

**Univerzitet u Beogradu
Fakultet organizacionih nauka**

**PROJEKTOVANJE INFORMACIONIH
SISTEMA
-Projektni rad-**

**Mentor : Sladan Babarogić
Studenti : Dalibor Vidović
 Jelena Đuknić
 Milesa Gordić
 Aleksandar Dabić**

Beograd
Jun 2004.

1. KORISNIČKI ZAHTEV

Opis problema:

Potrebno je implementirati deo informacionog sistema Fakulteta organizacionih nauka vezan za upis studenata. U sistemu postoji sedam glavnih procesa: 1. Definisanje uslova za upis, 2. Standardni upis, 3. Upis po prijemnom ispitu (prva godina), 4. Upis po završenoj višoj školi ili fakultetu, 5. Prelazak sa drugog fakulteta, 6. Promena nastavnog plana, 7. Upis mirovanja.

- *Definisanje uslova za upis* – Proces omogućuje definisanje uslova za upis studenata po svim mogućim osnovama. Najznačajniji uslov je broj nepoloženih ispita koje je moguće preneti u narednu godinu studija. Uslovi se definišu Zakonom, Statutom FON-a i odlukama odgovarajućih organa FON-a.
- *Standardni upis* – Proces obuhvata poslove upisa studenata po standardnim uslovima definisanim Zakonom, Statutom FON-a i odlukama odgovarajućih organa FON-a. Sastoji se od sledećih podprocesa: Upis naredne godine, Obnova godine, Promena statusa i Promena odseka.
- *Upis po prijemnom ispitu (prva godina)* – Proces omogućuje upis studenata u prvu godinu studija na osnovu rezultata prijemnog ispita.
- *Upis po završenoj višoj školi ili fakultetu* – Omogućava se upis studenata u odgovarajuću godinu studija (trenutno drugu ili treću), po završenoj višoj školi ili fakultetu, na osnovu važećih propisa.
- *Prelazak sa drugog fakulteta* – Proces obuhvata upis studenata u odgovarajuću godinu (trenutno drugu ili treću), prelaskom sa drugog fakulteta, na osnovu važećih propisa.
- *Promena nastavnog plana* – Proces omogućuje prelazak studenata sa starog na novi nastavni plan na osnovu važećih propisa.
- *Upis mirovanja* – Proces omogućuje upis mirovanja (zadržavanje postojećeg statusa) za odrađene kategorije studenata (vojnici, trudnice, studenti na dužem bolovanju,...), na osnovu važećih propisa.

2. STRUKTURNA SISTEMSKA ANALIZA

Strukturalna sistemska analiza (SSA) je jedna potpuna metodologija za specifikaciju informacionog sistema, odnosno softvera. Ona se na različite načine može povezati sa metodama drugih faza u neku specifičnu metodologiju celokupnog razvoja IS. Tako na primer, ona može biti polazna osnova za metodu Strukturnog projektovana programa, ili projektovanja logičke strukture baze podataka metodom normalizacije, ili se može tretirati kao metodološki postupak dekompozicije nekog sistema na podsisteme sa ciljem da se, nalaženjem modela podataka podsistema i njihovom integracijom, dođe do potpunog modela podataka posmatranog sistema.

Potpuna, tačna, formalna i jasna specifikacija IS, ili kako se to obično kaže, specifikacija zahteva korisnika, zahteva koje budući sistem treba da zadovolji, predstavlja bitan preduslov za uspešno dalje projektovanje i implementaciju sistema. Očigledno je zbog čega specifikacija IS treba da bude potpuna i tačna. Zahtev da specifikacija bude formalna iskazuje se zbog toga što je formalna specifikacija osnov za "transformaciono" projektovanje i implementaciju, za automatizovano generisanje baze podataka i programa iz nje, odnosno za korišćenje CASE sistema. Zahtev da specifikacija bude jasna iskazuje se zbog toga što u specifikaciji IS u velikoj meri učestvuju korisnici sistema, neinformatičari, pa jezik specifikacije mora biti i njima prihvatljiv. Originalna SSA čiji su tvorci Yourdon i njegovi saradnici (DeMarco i drugi) poseduje veoma jednostavne, grafičke, pa samim tim i jasne koncepte. Ovde su svi ovi koncepti zadržani, a strožija formalizacija je dodata samo za opis strukture tokova i skladišta podataka, da bi se obezbedio specifičan transformacioni razvoj IS koji Standardna metodologija zagovara.

Kao što je već ranije rečeno, specifikacija IS treba da prikaže (potpuno, tačno, formalno i jasno) šta budući informacioni sistem treba da radi. Veoma je bitno odmah istaći da specifikacija IS prikazuje **ŠTA** IS treba da da, a ne i **KAKO** to treba da ostvari. Očigledno je da prerano definisanje "kako", odnosno davanje nekih projektantskih rešenja u okviru specifikacije, ograničava kasniji mogući izbor (optimizaciju) načina implementacije sistema. Odgovor na pitanje "**kako**" daje se za konkretno okruženje, za definisanu tehnologiju i organizaciju u kojoj se sistem implementira. Da specifikacija ne bi sadržala tehnološki i organizaciono ograničena rešenja, obično se kaže da ona treba da opiše funkcionisanje IS u "idealnoj tehnologiji", gde praktično nikakva ograničenja ne postoje.

SSA posmatra informacioni sistem kao funkciju (proces obrade) koja, na bazi ulaznih, generiše izlazne podatke. Ulazni podaci se dovode u proces obrade, a izlazni iz njega odvođe preko tokova podataka. Tok podataka se tretira kao vod ili kao pokretna traka kroz koji stalno teku ili koja stalno nosi podatke na najrazličitijim nosiocima - papirni dokumenti, niz poruka koje čovek unosi preko tastature terminala, "paket" informacija dobijen preko neke telekomunikacione linije ili slično. Imajući u vidu zahtev da specifikacija treba da se oslobodi svih implementacionih detalja od interesa su samo sadržaj i struktura ulaznog toka, a ne i medijum nosilac toka.

Izvori ulaznih, odnosno ponori izlaznih tokova podataka mogu biti objekti van IS koji sa IS komuniciraju i koji se u SSA nazivaju interfejsi, drugi procesi u sistemu, ili tzv

skladišta. Skladišta podataka se posmatraju kao "tokovi u mirovanju", odnosno odloženi, akumulirani tokovi, različite vrste evidencija, arhiva, kartoteka i datoteka. I za skladišta kao i za tokove od interesa su isključivo njihov sadržaj i struktura.

Imajući u vidu sve rečeno, jednu potpunu specifikaciju IS čine:

- 1) Hijerarhijski organizovan skup dijagrama toka podataka;
- 2) Rečnik podataka koji opisuje sadržaj i strukturu svih tokova i skladišta podataka;

2.1 DIJAGRAMI

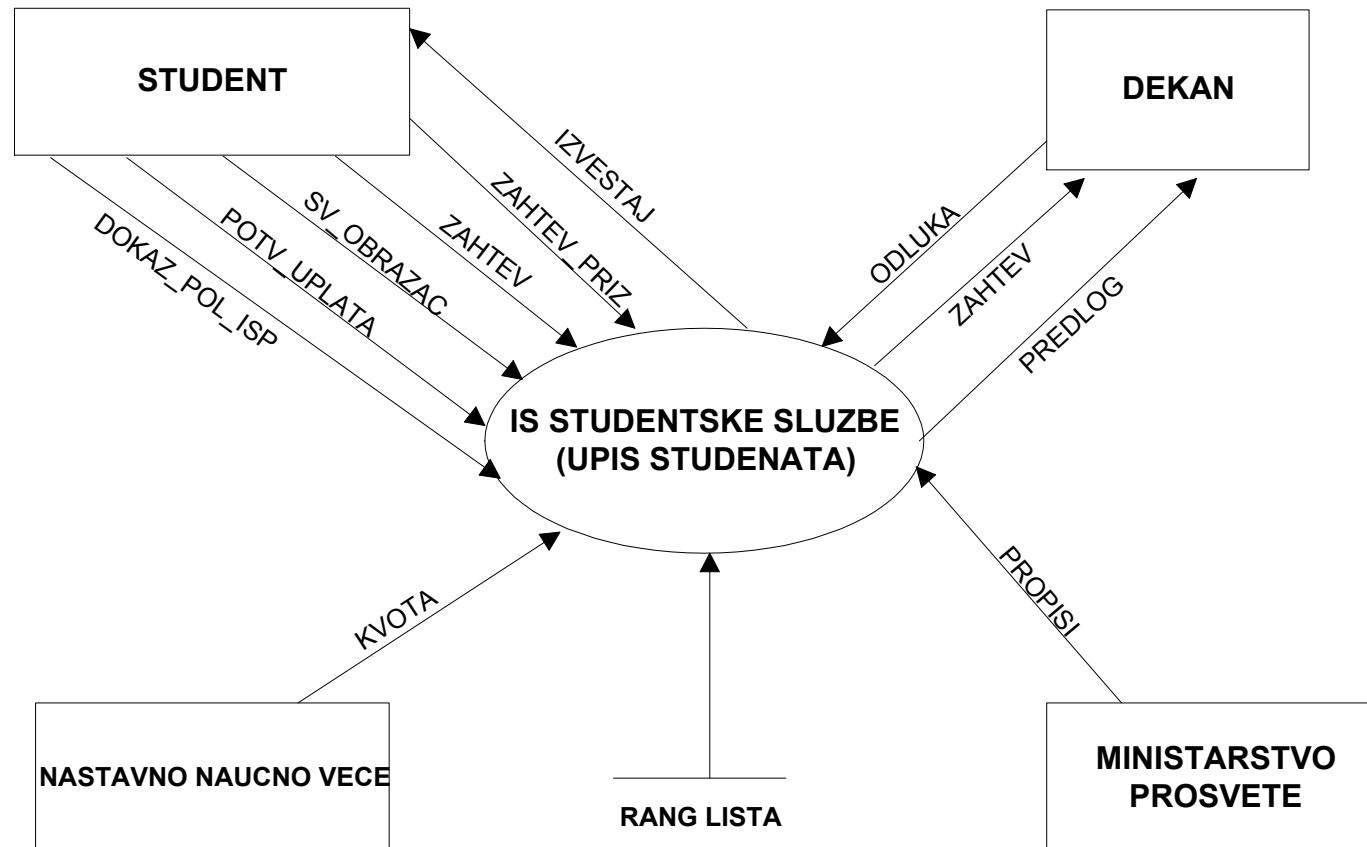
Osnovni koncepti za specifikaciju IS u SSA su, znači, funkcije, odnosno procesi obrade podataka, tokovi podataka, skladišta podataka i interfejsi. Njihov međusobni odnos se prikazuje preko *dijagrama toka podataka* koji prikazuje vezu interfejsa, odnosno skladišta kao izvora odnosno ponora podataka, sa odgovarajućim procesima, kao i međusobnu vezu procesa. U našem primeru koristimo sledeće graficke simbole:

- 1) krug ili elipsa predstavlja funkciju ili proces obrade podataka,
- 2) pravougaonik predstavlja interfejs,
- 3) usmerena linija predstavlja tok podataka,
- 4) dve paralelne linije ("otvoreni" pravougaonik) predstavlja skladište podataka.

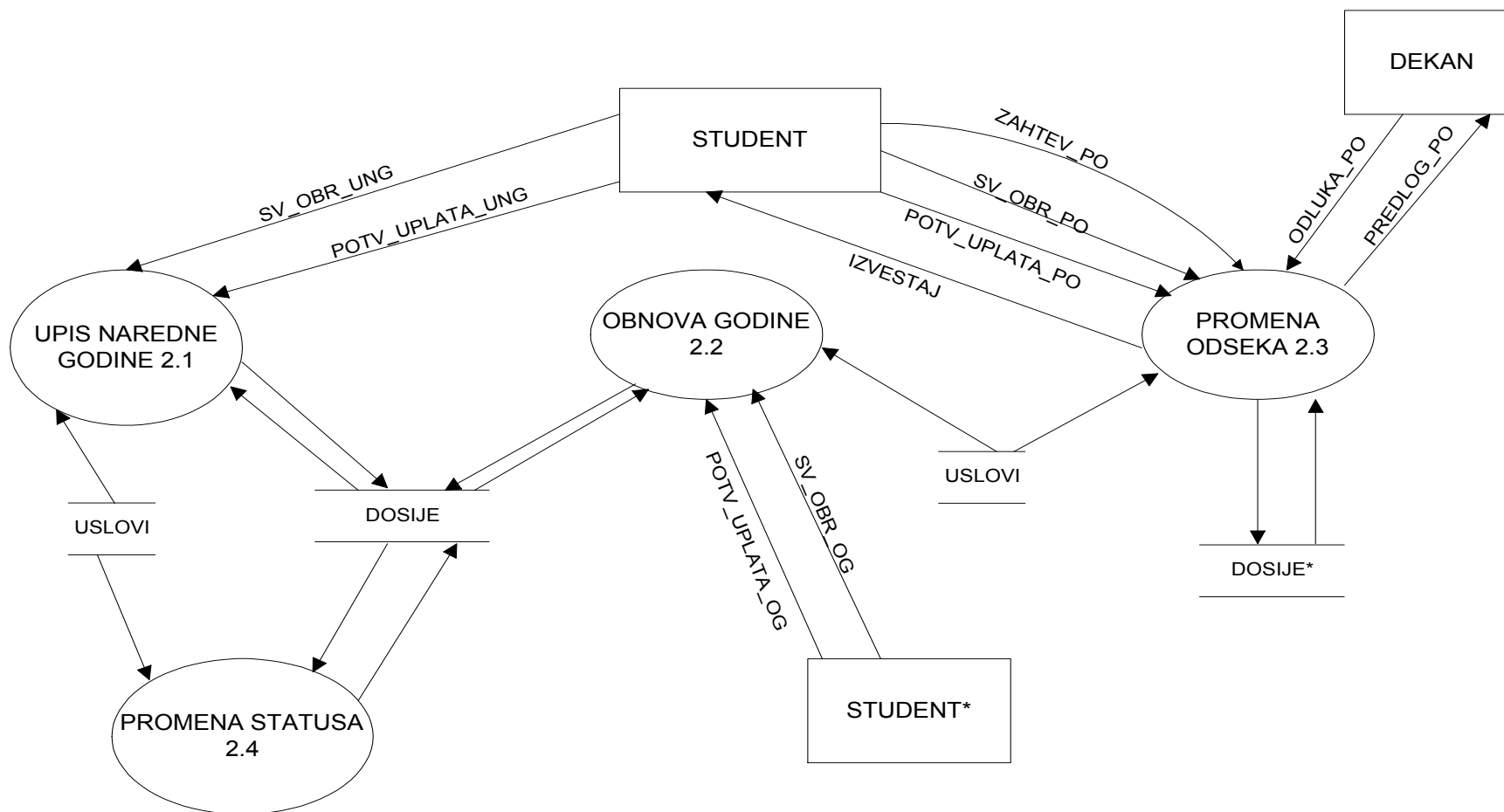
Očigledno je da se jedan IS sastoji iz mnoštva procesa, interfejsa, tokova i skladišta podataka. Specifikacija IS treba da bude potpuna (detaljna) i jasna. Kada bi se jedan sistem detaljno opisao i prikazao jednim dijagramom toka podataka, dobio bi se veoma nejasan opis sistema, paukova mreža procesa, tokova, skladišta i interfejsa. Istovremeno detaljan i jasan opis sistema zahteva opis na "različitim nivoima apstrakcije", odnosno hijerarhijski opis u kome se na višim nivoima sistem opisuje opštije, a na nižim, postepenim i organizovanim uvođenjem detalja, potpuno i detaljno. Hijerarhijski opis sistema u tehnici dijagrama tokova podataka se svodi na to da se na višim nivoima definišu globalniji procesi, a da se zatim svaki takav globalni proces, na sledećem nižem nivou, pretstavi novim dijagramom toka podataka.

Dijagram toka podataka na vrhu ovakve hijerarhije naziva se *dijagram konteksta*, a procesi na najnižem nivou (proces koji se dalje ne dekomponuju) nazivaju se *primitivni procesi*.

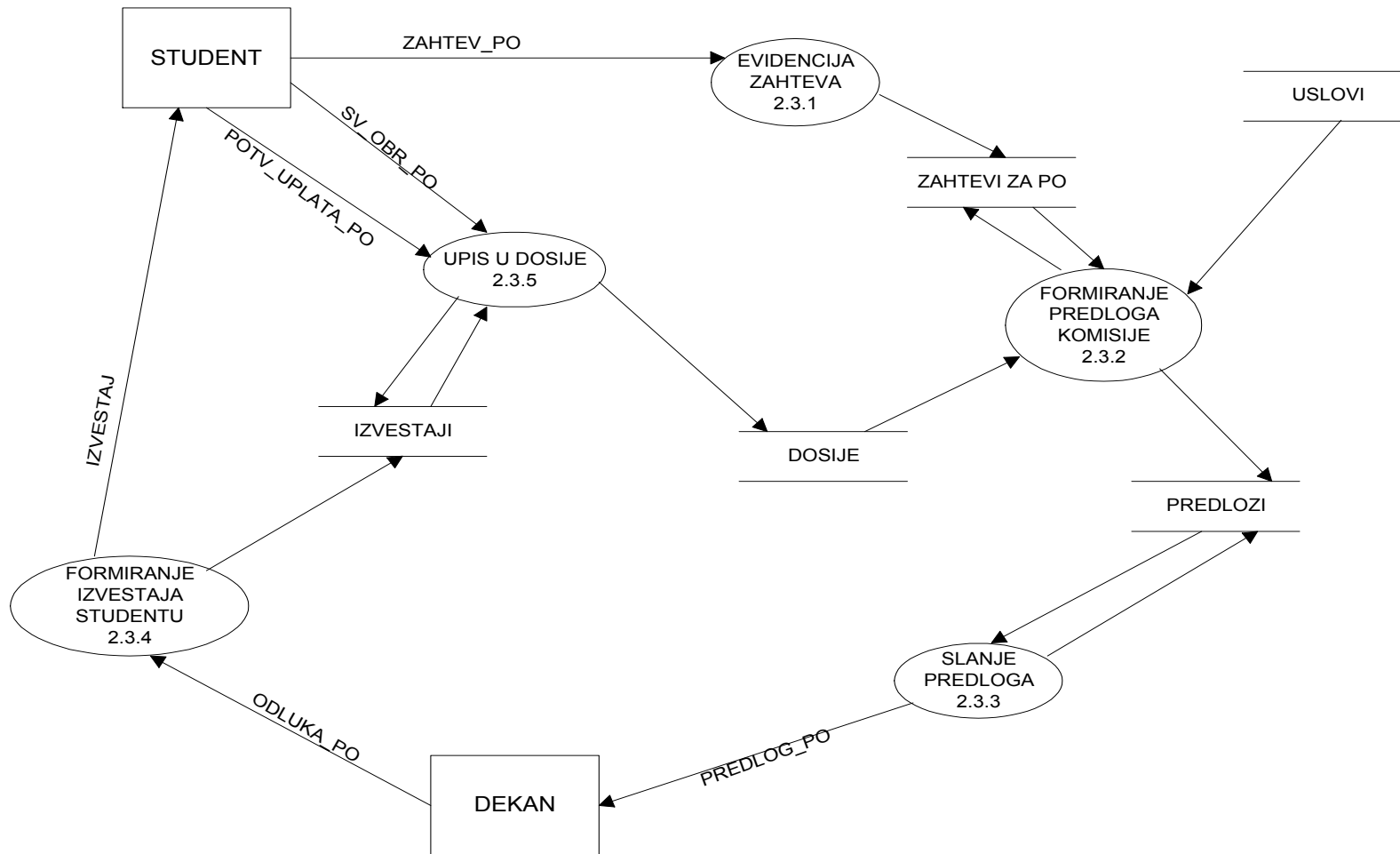
2.1.1 DIJAGRAM KONTEKSTA



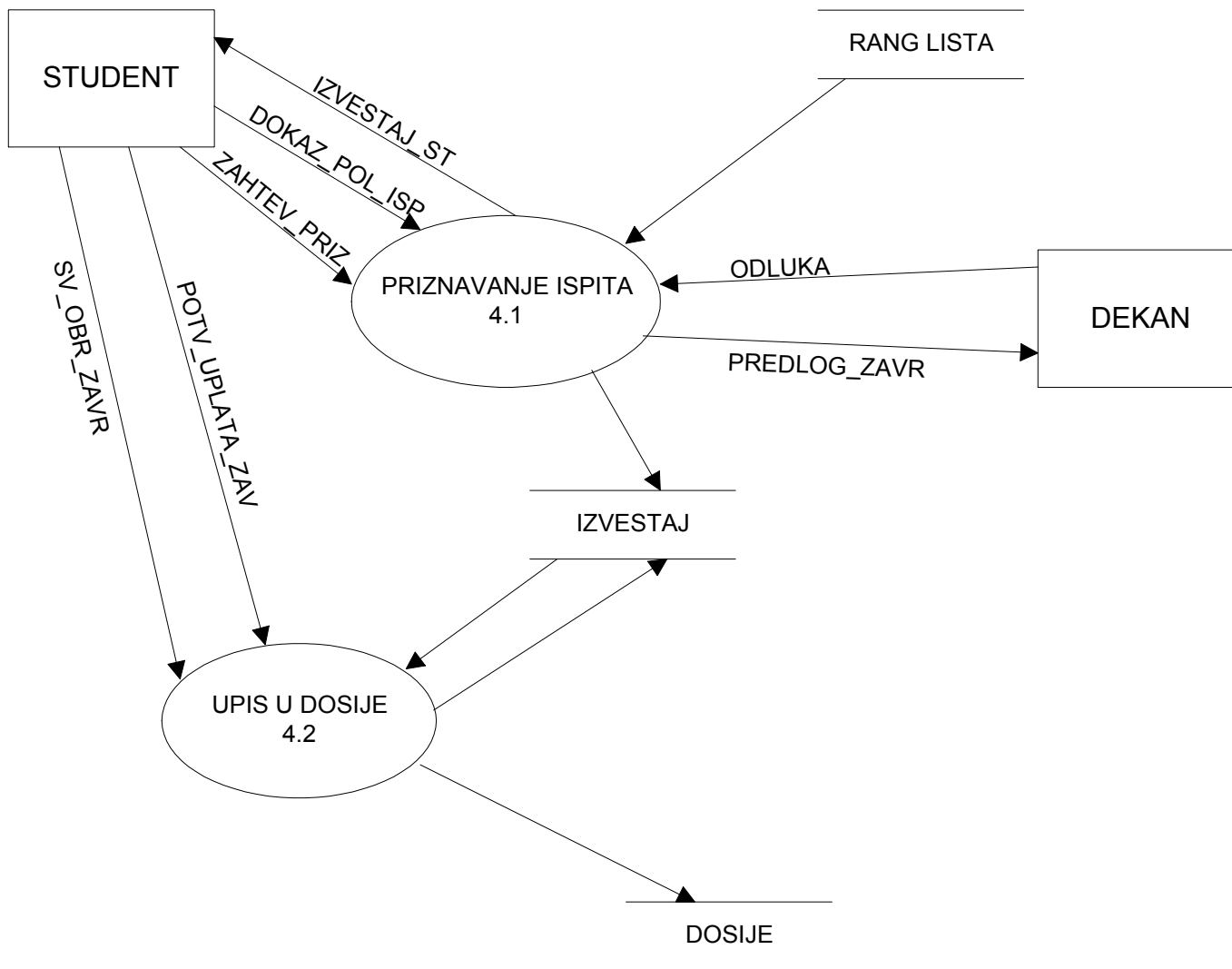
2.1.3 STANDARDNI UPIS



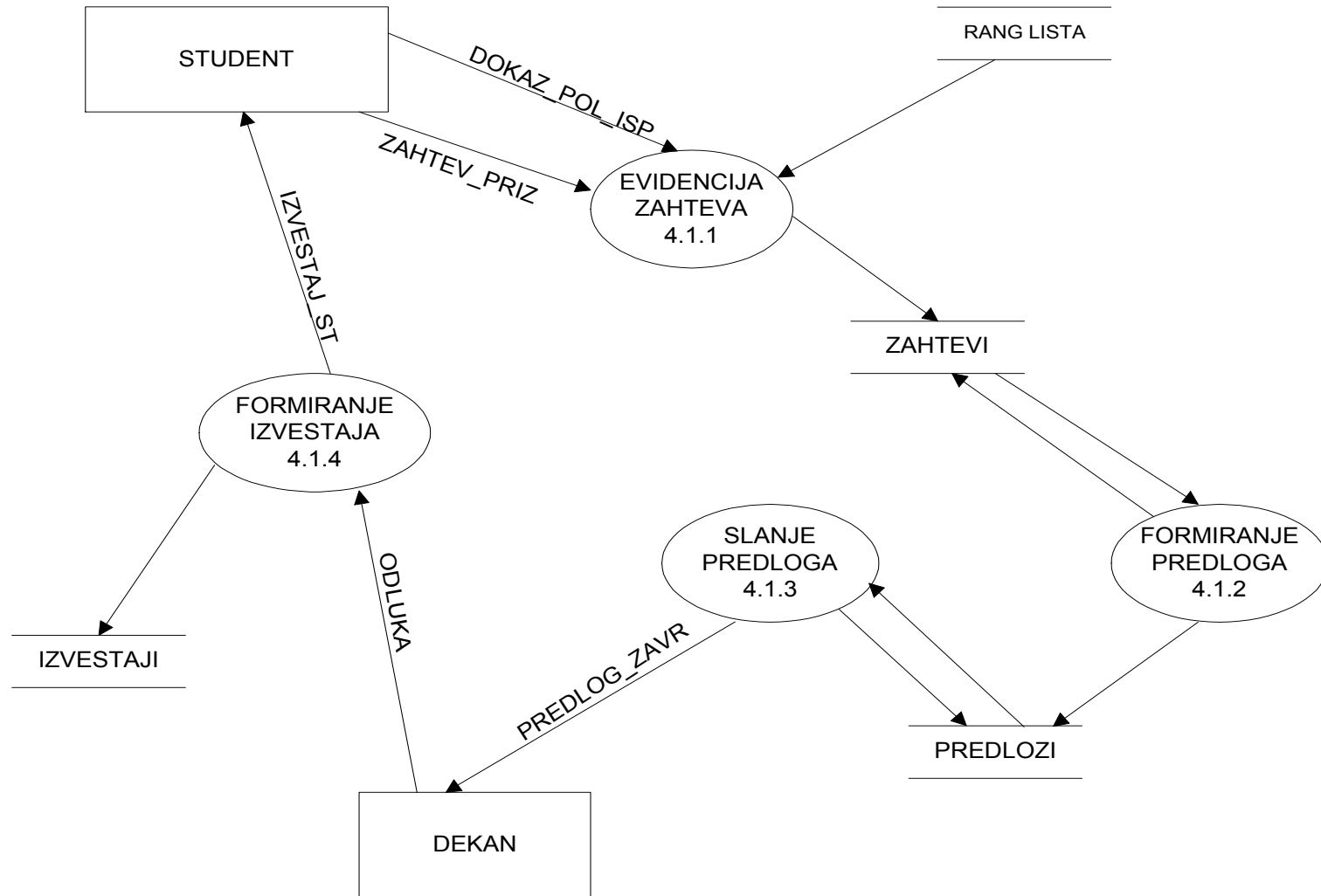
2.1.3.1 PROMENA ODSEKA



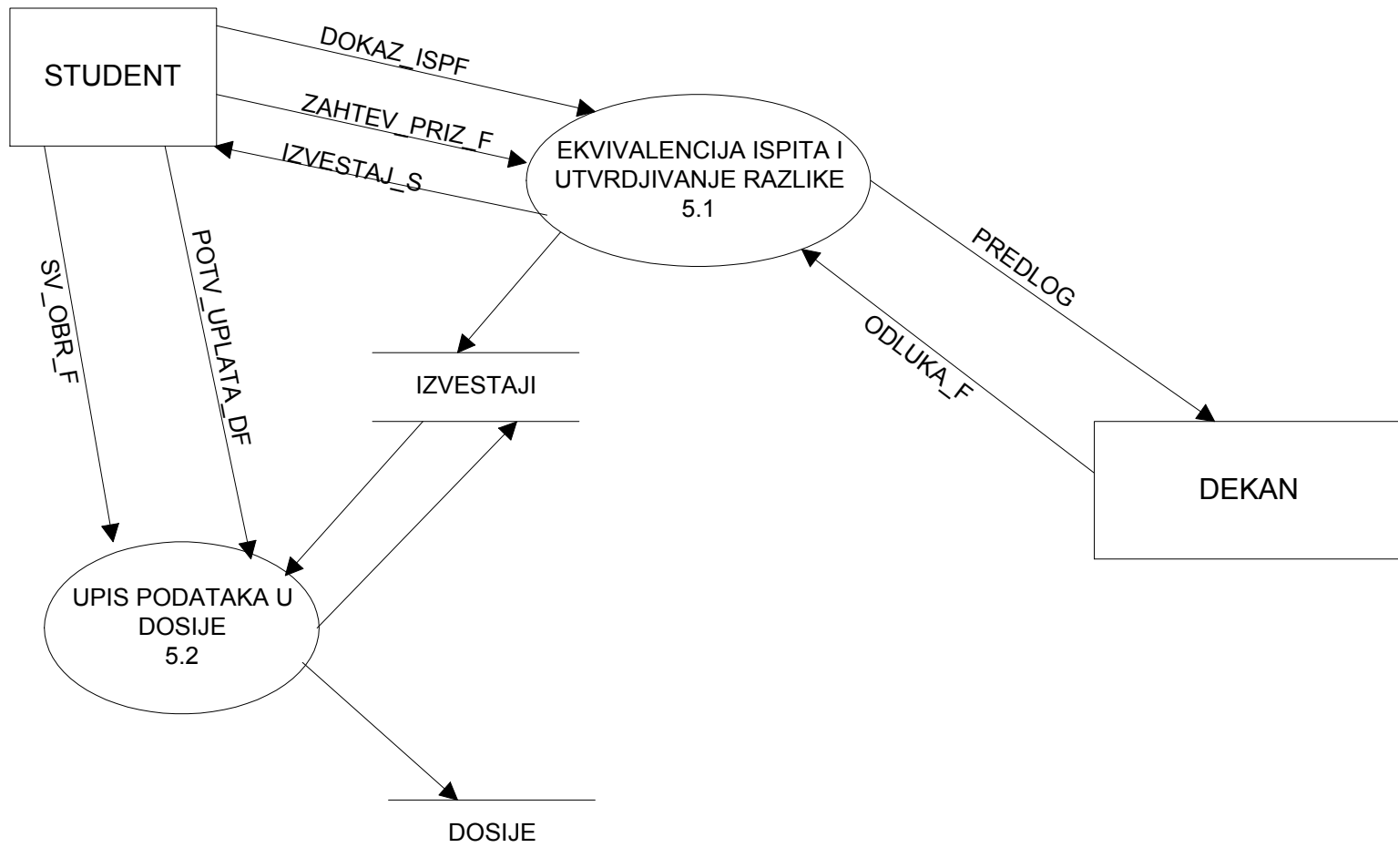
2.1.4 UPIS PO ZAVRSENOJ VISOKOJ SKOLI ILI FAKULTETU



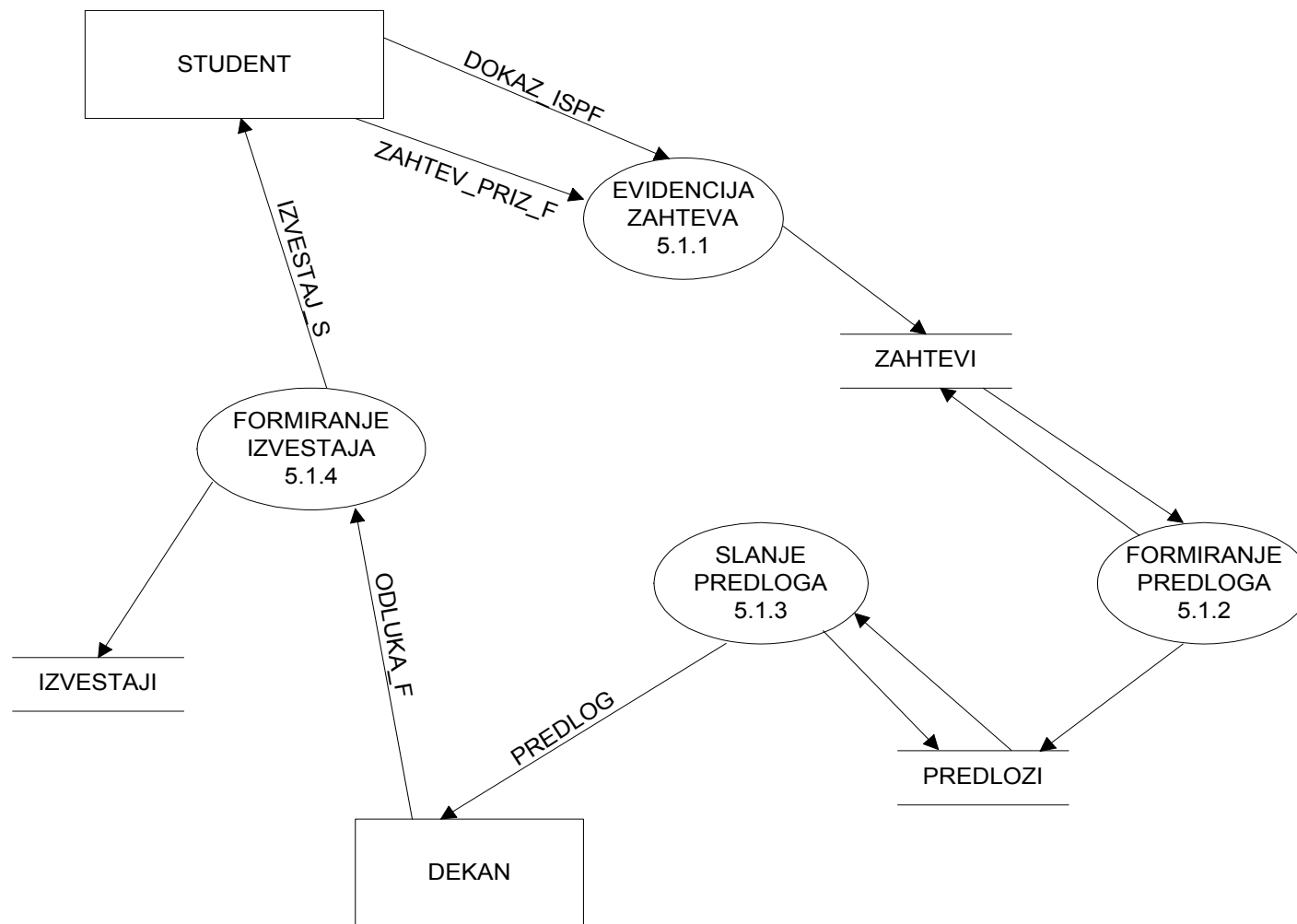
2.1.4.1 PRIZNAVANJE ISPITA



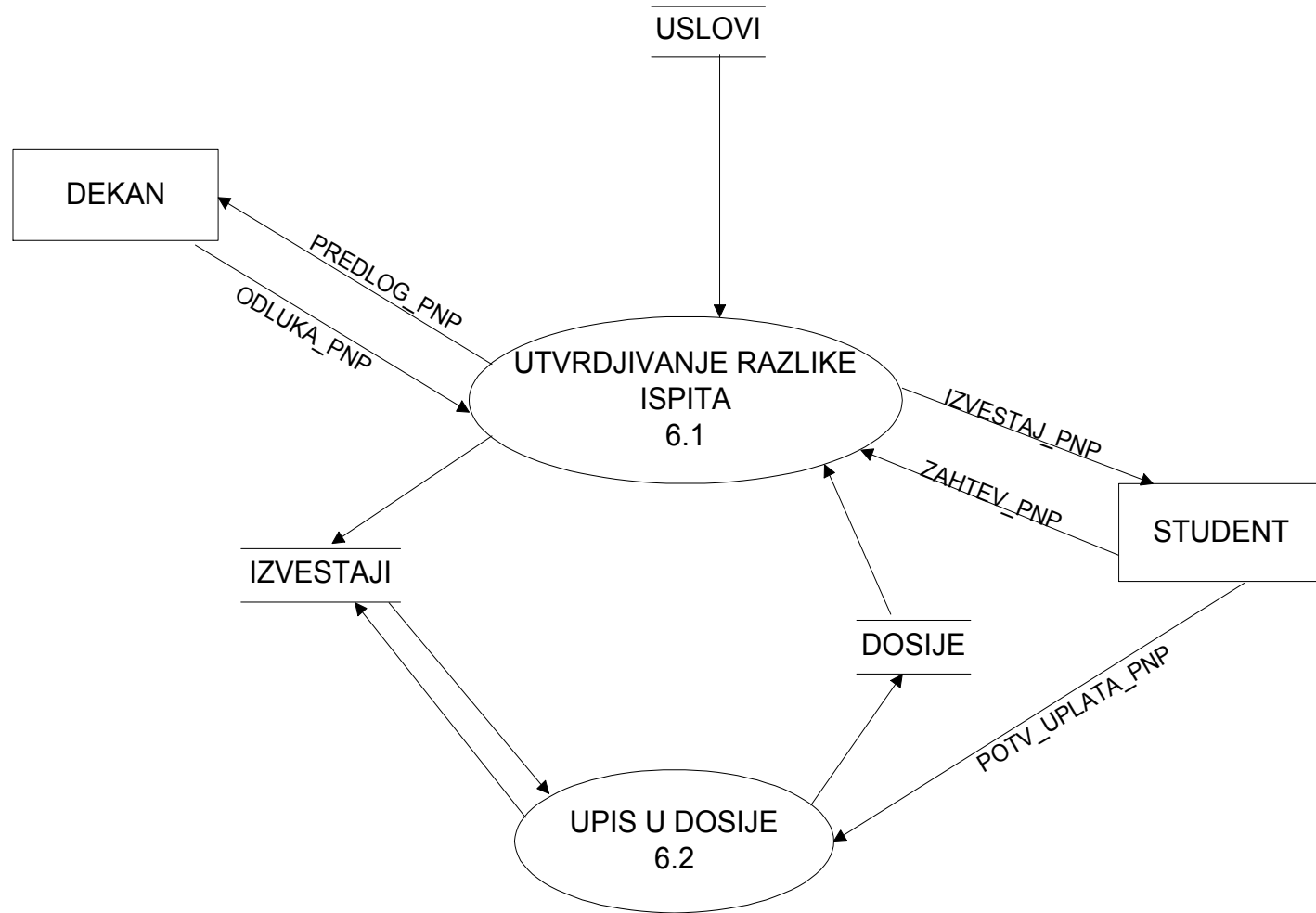
2.1.5 PRELAZAK SA DRUGOG FAKULTETA



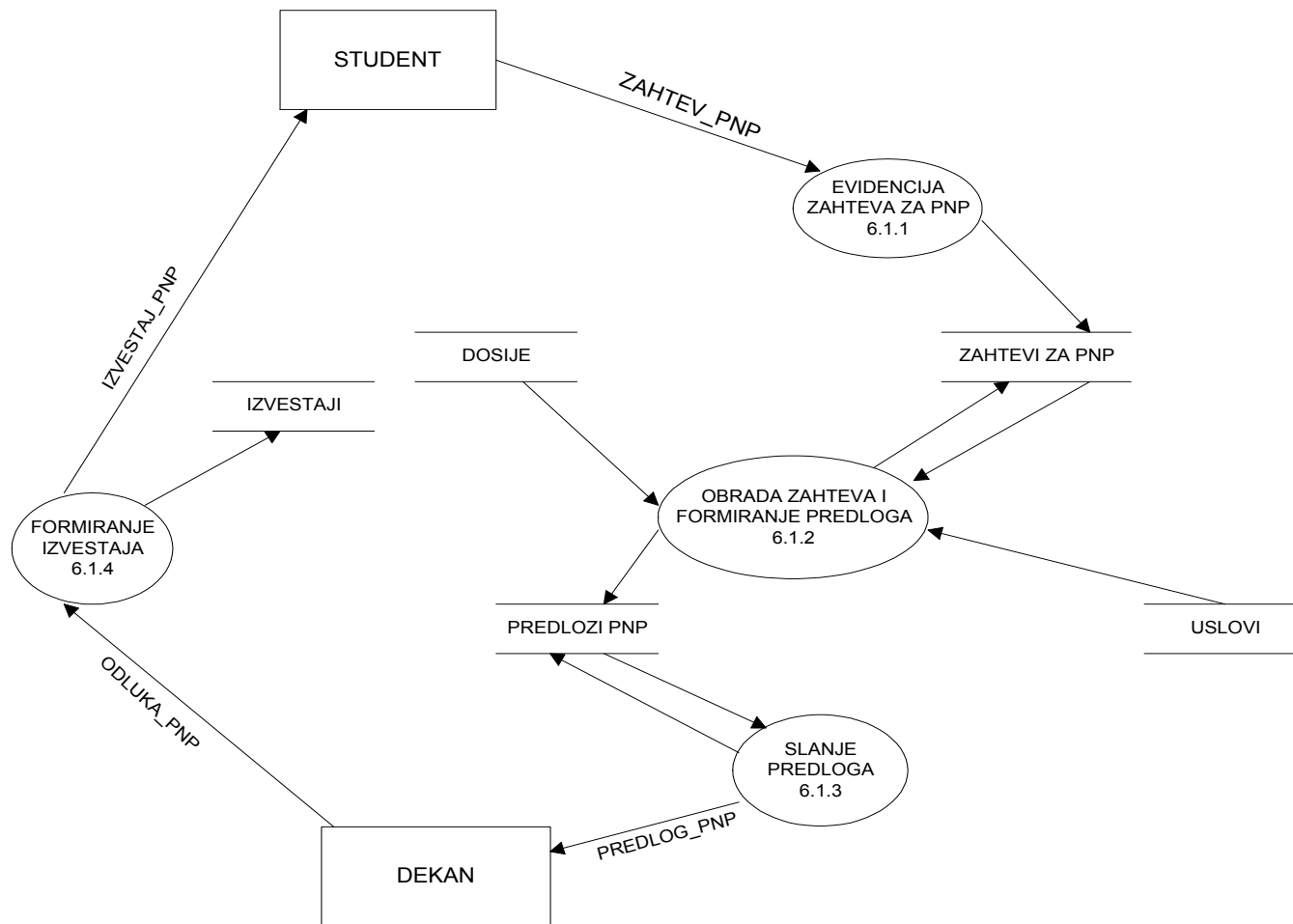
2.1.5.1 EKVIVALENCIJA ISPITA I UTVRDJIVANJE RAZLIKE



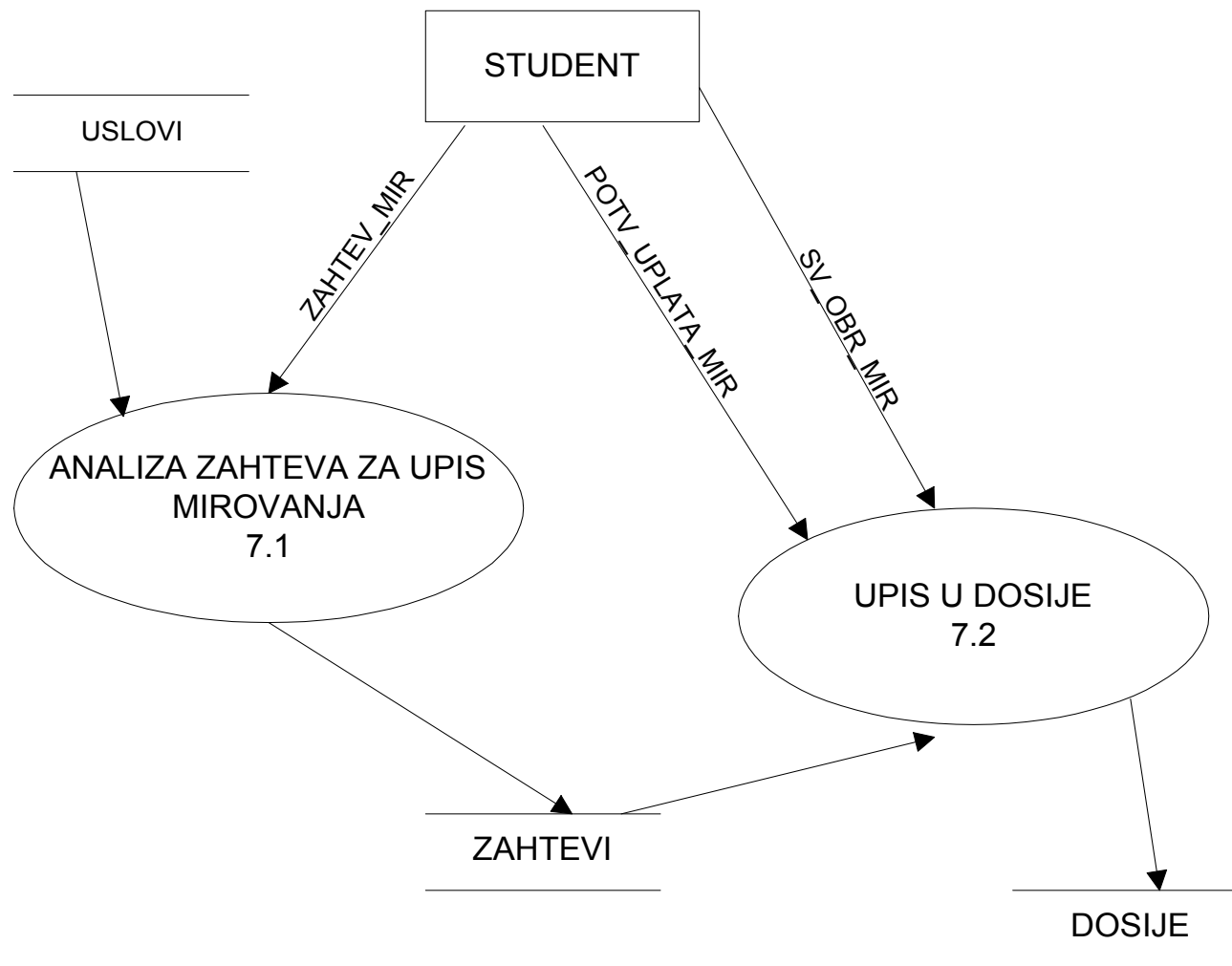
2.1.6 PROMENA NASTAVNOG PLANA



2.1.6.1 UTVRDJIVANJE RAZLIKE ISPITA



2.1.7 UPIS MIROVANJA



2.2 REČNIK PODATAKA

Rečnik podataka, kao što je ranije rečeno, daje opis strukture i sadržaja svih tokova i skladišta podataka. Bez obzira šta tok ili skladište podataka predstavljaju, papirni dokument, niz karaktera kao ulaz sa terminala, "paket" informacija dobijen telekomunikacionom linijom, kartoteku ili datoteku, kao logička struktura podataka oni predstavljaju neku kompoziciju polja. Da bi precizno definisali logičku strukturu skladišta i tokova i definisali sintaksu rečnika neophodno je da uvedemo definicije svi koncepta rečnika:

1) *Polje* je elementarna (atomska) struktura koja se dalje ne dekomponuje i koja ima svoju vrednost.

2) Polja svoje vrednosti uzimaju iz skupova vrednosti koji se nazivaju *domenima*. Domeni mogu biti:

- "predefinisani", odnosno standardni programsko-jezički domen, kao što su INTEGER, CHARACTER, REAL, LOGICAL i DATE.
- "semantički", kada se definišu posebno, preko svoga imena, predefinisane domene i, eventualno, ograničenja na mogući skup vrednosti predefinisane domena.

3) Pored ograničenja na vrednosti polja, odnosno vrednosti domena koja su data u primerima definišu se i druga. *Ograničenja* mogu biti prosta i složena. Lista dozvoljenih prostih ograničenja je:

(a) Θ konstanta, gde je Θ bilo koji operator poređenja koji se na datom domenu može definisati (na primer, $<$, $>$, $=$, \leq , \geq za brojne domene), a konstanta je neka definisana vrednost iz datog domena.

(b) BETWEEN konstanta, konstanta, gde su konstante vrednosti iz datog domena.

(c) IN (lista vrednosti), gde se lista formira od konstanti iz odgovarajućeg domena.

(d) NOT NULL, kada dato polje ne može da dobije "nulla vrednost", odnosno mora uvek da ima vrednost.

STRUCTURES

Rečnik podataka za **zahtev (za studente FON – a)**

zahtev : < Br_zah , Datum_zah , Naziv_zah , Ime_Prez_stud , Br_dos , Sadržaj_zah ,
Napom_zah , Potpis_stud , Potpis_ov_lica , Ovl_ID >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Broj zahteva : Br_zah – int – not null
Datum zahteva : Datum_zah – date
Naziv zahteva : Naziv_zah – char (40)
Ime i prezime studenta : Ime_Prez_stud – char (30)
Broj dosijea : Br_dos – string
Sadržaj zahteva : Sadržaj_zah – string
Napomena : Napom_zah – string
Potpis ovlaštenog lica : Potpis_ov_lica – char (30)
ID ovlaštenog lica : Ovl_ID – int – not null
Potpis studenta : Potpis_stud – char (30)

Rečnik podataka za **dokaz o položenim ispitima**

dokaz_pol_isp : < Sif_dok , ImePrez_stud , JMBG , Datum_dok , SifraF , NazivF ,
{<RB_isp , Sif_isp , Naziv_isp , Ocena_isp , Datum_isp , Profesor_isp >} ,
Potpis_ov_lica , Ovl_ID >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Šifra dokaza o ispitima : Sif_dok – int – not null
Ime i prezime studenta : ImePrez_stud – char (30)
JMBG : JMBG – int – not null
Datum dokaza o ispitima : Datum_dok – date
Šifra fakulteta : SifraF – int-not null
Naziv fakulteta : NazivF – char (40)
Redni broj ispita : RB_isp – int – not null
Šifra ispita : Sif_isp – int – not null
Naziv ispita : Naziv_isp – char (40)
Ocena : Ocena_isp – int – in (6,7,8,9,10)
Datum polaganja ispita : Datum_isp - date
Profesor : Profesor_isp – char (30)
Potpis ovlaštenog lica : Potpis_ov_lica – char (30)
Šifra ovlaštenog lica : Ovl_ID – String

Rečnik podataka za **odluku dekana**

Odluka : < Broj_odl , Datum_odl , Naziv_odl , Ime_Prez_stud , Broj_dos , Sadržaj_odl , Napom_odl , Potpis_ovl_lica , Ovl_ID >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Broj odluke : Broj_odl – int – not null
Datum odluke : Datum_izv – Date
Naziv odluke : Naziv_odl – char (40)
Ime i prezime studenta : Ime_Prezime_stud – char (30)
Broj dosijea : Broj_dos - string
Sadržaj odluke : Sadržaj_odl– string
Napomena : Napom_odl – string
Potpis ovlaštenog lica : Potpis_ovl_lica – char (30)
ID ovlaštenog lica : Ovl_ID – int – not null

Rečnik podataka za **izveštaj o priznatim ispitima**

Izveštaj : <Broj_izv , Datum_izv , ImePrez_stud , JMBG , Sif_fk , Naziv_fk ,
{< RB_I , Ime_pol_ispita , Sif_priz_isp , Naziv_priz_isp >} ,
{<RB_K , Ime_cl_kom , Cl_komID >} ,
Potpis_ovl , Ovl_ID >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Broj izveštaja : Broj_izv – int – not null
Datum izveštaja : Datum_izv – Date
Ime i prezime studenta : ImePrez_stud – char(30)
Matični broj : JMBG – int(13) – not null
Šifra fakulteta : Sif_fk – int – not null
Naziv fakulteta : Naziv_fk - char
RB položenog ispita : RB_I – int
Naziv položenog ispita : Ime_pol_isp – char
Šifra priznatog ispita : Sif_priz_isp – int – not null
Naziv priznatog ispita : Naziv_priz_isp – char
RB člana komisije : RB_K – int – in(1,2,3)
Ime i prezime člana komisije : Ime_cl_kom – char
Šifra člana komisije : Cl_komID – int – not null
Potpis ovlaštenog lica : Potpis_ovl – char
Šifra ovlaštenog lica : Ovl_ID – int – not null

Rečnik podataka za **potvrdu o uplati**

potv_uplata : < ImePrezime_stud , BrDos , Adresa_stud , Svrha_upl , Sif_sv ,
Dat_prijem , Sif_plac, Iznos_upl , BrRac , Poz_br , Sif_p >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Ime i prezime studenta : ImePrez_stud - char
Broj dosijea : BrDos – String
Adresa studenta : Adresa_stud – char
Svrha uplate : Svrha_upl – char
Šifra svrhe uplate : Sif_sv – int – not null
Datum prijema : Dat_prijem – Date
Šifra placanja : Sif_pl – String
Iznos uplate : Iznos_upl – Real
Žiro račun primaoca : BrRac – String
Poziv na broj : Poz_br – int – not null
Šifra potvrde : Sif_p – int – not null

Rečnik podataka za **predlog komisije o priznavanju ispita**

predlog : < PredlogID , DatumPred , ImePrezime_stud , JMBG , Naziv_fkl , Sif_fkl ,
Univ , UnivID ,
{<Rb_I , Ime_pol_isp , Sif_priz_isp , Naziv_priz_isp >}
{<RB_K , ClanKomID , ImeClanaKom >} >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Šifra predloga : PredlogID – String
Datum predloga : DatumPred – Date
Ime i Prezime studenta: ImePrezime_stud – char
Matični broj studenta : JMBG – int(13) – not null
Naziv fakulteta : Naziv_fkl – char
Šifra fakulteta : Sif_fkl – String
Univerzitet : Univ – char
Šifra univerziteta : UnivID – String
Redni broj položenog ispita : Rb_I – int – not null
Naziv položenog ispita : Ime_pol_isp - char
Šifra priznatog ispita : Sif_priz_isp – String
Naziv priznatog ispita : Naziv_priz_isp – char
Redni broj člana komisije : RB_K – int – not null
Šifra člana komisije : ClanKomID - String
Ime i prezime člana komisije : ImeClanaKom – char

Rečnik podataka za ŠV obrazac

sv_obrazac : < Sk_god , BrDos , Sif_sv , MatBr_reg , RedBr_pl , Ime(R)Prez , JMBG , Fkl_Vs , Sif_fkl , Od , Sif_od , Sm , Sif_sm , Unv , Sif_unv , Mesto_Sk , Sif_ms , Pol , GodRodj , Mesto_rodj , Opst_rodj , Sif_mr , Stan_s , Opst_stan , Ul , BrTel , Sif_mstan , Mesto_stud , Opst_stud , Ul_stud , Tel_stud , Drz , Sif_drz , Nac , Sif_nac , Preth_zav_sk , Sif_skol , Opst_skol , Sif_opst_skol , StrJez , God_zav_skol , God_upis , Finans , Pon , God_1_upis , SS_Otac , SS_Maj , Akt_rod , ZanRod , Sif_zan_rod , ZanStud , Sif_zan_stud , RO , MestoUp , DatumUp , Potpis_stud , Nap >

Naziv podatka : Drugi naziv podatka – Ograničenje

Školska godina : Sk_god – String

Broj dosijea : BrDos – String

Šifra ŠV obrasca : Sif_sv – String

Matični broj registra : MatBr_reg – int(8) – not null

Redni broj prijavnog lista : RedBr_pl – int(5) – not null

Prezime(ime oca ili majke) i ime : Ime(R)Prez – char

JMBG studenta : JMBG – int(13) – not null

Naziv fakulteta- akademije ili više škole : Fkl_Vs – char

Šifra fakulteta(ili više škole) : Sif_fkl – int(4) – not null

Odsek : Od – char

Šifra odseka : Sif_od – int(2) – not null

Smer : Sm – char

Šifra smeru : Sif_sm – int(2) – not null

Univerzitet : Unv – char

Šifra univerziteta : Sif_unv - int(2) – not null

Mesto škole : Mesto_Sk – char

Šifra mesta škole : Sif_ms – int(5) – not null

Pol : Pol – int(1) – not null

Godina rođenja : GodRodj – int(4) – not null

Mesto rođenja studenta : Mesto_rodj – char

Opština(ili strana država) rođenja : Opst_rodj – char

Šifra mesta rođenja : Sif_mr – int(5) – not null

Mesto(naselje) stalnog boravka studenta : Stan_s – char

Opština(ili strana država) stalnog boravka : Opst_stan – char

Ulica i kućni broj : Ul – String

Broj telefona : BrTel – int

Šifra mesta stalnog boravka : Sif_mstan – int(5) – not null

Mesto stanovanja studenta za vreme studiranja : Mesto_stud – char

Opština(ili strana država) studenta za vreme studiranja : Opst_stud – char

Ulica i broj mesta studiranja : Ul_stud - String

Telefonski broj : Tel_stud – int

Državljanstvo : Drz – char

Šifra državljanstva : Sif_drz – int(3) – not null
 Nacionalna pripadnost : Nac – char
 Šifra nacionalne pripadnosti : Sif_nac – int(2) – not null
 Prethodno završena škola : Preth_zav_sk – char
 Šifra škole : Sif_skol – int(3) – not null
 Opština(ili strana država) škole : Opst_skol – char
 Šifra opštine škole : Sif_opst_skol – int(5) – not null
 Strani jezik učen u školi : StrJez – char
 Godina završetka škole : God_zav_skol
 Godina studija koja se upisuje : God_upis – int(1) – in(1,2,3,4)
 Način finansiranja studija : Finans – int – in(1,2,3)
 Ponovno upisivanje : Pon – int – in(1,2)
 Godina upisa na fakultet : God_1_upis – int – not null
 Školska sprema oca : SS_Otac – int – in(1,2,3,4,5,6)
 Školska sprema majke : SS_Maj – int – in(1,2,3,4,5,6)
 Aktivnost roditelja(izdržavaoca) : Akt_rod – int – BETWEEN 1,9
 Zanimanje roditelja(izdržavaoca) : ZanRod – char
 Šifra zanimanja roditelja : Sif_zan_rod – int – in(1,2)
 Zanimanje studenta : ZanStud – char
 Šifra zanimanja studenta : Sif_zan_stud – int – in(1,2)
 Student u radnom odnosu : RO – int – in(1,2)
 Mesto upisa : MestoUp – char
 Datum upisa : DatumUp – Date
 Potpis studenta : Potpis_stud – char
 Napomena : Nap – String

Rečnik podataka za **dosije studenta**

dosije : < Broj_dos , ImePrezime_stud , JMBG , Datum_rodj , Mesto_rodj , Status_stud ,
 Naziv_ods , Sifra_ods , Semestar , Sk_god , { < Sif_pred , Naziv_pred ,
 Ocena_pred , Datum_isp , Prof , ProfesorID > } , Adresa_stud >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Broj dosijea : Broj_dos – string
 Ime i prezime studenta : ImePrez_stud – char (30)
 Matični broj studenta : JMBG – int – not null
 Datum rođenja : Datum_rodj – Date
 Mesto rođenja : Mesto_rodj – char (20)

Status studenta : Status_stud – char (20)
Naziv odseka : Naziv_ods – char (30)
Šifra odseka : Sifra_ods – int – not null
Šifra predmeta : Sif_pred – int – not null
Naziv predmeta : Naziv_pred – char (40)
Ocena predmeta : Ocena_pred – int – in (6,7,8,9,10)
Datum ispita : Datum_isp – Date
Predsednik ispitne komisije : Prof – char (30)
Semestar : Semestar – int – not null
Školska godina : Sk_god – string
Šifra Profesora : ProfesorID - String
Adresa studenta : Adresa_stud - char

Rečnik podataka za **zahtev za priznavanje ispita sa drugog fakulteta ili više škole**

zahtev_priz : < Broj_z , Datum_z , ImePrez_stud , JMBG , Datum_rodj , Mesto_rodj ,
ImeFakult , SifFkl , Smer , Tekst_z , PotpisStud , Potpis_OL , SifOvl >

Naziv podatka : Drugi naziv podatka – Domen – Ograničenje

Broj zahteva : Broj_z – int – not null
Datum zahteva : Datum_z – Date
Ime i prezime studenta : ImePrezime_stud – char
Maticni broj studenta : JMBG – int(13) – not null
Datum rođenja studenta : Datum_rodj – Date
Mesto rođenja studenta : Mesto_rodj – char
Naziv fakulteta sa koga se vrši priznavanje ispita : ImeFakult – char
Smer : Smer – char
Tekst zahteva : Tekst_z – String
Potpis studenta : PotpisStud – char
Potpis ovlašćenog lica : Potpis_OL - char
Šifra fakulteta : SifFkl – int – not null
Šifra ovlašćenog lica : SifOvl – int – not null

SV_OBR_PRI : < SV_OBRAZAC >
SV_OBR_ZAVR : < SV_OBRAZAC >
SV_OBR_ZAV : < SV_OBRAZAC >
SV_OBR_MIR : < SV_OBRAZAC >

ZAHTEV_MIR : < ZAHTEV >
ZAHTEV_PNP : < ZAHTEV >
ZAHTEV_PRIZ_F : < ZAHTEV_PRIZ >
ZAHTEV_PO : < ZAHTEV_PRIZ >

IZVESTAJ_ST : < IZVESTAJ >
IZVESTAJ_S : < IZVESTAJ >
IZVESTAJ_PNP : < IZVESTAJ >
IZVESTAJ_PO : < IZVESTAJ >

POTV_UPLATA_PRI : < POTV_UPLATA >
POTV_UPLATA_ZAV : < POTV_UPLATA >
POTV_UPLATA_DF : < POTV_UPLATA >
POTV_UPLATA_PNP : < POTV_UPLATA >
POTV_UPLATA_PO : < POTV_UPLATA >
POTV_UPLATA_MIR : < POTV_UPLATA >

PREDLOG_ZAVR : < PREDLOG >
PREDLOG_PO : < PREDLOG >

ODLUKA_F : < ODLUKA >
ODLUKA_PNP : < ODLUKA >
ODLUKA_PO : < ODLUKA >

DOKAZ_ISPF : < DOKAZ_POL_ISP >

3. NORMALIZACIJA

Normalizacija je postupak projektovanja logičke strukture baze podataka. Uobičajeno je da se koristi za projektovanje logičke strukture relacionog modela, pa će i ovde normalne forme biti definisane u terminologiji relacionog modela.

Najopštije rečeno, dobra je ona struktura baze podataka u kojoj je logička redundansa minimalna. Loše projektovana logička struktura baze podataka ne dovodi samo do anomalija u njenom održavanju i to do :

- anomalija u dodavanju
- anomalija u izbacivanju
- anomalija u ažuriranju

Postupkom normalizacije logička struktura baze podataka se dovodi u takav oblik (ili, drugim rečima, relacije se dovode u normalne forme) u kome se izbegavaju anomalije u održavanju i problemi u izveštavanju.

Relacija R je u *Prvoj normalnoj formi (1NF)* ako su sve vrednosti njenih atributa atomske.

Definicija *funkcionalne zavisnosti* se može dati i na sledeći način:

Atribut Y relacije R je funkcionalno zavisian od atributa X relacije R ako i samo ako kad god dve n-torke relacije R imaju istu x-vrednost one moraju imati istu i y-vrednost.

Koncept *potpune funkcionalne zavisnosti* se definiše na sledeći način:

Atribut Y relacije R je potpuno funkcionalno zavisian od atributa X relacije R ako je funkcionalno zavisian od atributa X, a nije funkcionalno zavisian ni od jednog pravog podskupa atributa X.

Koncept *tranzitivne funkcionalne zavisnosti* se definiše na sledeći način:

Data je relacija R sa atributima A, B i C, moguće složenim. Ako u relaciji R važi:

A ---> B
B ---> C
A ---> C
B -/-> A
C -/-> A

atribut C je tranzitivno funkcionalno zavisian od atributa A.

Jednostavnije rečeno, atribut C je tranzitivno funkcionalno zavisian od atributa A ako je funkcionalno zavisian od A i ako je funkcionalno zavisian od nekog atributa B koji je i sam funkcionalno zavisian od A.

Relacija R je u *Drugoj normalnoj formi (2NF)* ako i samo ako je u 1NF i svi njeni neključni atributi potpuno i funkcionalno zavise od primarnog ključa. Iz definicije 2NF i definicije potpune funkcionalne zavisnosti očigledno je da je svaka relacija sa prostim primarnim ključem u 2NF, jer prosti ključ nema semantički moguć pravi podskup.

Relacija R je u *Trećoj normalnoj formi (3NF)* ako i samo ako je u 2NF i ako svi njeni neključni atributi netranzitivno funkcionalno zavise od primarnog ključa. Relacija R je u 3NF ako svi njeni atributi daju jednoznačne činjenice o celom ključu i samo o celom ključu.

Determinanta relacije R je bilo koji atribut, prost ili složen, od koga neki drugi atribut u relaciji potpuno funkcionalno zavisi.

Relacija R je u *Boyce-Codd-ovoj normalnoj formi (BCNF)* ako i samo ako su sve determinante u relaciji i kandidati za ključ. Definicija BCNF je striktno stroža od definicije 2NF i 3NF. To znači da je svaka relacija koja je u BCNF sigurno i u 2NF i 3NF. Obrnuto ne važi.

Formalna definicija *višeznačnih zavisnosti* može se dati na sledeći način:

U relaciji R(A,B,C) postoji višeznačna zavisnost $A \twoheadrightarrow B$ ako i samo ako kad god u njoj postoje n-torke $\langle a,b,c \rangle$ i $\langle a,b',c' \rangle$, postoje takođe i n-torke $\langle a,b,c' \rangle$ i $\langle a,b',c \rangle$. Atributi A, B i C mogu biti složeni.

Relacija R je u *Četvrtoj normalnoj formi (4NF)* ako i samo ako kad god postoji višeznačna funkcionalna zavisnost, na primer $A \twoheadrightarrow B$, tada svi atributi relacije moraju takođe biti funkcionalno zavisni od A. Gornja definicija u osnovi kaže da u relaciji u 4NF sve funkcionalne i višeznačne zavisnosti moraju biti funkcionalne zavisnosti atributa od ključa. Ili, drugim rečima, relacija je u 4NF ako je u BCNF i ako su sve višeznačne zavisnosti funkcionalne zavisnost od primarnog ključa. Za praktičnu primenu može se dati i sledeća, neformalna i nedovoljno precizna definicija 4NF:

Relacija R je u 4NF ako u njoj nisu date dve (ili više) nezavisne višeznačne činjenice.

Relacija je u *Petoj normalnoj formi 5NF* onda kad se njen informacioni sadržaj ne može rekonstruisati iz relacija nižeg stepena, s tim što se slučaj relacija nižeg stepena sa istim ključem isključuje. Kako je višeznačna zavisnost specijalan slučaj zavisnosti spajanja, ako je relacija u 5NF ona je sigurno i u 4NF, pa samim tim i u svim ostalim. Obrnuto ne važi. 5NF se često naziva i "projekcija-spajanje normalna forma".

Primena normalizacije u projektovanju baze podataka

Projektovanju logičke strukture baze podataka, na osnovu teorije normalizacije, može se pristupiti na dva načina:

(1) Analiza relacija. Polazi se od nekog skupa nenormalizovanih relacija, svaka relacija ovoga skupa se dekompozicijom bez gubljenja informacija, svodi na neku od normalnih formi, a zatim se vrši "konsolidacija relacija", integrisanjem onih relacija, iz tako dobijenog skupa, koje imaju isti ključ.

(2) Sinteza relacija. Baza podataka se tretira kao skup podataka i njihovih međusobnih veza. Polazi se od skupa tih podataka (atributa) i definisanih zavisnosti između njih (funkcionalne, višeznačne, zavisnosti spajanja) i primenom teorije ovih zavisnosti direktno sintetizuju relacije u nekoj od normalnih formi.

Normalizacija **zahteva**

zahtev : < Br_zah , Datum_zah , Naziv_zahtev , Ime_Prez_stud , Br_dos , Sadržaj_zah ,
Napom_zah , Potpis_stud , Potpis_ov_lica , Ovl_ID >

gde su funkcionalne zavisnosti :

Br_zah , Br_dos → Naziv_zahtev , Datum_zah , ImePrez_stud , Sadržaj_zah ,
Napom_zah , Potpis_stud , Ovl_ID , Potpis_ov_lica

Br_zah → Naziv_zahtev , Datum_zah

Br_dos → ImePrez_stud , Potpis_stud

Ovl_ID → Potpis_ov_lica

Relacija zahtev_f je u 1NF jer ne postoji grupa sa ponavljanjem , ali nije u 2NF jer postoje neključni atributi koji nisu u potpunoj funkcionalnoj zavisnosti od ključnih atributa.

Zahtev1 (Br_zah , Naziv_zahtev , Datum_zah)

Dosije (Br_dos , ImePrezime_stud , Potpis_stud)

Sastav (Br_zah , Br_dos , Sadržaj_zah , Napom_zah , Ovl_ID , Potpis_ov_lica)

Sada su relacije u 2NF jer su svi neključni atributi u potpunoj funkcionalnoj zavisnosti od ključnih atributa. Relacija Sastav nije u 3NF jer postoji tranzitivna zavisnost neključnih atributa od primarnog ključa.

Zahtev1 (Br_zah , Naziv_zahtev , Datum_zah)

Dosije (Br_dos , ImePrezime_stud , Potpis_stud)

Sastav1 (Br_zah , Br_dos , Sadržaj_zah , Napom_zah , Ovl_ID)

Odobrio (Ovl_ID , Potpis_ov_lica)

Sada su sve relacije u 3NF , a takodje i u BCNF jer su sve determinante i kandidati za ključ.

Normalizacija dokaza o položenim ispitima

dokaz_pol_isp : < Sif_dok , ImePrez_stud , JMBG , Datum_dok , SifraF , NazivF ,
{<RB_isp , Sif_isp , Naziv_isp , Ocena_isp , Datum_isp , Profesor_isp ,
ProfesorID >} , Potpis_ov_lica , Ovl_ID >

gde su funkcionalne zavisnosti :

Sif_dok → ImePrez_stud , JMBG , Datum_dok , SifraF , NazivF , Ovl_ID ,
Potpis_ov_lica

Sif_dok , RB_isp → Sif_isp , Naziv_isp , Ocena_isp , Datum_isp , Profesor_isp ,
ProfesorID

JMBG → ImePrez_stud

SifraF → NazivF

Sif_isp → Naziv_isp

ProfesorID → Profesor_isp

Ovl_ID → Potpis_ov_lica

Relacija dokaz_pol_isp nije u 1NF zato što postoji grupa sa ponavljanjem.

Dokaz (Sif_dok , ImePrez_stud , JMBG , Datum_dok , SifraF , NazivF , Ovl_ID ,
Potpis_ov_lica)

Stavka (Sif_dok , RB_isp , Sif_isp , Naziv_isp , Ocena_isp , Datum_isp , Profesor_isp ,
ProfesorID)

Relacije Dokaz i Stavka su u 1NF jer više nema grupe sa ponavljanjem, a takođe su u 2NF jer svi njihovi neključni atributi potpuno funkcionalno zavise od ključnih atributa. Nisu u 3NF jer postoji tranzitivna zavisnost neključnih atributa od primarnog ključa.

Dokaz1 (Sif_dok , JMBG , Datum_dok , SifraF , Ovl_ID)

Student (JMBG , ImePrezime_stud)

Fakultet (SifraF , NazivF)

Stavka1 (Sif_dok , RB_isp , Sif_isp , Ocena_isp , Datum_isp , Profesor_ID)

Predmet (Sif_isp , Naziv_isp)

Profesor (ProfesorID , Profesor_isp)

Potpisao (Ovl_ID , Potpis_ov_lica)

Sada su relacije u 3NF a i u BCNF jer su sve determinante i kandidati za ključ.

Normalizacija odluke dekana

Odluka : < Broj_odl , Datum_odl , Naziv_odl , ImePrez_stud , Broj_dos , Sadržaj_odl , Napom_odl , Potpis_ovl_lica , Ovl_ID >

gde važe sledeće funkcionalne zavisnosti :

Broj_odl , Broj_dos → Naziv_odl , Ime_Prez_stud , Broj_dos , Sadržaj_odl , Napom_odl , Datum_odl , Potpis_ovl_lica , Ovl_ID

Broj_odl → Naziv_odl , Datum_odl

Broj_dos → ImePrez_stud

Ovl_ID → Potpis_ovl_lica

Relacija Odluka je u 1NF jer su vrednosti njenih atributa atomske.

Međutim, relacija nije u 2NF jer postoji nepotpuna funkcionalna zavisnost neključnih atributa od primarnog ključa.

Odluka1 (Broj_odl , Broj_dos , Sadržaj_odl , Napom_odl , Potpis_ovl_lica , Ovl_ID)

Sifra (Broj_odl , Naziv_odl , Datum_odl)

Student (Broj_dos , ImePrez_stud)

Sada su sve relacije u 2NF jer svi neključni atributi potpuno funkcionalno zavise od primarnog ključa.

Nisu u 3NF jer postoji tranzitivna funkcionalna zavisnost neključnih atributa od primarnog ključa.

Odluka2 (Broj_odl , Broj_dos , Sadržaj_odl , Napom_odl , Ovl_ID)

Sifra (Broj_odl , Naziv_odl , Datum_odl)

Student (Broj_dos , ImePrez_stud)

Potpisao (Ovl_ID , Potpis_ovl_lica)

Sve relacije su u 3NF i BCNF jer su sve determinante i kandidati za ključ.

Normalizacija izveštaja o priznatim ispitima

Izvestaj : <Broj_izv , Datum_izv , ImePrez_stud , JMBG , Sif_fk , Naziv_fk ,
{<RB_I , Ime_pol_ispita , Sif_priz_isp , Naziv_priz_isp >} ,
{<RB_K , Ime_cl_kom , Cl_komID , >} ,
Potpis_ovl , Ovl_ID >

gde važe sledeće funkcionalne zavisnosti :

Broj_izv , RB_I → Ime_pol_ispita , Sif_priz_isp , Naziv_priz_isp
Broj_izv , RB_K → Ime_cl_kom , Cl_komID
Broj_izv → Datum_izv , ImePrez_stud , JMBG , Sif_fk , Naziv_fk , Potpis_ovl , Ovl_ID
Ovl_ID → Potpis_ovl
JMBG → ImePrezime_stud
Sif_fk → Naziv_fk
Sif_priz_isp → Naziv_priz_isp
Cl_komID → Ime_cl_kom

Relacija Izvestaj nije u 1NF jer postoje dve grupe sa ponavljanjem.

Ispiti (Broj_izv , RB_I , Ime_pol_ispita , Sif_priz_isp , Naziv_priz_isp)
Komisija (Broj_izv , RB_K , Ime_cl_kom , Cl_komID)
Izvestaj1 (Broj_izv , Datum_izv , ImePrez_stud , JMBG , Sif_fk , Naziv_fk , Potpis_ovl ,
Ovl_ID)

Relacije Ispiti , Komisija i Izvestaj1 su u 1NF jer su eliminisane grupe sa ponavljanjem. Takođe te iste relacije su u 2NF jer svi neključni atributi potpuno funkcionalno zavise od primarnog ključa. Relacije nisu u 3NF jer postoji tranzitivna funkcionalna zavisnost.

Ispiti1 (Broj_izv , RB_I , Ime_pol_ispita , Sif_priz_isp)
PriznatIspit (Sif_priz_isp , Naziv_priz_isp)
Komisija1 (Broj_izv , Rb_K , Cl_komID)
Clan (Cl_komID , Ime_cl_kom)
Izvestaj2 (Broj_izv , Datum_izv , JMBG , Sif_fk , Ovl_ID)
Student (JMBG , ImePrez_stud)
Fakultet (Sif_fk , Naziv_fk)
Potpisao (Ovl_ID , Potpis_ovl)

Sada su sve relacije u 3NF jer nema više tranzitivne zavisnosti. Relacije su takođe u BCNF jer su sve determinante i kandidati za ključ.

Normalizacija **potvrde o uplati**

potv_uplata : < ImePrezime_stud , BrDos , Adresa_stud , Svrha_upl , Sif_sv ,
Dat_prijem , Sif_plac , Iznos_upl , BrRac , Poz_br , Sif_p >

gde su funkcionalne zavisnosti :

BrDos , Sif_p → ImePrezime_stud , Adresa_stud , Svrha_upl , Sif_sv , Dat_prijem ,
Sif_plac , Iznos_upl , BrRac , Poz_br
BrDos → ImePrezime_stud , Adresa_stud
Sif_sv → Svrha_upl

Relacija potv_o_upl jeste u 1NF zato što ne postoji grupa sa ponavljanjem.
Nisu u 2NF jer postoje neključni atributi koji nepotpuno funkcionalno zavise od primarnog ključa.

Potvrda (BrDos , Sif_p , Sif_sv , Svrha_upl , Dat_prijem , Sif_plac , Iznos_upl , Poz_br ,
BrRac)
Uplatilac (BrDos , ImePrezime_stud , Adresa_stud)

Relacije Potvrda i Uplatilac su u 2NF jer svi neključni atributi potpuno funkcionalno zavise od primarnog ključa. Međutim, nisu u 3NF jer postoji tranzitivna funkcionalna zavisnost.

Potvrda1 (BrDos , Sif_p , Sif_sv , Dat_prijem , Sif_plac , Iznos_upl , Poz_br , BrRac)
Svrha (Sif_sv , Svrha_upl)
Uplatilac (BrDos , ImePrezime_stud , Adresa_stud)

Sada su relacije u 3NF jer svi neključni atributi netranzitivno zavise od primarnog ključa.
Takodje su u BCNF jer su sve determinante i kandidati za ključ.

Normalizacija za **dosije studenta**

dosije : < Broj_dos , ImePrezime_stud , JMBG , Datum_rodj , Mesto_rodj , Status_stud , Naziv_ods , Sifra_ods , Semestar , Sk_god { < Sif_pred , Naziv_pred , Ocena_pred , Datum_isp , Prof , ProfesorID > } >

gde su funkcionalne zavisnosti :

Broj_dos → ImePrezime_stud , JMBG , Adresa_stud , Datum_rodj , Mesto_rodj , Status_stud , Naziv_ods , Sifra_ods , Semestar , Sk_god
Broj_dos , Sif_pred → Naziv_pred , Ocena_pred , Datum_isp , Prof , ProfesorID
JMBG → ImePrezime_stud , Datum_rodj , Mesto_rodj
Sif_pred → Naziv_pred
ProfesorID → Profesor
Sifra_ods → Naziv_ods

Relacija dosije nije u 1NF jer postoji grupa sa ponavljanjem.

Dosije1 (Broj_dos , ImePrezime_stud , JMBG , Adresa_stud , Datum_rodj , Mesto_rodj , Status_stud , Naziv_ods , Sifra_ods , Semestar , Sk_god)
Prijava (Broj_dos , Sif_pred , Naziv_pred , Ocena_pred , Datum_isp , Prof , ProfesorID)

Relacije Dosije1 i Prijava su u 1NF jer nema više grupe sa ponavljanjem.

Relacija Dosije1 je u 2NF jer sadrži prost ključ, dok Prijava nije u 2NF jer imamo situaciju da Naziv_pred funkcionalno zavisi od atributa Sif_pred koji je, kako se vidi ,deo primarnog ključa.

Prijava1 (Broj_dos , Sif_pred , Ocena_pred , Datum_isp , Prof , ProfesorID)
Predmet (Sif_pred , Naziv_pred)

Sada su relacije Prijava1 i Predmet u 2NF jer svi njihovi neključni atributi potpuno funkcionalno zavise od primarnog ključa.

Relacije Dosije1 , Prijava1 nisu u 3NF jer postoji tranzitivna funkcionalna zavisnost neključnih atributa od primarnog ključa.

Dosije2 (Broj_dos , JMBG , Status_stud , Sifra_ods , Semestar , Sk_god)
Student (JMBG , ImePrezime_stud , Adresa_stud , Datum_rodj , Mesto_rodj)
Odsek (Sifra_ods , Naziv_ods)
Prijava2 (Broj_dos , Sif_pred , Ocena_pred , Datum_isp , ProfesorID)
Profesor (ProfesorID , Prof)
Predmet (Sif_pred , Naziv_pred)

Sve relacije su u 3NF jer svi njihovi neključni atributi netranzitivno zavise od primarnog ključa. Takodje su u BCNF jer su sve determinante i kandidati za ključ.

Normalizacija zahteva za priznavanje ispita sa drugog fakulteta ili više škole

zahtev_priz : < Broj_z , Datum_z , ImePrez_stud , JMBG , Datum_rodj , Mesto_rodj ,
ImeFakult , SifFkl , Smer ,SifSmer , Tekst_z , , Potpis_OL , SifOvl >

gde su funkcionalne zavisnosti :

Broj_z , JMBG → Datum_z , ImePrez_stud , Datum_rodj , Mesto_rodj ,
ImeFakult , SifFkl , Smer ,SifSmer , Tekst_z , Potpis_OL , SifOvl >

Broj_z → Datum_z

JMBG → ImePrez_stud , Datum_rodj , Mesto_rodj , SifFkl , ImeFakult , SifSmer , Smer

SifFkl → ImeFakult

SifSmer → Smer

SifOvl → Potpis_OL

Relacija zahtev_priz jeste u 1NF jer ne postoji grupa sa ponavljanjem.

Nije u 2NF jer postoje takvi neključni atributi koji nepotpuno funkcionalno zavise od primarnog ključa.

Sadrzaj (Broj_z , JMBG , Tekst_z , SifOvl , Potpis_OL)

Student (JMBG , ImePrez_stud , Datum_rodj , Mesto_rodj , SifFkl , ImeFakult , SifSmer , Smer)

Zahtev1 (Broj_z , Datum_z)

Relacije Zahtev1, Sadrzaj i Student su u 2NF jer svi neključni atributi potpuno funkcionalno zavise od primarnog ključa. To se u slučaju relacija Student i Zahtev1 podrazumeva jer sadrže prost ključ pa su one sigurno u 2NF. Relacija Zahtev1 jeste u 3NF dok ostale nisu jer postoji tranzitivna funkcionalna zavisnost neključnih atributa od primarnog ključa.

Sadrzaj1 (Broj_z , JMBG , Tekst_z , SifOvl)

Potpisao (SifOvl , Potpis_OL)

Student 1(JMBG , ImePrez_stud , Datum_rodj , Mesto_rodj , SifFkl , SifSmer)

Fakultet (SifFkl , ImeFakult)

Odsek (SifSmer , Smer)

Zahtev1 (Broj_z , Datum_z)

Sada su sve relacije u 3NF jer je eliminisana tranzitivna zavisnost. Takođe su u BCNF jer su sve determinante i kandidati za ključ.

Normalizacija **predloga komisije o priznavanju ispita** (sa drugog fakulteta)

predlog : < PredlogID , DatumPred , ImePrezime_stud , JMBG , Naziv_fkl , Sif_fkl , Univ , UnivID , Drzava , DrzavaID ,
{<Rb_I , Ime_pol_isp , Sif_priz_isp , Naziv_priz_isp >}
{<RB_K , ClanKomID , ImeClanaKom >}>

gde važe sledeće funkcionalne zavisnosti :

PredlogID → DatumPred , ImePrezime_stud , JMBG , Naziv_fkl , Sif_fkl , Univ , UnivID , Drzava , DrzavaID
PredlogID , Rb_I → Ime_pol_isp , Sif_priz_isp , Naziv_priz_isp
PredlogID , RB_K → ClanKomID , ImeClanaKom
JMBG → ImePrez_stud
Sif_fkl → Naziv_fkl
UnivID → Univ
DrzavaID → Drzava
Sif_priz_isp → Naziv_priz_isp
ClanKomID → ImeClanaKom

Relacija predlog nije u 1NF jer postoje dve grupe sa ponavljanjem.

Predlog1 (PredlogID , DatumPred , ImePrezime_stud , JMBG , Naziv_fkl , Sif_fkl , Univ , UnivID , Drzava , DrzavaID)
StavkaIspit (PredlogID , Rb_I , Ime_pol_isp , Sif_priz_isp , Naziv_priz_isp)
StavkaKomisija (PredlogID , RB_K , ClanKomID , ImeClanaKom)

Sada su relacije u 1NF jer nema više grupa sa ponavljanjem. Relacije su takođe u 2NF jer svi neključni atributi potpuno funkcionalno zavise od primarnog ključa. Nisu u 3NF jer postoji tranzitivna funkcionalna zavisnost neključnih atributa od primarnog ključa.

Predlog2 (PredlogID , DatumPred , JMBG , Sif_fkl , UnivID , DrzavaID)
Student (JMBG , ImePrezime_stud)
Fakultet (Sif_fkl , Naziv_fkl)
Univerzitet (UnivID , Univ)
StavkaIspit1 (PredlogID , Rb_I , Ime_pol_isp , Sif_priz_isp)
Predmet (Sif_priz_isp , Naziv_priz_isp)
StavkaKomisija1 (PredlogID , RB_K , ClanKomID)
Clan (ClanKomID , ImeClanaKom)

Sada su sve relacije u 3 NF , a takođe u BCNF jer su sve determinante i kandidati za ključ.

4. DIJAGRAMI OBJEKTI-VEZE

Model objekti-veze je najpopularniji i u praksi najviše korišćeni semantički model podatka (model podataka treće generacije). Postoji više različitih verzija ovog modela. Ovde se izlaže jedna specifična verzija ovog modela, Prošireni model objekti-veze (PMOV) u kome se definišu i jezik za specifikaciju ograničenja i operacije modela, čime se dobija alat za formalnu specifikaciju IS.

Kao što je ranije rečeno, model podataka predstavlja intelektualno sredstvo pomoću koga se prikazuje kako su podaci o nekom realnom sistemu međusobno povezani. Model podataka obezbeđuje interpretaciju podataka o posmatranom realnom sistemu. Interpretacija podataka se u nekom modelu podataka ostvaruje kroz tri njegove osnovne komponente:

- (1) Strukturu podataka, preko koje se predstavljaju statičke karakteristike sistema;
- (2) Ograničenja - logička ograničenja na vrednosti podatka koja u svakom trenutku posmatranja (stacionarnom stanju) treba da budu zadovoljena. Ova dodatna ograničenja na podatke koja nisu obuhvaćena samom strukturom nazivaju se i vrednosnim pravilima integriteta modela podataka.
- (3) Operacije nad konceptima strukture modela, preko kojih je moguće opisati dinamiku sistema, odnosno dati interpretaciju podataka kroz obradu podataka u modelu podataka.

Struktura Proširenog modela objekti-veze

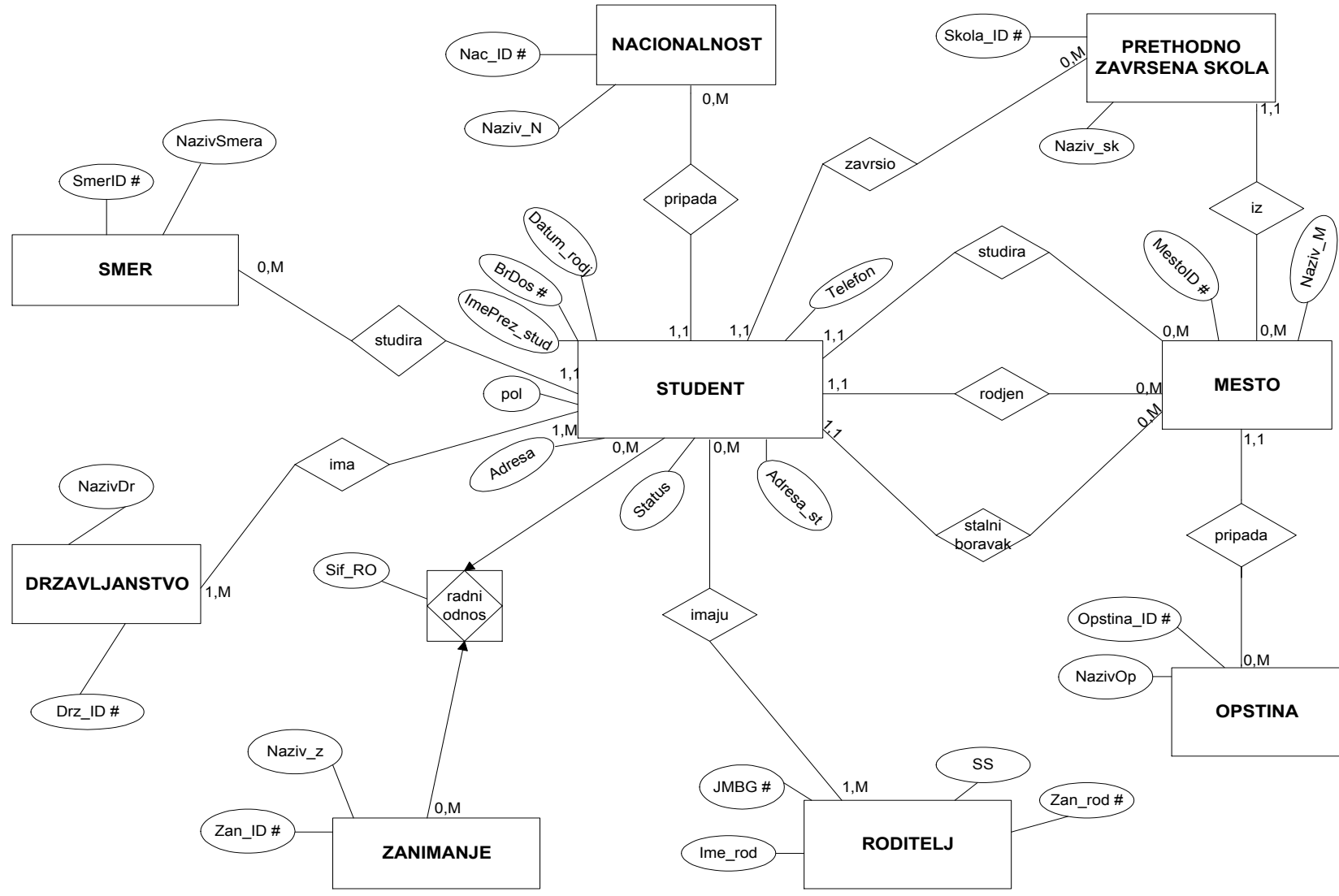
Struktura PMOV predstavlja se *dijagramima objekti-veze* (DOV).

U modelu PMOV sistem se opisuje kao skup objekata i njihovih veza. Objekat u modelu može da predstavlja neki fizički objekat realnog sistema (konkretan proizvod, konkretnog radnika (Jovan Jovanović), vremenski trenutak ili period i slično), ili neki koncept (klasa daktilografije, smer studija i slično)

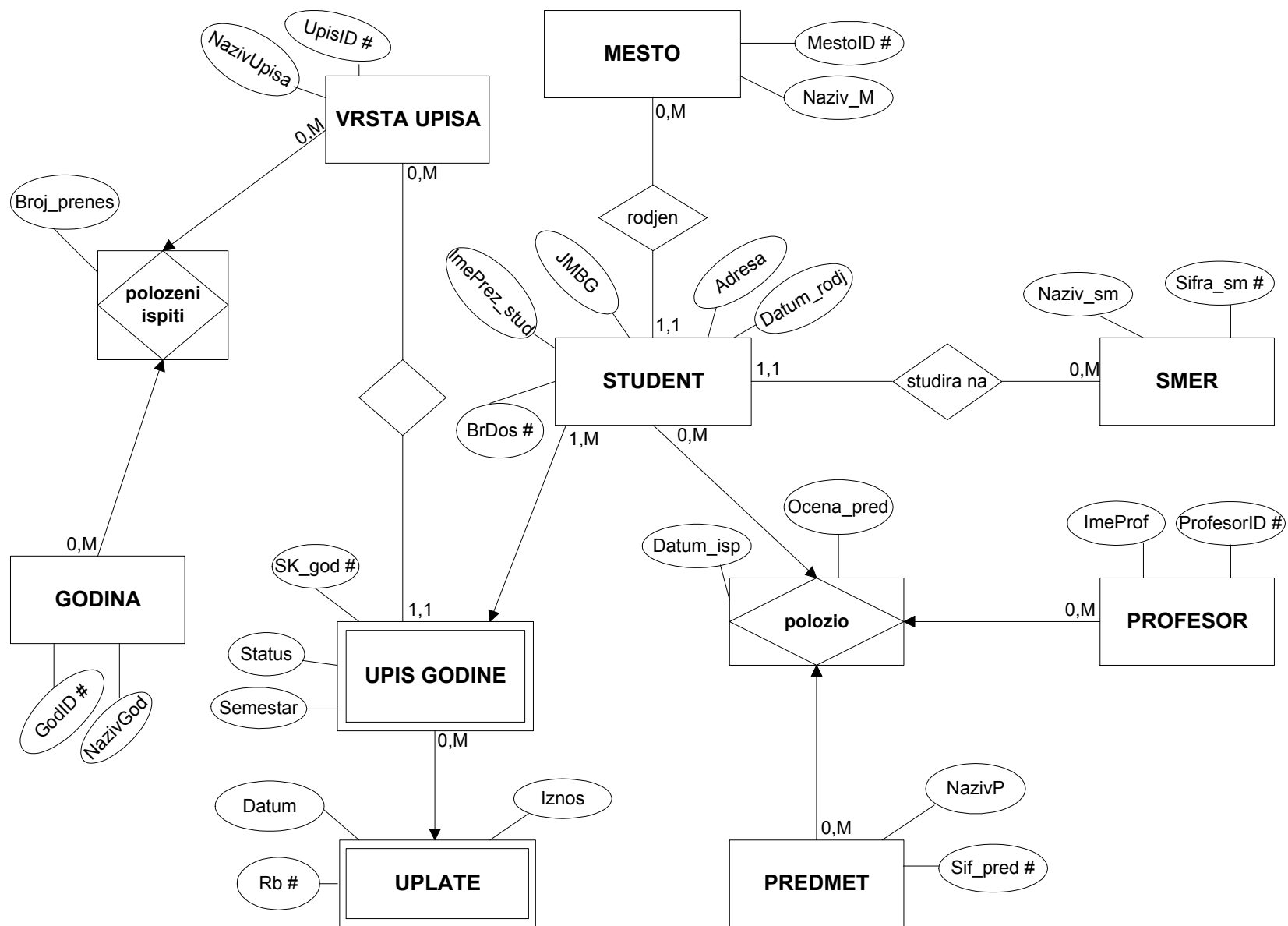
Na dijagramima objekti veze (DOV) klase objekata se prikazuju pravougaonicima. Svaka klasa definiše istovremeno i tip objekta.

Veze u modelu opisuju način povezivanja (uzajamna dejstva) objekata. Apstrakcija klasifikacije može se primeniti i na veze i definisati pojmovi tipa, klase i pojavljivanja veze.

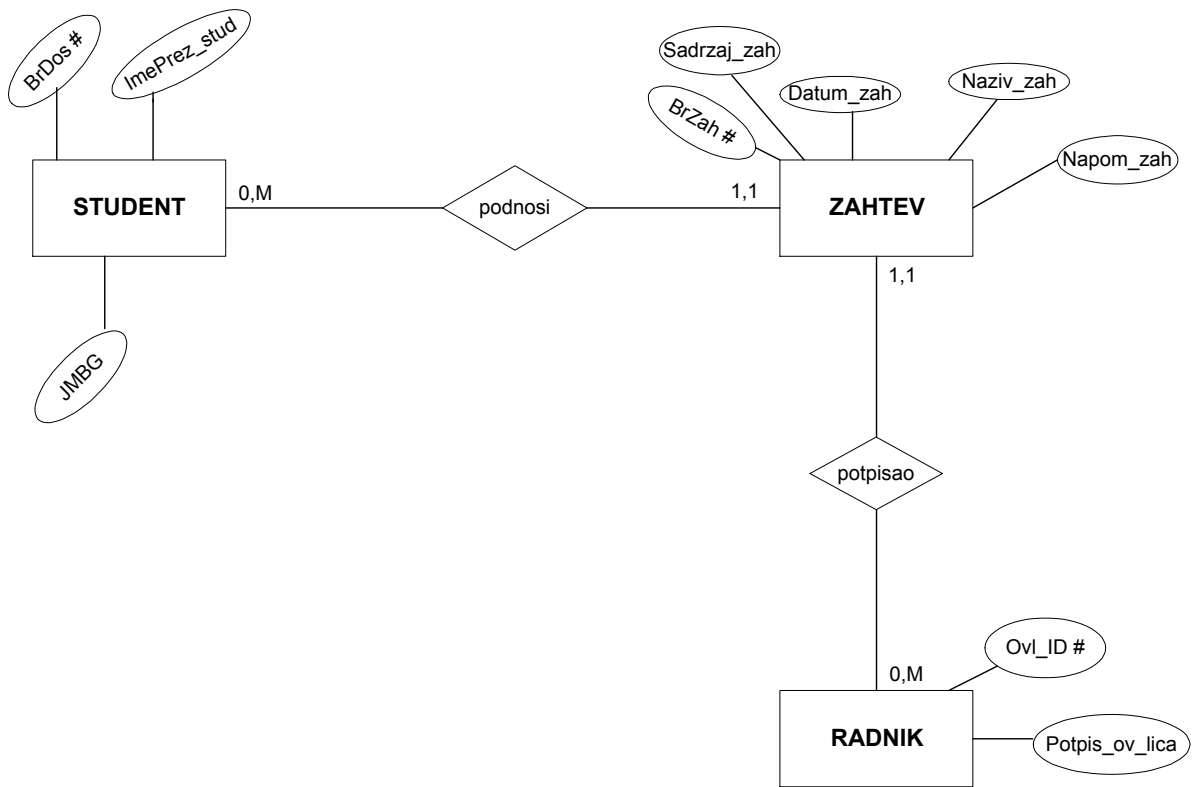
3.1 ŠV obrazac



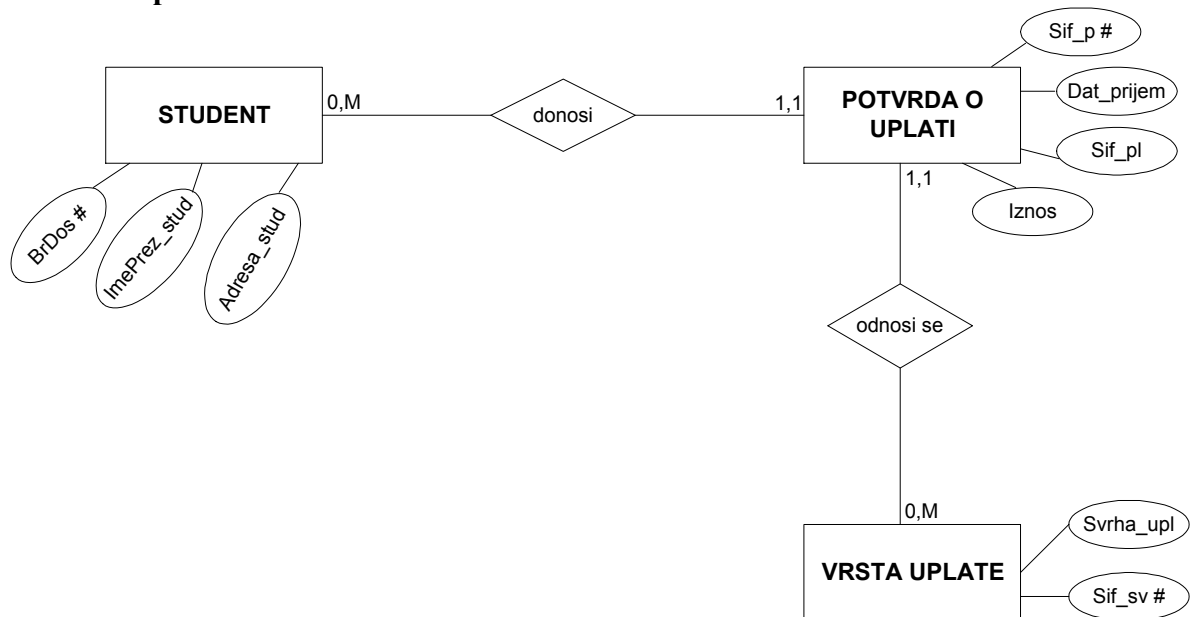
3.2 Dosije studenta



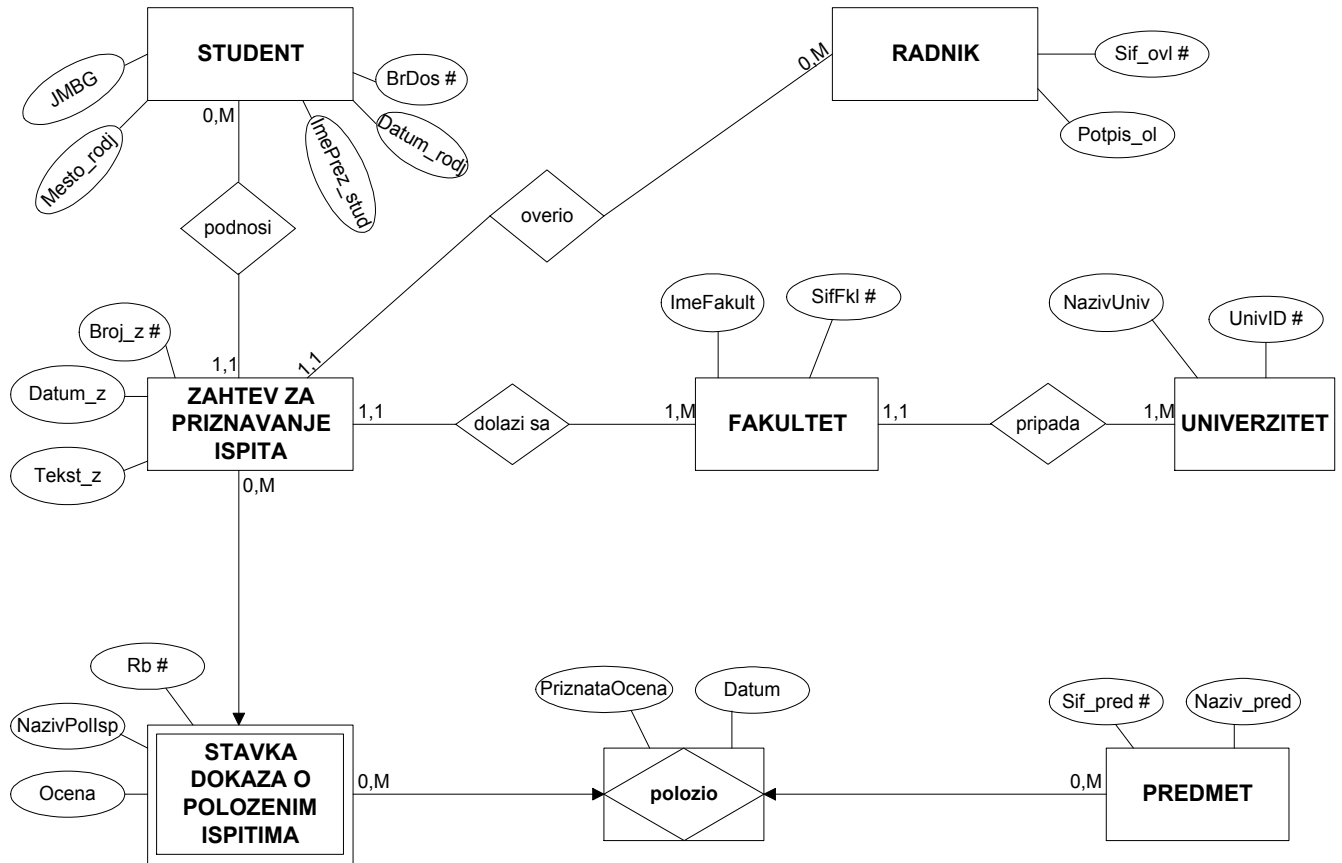
Zahtev



Potvrda o uplati



Zahtev za priznavanje ispita



5. RELACIONI MODEL

Sledeće dve bitne karakteristike čine *relacioni model* najpopularnijim modelom baza podataka:

1) Struktura modela je veoma jednostavna, prihvatljiva svakom korisniku, jer relaciona baza podataka predstavlja skup tabela. I same operacije, koje iz skupa datih tabela (baze podataka) generišu izlaz (takođe tabelu), su jednostavne i lako prihvatljive.

2) Moguća je formalno-matematička interpretacija tabela. Tabela se može definisati kao matematička relacija i zatim iskoristiti bogata teorijska osnova odgovarajućeg matematičkog aparata.

Kartezijanski (Dekartov) proizvod. Neka je data kolekcija skupova D_1, D_2, \dots, D_n (ne neophodno različitih). Kartezijanski proizvod ovih n skupova

$$D_1 \times D_2 \times \dots \times D_n$$

je skup svi mogućih uređenih n -torki

$$\langle d_1, d_2, \dots, d_n \rangle, \text{ tako da je } d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n.$$

Relacija. Relacija definisana na n skupova je podskup Dekartovog proizvoda tih n skupova.

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

Podskup sadrži one n -torke Dekartovog proizvoda koje zadovoljavaju zadatu relaciju.

Domen relacije. Skupovi D_1, D_2, \dots, D_n se nazivaju domenima relacije R .

Stepen relacije. Broj domena na kojima je definisana neka relacija se naziva stepen relacije. (Razlikujemo unarne (na jednom domenu), binarne (na dva domena) i n -arne relacije).

Kardinalnost relacije je broj n -torki u relaciji.

Atribut relacije. Imenovani domen, sa imenom koje definiše ulogu domena u relaciji se naziva atribut relacije.

Pošto je relacija skup, a svaka tabela nije, definišu se sledeći *uslovi koje tabela mora da zadovolji da bi bila relacija*:

- (1) Ne postoje duplikati vrsta tabele;
- (2) Redosled vrsta nije značajan;
- (3) Redosled kolona nije značajan.

Pored toga, da bi se mogao definisati jednostavan skup operacija nad relacijama, definiše se sledeći dodatni uslov:

- (4) Sve vrednosti atributa u relacijama su atomske.

Ako relacija zadovoljava uslov (4) tada je ona u Prvoj normalnoj formi. Svaka relacija u relacionom modelu mora biti u prvoj normalnoj formi. Termin "normalizovana relacija" se koristi za relacije u prvoj normalnoj formi. (Za ostale normalne forme mora se precizirati o kojoj normalnoj formi se radi.)

Ključ relacije se definiše na sledeći način:

Ključ relacije R je takva kolekcija K njenih atributa koja zadovoljava sledeća dva uslova:

- Osobina jedinstvenosti. Ne postoje bilo koje dve n -torke sa istom vrednošću K .
- Osobina neredundantnosti. Ako se bilo koji atribut izostavi iz K , gubi se osobina jedinstvenosti.

Ona kolekcija atributa K koja zadovoljava samo osobinu jedinstvenosti naziva se *nadključ relacije*. Može, u jednoj relaciji postojati više različitih kolekcija K atributa koje zadovoljavaju definiciju ključa. Sve takve kolekcije se nazivaju *kandidati za ključ*. Jedan od kandidata koji se izabere da praktično služi za identifikaciju n-torke relacije se tada naziva *primarni ključ*. Ostali (neizabrani) kandidati se tada nazivaju *alternativnim ključevima*.

Atributi koji učestvuju u ključevima (koji su deo kandidata za ključ) nazivaju se *ključnim atributima*. Ostali atributi u relaciji su *neključni (ili sporedni) atributi*.

Spoljni ključ je atribut (ili grupa atributa) u relaciji R1 čija se vrednost koristi za povezivanje sa vrednošću primarnog ključa u nekoj relaciji R2. Spoljni ključ i njemu odgovarajući primarni ključ moraju biti definisani nad istim domenom. Spoljni ključevi služe da uspostave veze između relacija u relacionoj bazi podataka.

Relaciona baza podataka je kolekcija vremenski promenljivih relacija.

Izvedena relacija (pogled) je relacija koja se može izvesti iz skupa datih baznih i izvedenih relacija, preko operacija koje se definišu nad relacijama.

Bazna relacija je relacija koja se ne može izvesti iz ostalih relacija u relacionoj bazi podataka.

Relacija se može predstaviti kao tabela.

Ekstenzija relacije je skup svih n-torki date relacije, odnosno predstavljanje tabele navođenjem svih vrsta.

Intenzija relacije je generalizacija ekstenzije, ona je vremenski nepromenljiva i označava se na sledeći način:

(Ispred zagrade je naziv relacije, u zagradi su navedeni atributi, a primarni ključ obeležen je masnim otiskom).

Šema relacione baze podataka je predstavljanje strukture relacione baze kao skupa intenzija relacija.

Nula vrednosti. Termin "nula vrednost" (koji ćemo obeležavati sa ?) se koristi da označi "još nepoznatu vrednost" za neki atribut ili "neprimenljivo svojstvo" za neki objekat ili vezu koje predstavlja tabela.

Ograničenja u relacionom modelu

U relacionom modelu je neophodno definisati i ograničenja na vrednosti nekih atributa u pojedinim relacijama, da bi se korektnije opisao realni sistem koga baza podataka modelira. Specifikacija domena atributa već sama po sebi predstavlja ograničenje na vrednosti atributa. Mogu se davati i složenija ograničenja koja definišu granice nekih izvedenih promenljivih, ili koja definišu granice vrednosti jednog atributa u zavisnosti od vrednosti nekih drugih atributa. Ovakva ograničenja zavise od konkretnog sistema i moraju u bazi podataka da budu uvek zadovoljena da bi se očuvao integritet baze podataka.

Postoje *i opšta ograničenja* koja važe za bilo koji relacioni model, koja proizilaze iz načina opisa realnog sistema u relacionom modelu i koja se nazivaju pravilima integriteta relacionog modela. Definišu se dva opšta pravila integriteta relacionog modela:

Pravilo integriteta 1 - Integritet entiteta. Ni jedan atribut koji je primarni ključ ili deo primarnog ključa neke baze relacije nemože da uzme nula vrednost.

Pravilo integriteta 2 - Referencijalni integritet. Ako neka bazna relacija (recimo R2) poseduje spoljni ključ (recimo SK) koji ovu relaciju povezuje sa nekom drugom baznom relacijom (recimo R1), preko primarnog ključa (recimo PK), tada svaka vrednost SK mora biti bilo jednaka nekoj vrednosti PK, ili biti nula vrednost. Relacije R1 i R2 ne moraju biti različite. Referencijalni integritet obezbeđuje korektno povezivanje objekata koji su predstavljeni u relacionom modelu i neformalno se može iskazati i na sledeći način: Ne može objekat koji nije predstavljen u odgovarajućem skupu objekata u bazi podataka da učestvuje u nekoj od veza predstavljenih u bazi podataka.

Komercijalni SUBP, mada se zasnivaju na relacionoj algebri i/ili relacionom računu, poseduju *upitne jezike* sa konstrukcijama koje su mnogo bliže korisniku, mnogo bliže prirodnom jeziku, nego što su konstrukcije same relacione algebre, odnosno računa. Najpoznatiji relacioni upitni jezici su: SQL koji se bazira na kombinaciji relacione algebre i relacionog računa, Quel, koji predstavlja implementaciju relacionog računa n-torki i QBE, koji predstavlja implementaciju relacionog računa domena.

RELACIJE

Dosije studenta

STUDENT (BrDos, Sifra_sm, MestoID, ImePrez_stud, JMBG, Adresa, Datum_rodj)
MESTO (MestoID, Naziv_M)
SMER (Sifra_sm, Naziv_sm)
POLOZENI ISPITI (UpisID, GodID, Broj_prenes)
GODINA (GodID, NazivGod)
VRSTA UPISA (UpisID, NazivUpisa)
UPIS GODINE (BrDos, Skgod, Status, Semestar, UpisID)
UPLATE (Rb, SK_god, BrDos, Datum, Iznos)
POLOZIO (BrDos, Sif_pred, ProfesorID, Datum_isp, Ocena_pred)
PROFESOR (ProfesorID, ImeProf)
PREDMET (Sif_pred, NazivP)

SV obrazac

STUDENT (BrDos, SmerID, MestoID, Nac_ID, Skola_ID, Status
ImePrez_stud, Pol, Adresa, Adresa_st, Datum_rodj)
MESTO (MestoID, Opstina_ID, Naziv_M)
SMER (SmerID, NazivSmera)
NACIONALNOST (Nac_ID, Naziv_N)
PRETHODNO ZAVRŠENA ŠKOLA (Skola_ID, MestoID, Naziv_sk)
OPŠTINA (Opstina_ID, Nazivop)
RODITELJ (Zan_rod, JMBG, BrDos, SS, Ime_Rod)
RADNI ODNOS (BrDOS, Zan_ID, Sif_RO)
ZANIMANJE (Zan_ID, Nazivz)
DRZAVLJANSTVO (Drz_ID, NazivDr)
IMA (BrDos, DrzID)
IMAJU (BrDos, JMBG, Zan_rod)

Zahtev za priznavanje ispita

STUDENT (BrDos, ImePrez_stud, JMBG, Datum_rodj, Mesto_rodj)
RADNIK (Sif_ovl, Potpis_ol)
ZAHTEV ZA PRIZNAVANJE ISPITA (Broj_z, Sif_ovl, SifFkl, BrDos, Datum_z,
Tekst_z)
FAKULTET (SifFkl, UnivID, ImeFakult)
UNIVERZITET (UnivID, NazivUniv)
STAVKA DOKAZA O POLOZENIM ISPITIMA (Broj_z, Rb, NazivPollsp, Ocena)
POLOZIO (Rb, Sif_pred, Broj_z, PriznataOcena, Datum)
PREDMET (Sif_pred, Naziv_pred)

Zahtev

STUDENT (BrDos, ImePrez_stud, JMBG)

RADNIK (Ovl_ID, Potpis_ov_lica)

ZAHTEV (BrZah, BrDos, Ovl_ID, Sadržaj_zah, Datum_zah, Naziv_zah, Napom_zah)

Potvrda o uplati

STUDENT (BrDos, ImePrez_stud, Adresa_stud)

VRSTA UPLATE (Sif_sv, Svrha_upl)

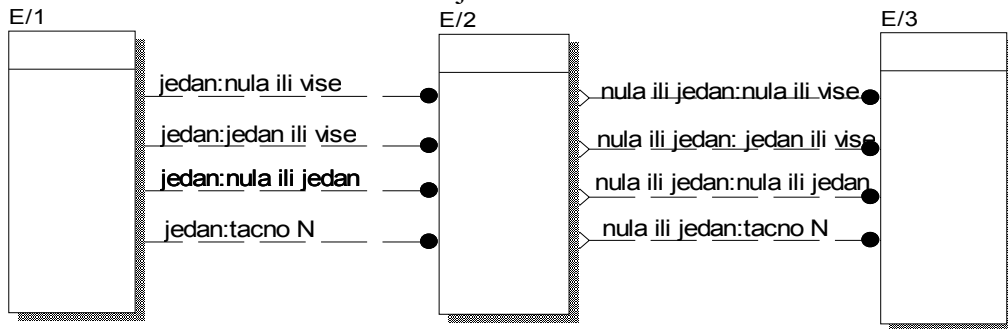
POTVRDA O UPLATI (Sif_p, BrDos, Sif_sv, Dat_prijem, Sif_pl, Iznos)

6. IDEF1X DIJAGRAMI

Prošireni model objekti_ veze (PMOV) i IDEF1X koriste Dijagram objekti_ veze (DOV) ,kao osnovno grafičko sredstvo za iskazivanje svoje strukture.

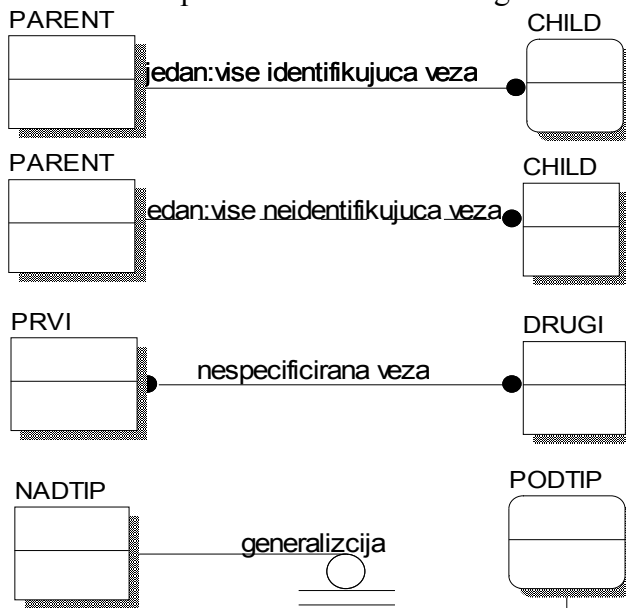
Objekti po IDEF1X standardu mogu biti nezavisni (jaki) i zavisni (slabi). Jaki objekti se predstavljaju simbolom pravougaonika sa oštrim ivicama, a slabi pravougaonikom sa zaobljenim ivicama.

Po IDEF1X sintaksi dozvoljene su sledeće *kardinalnosti*:



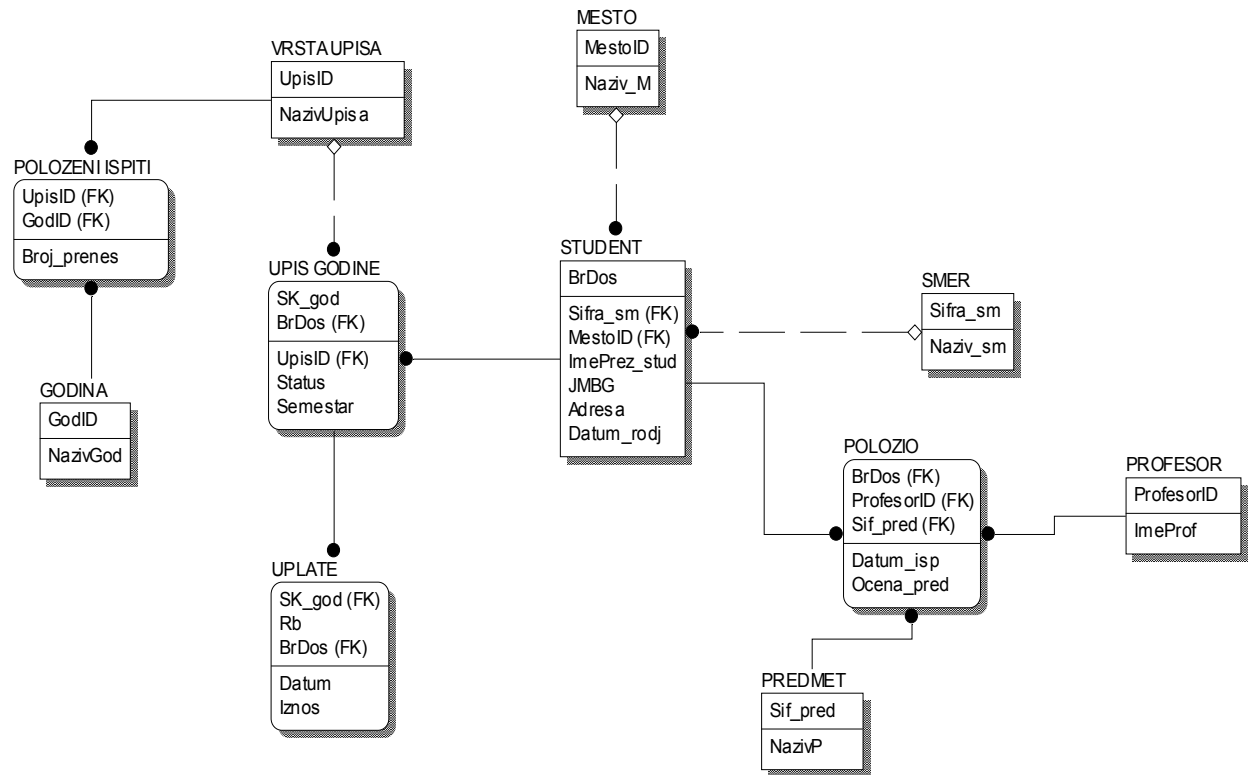
Intervalne kardinalnosti nisu podržane pa se indirektno predstavljaju preko nespecificirane veze gde su kao komentari navedene neophodne kardinalnosti .

Veze po IDEF1X sintaksi mogu biti:

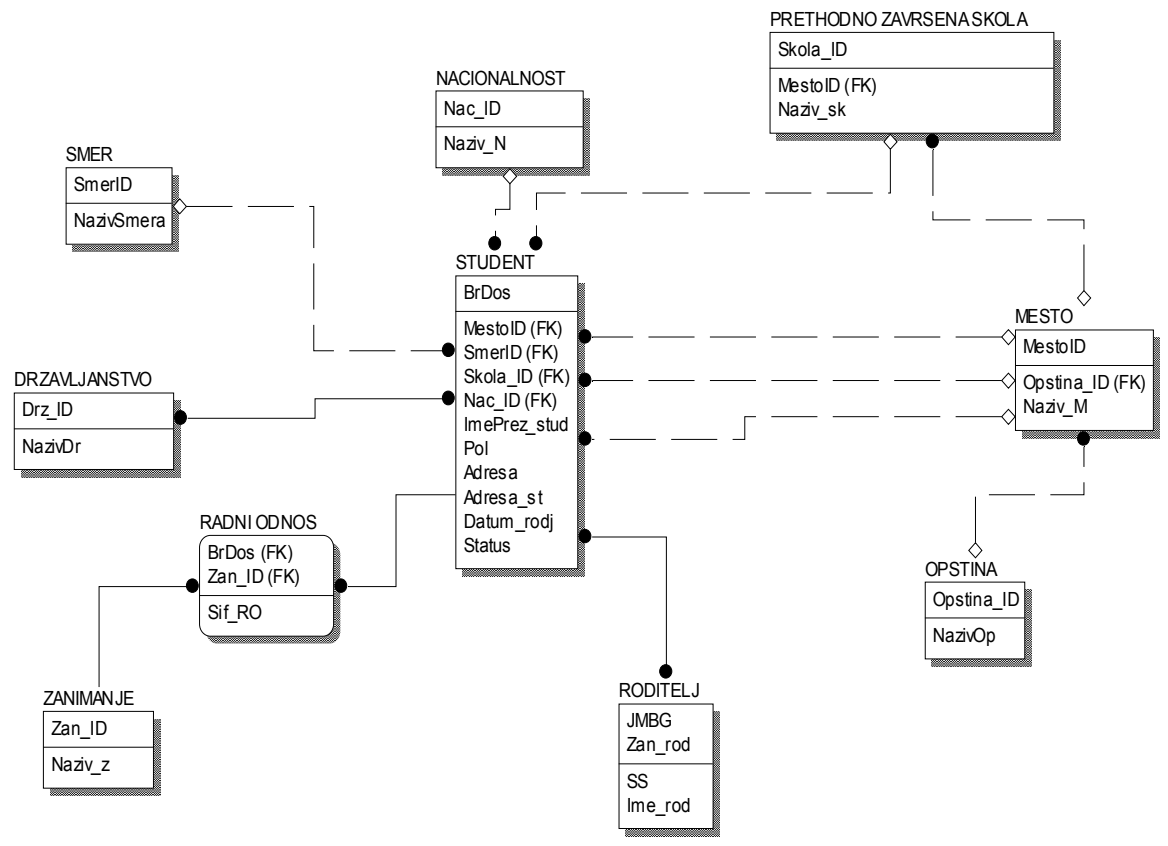


Atributi po IDEF1X sintaksi nisu obavezni Atributi se navode unutar simbola za objekat i to ključni atributi u gornjem delu, a opisni u donjem delu simbola.

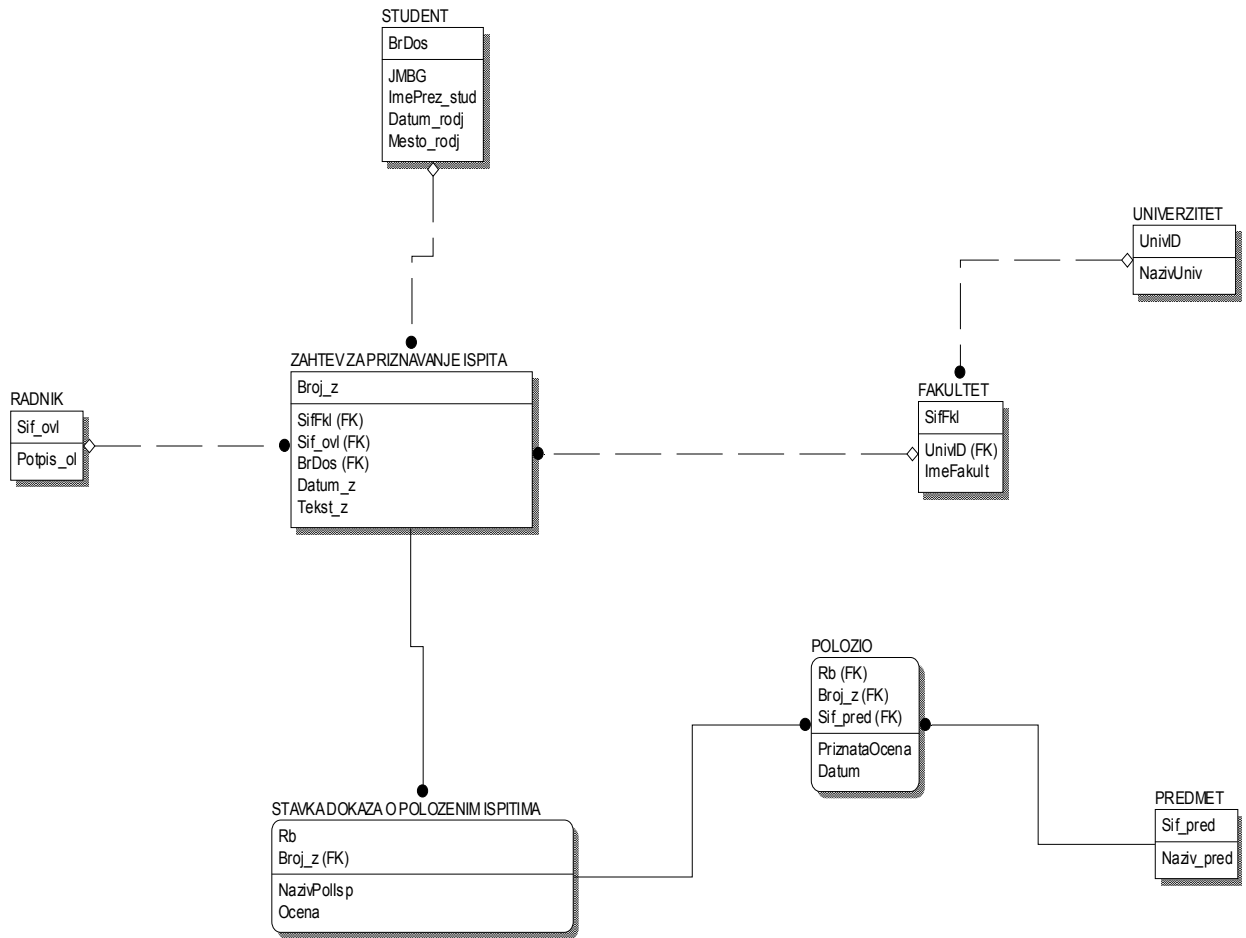
1. Dosije studenta



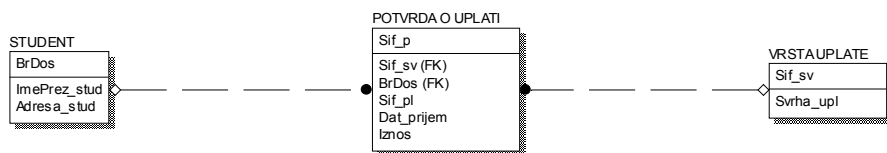
2. SV obrazac



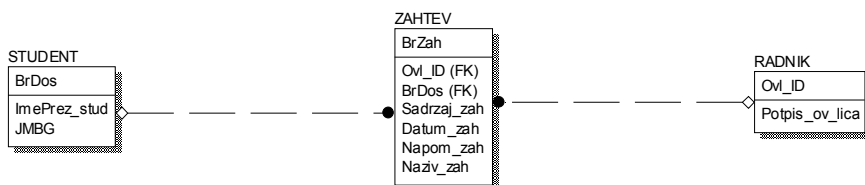
3. Zahtev za priznavanje ispita



4. Potvrda o uplati



5. Zahtev



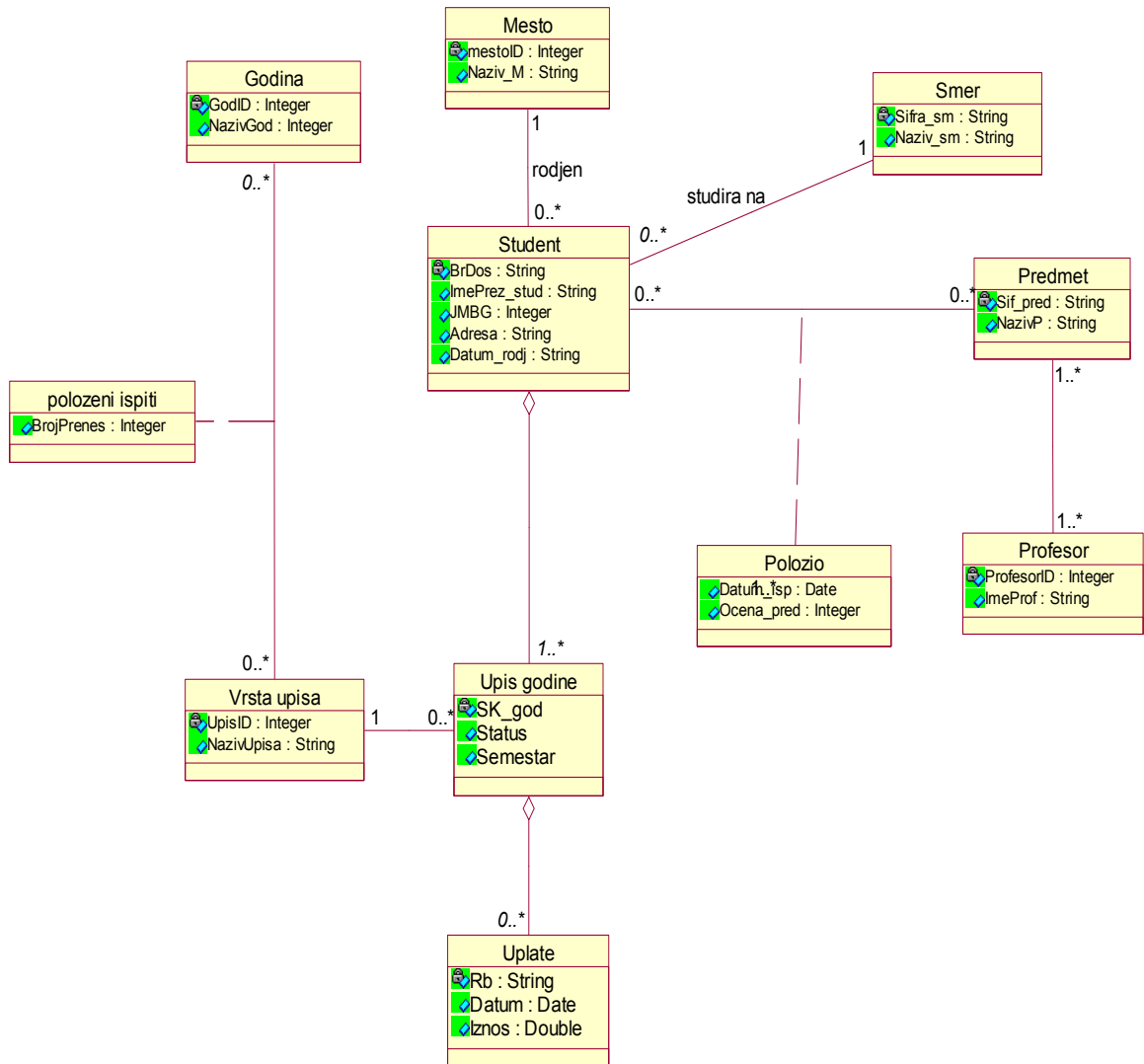
7. KONCEPTUALNI MODEL

Alatka Rational Rose

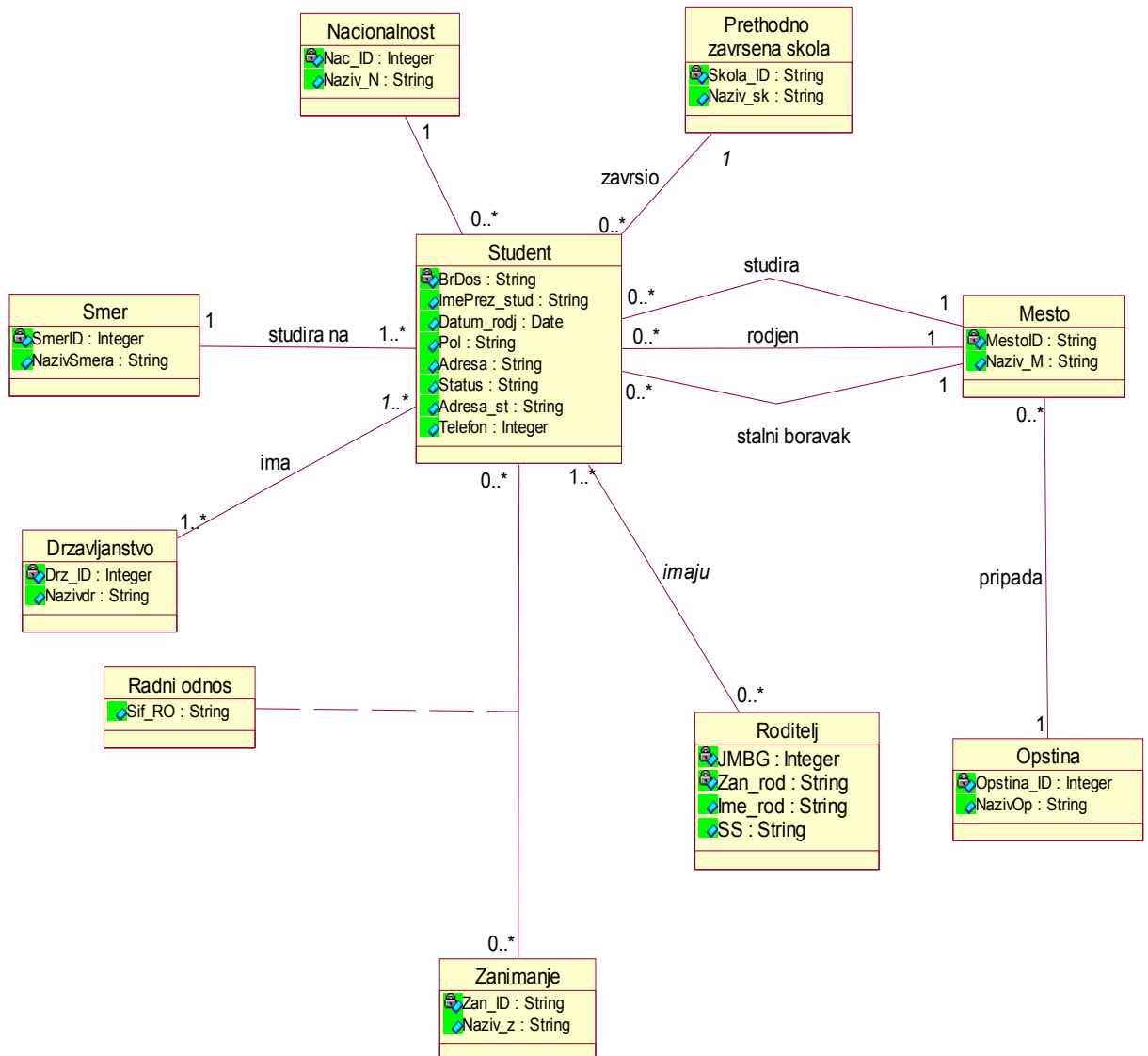
Familija programa *Rational Rose* projektovana je tako da programerima obezbedi komplet alata za vizuelno modelovanje koji omogućuju razvoj robusnih, efikasnih rešenja za stvarne poslovne potrebe. Proizvodi *Rational Rosea* koriste opšti univerzalni standard i time približavaju modelovanje kako programerima koji modeluju logiku aplikacije, tako i neprogramerima koji bi hteli da modeluju poslovne procese.

Modelovanje omogućava bolje razumevanje zahteva, čisti dizajn i sisteme pogodnije za održavanje. Notacija ima bitnu ulogu u svakom modelu. *Unified Modeling Language (UML)* obezbeđuje veoma robusnu notaciju koja iz analize izrasta u proces projektovanja. *UML* je jezik kojim se definišu, vizualizuju i dokumentuju tvorevine objektno orjentisanog sistema koji se razvija.

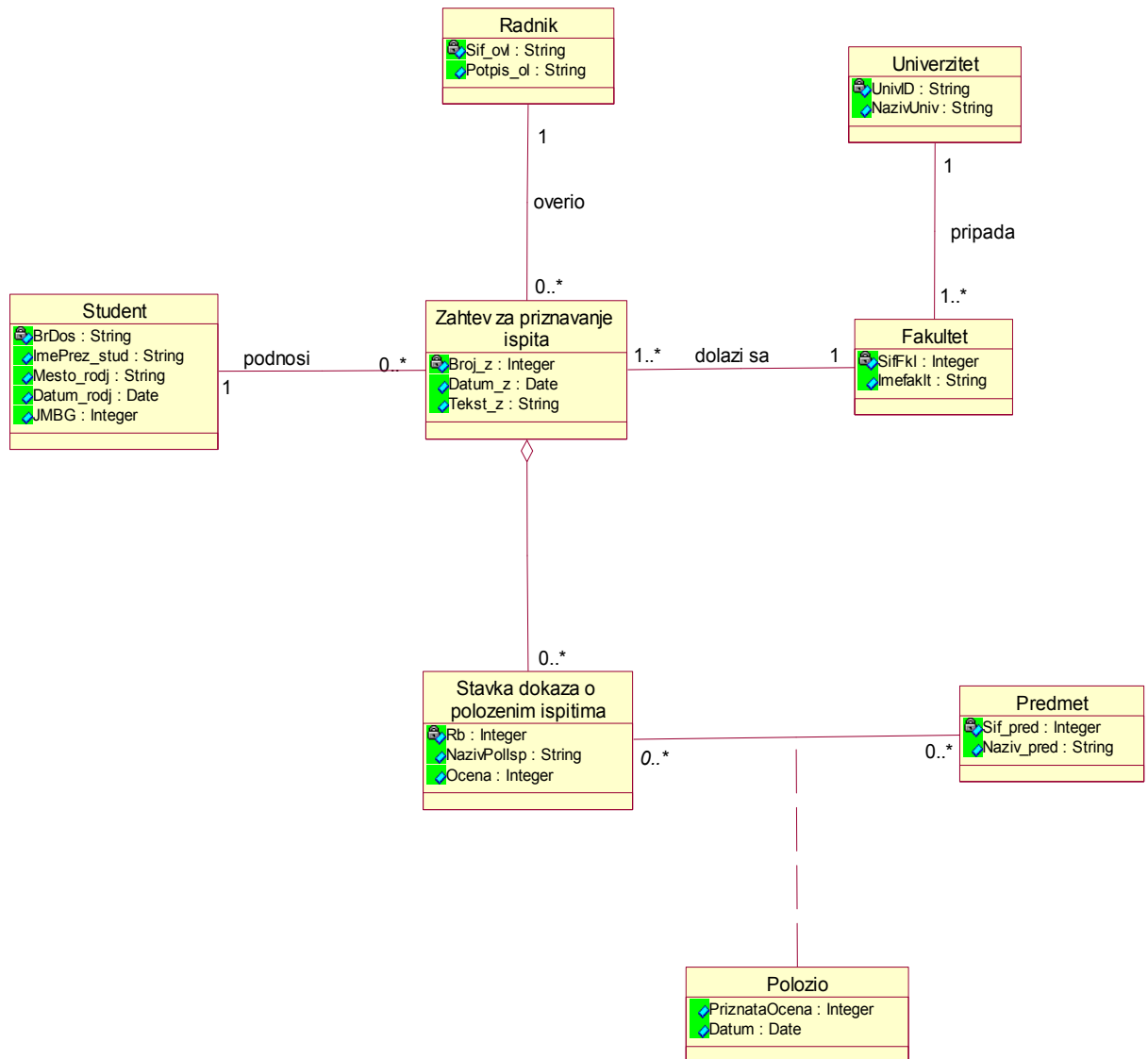
1. Dosije studenta



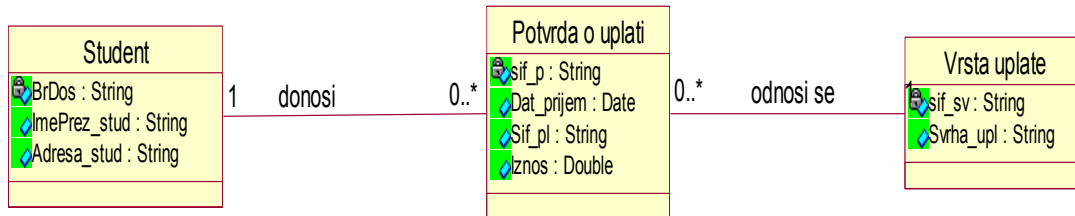
2. SV obrazac



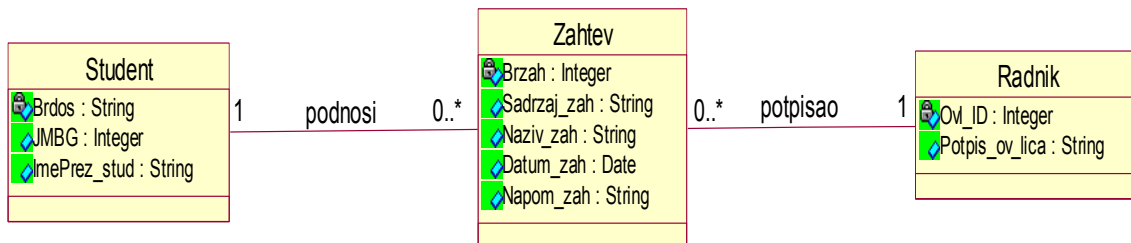
3. Zahtev za priznavanje ispita



4. Potvrda o uplati



5. Zahtev



8. SLUČAJEVI KORIŠĆENJA

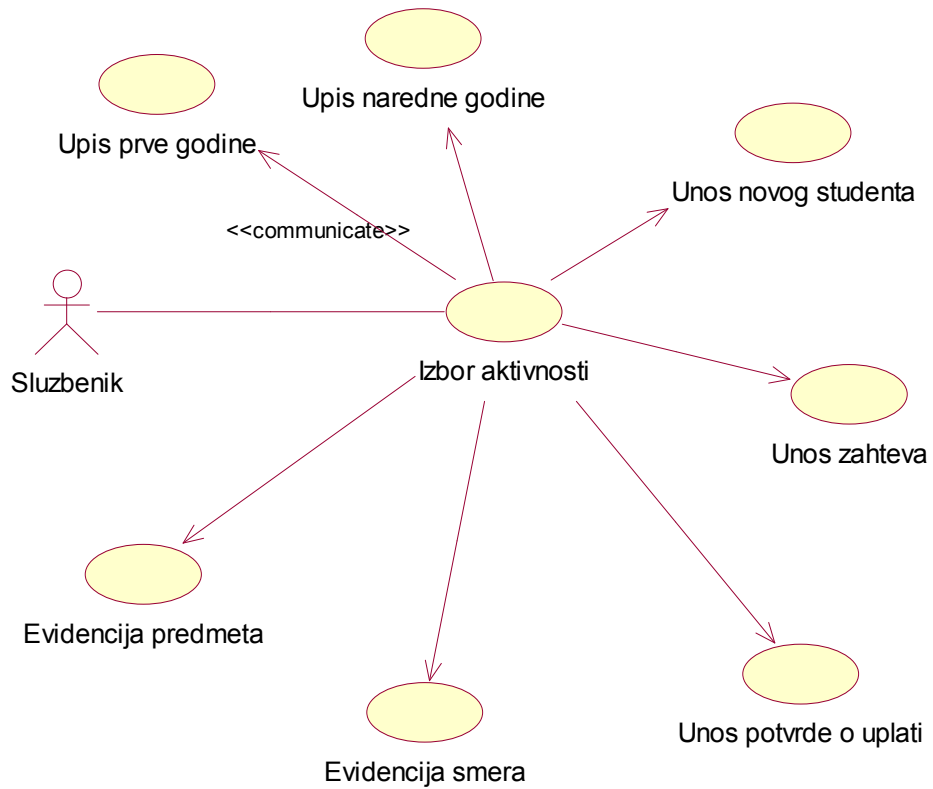
Slučajevi korišćenja modeluju dijalog između izvođača i sistema. One predstavljaju funkcionalnost koju obezbeđuje sistem. Skupina slučajeva korišćenja za neki sistem ustanovljava sve definisane načine na koje taj sistem može biti korišćen. Formalna definicija slučajeva korišćenja glasi: slučaj korišćenja je niz transakcija koje izvodi sistem, koji daje merljive rezultate od vrednosti za pojedinačnog izvođača.

Između izvođača i slučaja korišćenja može postojati relacija asocijacije, koja se još naziva i komunikacionom asocijacijom. Mogu postojati dva tipa relacija između slučajeva korišćenja: *include* (uključiti) i *extend* (proširi). Relacije *include* se formiraju između novog slučajeva korišćenja i svakog drugog slučaja korišćenja koji koristi njegovu funkcionalnost. Relacija *extend* koristi se da prikaže :

- Opciono ponašanje
- Ponašanje koje se pokreće samo pod određenim uslovima
- Nekoliko različitih tokova koji mogu biti pokrenuti na osnovu izbora izvođača

Dijagram slučajeva korišćenja je grafički prikaz pojedinih ili svih izvođača, slučajeva korišćenja i njihovih interakcija.

1. IZBOR AKTIVNOSTI



Naziv slučaja korišćenja : Izbor aktivnosti

Opis slučaja korišćenja : Odabir aktivnosti koje službenik želi da obavlja

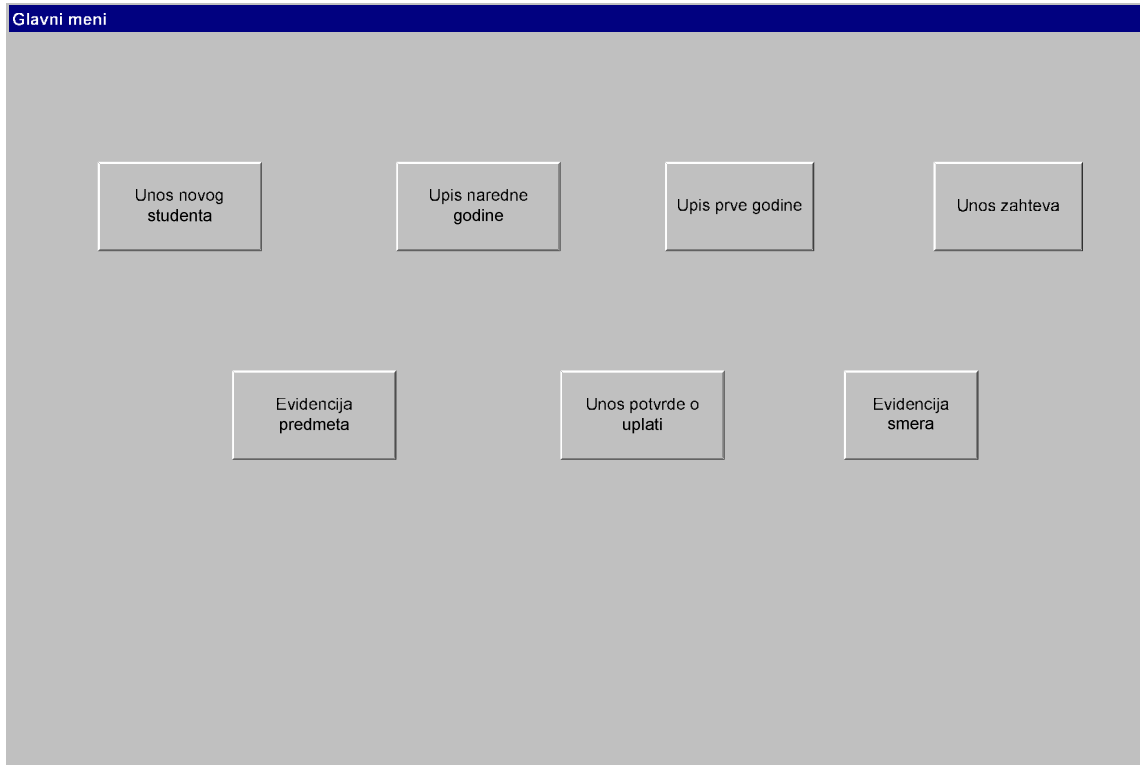
Akteri : Prioritet:1

Autor : Dalibor Vidović

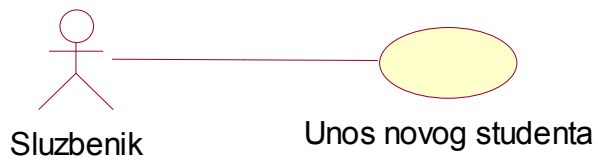
Status : U izradi

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
Bira aplikaciju	Pokrece aplikaciju



2. UNOS NOVOG STUDENTA



Naziv slučaja korišćenja : Unos novog studenta

Opis slučaja korišćenja : Službenik unosi podatke o studentu

Akteri : Prioritet:1

Autor : Dalibor Vidović

Status : U izradi

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
-Pokrece aplikaciju -Unosi podatke o studentu na osnovu SV obrazca -Bira opciju potvrdi/otkazi	-Otvara odgovarajucu formu

Postoji alternativno funkcionisanje : U slučaju da se ne unesu odgovarajući podaci u određena polja sistem javlja poruku u grešci.

Unos novog studenta

Unesite osnovne podatke o studentu sa SV obrasca

Ime I prezime studenta Pol

Broj indeksa /

JMBG

Datum rođenja

Mesto rođenja studenta Opština(ili strana država)

Ime jednog od roditelja Zanimanje roditelja

Mesto stalnog boravka studenta Opština

Ulica I kućni broj Telefon

Mesto stanovanja studenta za vreme studiranja Opština

Ulica I kućni broj Telefon

Državljanstvo Nacionalnost

Zanimanje studenta (ako postoji)

Nacin finansiranja Smer

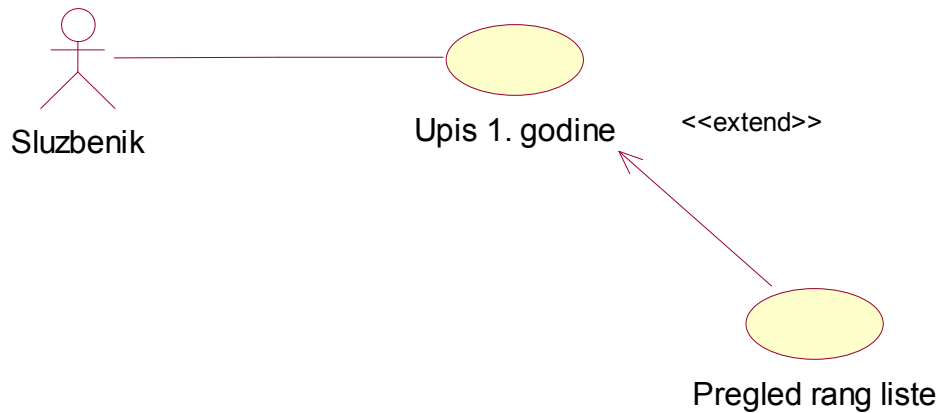
Naredna polja se popunjavaju samo za studente koji prvi put upisuju prvu godinu

Prethodno završena skola Godina završetka

Mesto škole Opština

POTVRDI ODUSTANI

3. UPIS PRVE GODINE (izbor studenta sa rang liste)



Naziv slučaja korišćenja : Upis prve godine

Opis slučaja korišćenja : Na osnovu položenog prijemnog ispita studentu se omogućava upis u prvu godinu studija

Akteri : Prioritet:1

Autor : Mileša Gordić

Status : U izradi

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
Korisnik bira aplikaciju upis godine	
Unosi ime i prezime, JMBG i redni broj prijavnog lista	Otvora formu za upis
Upisuje dodatne podatke i bira opciju potvrdi	Pronalazi studenta u bazi i vraća podatke

Postoji i alternativno funkcionisanje: U slučaju da se student ne nalazi na listi primljenih sistem javlja odgovarajuću poruku.

Unesite podatke kandidata

Ime i prezime JMBG Broj prijavnog lista

PRONADJI

ODUSTANI

Kandidat Pozicija na rang listi Ukupan broj bodova Broj indeksa / Nacin finansiranja UPISI
STUDENTAOTKAZI
UPIS

4. EVIDENCIJA SMERA



Naziv slučaja korišćenja : Evidencija smer

Opis slučaja korišćenja : Službenik unosi naziv i šifru smer

Akteri : Prioritet:1

Autor : Milesa Gordić

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
Korisnik bira aplikaciju unos smer	
Unosi smer i sifru smer i bira opciju potvrdi	Pokrece odgovarajucu aplikaciju

Evidencija smer

Unesite podatke o smeru

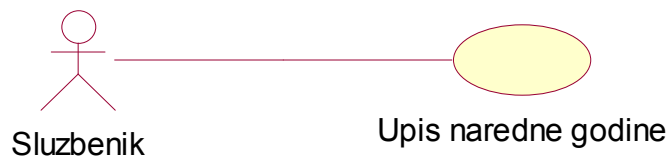
Sifra smer

Naziv smer

Datum uvodjenja

POTVRDI OTKAZI

5. UPIS NAREDNE GODINE



Naziv slučaja korišćenja : Upis naredne godine

Opis slučaja korišćenja : Studentu se omogućava da na osnovu zadovoljenog uslova upise narednu godinu studija

Akteri : Prioritet:1

Autor : Aleksandar Dabić

Status : U izradi

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
Pokrece aplikaciju upis naredne godine	
Unosi broj indeksa i ime i prezime studenta	Otvora odgovarajucu formu
Bira godinu upisa	Otvora formu sa osnovnim podacima o studentu
	Proverava zadovoljenje uslova i izvrsava upis

Postoji alternativno funkcionisanje : U slučaju da nije zadovoljen uslov sistem štampa odgovarajuću poruku.

Upis naredne godine

Broj indeksa /

Ime i prezime studenta

PRIHVATI ODUSTANI

Upis naredne godine

Student Broj indeksa /

Godina

Polozeni ispiti

Ukupan broj polozenih ispita

Odaberi godinu koju student upisuje

IZVRSI UPIS

OTKAZI

6. UNOS ZAHTEVA



Naziv slučaja korišćenja : Unos zahteva

Opis slučaja korišćenja : Službenik unosi podatke vezane za zahtev

Akteri : Prioritet:1

Autor : Aleksandar Dabić

Status : U izradi

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
Korisnik bira aplikaciju unos zahteva Unosi naziv zahteva, datum i sadrzaj zahteva, ime i prezime studenta i broj dosijea Bira opciju potvrdi	Pokrece odgovarajucu aplikaciju i generise broj zahteva

Zahtev

Naziv zahteva Broj zahteva

Ime i prezime podnosioca

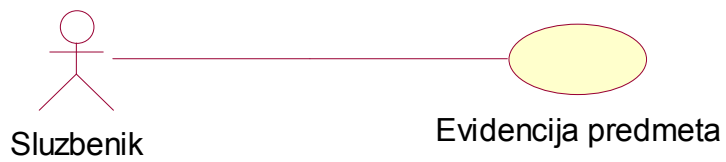
Broj indeksa / Datum

Sadrzaj zahteva

Napomena

POTVRDI ODUSTANI

7. EVIDENCIJA PREDMETA



Naziv slučaja korišćenja : Evidencija predmeta

Opis slučaja korišćenja : Službenik unosi naziv i šifru predmeta

Akteri : Prioritet:1

Autor : Jelena Đuknić

Status : U izradi

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
Korisnik bira aplikaciju unos predmeta	
Unosi naziv i sifru predmeta i bira opciju potvrdi	Pokrece odgovarajucu aplikaciju

Evidencija predmeta

Unesite podatke o predmetu

Sifra predmeta

Naziv predmeta

Datum uvodjenja

8. UNOS POTVRDE O UPLATI



Naziv slučaja korišćenja : Unos potvrde o uplati

Opis slučaja korišćenja : Evidentira se dokaz da je student izmirio određene finansijske obaveze prema fakultetu

Akteri : Prioritet:1

Autor : Jelena Đuknić

Status : U izradi

Opis normalnog funkcionisanja :

KORISNIK	SISTEM
Korisnik bira aplikaciju za unos	
Unosi broj dosijea, datum,svrhu i iznos uplate bira opciju potvrdi	Otvora formu za unos podataka o studentu

Unos potvrde o uplati

Sifra uplate

Broj indeksa /

Svrha uplate

Iznos

Datum

POTVRDI OTKAZI

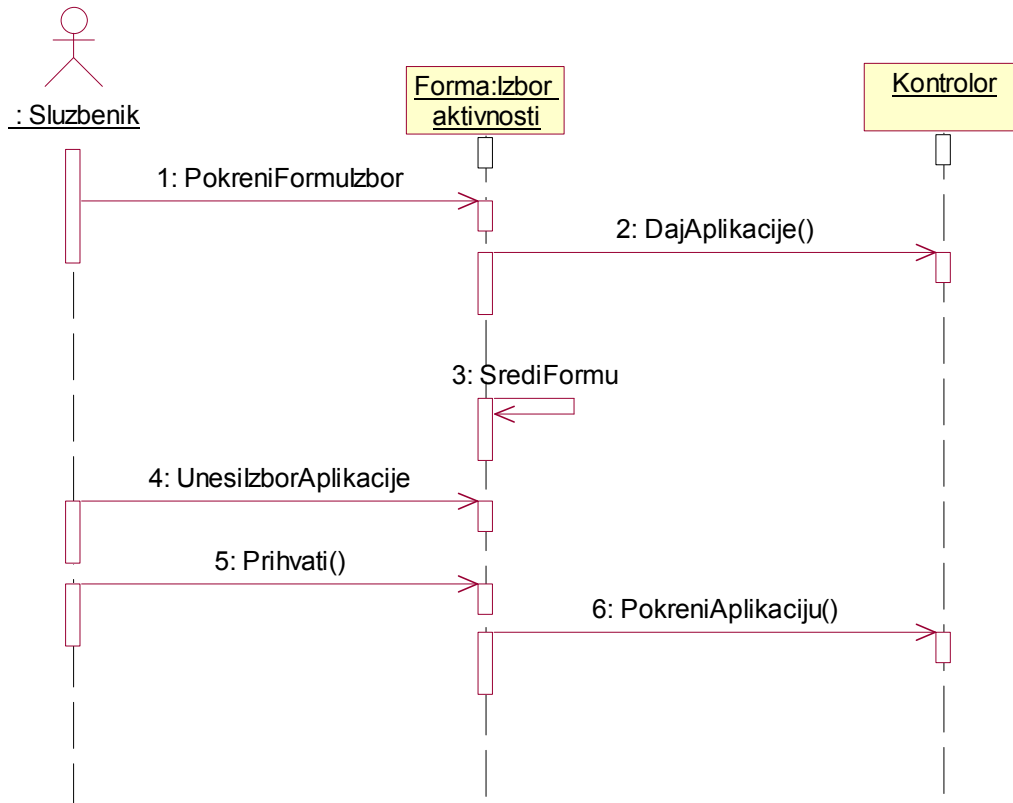
9. SEKVENCIJALNI DIJAGRAMI

Postoje dva tipa dijagrama interakcije - *dijagrame sekvenci i dijagrami kolaboracije*. Svaki taj dijagram je grafički prikaz scenarija, konkretnog primerka korisničke funkcije.

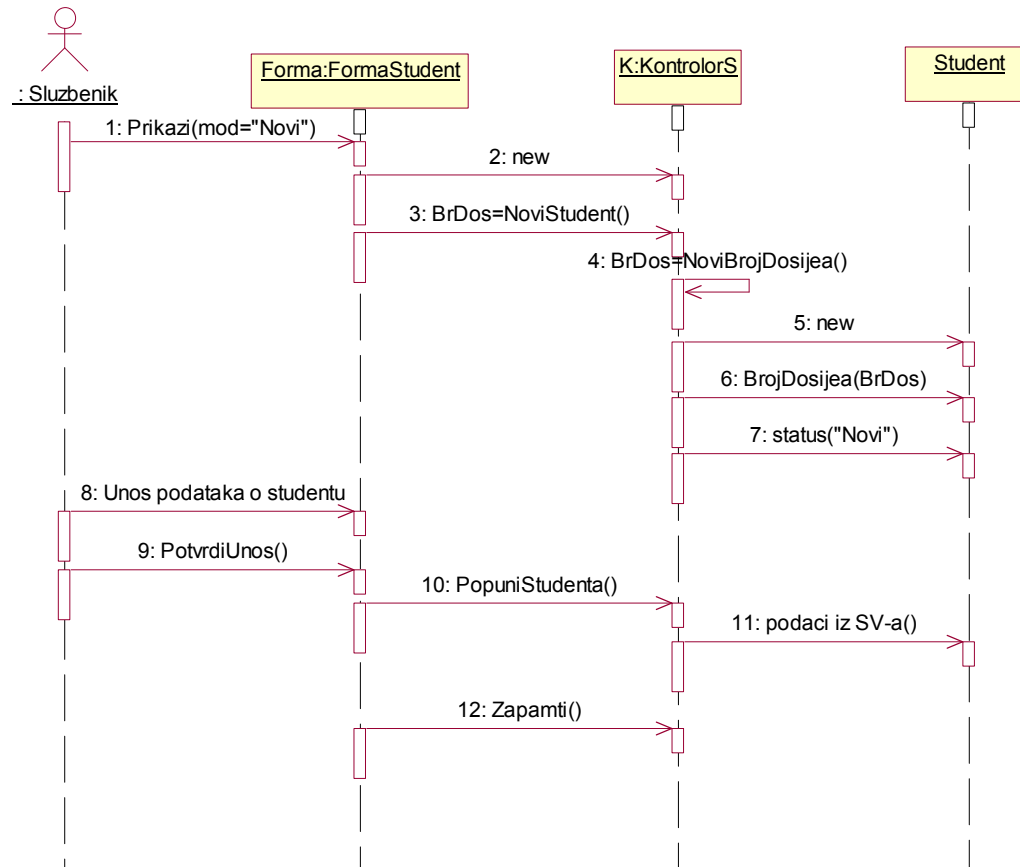
Sekvencijalni dijagram prikazuje interakcije između objekata poredane po vremenskom redosledu. On prikazuje objekte i klase u vezi sa scenarijom, kao i redosled poruka koje objekti razmenjuju. Dijagram sekvenci se obično vezuju za realizaciju korisničkih funkcija u *Logical View* sistema u razvoju.

U UML-u se objekat u dijagramu sekvenci predstavlja pravougaonikom koji sadrži podvučeno ime objekta. Svaki objekat ima svoju vremensku liniju predstavljenu isprekidanom linijom ispod objekta. Poruke koje objekti razmenjuju predstavljene su u vidu strelica usmerenih od pošiljaoca ka primaocu poruke.

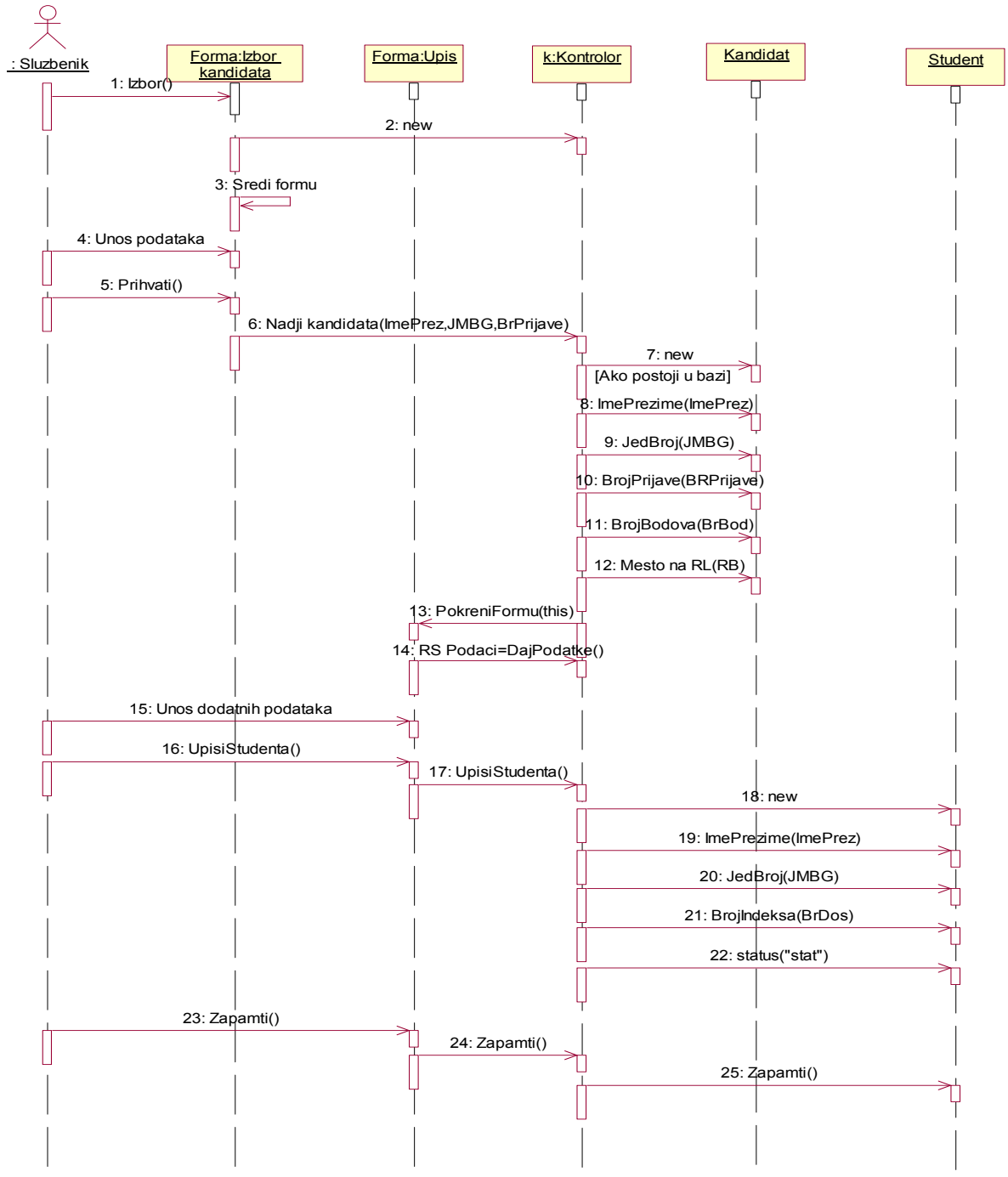
1. IZBOR AKTIVNOSTI



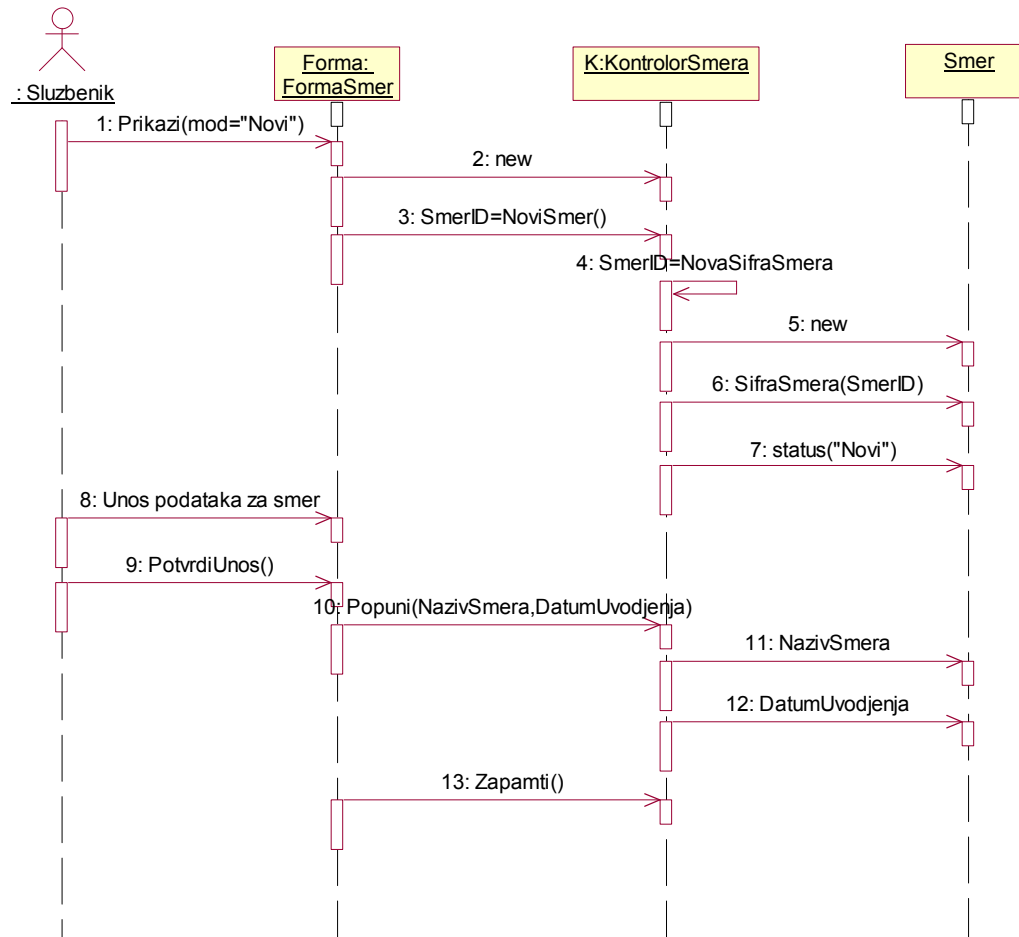
2. UNOS NOVOG STUDENTA



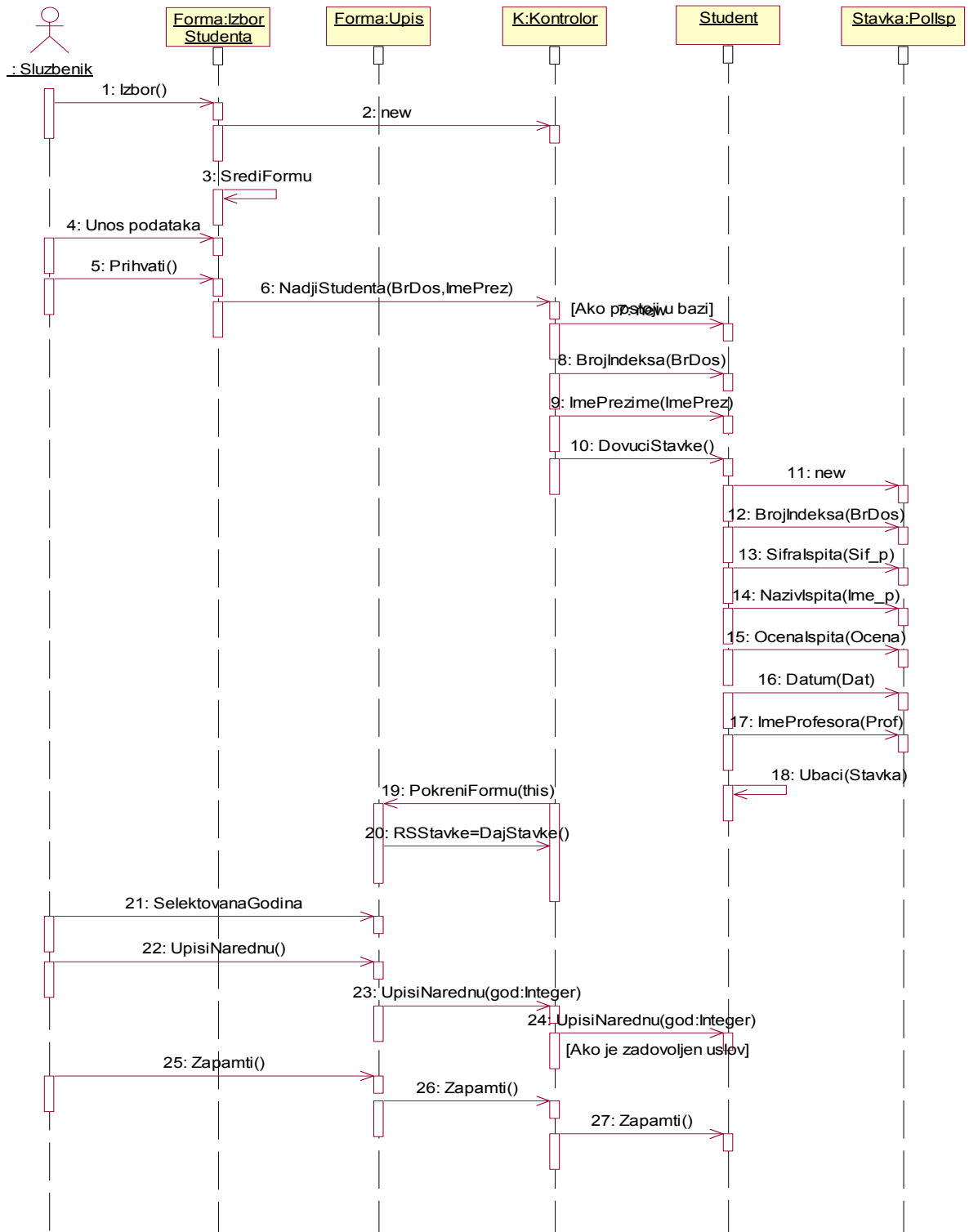
3. UPIS PRVE GODINE



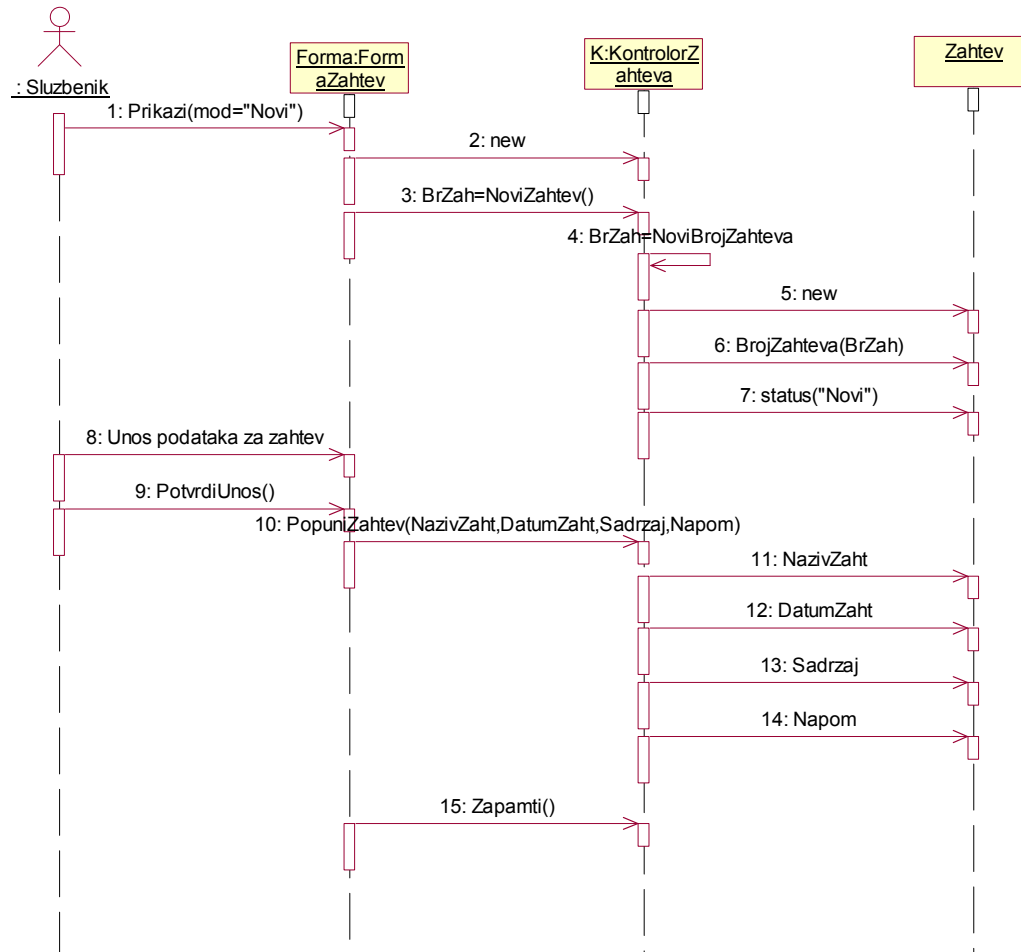
4. EVIDENCIJA SMERA



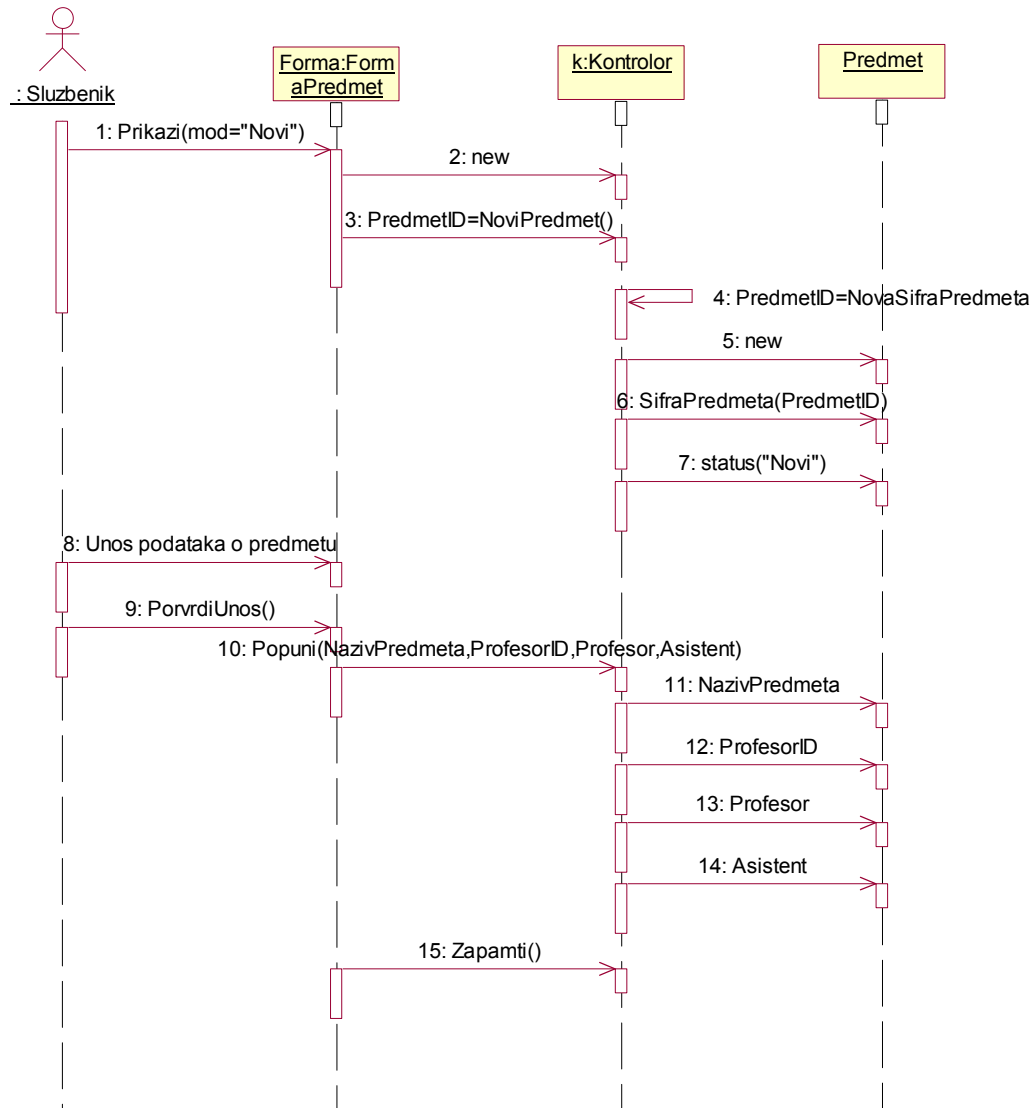
5. UPIS NAREDNE GODINE



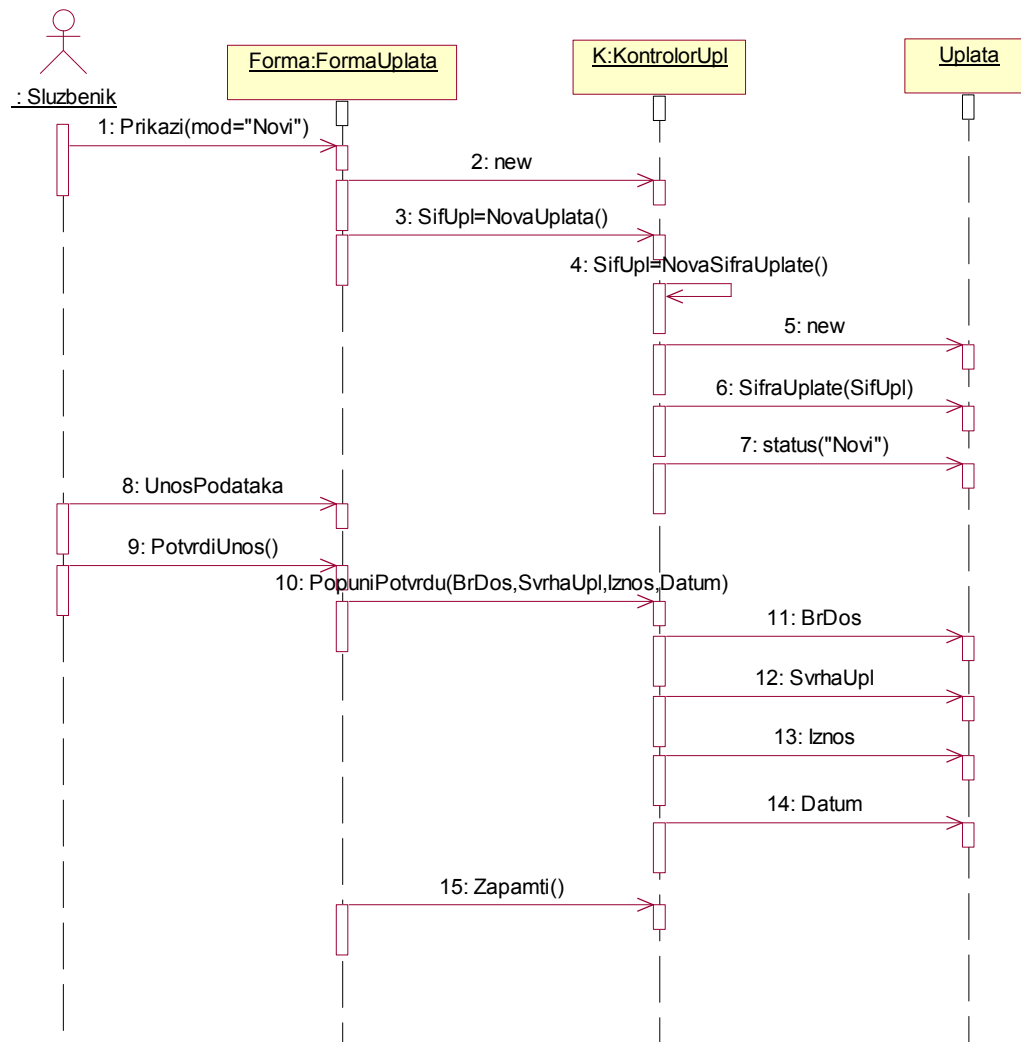
6. UNOS ZAHEVA



7. EVIDENCIJA PREDMETA



8. UNOS POTVRDE O UPLATI

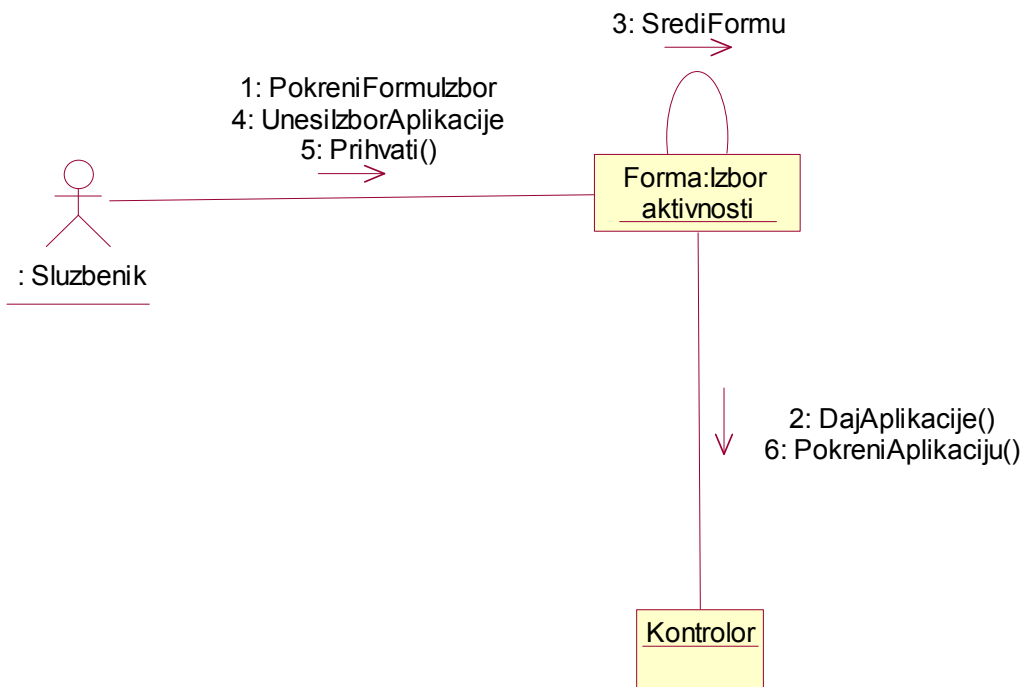


10. DIJAGRAMI KOLABORACIJE

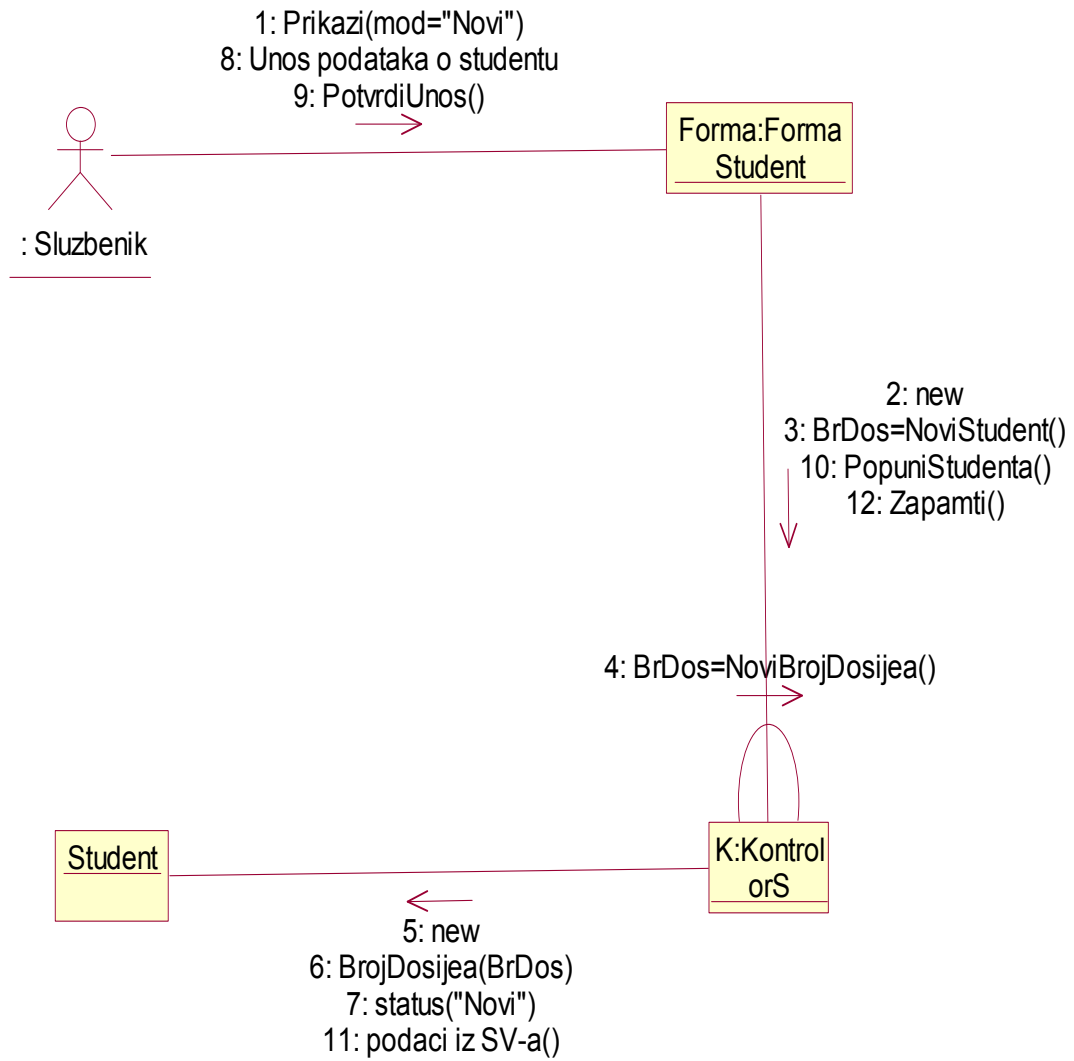
Dijagram kolaboracije predstavlja alternativni način da se prikaže scenario. Taj tip dijagrama prikazuje interakcije objekata pomoću tih objekata njihovih veza međusobnih veza. Dijagram kolaboracije sadrži:

- Objekte prikazane pravougaonicima
- Veza među objektima prikazane linijama koje spajaju objekte
- Poruke prikazane tekstem i strelicom usmerenom od pošiljaoca ka primaocu poruke

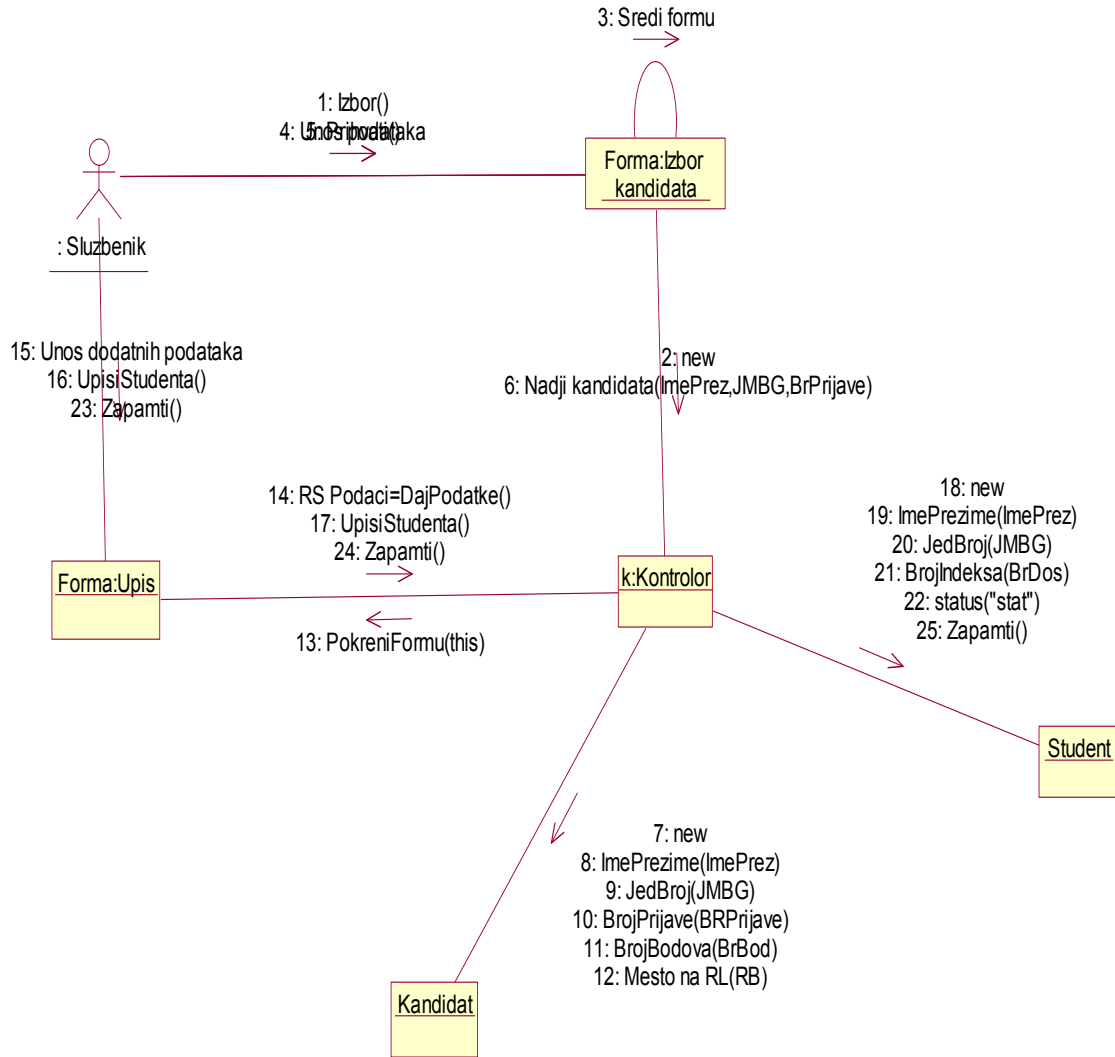
1. IZBOR AKTIVNOSTI



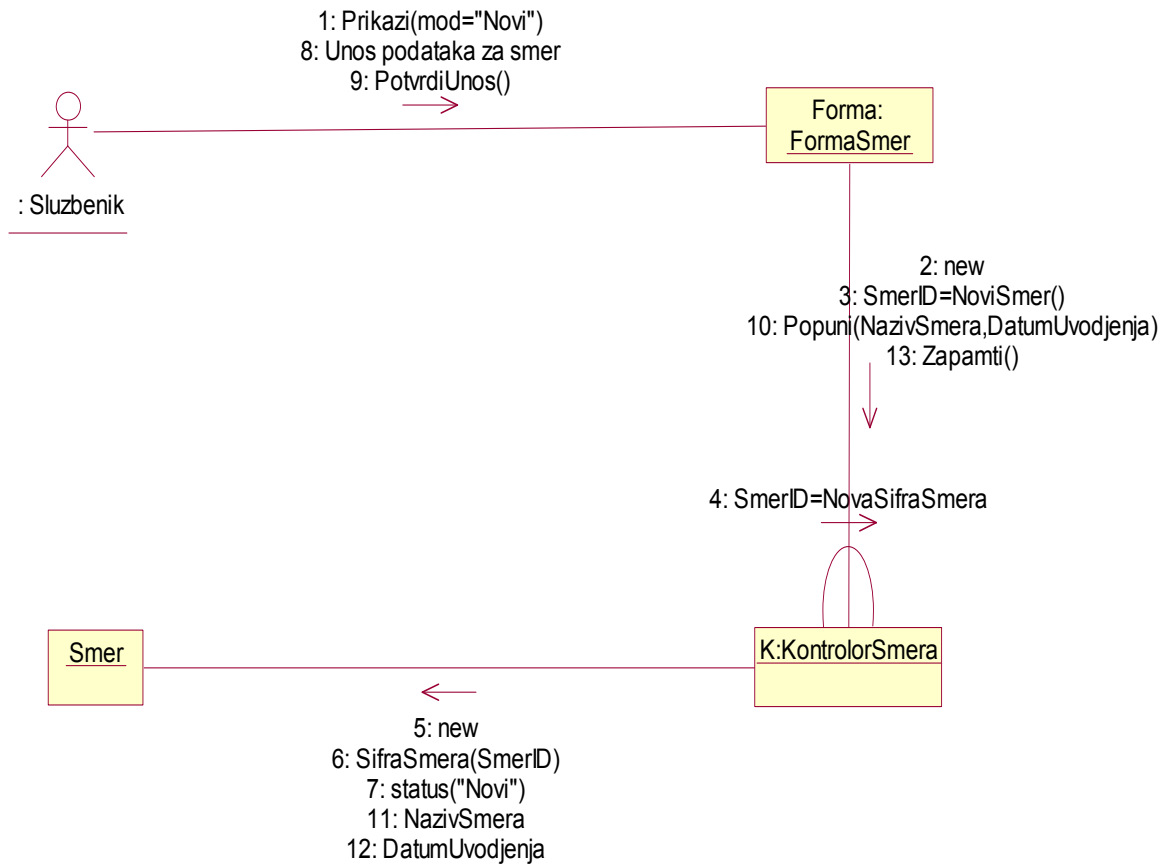
2. UNOS NOVOG STUDENTA



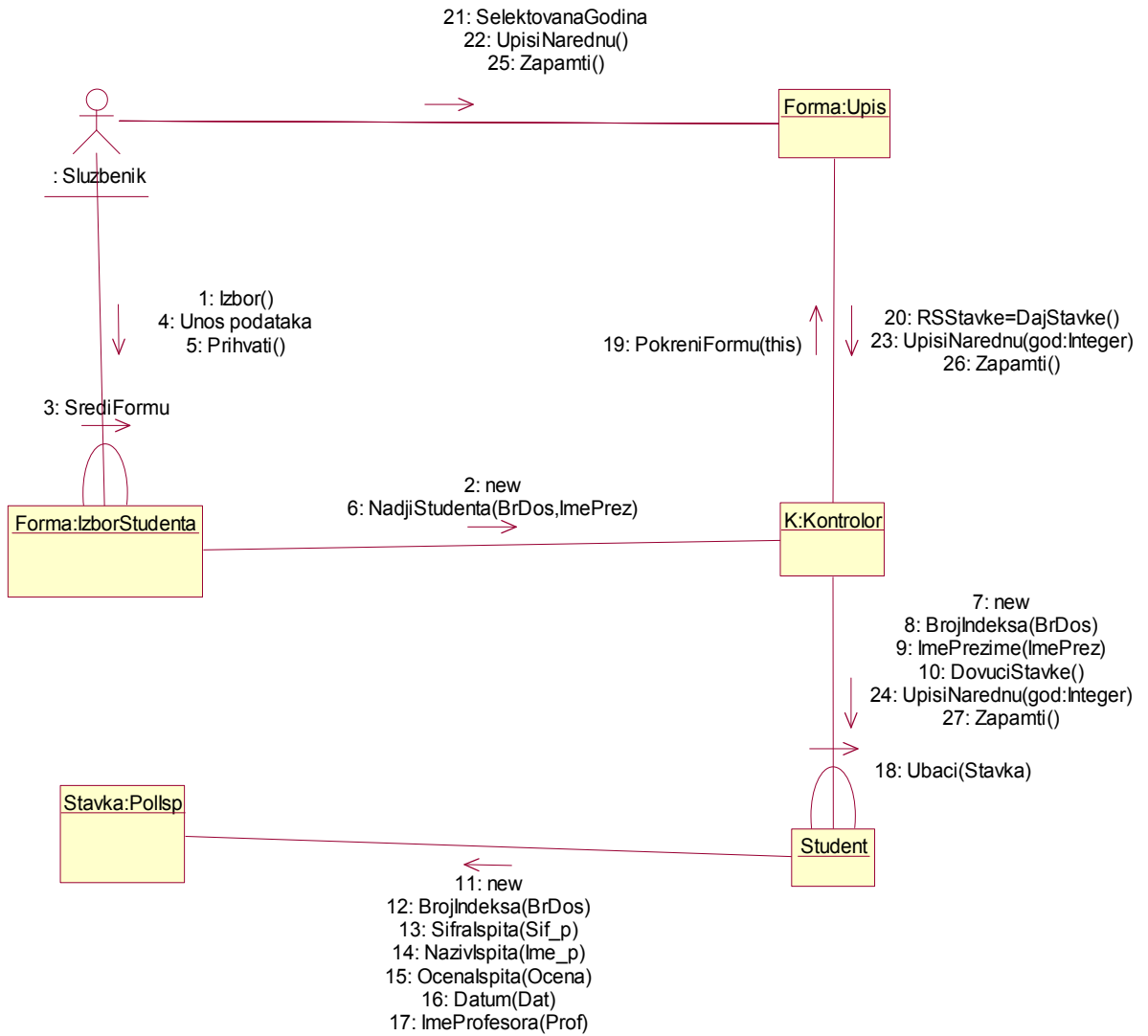
3. UPIS PRVE GODINE



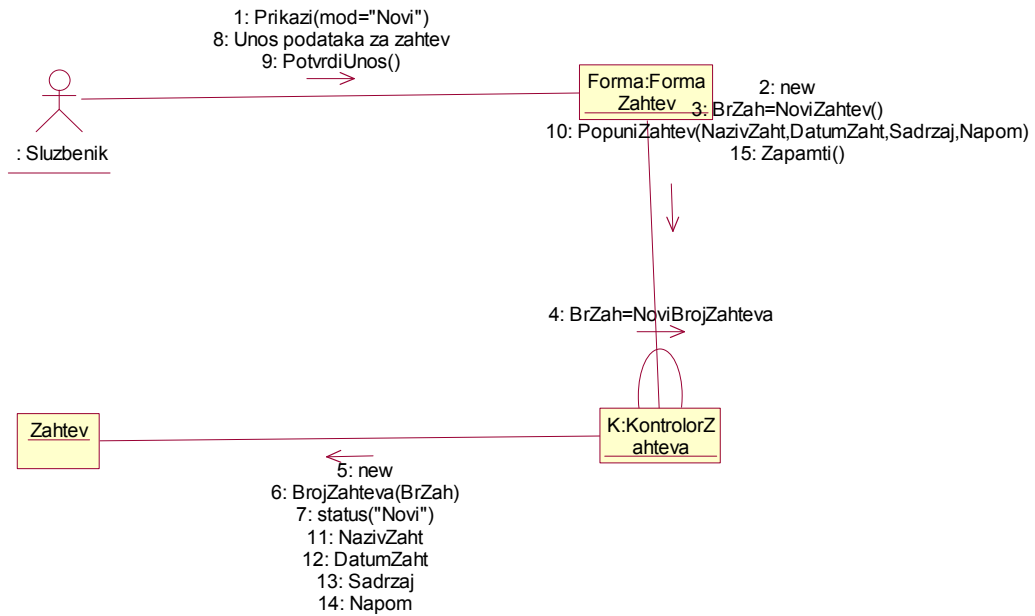
4. EVIDENCIJA SMERA



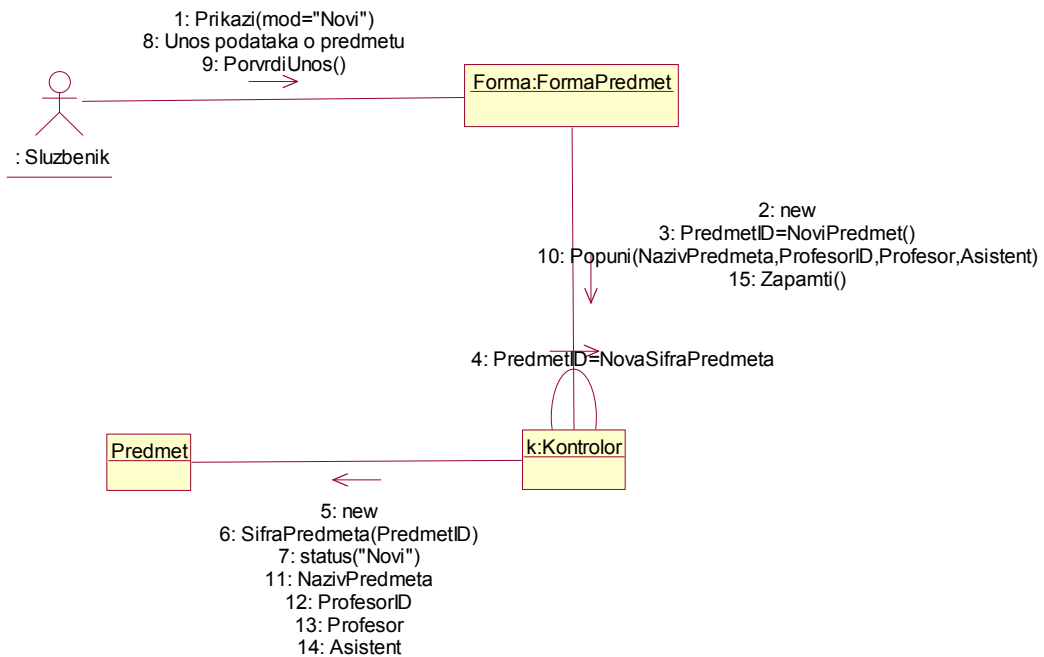
5. UPIS NAREDNE GODINE



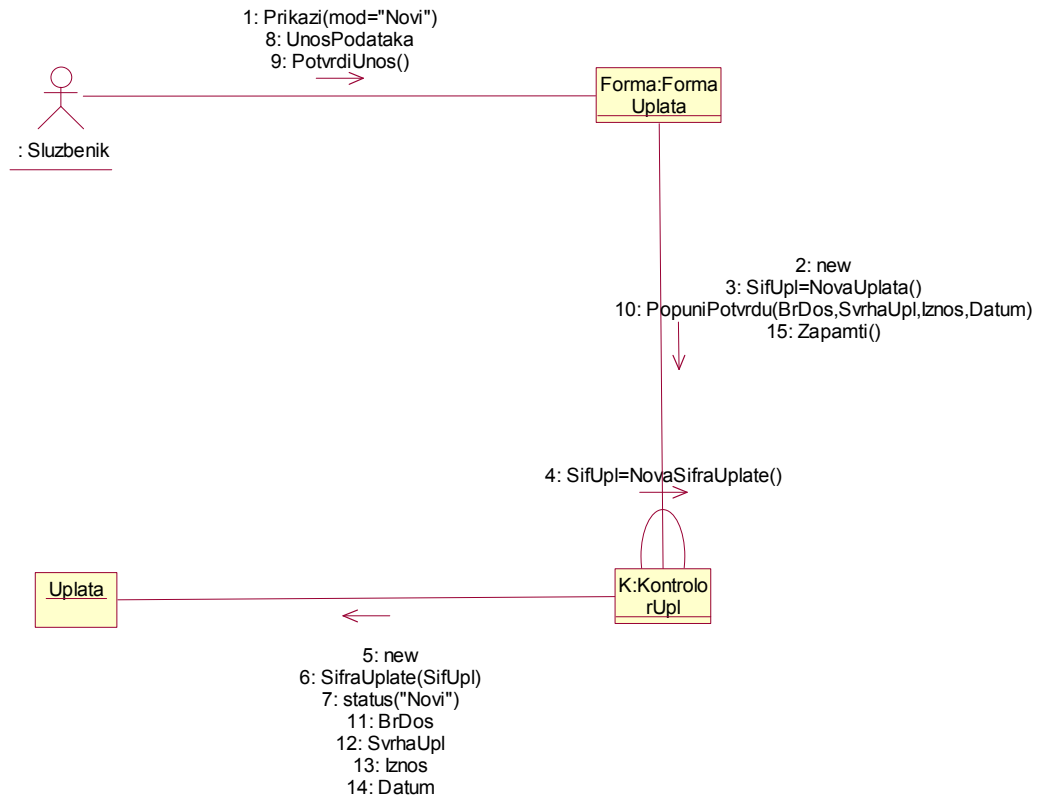
6. UNOS ZAHTEVA



7. EVIDENCIJA PREDMETA



8. UNOS POTVRDE O UPLATI

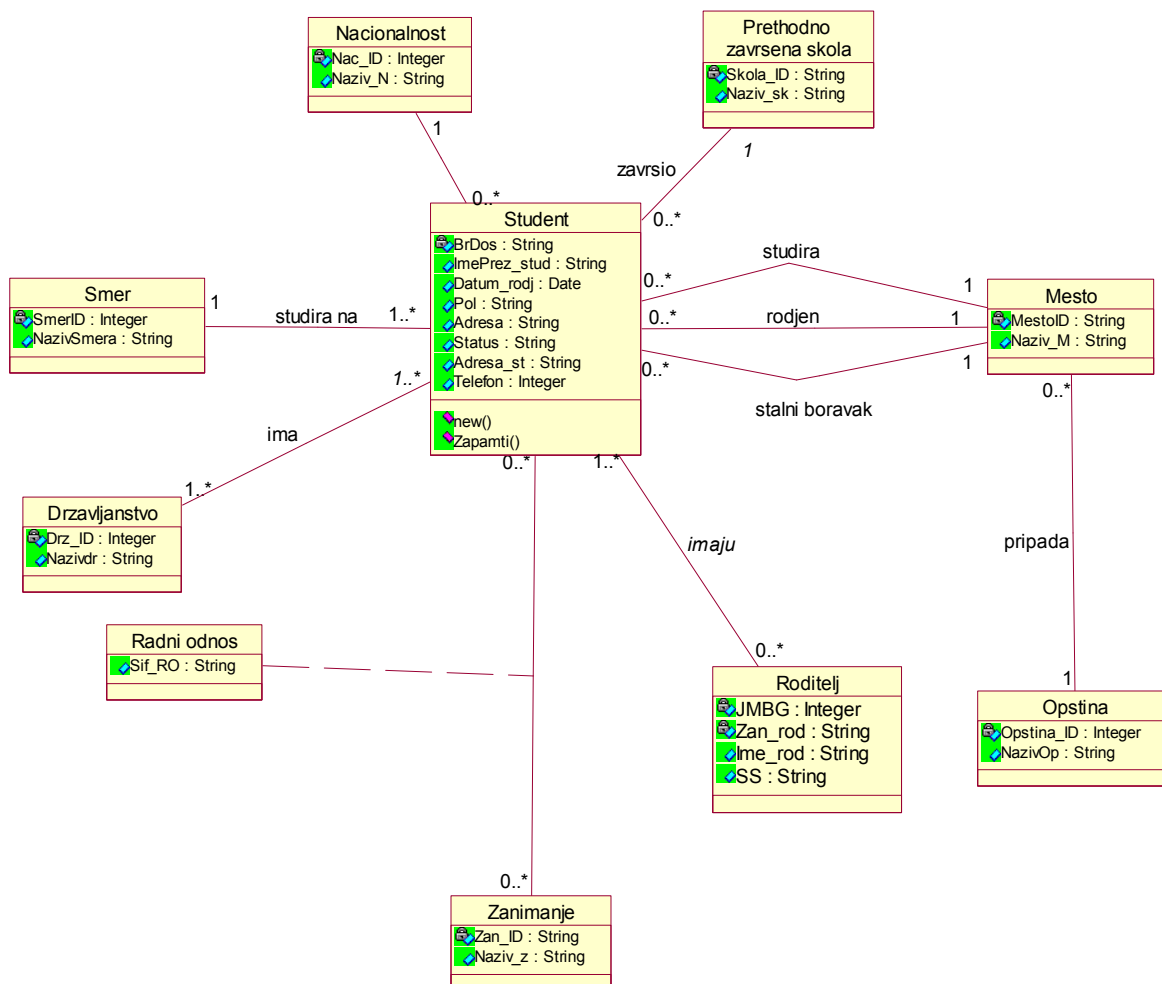


11. DIJAGRAMI KLASA

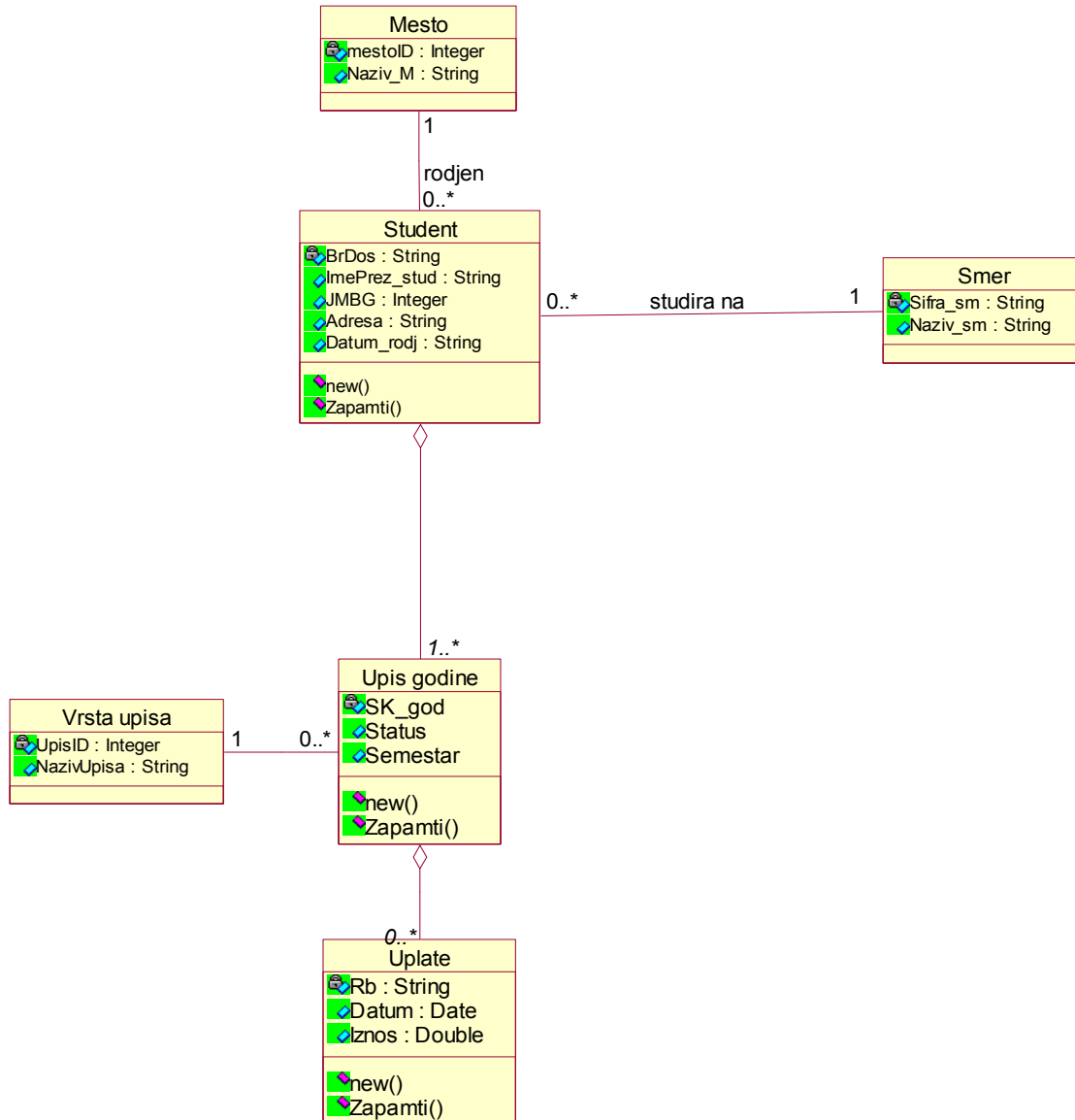
Kako se modelu dodaje sve više klasa, javlja se sve veća i veća potreba da se one prikažu na još neki način, a ne samo tekstualno. Prave se dijagrami klasa kako bi obezbedili sliku ili pregled nekih ili svih klasa u modelu.

Glavni dijagram klasa u logičkom prikazu modela obično je slika paketa u sistemu. Svaki paket ima takođe svoj glavni dijagram klasa, koji obično prikazuje javne klase paketa. Ostali dijagrami se prave ako je to potrebno.

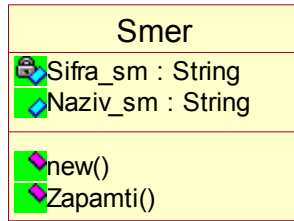
1. UNOS NOVOG STUDENTA



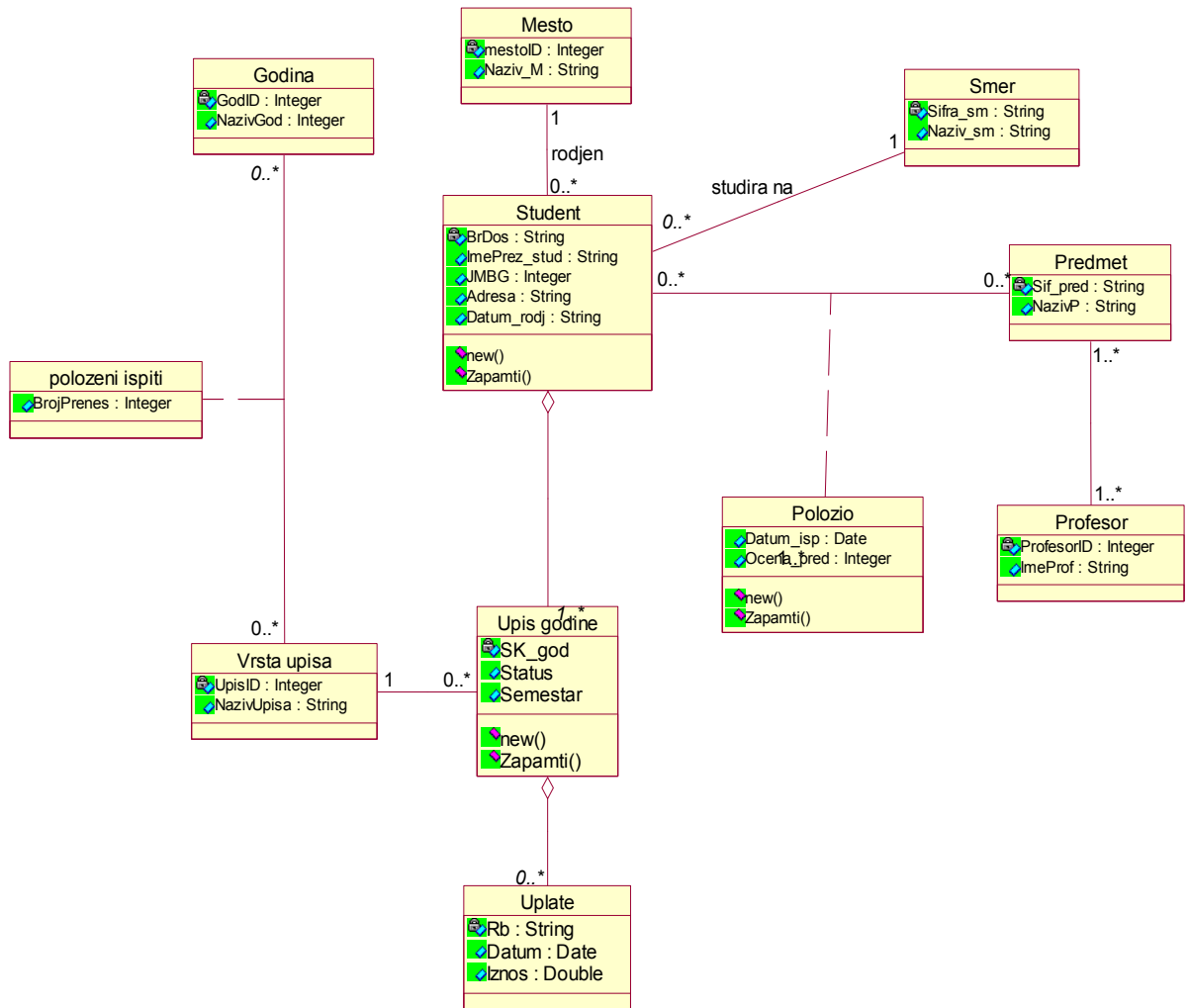
2. UPIS PRVE GODINE



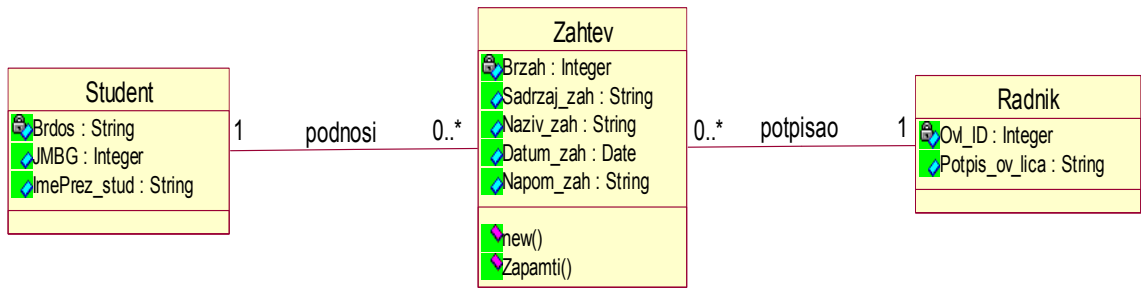
3. EVIDENCIJA SMERA



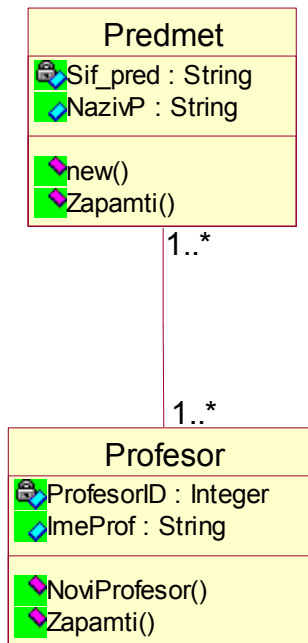
4. UPIS NAREDNE GODINE



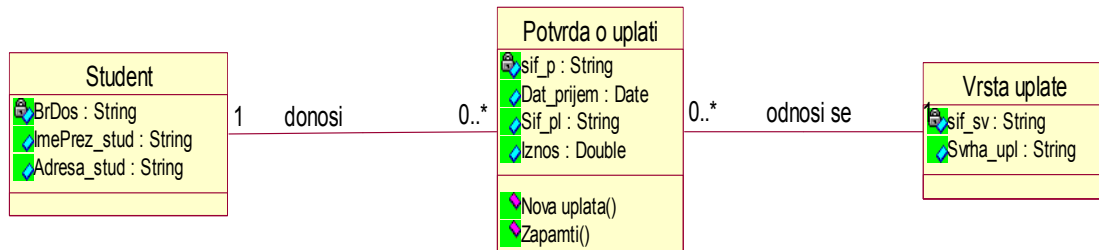
5. UNOS ZAHEVA



6. EVIDENCIJA PREDMETA



7. UNOS POTVRDE O UPLATI



Literatura

1. Zoran Marjanović - "Baze podataka", Beograd, 2003.
2. Vidojko Ćirić , Siniša Vlajić -"Razvoj softvera, programski jezik JAVA", Beograd 2002.
3. Craig Larman : "Applying UML and Patterns", Prentice Hall, 1997.
4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides : "Design patterns", Addison - Wesley, September 1999
5. G. Booch, J. Rumbaugh, I. Jacobson "UML Vodič za korisnike" CET . Beograd 2002
6. www.rational.com