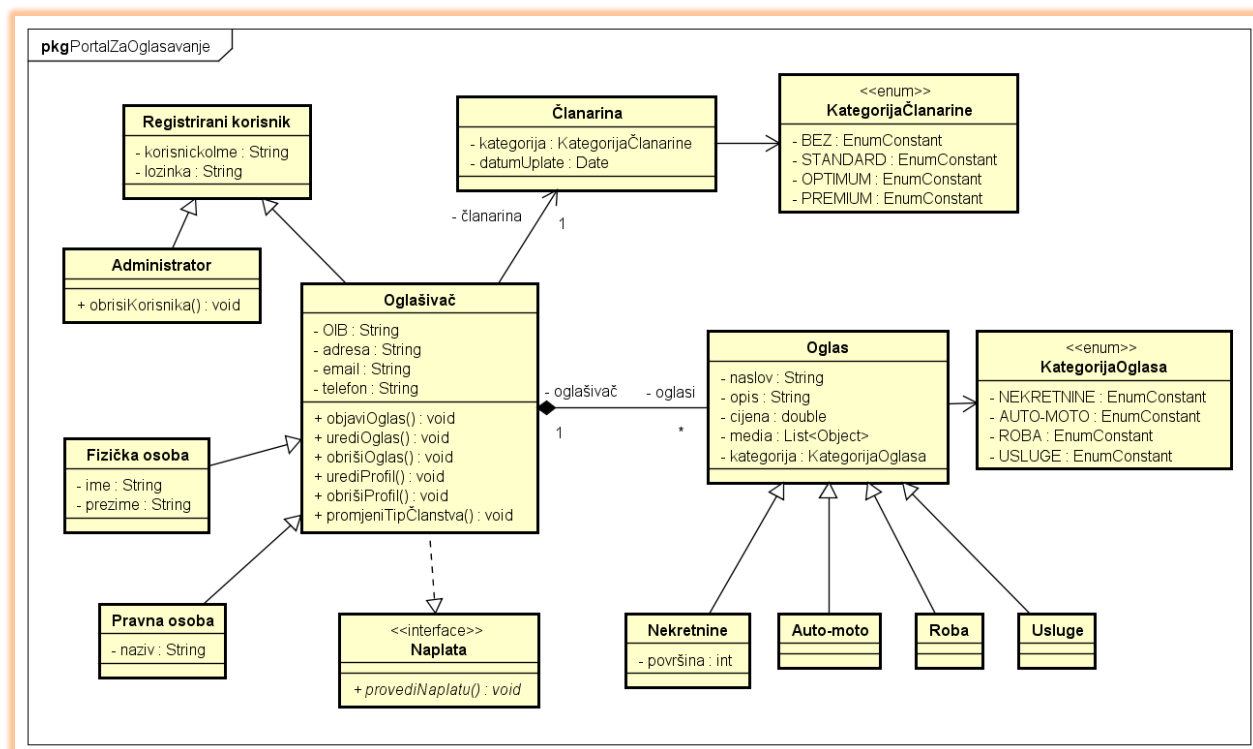
2.4. Programska potpora za licenciranje



Komentar:

Budući da adresa sadrži nekoliko elemenata, nije ih preporučljivo pisati kao jedan niz znakova (String) nego su izdvojeni u cjelinu kao zaseban razred. Metoda `kupiLicencu()` razreda `Kupac` predstavlja odgovornost u domenskom modelu, međutim u konkretnoj implementaciji (npr. prema arhitekturi MVC), ova odgovornost može prijeći u upravljački sloj (Controller). `Licenca` sadrži ključ koji dobiva kupac i koji može biti alfanumerički kod ili fizički USB ključ. Kako se način generiranja ključa razlikuje ovisno o vrsti ključa, ostvarene su dvije konkretne implementacije sučelja `GeneratorKljuca`: `AlfaNumGenerator` i `HWGenerator`.

2.5. Portal za oglašavanje



Komentar:

Metode razreda oglašivač i Administrator predstavljaju odgovornosti u domenskom modelu, međutim u konkretnoj implementaciji (npr. prema arhitekturi MVC), ova odgovornost može prijeći u upravljački sloj (Controller). Rješenje koje ne sadrži enumeraciju kategorije oglasa bi također bilo prihvatljivo, no s obzirom da o vrsti oglasa ovisi hoće li oglašivač morati platiti članarinu ili ne, s implementacijske strane je elegantnije rješenje u kojem enumeracija postoji. Kategorije članarine su istaknute samo kao enumeracija kategorijačlanarine, budući da razred Članarina ne mijenja attribute niti odgovornosti ovisno o tipu članarine te u tom slučaju nije nužno uvoditi nasljeđivanje.

3. Dijagrami stanja

➤ Zadaci

3.1. Oglas na portalu za oglašavanje

Modelirajte stanja oglasa na portalu za oglašavanje iz zadatka 1.5. i pri tome uzmite sljedeće napomene u obzir.

Oglas je nakon objave aktivan. Oglašivač ga može po želji deaktivirati i reaktivirati, a može ga i obrisati. Dok je oglas aktivan broje se pregledi tog oglasa. Kada se oglas deaktivira, on se više ne prikazuje na portalu i vidljiv je samo oglašivaču koji ga je stvorio i koji ga može reaktivirati. Prilikom reaktivacije oglasa, broj pregleda se resetira na nulu. Kada se oglas stvori ili reaktivira, prvih 24 sata kraj njega stoji status „Novo!“. Ukoliko je oglas u kategoriji koja se plaća, a oglašivaču je istekla članarina, oglas će se automatski deaktivirati. Oglašivač može svoj oglas obrisati i tada se on trajno uklanja iz sustava.

3.2. Tablični kalkulator

Modelirajte stanje ćelije u tabličnom kalkulatoru pomoću UML dijagrama stanja.

Ćelija u tabličnom kalkulatoru je nefokusirana sve dok se na nju ne klikne mišem ili pozicionira putem tipkovnice. Fokusiranjem ćelije podeblja se njezin okvir. Kada je ćelija fokusirana, moguć je unos podataka koji počinje unosom bilo kojeg znaka različitog od „Enter“, „Tab“ ili „Esc“ i traje sve dok se ne unese neki od tih znakova ili dok se ne pomakne fokus na neku drugu ćeliju klikom miša ili pozicioniranjem putem tipkovnice. Svaki novi uneseni znak se dodaje na kraj niza novog unosa. Prilikom unosa podataka u ćeliju, na zaslonu se prikazuju novo uneseni podaci, ali se zapamćeni sadržaj ćelije ne mijenja. Sadržaj ćelije se osvježava tek nakon što se unesu znakovi „Enter“ ili „Tab“ ili se pomakne fokus na neku drugu ćeliju klikom miša ili tipkovnicom. Ukoliko se unese znak „Esc“, sadržaj ćelije ostaje nepromijenjen tj. novi unos se odbacuje, ali ona i dalje ostaje fokusirana.

3.3. Izvođenje poslova na procesoru

Dijagramom stanja modelirajte izvođenje poslova na dvojezgrenom procesoru. Zasebno prikažite:

a) stanje posla na procesoru i b) stanje procesora.

Pretpostavite da svaki posao ima zabilježenu količinu priručne memorije koju je zauzeo i približnu količinu vremena potrebnog za izvođenje. Posao se na početku nalazi u priručnoj memoriji u stanju čekanja i ostaje u tom stanju sve dok se jedna procesorska jezgra ne oslobodi. Kad se jedna jezgra na procesoru oslobodi, posao prelazi u stanje razmatranja. U tom stanju, onaj posao koji zauzima najviše priručne memorije se odabire za izvršavanje. Svi ostali poslovi se vraćaju u stanje čekanja. Svaki posao pri izvršavanju na procesoru mora proći kroz tri stanja: 1) predobradu, u kojoj se posao optimira, 2) izvođenje i 3) zapis rezultata kada se rezultati zapisuju u glavnu memoriju te se postavlja signalna zastavica da je jezgra prazna.

Pretpostavite da jedna jezgra izvodi samo jedan posao u nekom trenutku te da se proces razmatranja, tj. odabira posla za izvršavanje izvodi na jezgri koja se oslobodila. Također pretpostavite da se u slučaju gašenja procesora svi poslovi koji su trenutno u izvršavanju prekidaju i brišu.

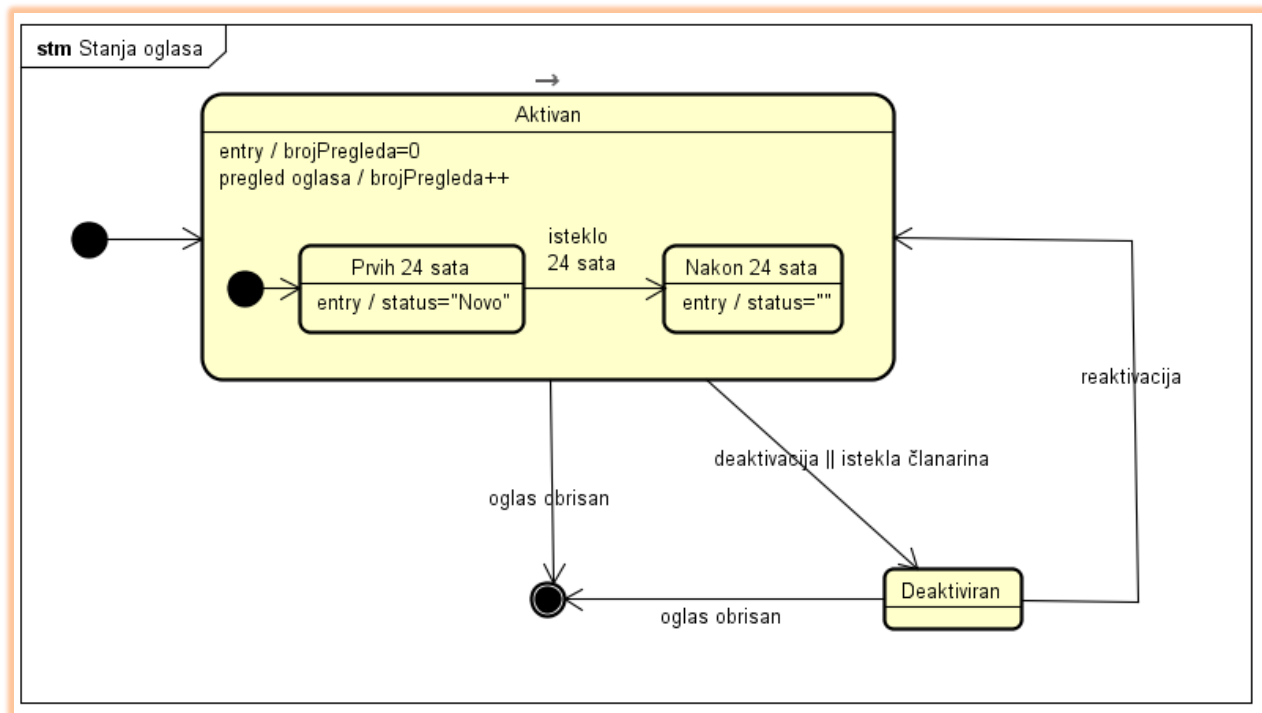
3.4. Korisničko sučelje web trgovine

Modelirajte stanja korisničkog sučelja prilikom stvaranje narudžbe u web trgovini iz zadatka 2.1 za registriranog i prijavljenog korisnika.

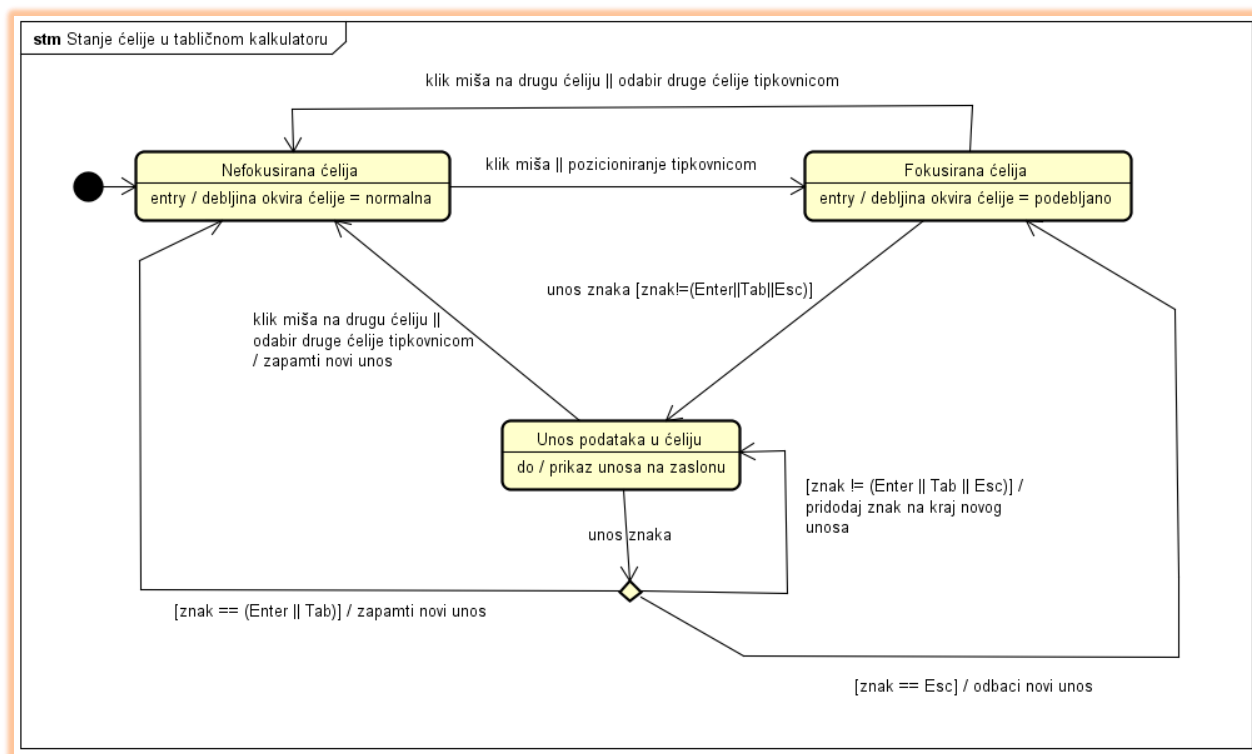
Nakon prijave u sustav korisniku se prikazuje katalog, a košarica je prazna. Korisnik započinje kupnju dodavanjem proizvoda iz kataloga u košaricu. U svakom trenutku korisnik može pregledati sadržaj košarice te ubačene proizvode izbaciti ili im promijeniti količinu. Prilikom pregleda košarice, korisnik može odabrati stvaranje narudžbe ako je gotov s ubacivanjem svih proizvoda ili može odabrati nastavak kupnje čime se vraća na katalog i može dodavati još proizvoda. Ukoliko odabere stvaranje narudžbe, prikazuje mu se forma za unos podataka za plaćanje. Nakon potvrde unosa podataka za plaćanje, omogućuje se unos podataka za dostavu, s time da se prvo dohvaćaju podaci o korisnikovoj adresi iz sustava, a zatim se ostavlja mogućnost da ih korisnik ispravi po želji. Kada korisnik potvrdi podatke za dostavu, narudžba je stvorena te se korisniku ispisuje obavijest o uspješnoj kupovini potvrda, košarica se prazni i vraća ga se na pregled kataloga. U svakom trenutku prije potvrde podataka za dostavu, korisnik može odustati od narudžbe čime se vraća na katalog, a svi proizvodi i dalje ostaju u košarici. Korisnik može odabrati i opciju pražnjenja košarice čime se uklanjaju svi proizvodi iz košarice. Ukoliko napusti web trgovinu (zatvori preglednik ili se odjavi) u bilo kojem trenutku prije potvrde podataka za dostavu, podaci o artiklima dodanim u košaricu se pamte i pri ponovnoj prijavi u aplikaciju korisniku se prikazuje stranica kataloga sa očuvanim sadržajem košarice.

➤ Rješenja

3.1. Oglas na portalu za oglašavanje



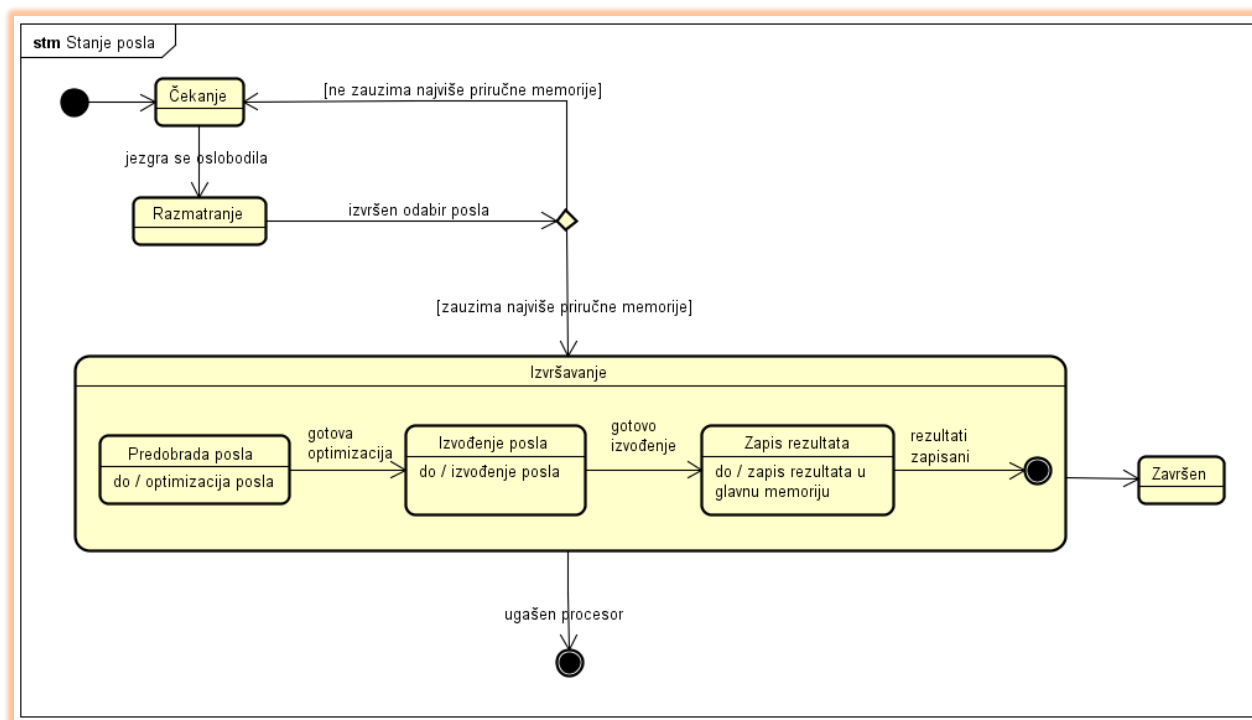
3.2. Tablični kalkulator



Komentar:

Na prijelazima koji su uzrokovani unosom znaka potrebno je razlikovati događaj (unos znaka) od provjere uvjeta koji je znak unesen (npr. je li unesen znak „Esc“). To je iz razloga što svi moderni radni okviri koji služe za razvoj korisničkog sučelja imaju metode za osluškivanje (engl. listener) unosa znaka (engl. key) na tipkovnici (bilo kojeg znaka) što predstavlja događaj, a zatim razvojni programer piše kôd u kojem se provjerava koji je znak unesen.

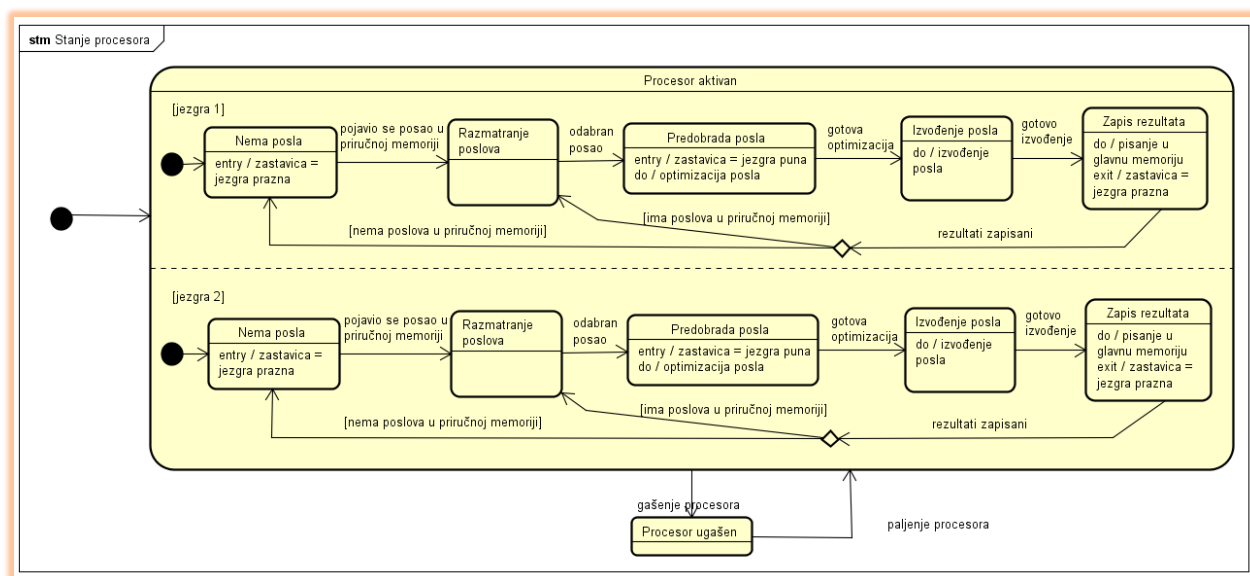
3.3. Izvođenje poslova na procesoru



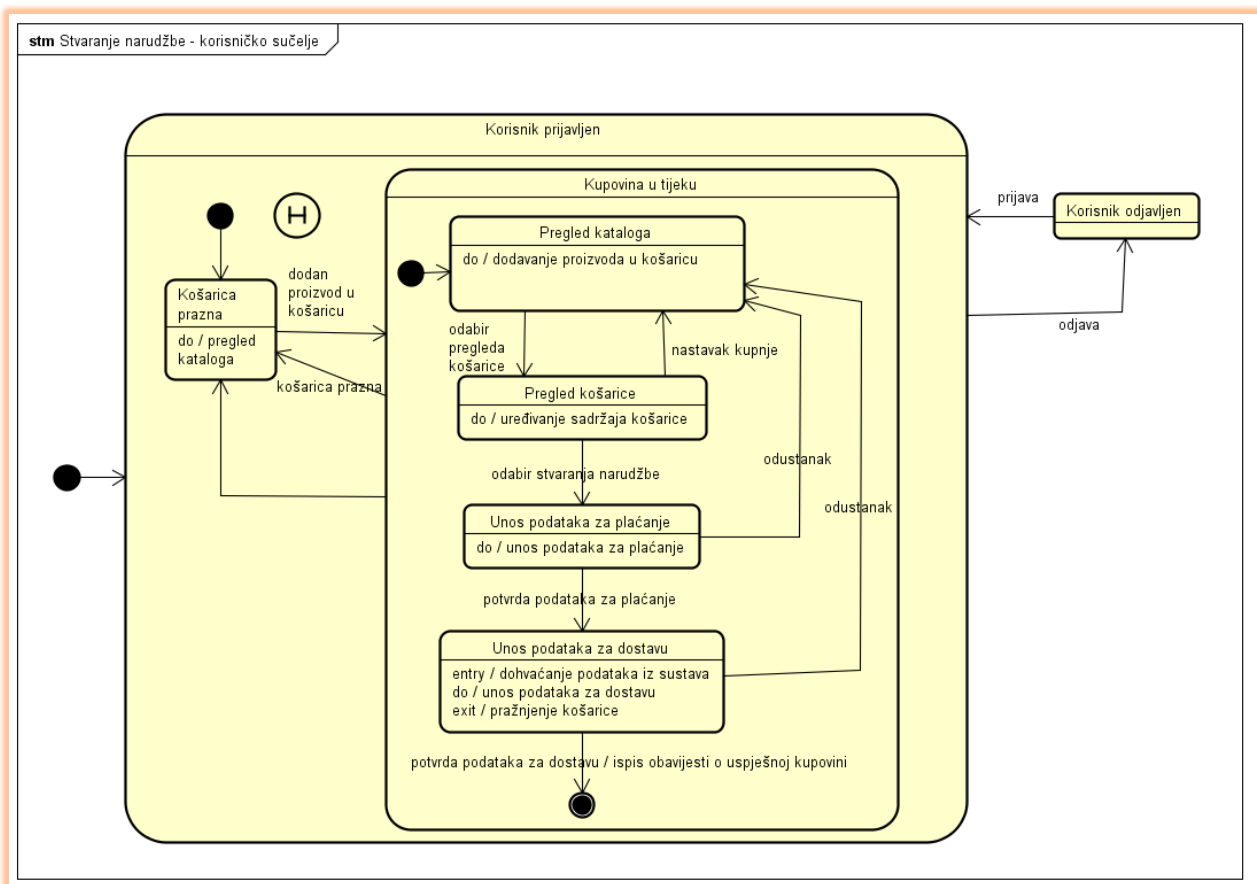
Komentar:

Na dijagramu stanja posla potrebno je razlikovati stanje kada je posao uspješno završen i kada je izvršavanje prekinuto te se prema naputku u zadatku posao briše čime nestaje instanca tog posla (čvor završnog stanja).

Na prijelazu između stanja „Izvršavanje“ i „Završen“ nije naveden događaj (okidač) zato što je „Izvršavanje“ složeno stanje unutar kojeg je eksplicitno naznačen završetak složenog stanja (nakon što su rezultati zapisani) te se podrazumijeva da se taj prijelaz automatski izvršava kada se složeno stanje završi.



3.4. Korisničko sučelje web trgovine



Komentar:

Budući da postoji pamćenje stanja košarice za korisnike koji su prijavljeni, u stanje „Korisnik prijavljen“ se uvode dva podstanja vezana uz stanje košarice: „Košarica prazna“ i „Kupovina u tijeku“ te pseudostanje plitke povijesti (engl. shallow history). U slučaju kada ne bi postojalo podstanje „Kupovina u tijeku“, nego bi stanja sadržana u njemu bila na istoj razini kao i pseudostanje plitke povijesti, onda bi se nakon ponovne prijave korisnik vraćao na zadnji korak u kojem je bio prije nego što se odjavio (npr. unos podataka za dostavu), a u zadatku je eksplicitno zadano da se vraća na pregled kataloga.

Na prijelazu između stanja „Kupovina u tijeku“ i „Košarica prazna“ nije naveden događaj (okidač) zato što je „Kupovina u tijeku“ složeno stanje unutar kojeg je eksplicitno naznačen završetak složenog stanja (potvrda podataka za dostavu) te se podrazumijeva da se taj prijelaz automatski izvršava kada se složeno stanje završi.

4. Dijagrami komponenti

➤ Zadaci

4.1. MVC

Modelirajte arhitekturni obrazac MVC (engl. *Model-View-Controller*). Komponenta Model sadrži sve podatke i poslovnu logiku, komponenta View je zadužena za prikaz podataka i interakciju s korisnikom, a komponenta Controller je posrednik između komponenti Model i View. Komponenta View poziva komponentu Controller preko sučelja IEvent. Komponenta Controller osvježava podatke koristeći sučelje IUpdate komponente Model. Nakon svake promjene podataka, komponenta Model obavještava komponentu View pozivajući metode sučelja INotify koje ostvaruje komponenta View. Osvježene podatke iz Modela komponenta View dobiva preko sučelja IData.

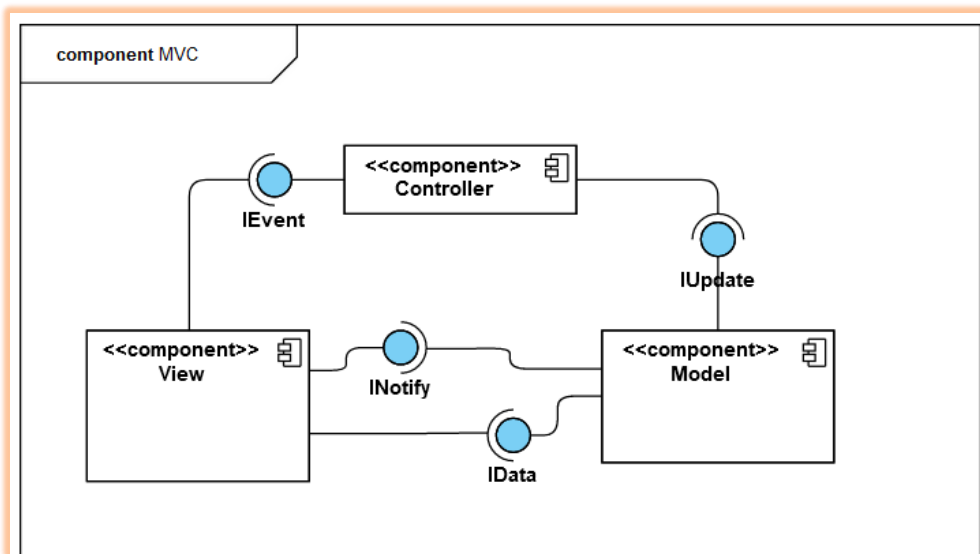
4.2. Portal za oglašavanje

Modelirajte web aplikaciju portala za oglašavanje iz zadatka 1.5. uz pretpostavku sljedećeg opisa strukture aplikacije.

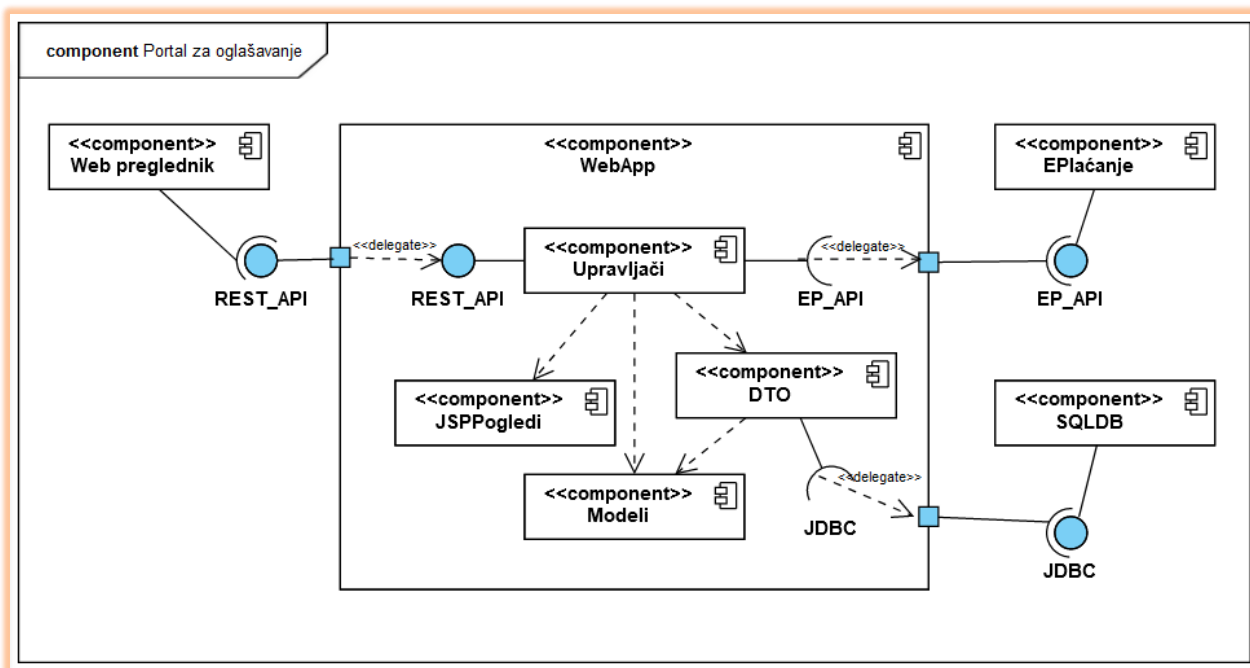
Web aplikacija se sastoji od 4 komponente: Upravljači, Modeli, JSPPogledi i DTO. Upravljači pružaju REST_API sučelje na koje vanjski web preglednik može slati zahtjeve i putem kojeg šalju JSPPoglede kao odgovor. Upravljači se također povezuju s vanjskom uslugom EPlaćanje putem sučelja EP_API. Upravljači koriste komponentu DTO za pristup SQL bazi podataka koja ostvaruje sučelje JDBC. DTO upisuju podatke iz baze podataka u komponentu Modeli, a Upravljači koriste Modele za manipulaciju podacima.

➤ Rješenja

4.1. MVC



4.2. Portal za oglašavanje



Komentar: Gledano iz perspektive komponenti Web preglednik, EPlaćanje i SQLDB, unutarnje komponente Web aplikacije i njihova sučelja su skriveni te sva interakcija s njima mora ići preko portova. Iz tog razloga komponenta WebApp mora imati vanjska sučelja REST_API, EP_API i JDBC koja se onda preko portova delegiraju na unutarnja sučelja komponenti. Za komponente za koje nije eksplicitno definirano sučelje, kao npr. u slučaju kad Upravljači koriste komponentu DTO, komponente se povezuje vezom ovisnosti.

5. Dijagrami razmještaja

➤ Zadaci

5.1. Klijent – poslužitelj

Modelirajte sustav u kojem klijent komunicira s poslužiteljem. Na klijentskoj strani je PC računalo s operacijskim sustavom Linux Debian na kojem je pokrenut web preglednik Mozilla Firefox. Klijent se protokolom HTTP spaja na poslužitelj. Na poslužitelju je instaliran operacijski sustav Linux CentOS. Poslužiteljska aplikacija pod nazivom WebAPP pokrenuta je unutar poslužiteljskog programa Apache HTTP.

Nacrtajte jedan specifikacijski dijagram razmještaja (engl. *Specification Level Deployment Diagram*) i jedan dijagram razmještaja instanci (engl. *Instance Level Deployment Diagram*).

5.2. Trorazinska arhitektura

Modelirajte specifikacijskim dijagramom razmještaja (engl. *Specification Level Deployment Diagram*) trorazinsku arhitekturu: klijent – poslužitelj - baza podataka. Na klijentskoj stranikoristi se PC računalo na kojem je pokrenuta aplikacija ClientAPP. Poslužiteljska aplikacija ServerAPP se nalazi na računalu s operacijskim sustavom RHEL. Baza podataka je implementirana kao usluga u oblaku Heroku Postgres. Klijent komunicira s poslužiteljem protokolom SSH, a poslužitelj komunicira s bazom protokolom HTTPS.

5.3. Vremenska prognoza

Modelirajte specifikacijskim dijagramom razmještaja (engl. *Specification Level Deployment Diagram*) informacijski sustav za davanje vremenske prognoze. Klijent se putem mobilne aplikacije MeteoAPP za Android OS spaja na web poslužitelj MeteoServer. Poslužiteljska web aplikacija MeteoWeb.war je pokrenuta pomoću poslužiteljskog programa Apache Tomcat na računalu na kojem je instaliran operacijski sustav WinServer 2019. Web poslužitelj dobavlja podatke od vanjskog servisa EUMeteo. U slučaju nedostupnosti servisa EUMeteo, web poslužitelj dobavlja informacija od servisa USMeteo. Za komunikaciju između klijenta i poslužitelja te poslužitelja i servisa se koristi protokol HTTP.

5.4. Web trgovina

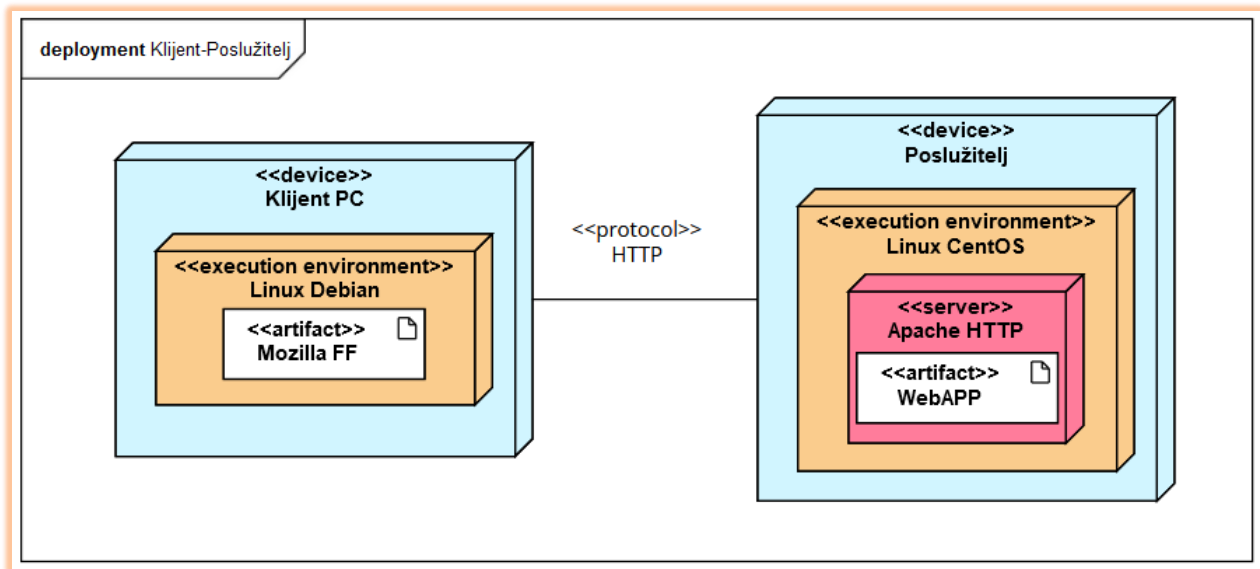
Modelirajte sustav web trgovine. Web aplikacija je ostvarena iz dva dijela: korisničkog sučelja (engl. *front-end*) i poslužiteljske aplikacije u pozadini (engl. *back-end*). Korisničko sučelje je implementirano u radnom okviru Angular i pokrenuto je na platformi Heroku. Poslužiteljska aplikacija je implementirana koristeći Node.js na platformi Firebase, a pokrenuta je unutar kontejnera Docker. Baza podataka, u koju poslužiteljska aplikacija sprema sve podatke, također se nalazi na platformi Firebase, izvodi se koristeći uslugu Firestore i sadrži sheme Korisnici, Proizvodi i Narudžbe. Klijent se putem web preglednika spaja na korisničko sučelje (front-end) preko kojeg komunicira s poslužiteljskom aplikacijom (back-end). Za komunikaciju se koristi protokol HTTPS.

Nacrtajte specifikacijski dijagram razmještaja (engl. *Specification Level Deployment Diagram*). Također nacrtajte i jedan dijagram razmještaja instanci (engl. *Instance Level Deployment Diagram*) na kojem ćete prikazati pristup jednog klijenta s PC računala i jednog klijenta s mobilnog uređaja te dvije instance kontejnera Docker koje se koriste radi balansiranja opterećenja.

➤ Rješenja

5.1. Klijent – poslužitelj

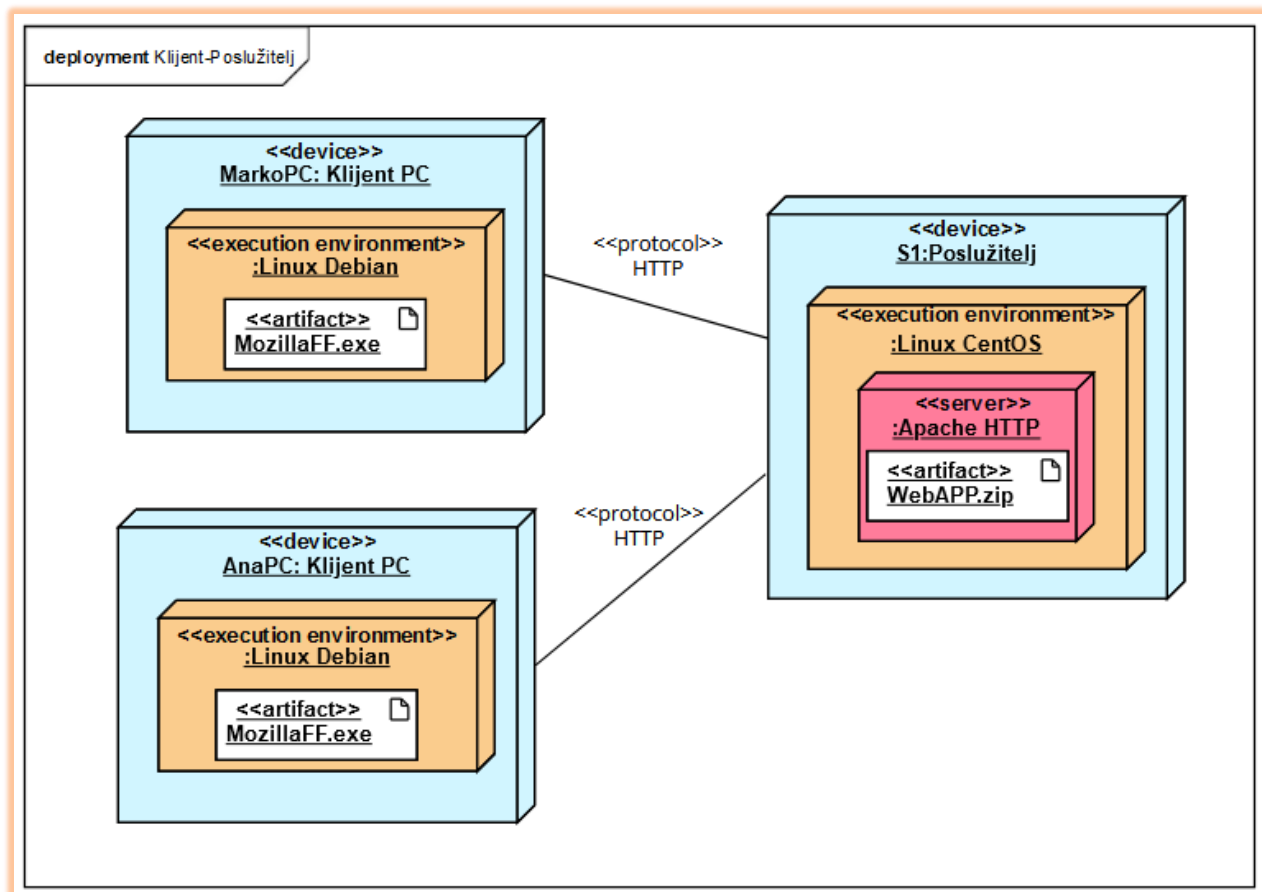
Specifikacijski dijagram:



Komentar: Okolina izvođenja (operacijski sustav, poslužiteljski programi kao npr. Tomcat, Apache PHP i sl.) se uvijek prikazuje kao čvor. Moguće je koristiti generički stereotip <<execution environment>> ili konkretnije specificirati npr. <<server>>, <<service>> itd.

Na dijagramima razmještaja je NUŽNO koristiti stereotipe za opis svih čvorova i komponenti. Generički stereotip za sve komponente (aplikacije koje se izvode na čvorovima) je <<artifact>>, ali moguće je koristiti i neke specifičnije kao npr. document, executable, file, library, script, source itd.

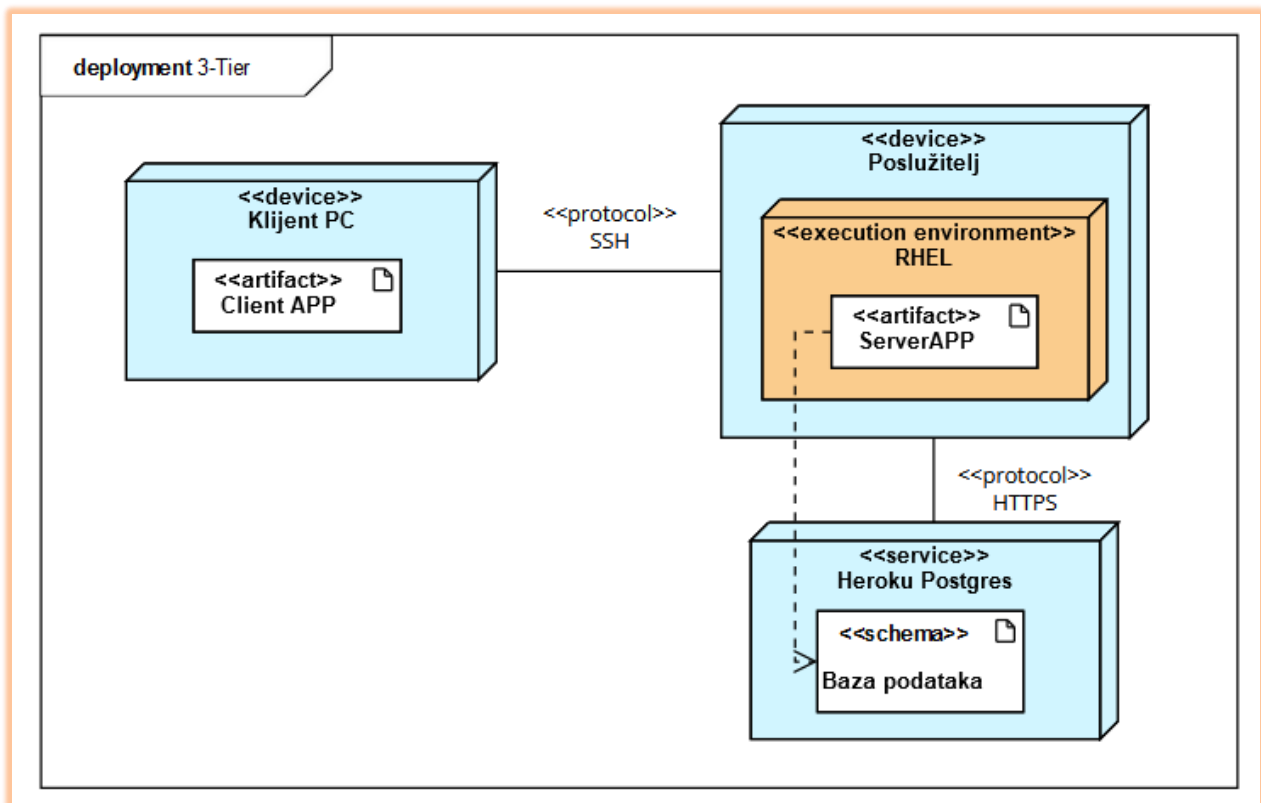
Dijagram instanci:



Komentar: Uočiti sintaksu na dijagramu instanci „NazivInstance: Tip“. Moguće je imati više instanci istog tipa, npr. „MarkoPC“ i „AnaPC“ su primjeri dvije instance istog tipa čvora „Klijent PC“.

Nije nužno uvijek pisati naziv instance. Ukoliko naziv nije poznat (ili nije važan) sintaksa je oblika „:Tip“, npr. „:LinuxDebian“ označava da je okolina izvođenja tipa „Linux Debian“, ali nema neki poseban naziv.

5.2. Trorazinska arhitektura

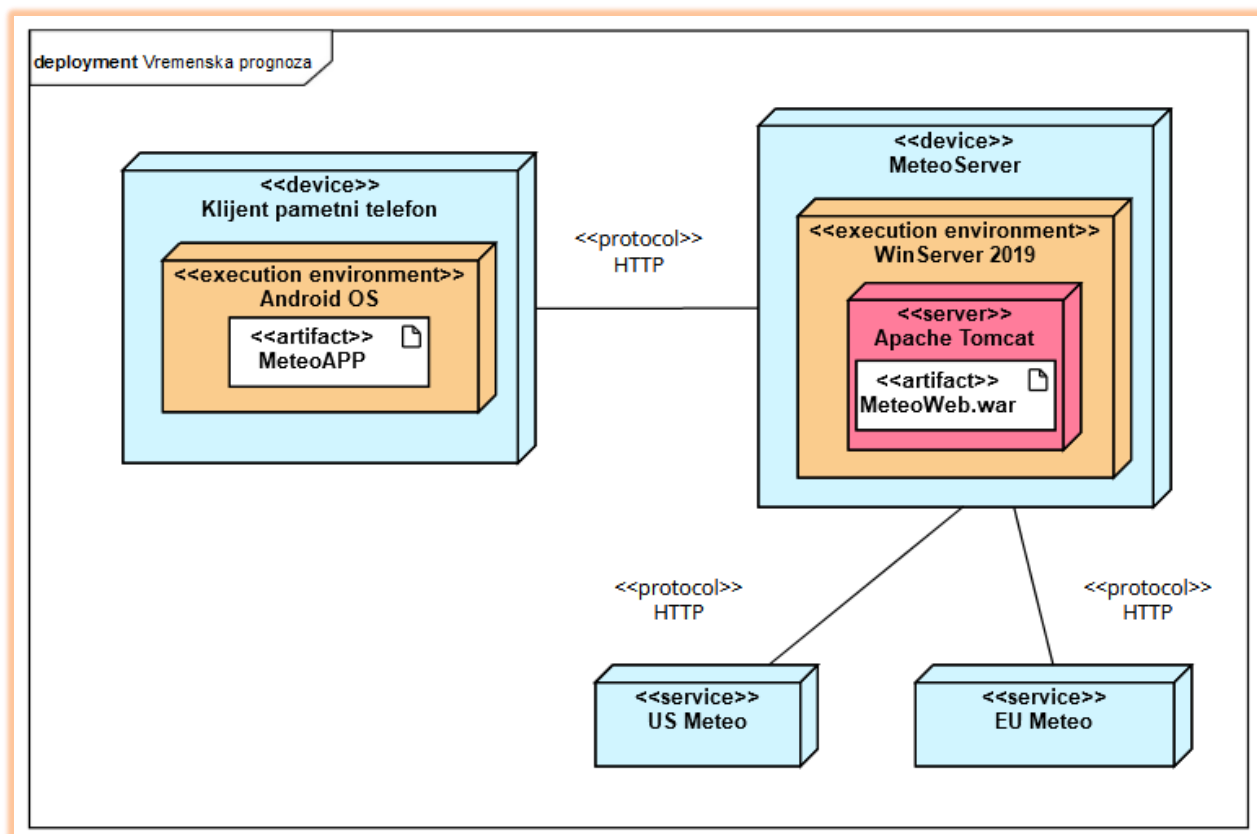


Komentar: Osim na fizičke čvorove (<<device>>), dijelovi sustava se mogu pustiti u pogon koristeći usluge oblaka kao npr. Heroku Postgres za poslužitelj baze podataka. U tom slučaju se najčešće koristi stereotip <<service>> ili <<cloud service>> kako bi se naznačila razlika u odnosu na fizičke čvorove. Također, kod korištenja usluga u oblaku vrlo često nisu dostupne informacije o detaljima okoline izvođenja kao što je operacijski sustav i sl. pa onda to nije ni potrebno modelirati.

Na ovom dijagramu je potrebno uočiti da, za razliku od prethodnog primjera, nije naznačen operacijski sustav na klijentskom računalu. Osnovni razlog jest zato što ta informacija nije dana u tekstu zadatka, a također se može tumačiti i da OS na klijentskoj strani nije bitan, tj. klijentska aplikacija u ovom slučaju radi neovisno o OS-u.

Također uočiti ovisnost između artefakata ServerAPP i Baze podataka.

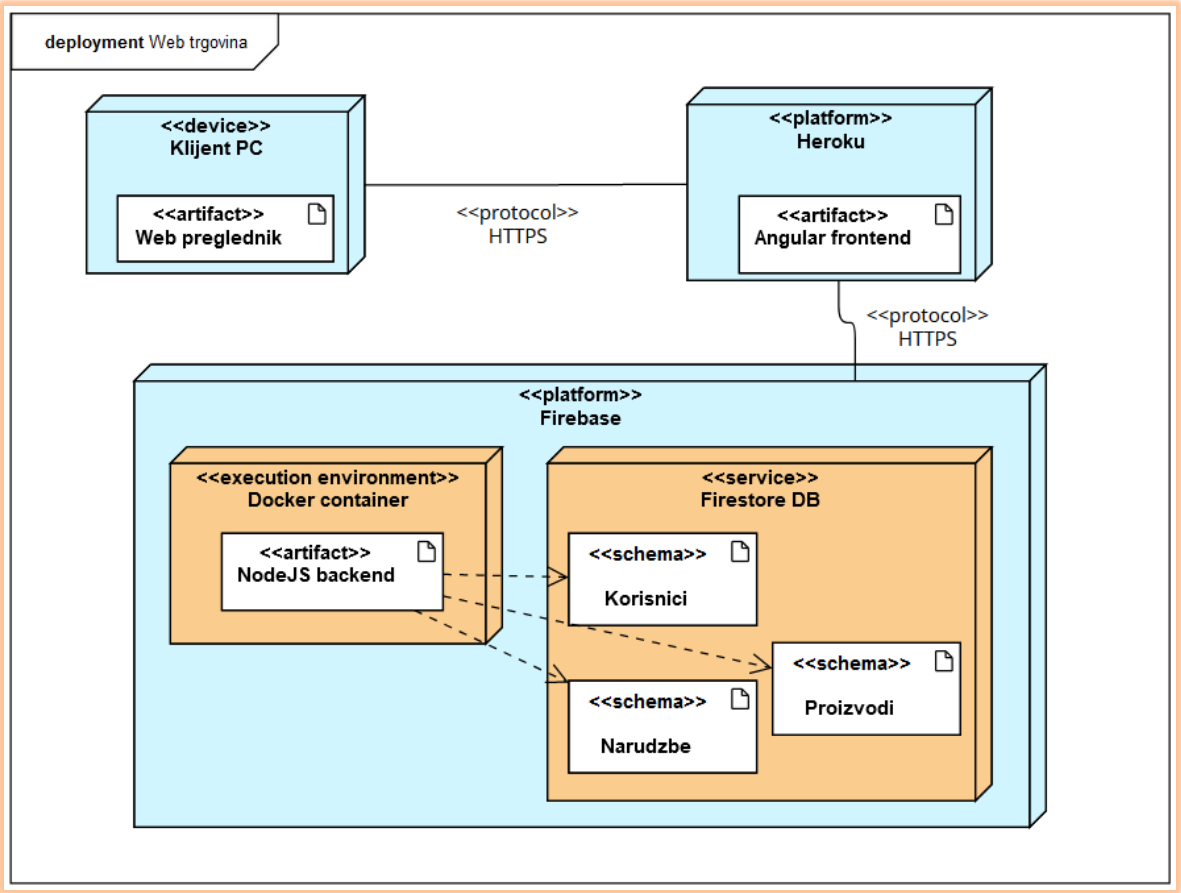
5.3. Vremenska prognoza



Komentar: O vanjskim servisima na koje se spaja MeteoServer nemamo nikakvih dodatnih informacija pa ih prikazujemo kao jednostavne čvorove sa stereotipom <<service>>.

5.4. Web trgovina

Specifikacijski dijagram:



Dijagram instanci:

