Poslovna aplikacija za upravljanje sastancima u poduzeću nad relacijskom bazom podataka (Clarion)

Kaluža, Marina

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:195:608925

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Sveučilište u Rijeci
Fakultet informatike i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

zir.nsk.hr

DIGITALNA KNJIŽNICA

Image not found or type unknown

Sveučilište u Rijeci – Odjel za informatiku

Jednopredmetna informatika

Marina Kaluža

Poslovna aplikacija za upravljanje sastancima u poduzeću nad relacijskom bazom podataka (Clarion)

Završni rad

Mentor: doc. dr. sc. Danijela Jakšić

Rijeka, rujan 2020.

ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI

Rijeka, 15.6.2020.

Zadatak za završni rad

Pristupnik: Marina Kaluža

Naziv završnog rada: Poslovna aplikacija za upravljanje sastancima u poduzeću nad relacijskom bazom podataka (Clarion)

Naziv završnog rada na eng. jeziku: Business application with relational database for planning the business meetings (Clarion)

Sadržaj zadatka:

Baza podataka (BP) predstavlja kolekciju podataka, ograničenja i operacija koji reprezentiraju neke aspekte realnoga svijeta. Dakle, BP je model neke aplikacijske domene. Cilj rada je izgraditi aplikaciju za potrebe procesa upravljanja sastancima u nekom poduzeću. Aplikacija će biti izgrađena nad relacijskom BP pomoću RAD alata Clarion. Na početku će biti specificirani zahtjevi i potrebne značajke aplikacije. Sljedeći korak bit će dizajn baze podataka, odnosno izrada modela entiteti i veze. Transformacijom modela entiteti i veze konstruirat će se relacijski model. Na temelju relacijskog modela izgradit će se SQL skripta. Nakon toga kreće izrada same aplikacije. Aplikacija će raditi nad relacijskom BP u navedenoj aplikacijskoj domeni (upravljanje sastancima).

Mentor

doc.dr.sc. Danijela Jakšić

Voditelj za završne radove

doc. dr. sc. Miran Pobar

Zadatak preuzet:

(potpis pristupnika)

SADRŽAJ

1 UVOD 1
2 SPECIFIKACIJA ZAHTJEVA 2
3 ANALIZA DOKUMENTA 2
4 MODEL PODATAKA 3
4.1 MODEL ENTITETI-VEZE 4
4.2 RELACIJSKI MODEL 5
4.3 SQL SKRIPTA 7
5 IZRADA SOFTVERA 9
5.1 IZRADA RIJEČNIKA 9
5.2 IZRADA APLIKACIJE 18
5.2.1 Procedura za glavni izbornik 18
5.2.2 Procedura za popis 22
5.2.3 Procedura za ažuriranje 24
5.2.4 Procedura za izvješće 29
6 KORIŠTENJE APLIKACIJE 32
7 ZAKLJUČAK 38
LITERATURA 39
POPIS TABLICA 39
POPIS SLIKA 39

1 UVOD

Relacijska baza podataka je bazirana na relacijskom modelu i u njoj su podaci raspoređeni u relacije. Relacijski model je predložio E.F.Codd 1970. godine [5]. Do devedesetih godina prošlog stoljeća relacijske baze podataka su u velikoj mjeri zamijenile hijerarhijske i mrežne baze podataka koje su do tada bile standard za izradu poslovnih aplikacija [8]. Sustav relacijske baze podataka se originalno definirao pomoću Coddovih 12 pravila, ali niti jedna komercijalna implementacija relacijskog modela nije u skladu sa svim pravilima, tako da je pojam relacijske baze podataka počeo opisivati širu klasu sustava baza podataka od početno definiranog [7].

U današnje vrijeme relacijske baze podataka su toliko raširene da je gotovo nemoguće zamisliti modernu tvrtku bez nje. Relacijske baze podataka mogu se definirati i kao one baze podataka koje se mogu ispitivati pomoću SQL-a (engl. Structured Query Language). Upotreba SQL-a olakšava rad s više relacijskih baza podataka. Sveprisutnost aplikacija koje koriste relacijske baze podataka jamči da će se relacijske baze podataka koristiti i u budućnosti [8].

James Martin je 1991. izumio RAD (engl. Rapid Application Development) pristup. RAD spada pod agilne metoda za razvoj softvera koja daje prednost brzim izdanjima i iteracijama prototipa. Za razliku od tradicionalnog pristupa razvoju softvera, gdje se najviše pažnje posvećuje planiranju izrade i specificiranju zahtjeva, kod RAD pristupa se naglašava važnost povratne informacije korisnika koja daje uputstva što treba u sljedećoj iteraciji prototipa nadograditi ili promijeniti [9].

Vrlo dobar RAD alat za izradu softvera je Clarion. Clarion posjeduje jezik 4. generacije kojeg je osmislio Bruce Barrington 1980. U početku on je bio osmišljen za računala za DOS operacijskim sustavom sve do 1994. kad je izašao Clarion za Windowse. Topspeed je razvijao Clarion do 2000. godine kada je SoftVelocity preuzeo razvoj proizvoda [10]. Clarion je jezik čije je integrirano razvojno okruženje (IDE engl. Integrated Development Environment) usko integrirano s bazama podataka. U IDE-u je omogućeno postavljanje tablica ili uvoz iz vanjskih izvora kao što su ODBC ili SQL baza podataka [10]. U izrade ove aplikacije korišten je uvoz pomoću ODBC-a.

Cilj ovog rada je napraviti aplikaciju za upravljanje sastancima u poduzeću. Da bi se došlo do funkcionalne aplikacije treba proći kroz više koraka razvoja. U 2. poglavlju „Specifikacija zahtjeva“ će se navesti što se želi postići aplikacijom, odnosno navesti će se korisnički zahtjevi. U 3.poglavlju će biti naveden glavni dokument sustava. 4. poglavlje će sadržati modela prikaza sustava, prvo model entiteti – veze iz kojeg se pravilima prevođenja dolazi do relacijskog modela. Zadnje potpoglavlje bit će izrada SQL skripte. Nakon toga u 5. poglavlju bit će opširno opisan način izrade rječnika i aplikacije u Clarionu. Zadnje će se pokazati način korištenja aplikacije.

2 SPECIFIKACIJA ZAHTJEVA

Kako bi se specificirali zahtjevi sustava, sustav se analizirao korisničkim pričama. U korisničkim pričama se na jednostavan način zapisuje što će softver moći obavljati. Često su pisane iz perspektive korisnika [1]. Korisničke priče su zapisi potreba korisnika. One obično slijede jednostavan predložak: Kao __(vrsta korisnika)__ želim da __(neki cilj)__ kako bi __(neka svrha)_ [11].

Korisničke priče su dio agilnog pristupa razvoju informacijskih sustava. Agilni razvoj fokusira se na prilagodljivosti procesa razvoja i cilj je u što kraćem roku korisniku isporučiti softver [2]. U razvoju aplikacije nije bitno da bude točno navedeno na koji način će se nešto u softveru napraviti, nego je bitno isporučiti softver ili prototip softvera za koji onda korisnik daje povratnu informaciju što treba nadodati, prepraviti.

U sljedećoj tablici su prikazane korisničke priče kojima se specificiraju zahtjevi sustava.

Tablica 1 Korisničke priče

Kao	Želim da	Kako bi
Uprava	Mogu unijeti zaposlenike i njihova radna mjesta	Se znalo kojim kadrovima pripadaju
Uprava	Mogu unijeti dnevni red novog sastanka	Osobe koje trebaju sudjelovati na sastanku bile pravovremeno obavještene
Partner	Dobiti izvješće sastanka	Znali što je na sastanku zaključeno

3 ANALIZA DOKUMENTA

Kao glavni dokument traženog sustava izdvaja se izvještaj zapisnika sa sastanka. Na sljedećoj slici je prikazan primjer izvještaja zapisnika sastanka. Na zapisniku su promjenjivi podaci zaokruženi odnosno oni koje se treba spremati u bazu podataka. Tipovi podataka koji se trebaju moći spremiti u bazu podataka su datum sastanka, lokacija sastanka, partner koji sudjeluje na sastanku, podaci o partnerskoj firmi (ulica, mjesto), zaposlenici i partnerovi zaposlenici koji su sudjelovali na sastanku te koji su ciljevi sastanka odnosno točke dnevnog reda i što se na sastanku zaključilo.

IZVJEŠTAJ ZAPISNIKA SASTANKA

Datum: 20.8.2020.
Lokacija: Eksterni

Partner:
Sky tours
Vukovarska ulica 35
Rijeka

Zaposlenici:
Ana Anić
Ivo Ivić

Zaposlenici partnera:
Mara Marić
Marko Markić

Točke dnevnog reda:
1. Ugovaranje ponude – Što sve traženi sustav treba sadržati
2. Razno

Zaključci:
1. Proizvod će sadržati sve potrebno
2. Za mjesec dana potreban je sljedeći sastanak da se analizira napredak

Slika 1 Izvještaj zapisnika sastanka

4 MODEL PODATAKA

Model podataka je apstraktni model koji organizira podatke i standardizira njihov međusobni odnos i njihova svojstva. Model podatak se može odnositi na dva pojma. Prvi se odnosi na apstraktnu formalizaciju objekata i odnosa. Drugi slučaj se odnosi na skup pojmova koji se koriste za definiranje formalizacija npr. entitet, atribut [4].

Model podataka se može definirati i kao pojednostavljena reprezentacija sustava preko skupa entiteta, veza među entitetima, atributima entiteta te agregacijama entiteta. To je prikaz skupa podataka koji se modelom interpretiraju preko aspekta: strukture, ograničenja i operatora [3]. Struktura modela podataka je skup entiteta i veza koji interpretirajući podatke ih klasificira u tipove entiteta i tipove veza između tipova entiteta. Ograničenja modela podataka su koncepti strukture modela koji omogućuju daljnju interpretaciju podataka. Operatori modela čine skup koncepata kojima se omogućuju dinamičke karakteristike skupa podataka. Koncepti strukture i ograničenja spadaju pod statička svojstva poslovnog sustava, a operatori se vezuju sa promjenom stanja podataka u bazi podataka u skladu sa promjenom istih u poslovnome sustavu [3].

4.1 MODEL ENTITETI-VEZE

Model entiteti – veze grafički je prikaz međusobno povezanih podataka svrstanih u grupe iz promatranog sustava. Osnovni koncepti u modelu entiteti – veze su: entitet i tip entiteta, veza i tip veze, atribut, slabi tip entiteta i specijalni tipovi veze, agregirani tip entiteta, povratni tip entiteta i generalizacijski tip veze [3]. Entitet je nešto što se može jednoznačno imenovati, to može biti predmet, događaj, apstraktni pojam… Tip entiteta je skup entiteta istog tipa, entiteti pripadaju istom tipu entiteta ako imaju slične karakteristike [3]. Slično se dobiva i razlika između veze i tipa veze, pri čemu je veza odnos između dva entiteta, dok je tip veze skup veza između istih tipova entiteta [3].

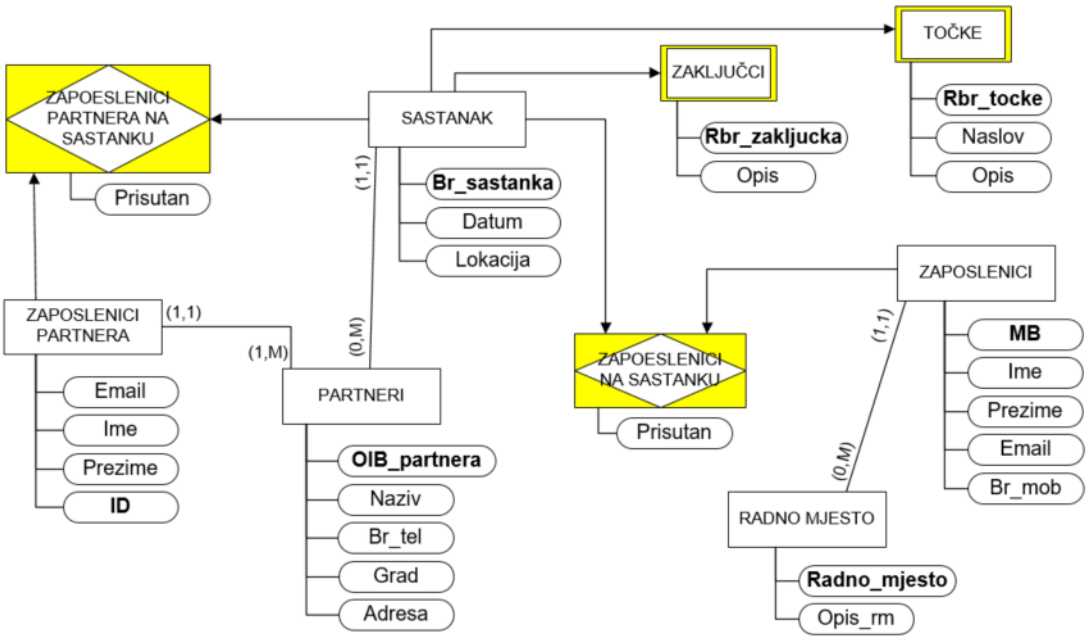
Za kreiranje odabranog sustava korišteni su jaki tip entiteta, slabi tip entiteta te agregacija.

Jaki tip entiteta nije ovisan o drugim tipovima entiteta i ima vlastiti primarni ključ (na slici 2 atributi prikazani podebljano). Jaki entiteti u izrađenom sustavu su: sastanak, partner, zaposlenici partnera, zaposlenici i radno mjesto.

Slabi tip entiteta je ovisan o drugom tipu entiteta, a veza između ta dva entiteta se prikazuje specijalnim tipom veze. Postoji više vrsta specijalnih tipova veze, a u ovom sustavu je korišten egzistencijalni-identifikacijski (E-I) tip veze u kojoj je slabi tip entiteta egzistencijalno i identifikacijski ovisan u odnosu na jak tip entiteta [3]. Npr. slabi entitet zaključci egzistencijalno i identifikacijski ovisi o jakom entitetu sastanci, i brisanjem nekog sastanka brišu se i svi zaključci spremljeni za taj sastanak. Isto vrijedi i za slabi entitet točke.

Agregacija ili agregirani tip entiteta se koristi kad između dva entiteta postoji veza (M,M) više prema više koju se treba razložiti na dvije (1,M) veze. To se vidi na primjeru zaposlenici na sastanku, pošto jedan ili više zaposlenika može prisutstvovati na nijednom, jednom ili više sastanka te se na jednom ili više sastanka može pojaviti jedan ili više zaposlenika, vidljivo je da se radi o (M,M) vezi između entiteta sastanak i zaposlenici i taj problem je riješen dodavanjem agregiranog tipa entiteta „zaposlenici na sastanku“.

Sljedeća slika prikazuje model entiteti – veze za odabrani sustav. Model je dobiven pomoću podataka koji su dobiveni u analizi dokumenata. Početni tip entiteta od kojeg se krenulo je bio „sastanak“. Tipovi entiteta „zaključci“ i „točke“ jesu slabi tipovi entiteta pošto su prvo određeni sa sastankom na koji se odnose. Tip entiteta „partner“ je povezan sa „sastanak“ kako bi se znalo, ako je partner prisutstvovao sastanku koji je to partner bio. Iz analiziranog dokumenta vidljivo je da se još trebalo dodati zaposlenike i partnerove zaposlenike. Zaključuje se da tip entiteta „zaposlenici“ je povezan sa tipom entiteta „sastanak“ vezom (M,M) pa je između ta dva entiteta dodana agregacija kako bi se ta veza mogla realizirati. Isto vrijedi i za partnerove zaposlenike. U tipu entiteta „zaposlenici_partnera“ se planira dodati „unique“ ograničenje kako bi partnerov zaposlenik bio jednoznačno određen „OIB-om partnera“ i „email-om“ zaposlenika. Ovaj tip entiteta ima primarni ključ samo polje „ID“ umjesto polja „OIB_partnera“ i „email“ kako bi ga bilo „jednostavnije“ povezati s agregacijom „zaposlenici partnera na sastanku“. I zadnje dodan je tip entiteta „radno_mjesto“ kako bi se znalo kojem radnom mjestu zaposlenici pripadaju.



Slika 2 Model entiteti-veze

4.2 RELACIJSKI MODEL

Relacijski model podataka ili relacijska metoda je metoda za modeliranje podataka u skladu s pravilima organizacije relacijskih baza podataka [5]. U relacijskom modelu podaci su raspoređeni u relacije, a relacije se sastoje od slogova [6]. U relacijama postoje dvije vrste ključeva primarni i vanjski ključevi. Primarni ili glavni ključ je onaj najmanji skup atributa koji funkcijski određuju ostale neključne atribute. Vanjski ključ je referenca na atribut koji predstavlja primarni ključ povezane relacije.

U relacijskom modelu između ostalih ne postoje slabi tipovi entiteta, agregacie i specijalni tipovi veza. Zbog toga prevođenje relacijskog modela u model entiteti – veze nije moguće [3].

Postoji 18 pravila za prevođenje EV modela u relacijski model [3].

1. pravilo – vrijednost u Dijagramu entiteti – veze (DEV) odgovara vrijednostima u relacijskom modelu (RM)
2. pravilo – tip vrijednosti u DEV-u odgovara konceptu domena u RM-u
3. pravilo – atribut u DEV-u odgovara konceptu atributa u RM-u
4. pravilo – entitet u DEV-u odgovara slogu (n-torki) relacije u RM-u
5. pravilo – kandidat za ključ tipa entiteta u DEV-u postaje kandidat za ključ tablice u RM-u
6. pravilo – svaki tip entiteta DEV-a postaje relacija u RM-u, tako da atributi tipa entiteta postaju atributi relacije, a primarni ključ tipa entiteta postaje primarni ključ relacije
7. pravilo – izvedeni atribut u DEV-u odgovara konceptu izvedeni atribut u RM-u

8. pravilo – u DEV-u nije dopušten viševrijednosni atribut, ali ako takav atribut ipak postoji tada se on pretvara u tip entiteta gdje brojnosti atributa postaju brojnosti novog tipa veze
9. pravilo – agregacija iz DEV-a postaje zasebna relacija u RM-u kojoj je primarni ključ složen od primarni ključeva entiteta koji sudjeluju u vezi
10. pravilo – svaki tip veze s brojnostima se ne prevodi u relaciju, nego se u entitet na čijoj strani veze je brojnost M umeće atribut (vanjski ključ) koji je primarni ključ u entitetu na čijoj je strani veza (1,1)
11. pravilo – svaki (1,1):(1,1) tip veze se ne prevodi u relaciju
12. pravilo – svaki (0,1):(1,1) tip veze i (0,1):(0,1) tip veze se prevodi po 11.pravilu
13. pravilo – ako tip veze povezuje više od dva tipa entiteta tada se taj tip veze razlaže u binarne tipove veza (više povezanih agregacija)
14. pravilo – specijalni tipovi veze „identifikacijska (ID)“, „egzistencijala (E)“ i „ID & E“ ne prevode se
15. pravilo – tip slabog entiteta prevodi se u relaciju na način da atributi slabog tipa entiteta postaju atributi relacije, a ključ relacije je složen od ključa jakog i slabog tipa entiteta
16. pravilo – tip veze definiran nad istim tipom entiteta (povratni tip veze) prevodi se u relacijski model kao i drugi tipovi veze, u ovisnosti o brojnosti tipa veze
17. pravilo – generalizacija se prevodi u RM tako da svaki tip entiteta postaje relacija, ključ svih relacija je jednak ključu tipa entiteta na vrhu generalizacije, a atributi tipova entiteta postaju atributi relacije
18. pravilo – slaba agregacija u DEV-u postaje relacija u RM-u, takva da atributi slabe agregacije postaju atributi relacije, a ključ te relacije je složen od ključeva tipova entiteta koji sudjeluju u agregacijskoj vezi i ključa slabog tipa entiteta u slaboj agregaciji.

Slijedi prikaz relacijskog modela dobivenog iz ranije navedenog DEV-a. Primarni ključevi relacija su prikazani podebljano, a vanjski ključevi su ukošeno. Atributi koji su prikazani podebljano i ukošeno jesu primarni ključevi u tim shemama relacija, ali isto tako njihove vrijednosti se dobivaju iz atributa drugih shema relacija (vanjski ključ) pa su zbog toga i ukošeni.

RADNO MJESTO (**radno_mjesto**, opis_rm)

ZAPOSLENICI (**MB**, ime, prezime, br_mob, e-mail, *radno mjesto*)

PARTNERI (**OIB_partnera**, naziv, br_tel, grad, adresa)

ZAPOSLENICI PARTNERA (*OIB_partnera*, e-mail, ime, prezime, **ID**)

SASTANAK (**br_sastanka**, datum, lokacija, *OIB_partnera*)

TOČKE DNEVNOG REDA (***br_sastanka*, rbr_tocke**, naslov, opis)

ZAKLJUČCI (***br_sastanka*, rbr_zakljucka**, opis)

ZAPOSLENICI_NA_SASTANKU (***br_sastanka, mb***, prisutan)

ZAPOSLENICI_PARTNERA_NA_SASTANKU (***br_sastanka, ID***, prisutan)

4.3 SQL SKRIPTA

Strukturirani upitni jezik engl. Structure Query Language (SQL) je jezik za rad s bazama podataka (BP) [6]. Jedan je od najčešće korištenih jezika za relacijske baze podataka. Kao jezik za rad s bazama podataka sadrži načine za obavljanje ove tri funkcije: kreiranje baze podataka Data Definition Language (DDL), izmjenu baze podataka Data Manipulation Language (DML), obradu podataka engl. Data Query Language (DQL) i upiti na bazu engl. Data Dictionary (DD) [6].

Slijedi skripta dobivena iz relacijskog modela za izrađivani sustav.

```
create table RADNO_MJESTO (
     radno_mjesto varchar(32) not null comment 'Radno mjesto',
     opis_rm text comment 'Opis radnog mjesta',
     primary key (radno_mjesto)
);
create table ZAPOSLENICI(
     MB int not null comment 'Matični broj',
     ime varchar(24) not null comment 'Ime radnika',
     prezime varchar(24) not null comment 'Prezime radnika',
     email varchar(40) not null comment 'Email radnika',
     br_mob varchar(9) comment 'Broj mobitela',
     radno_mjesto varchar(32) not null comment 'Radno mjesto',
     primary key (MB),
     foreign   key   (radno_mjesto)   references   RADNO_MJESTO
(radno_mjesto) on update cascade on delete restrict
);
create table PARTNERI(
     OIB_partnera varchar(12) comment 'OIB partnera',
     naziv varchar(24) not null comment 'Ime partnera',
     br_tel varchar(20) comment 'Broj telefona',
     grad varchar(32) comment 'Grad',
     adresa varchar(32) comment 'Adresa partnera',
     primary key (OIB_partnera)
);
create table SASTANAK(
     br_sastanka int not null comment 'Broj sastanka',
     datum date not null comment 'Datum sastanka',
     lokacija varchar(32) comment 'Lokacija sastanka',
     OIB_partnera varchar(12) not null comment 'OIB partnera',
     primary key (br_sastanka),
     foreign key (OIB_partnera) references PARTNERI (OIB_partnera)
          on update cascade on delete restrict
);
create table zaposlenici_na_sastanku(
     br_sastanka int not null comment 'Broj sastanka',
     MB int not null comment 'Matični broj',
     prisutan tinyint comment 'Prisutan na sastanku',
```

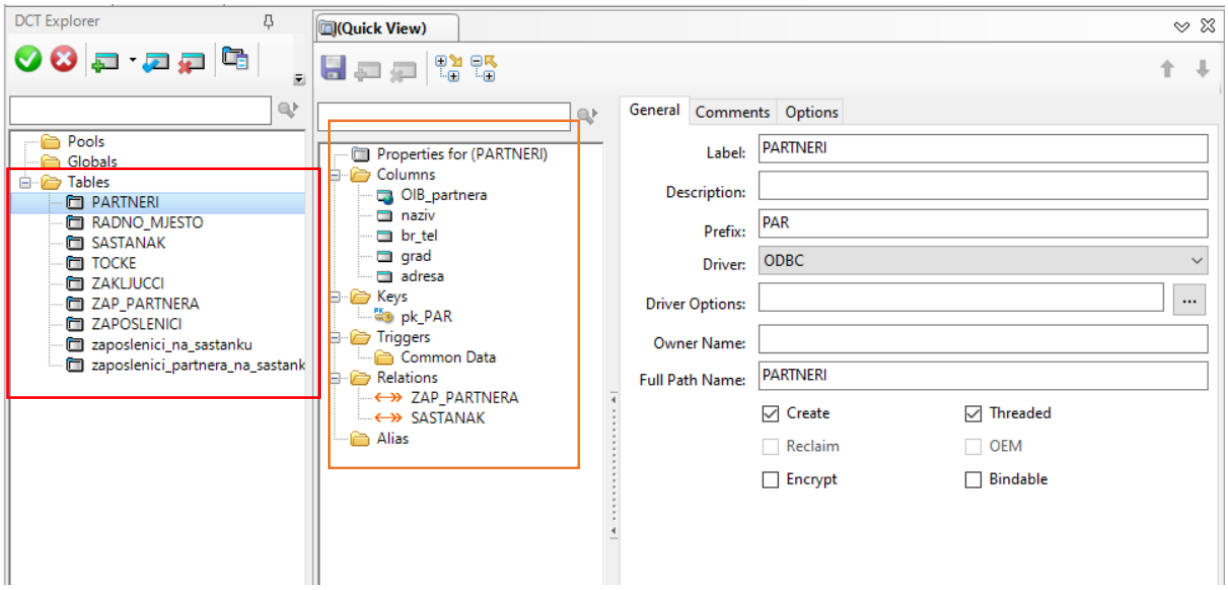
```
     primary key (br_sastanka, MB),
     index rk_ZNS (MB, br_sastanka),
     foreign key (br_sastanka) references SASTANAK (br_sastanka)
          on update cascade on delete cascade,
     foreign key (MB) references ZAPOSLENICI (MB)
          on update cascade on delete cascade
);
create table ZAP_PARTNERA(
     OIB_partnera varchar(12) not null comment 'OIB partnera',
     email varchar(40) not null comment 'Email radnika',
     ime   varchar(32)   not   null   comment   'Ime   partnerovog
zaposlenika',
     prezime varchar(24) not null comment 'Prezime partnerovog
zaposlenika',
     ID int not null comment 'ID zaposlenika partnera',
     primary key (ID),
     unique index (OIB_partnera, email),
     foreign key (OIB_partnera) references PARTNERI (OIB_partnera)
          on update cascade on delete cascade
);
create table zaposlenici_partnera_na_sastanku(
     br_sastanka int not null comment 'Broj sastanka',
     ID int not null comment 'ID zaposlenika partnera',
     prisutan tinyint comment 'Prisutan na sastanku',
     primary key (br_sastanka, ID),
     index rk_ZPNS (ID, br_sastanka),
     foreign key (br_sastanka) references SASTANAK (br_sastanka)
          on update cascade on delete cascade,
     foreign key (ID) references ZAP_PARTNERA (ID)
          on update cascade on delete cascade
);
create table TOCKE(
     br_sastanka int not null comment 'Broj sastanka',
     rbr_tocke int not null comment 'Redni broj točke',
     naslov varchar(24) not null comment 'Naslov točke',
     opis text comment 'Opis točke',
     primary key (br_sastanka, rbr_tocke),
     foreign key (br_sastanka) references SASTANAK (br_sastanka)
          on update cascade on delete cascade
);
create table ZAKLJUCCI(
     br_sastanka int not null comment 'Broj sastanka',
     rbr_zakljucka int not null comment 'Redni broj zaključka',
     opis text comment 'Opis točke',
     primary key (br_sastanka, rbr_zakljucka),
     foreign key (br_sastanka) references SASTANAK (br_sastanka)
          on update cascade on delete cascade );
```

5 IZRADA SOFTVERA

5.1 IZRADA RIJEČNIKA

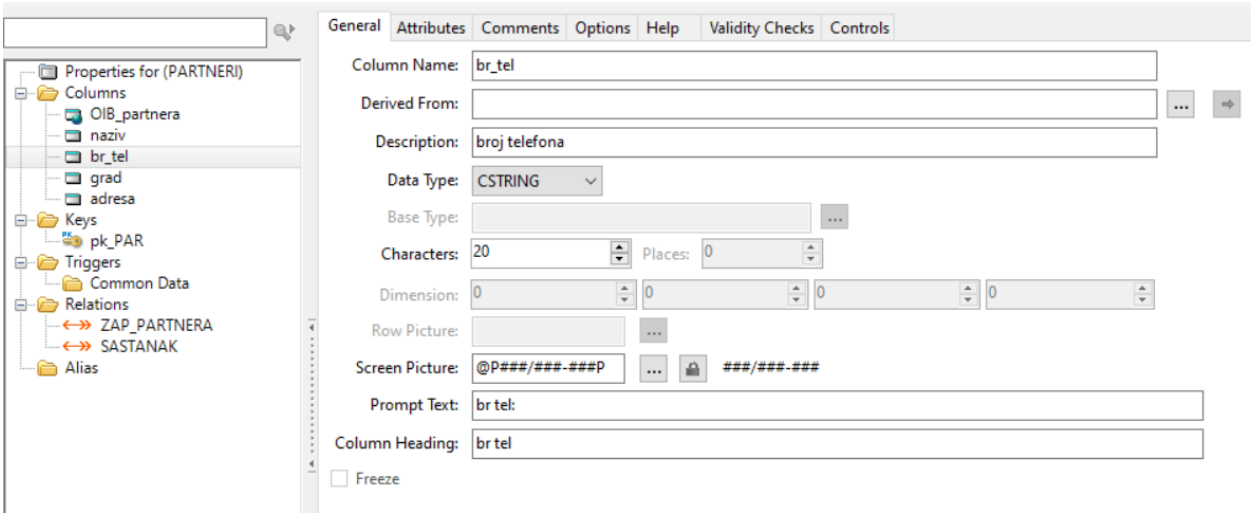
Izradom rječnika dodale su se dodatne restrikcije za neke atribute koje u prethodnim modelima nisu bile dostupne npr. ograničenje prikaza (engl. screen picture), prompt ime za formu za ažuriranje, ime headera za zaglavlje liste, dodatne prilagodbe prikaza polja za upis (postavke validacije (engl. validity checks) koje direktno utječu na aplikacijske kontrole prikaza podataka na prozoru. Dodan je i ključ za sortiranje.

Na sljedećoj slici se vidi popis svih tablica (označeno crvenom bojom) koji odgovara svim shemama relacija iz relacijskog modela. Označeno narančastom se vidi od čega se sastoji tablica partneri (stupci, ključevi i veze).

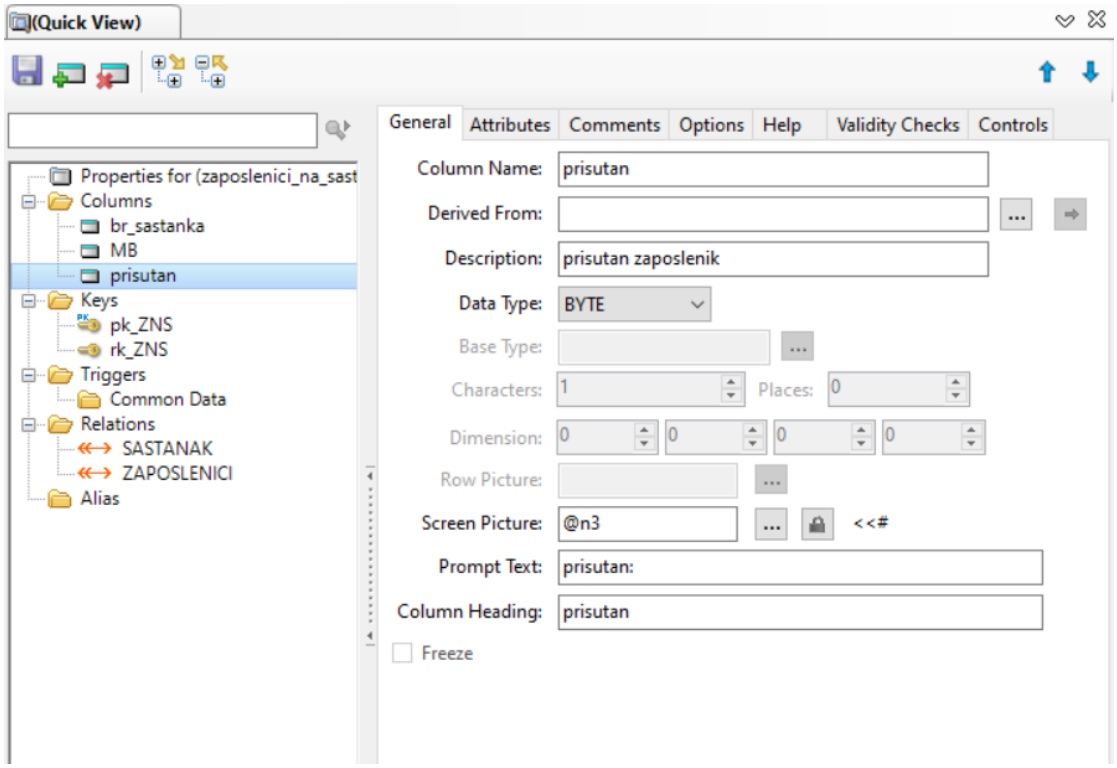


Slika 3 Lista svih tablica u rječniku i podaci o tablici partneri

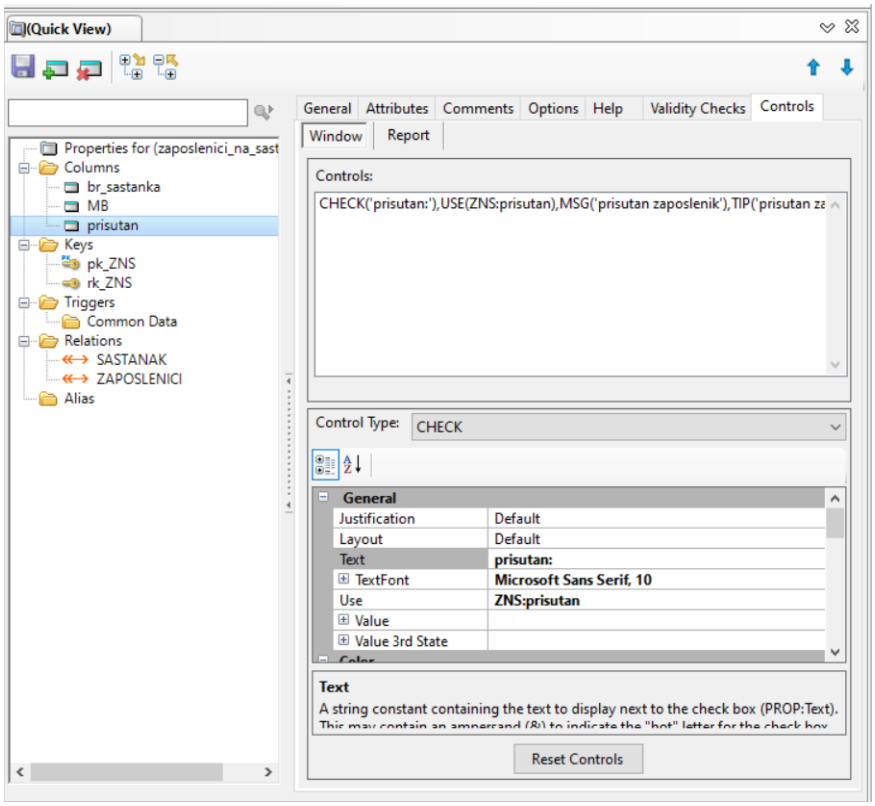
Na sljedećim slikama se vidi dodane restrikcije za neka polja. Na slici 4 je dodan „screen picture“ tako da se unosom broja telefona promijeni prikaz. Na slikama 5 i 6 vidimo restrikciju na polju „prisutan“ u tablici „zaposlenici na sastanku“, gdje je polje tipa byte , a unos je ostvaren pomoću checkboxa. Na slici 7 je unos u polje lokacija ograničeno s listom izbora kojom lokacija može biti samo „interni“ ili „eksterni“ sastanak.



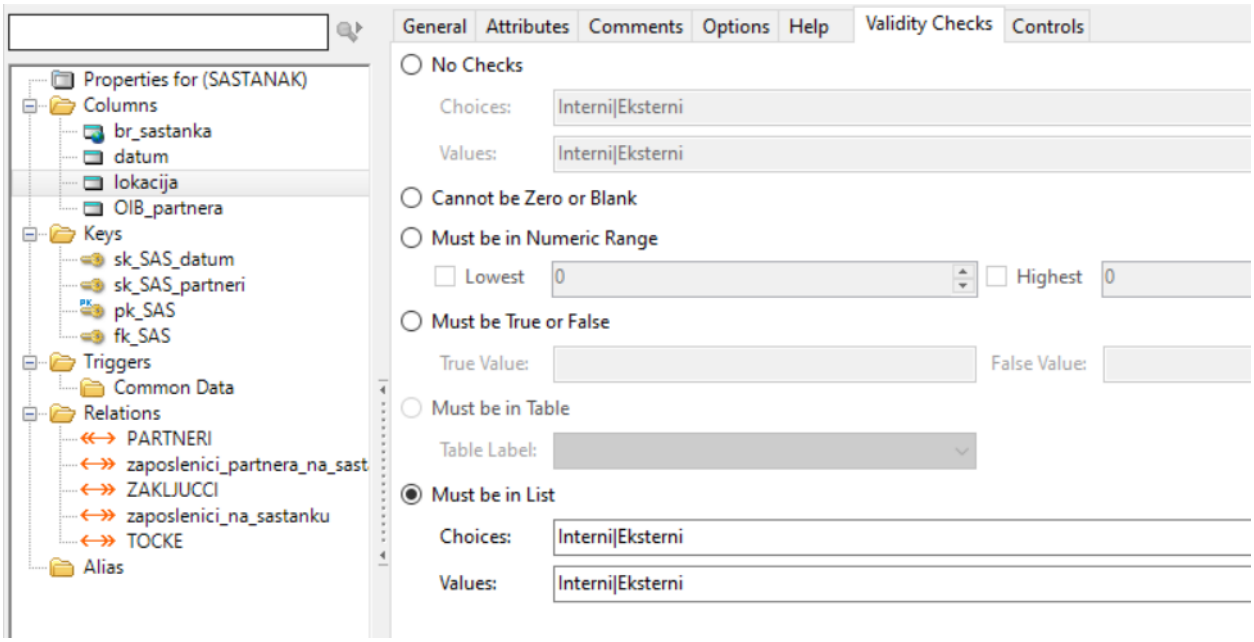
Slika 4 Restrikcija na atributu telefon



Slika 5 Polje tipa byte

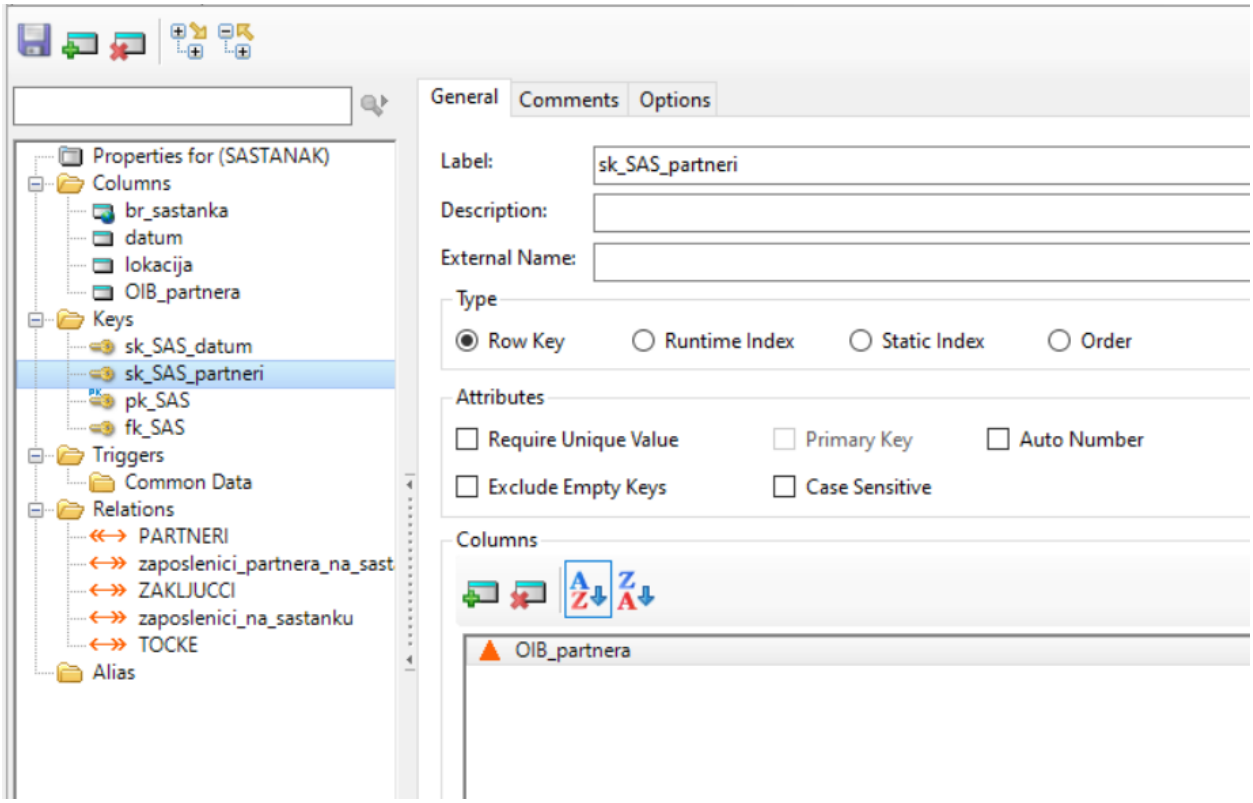


Slika 6 Unos pomoću checkboxa

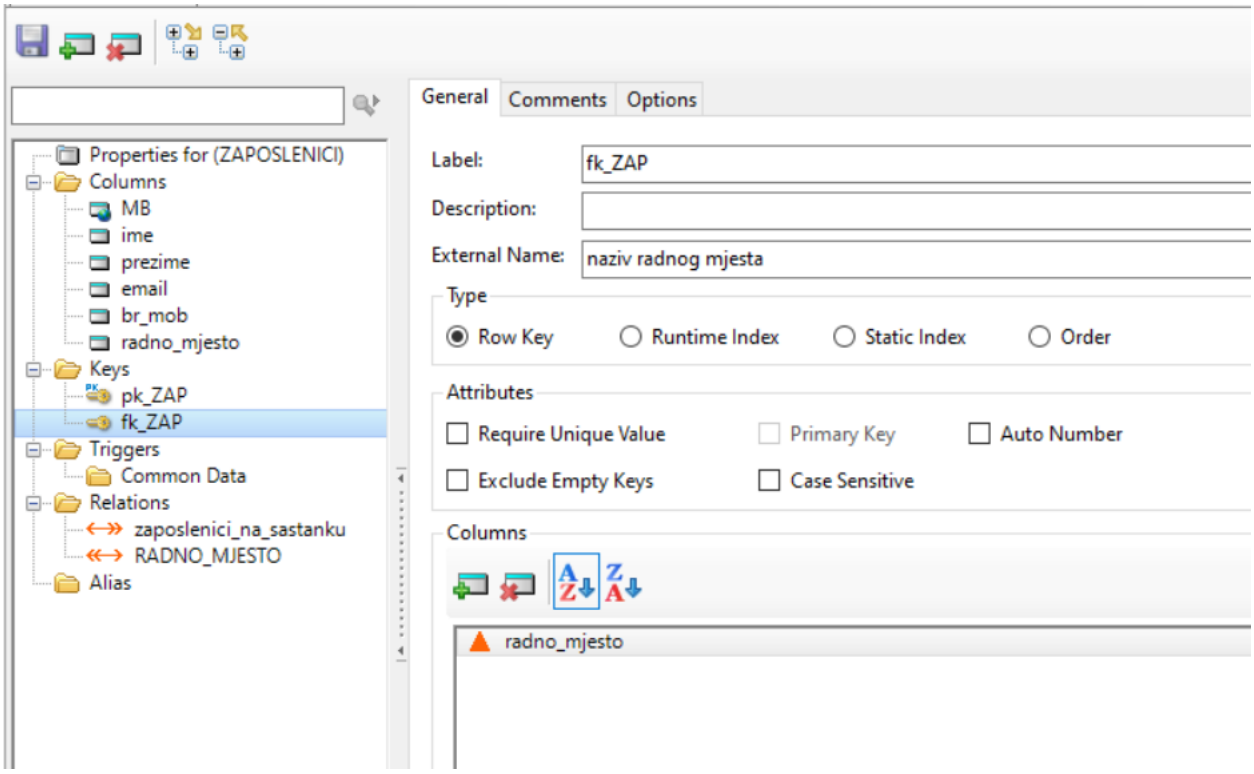


Slika 7 Unos pomoću liste opcija

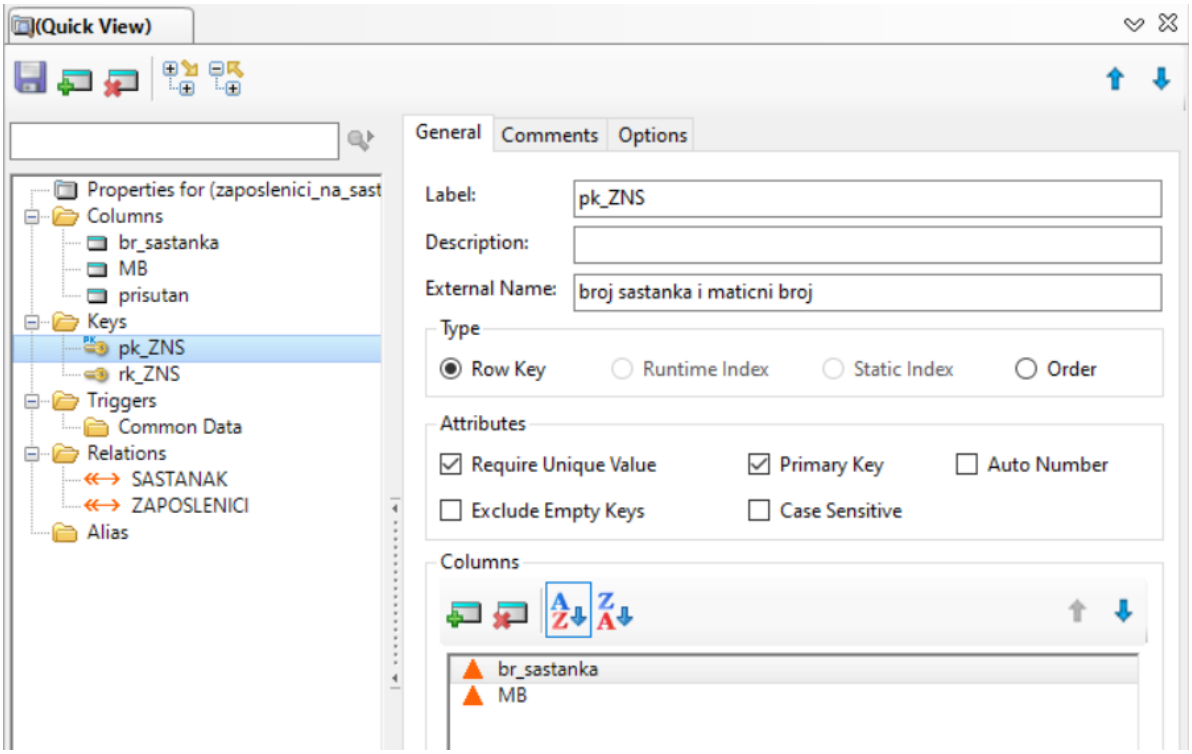
Na sljedećim slikama prikazani su ključevi. Na slici 8 je prikazan ključ za sortiranje koji se odnosi na polje „OIB_partnera“. Na slici 9 je primjer vanjskog ključa koji se odnosi na polje „radno_mjesto“ i tablici „zaposlenici“ koje se dobiva iz tablice „radno mjesto“. Na slikama 10 i 11 se nalaze primarni i rezervni ključevi koji su potrebni kako bi se mogle napraviti veze sa dvije tablice pomoću čijih je primarnih ključeva agregacija napravljena. Slika 12 prikazuje uključenu opciju autonumber u primarnom ključu pomoću koje će se polje „rbr_tocke“ automatski povećavati za svaku sljedeću točku dnevnog reda u sastanku.



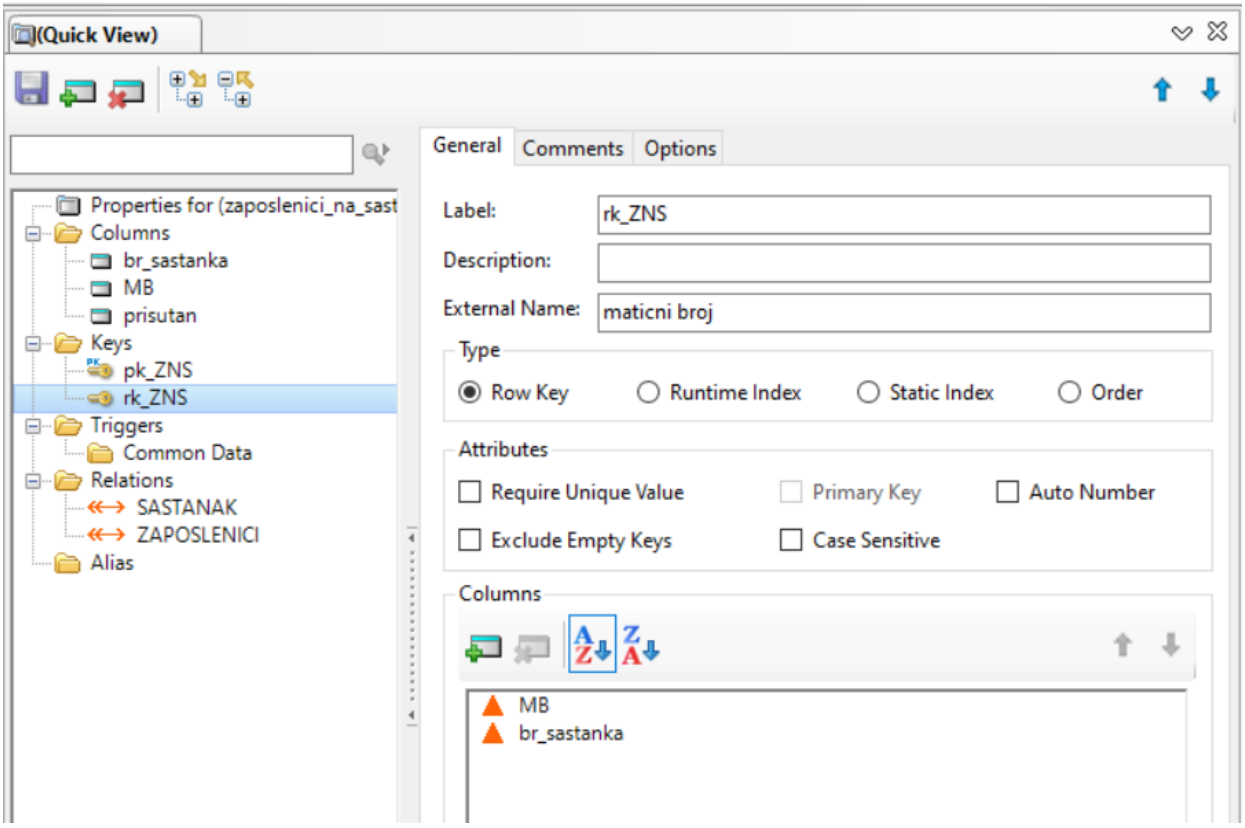
Slika 8 Ključ za sortiranje u tablici sastanaka



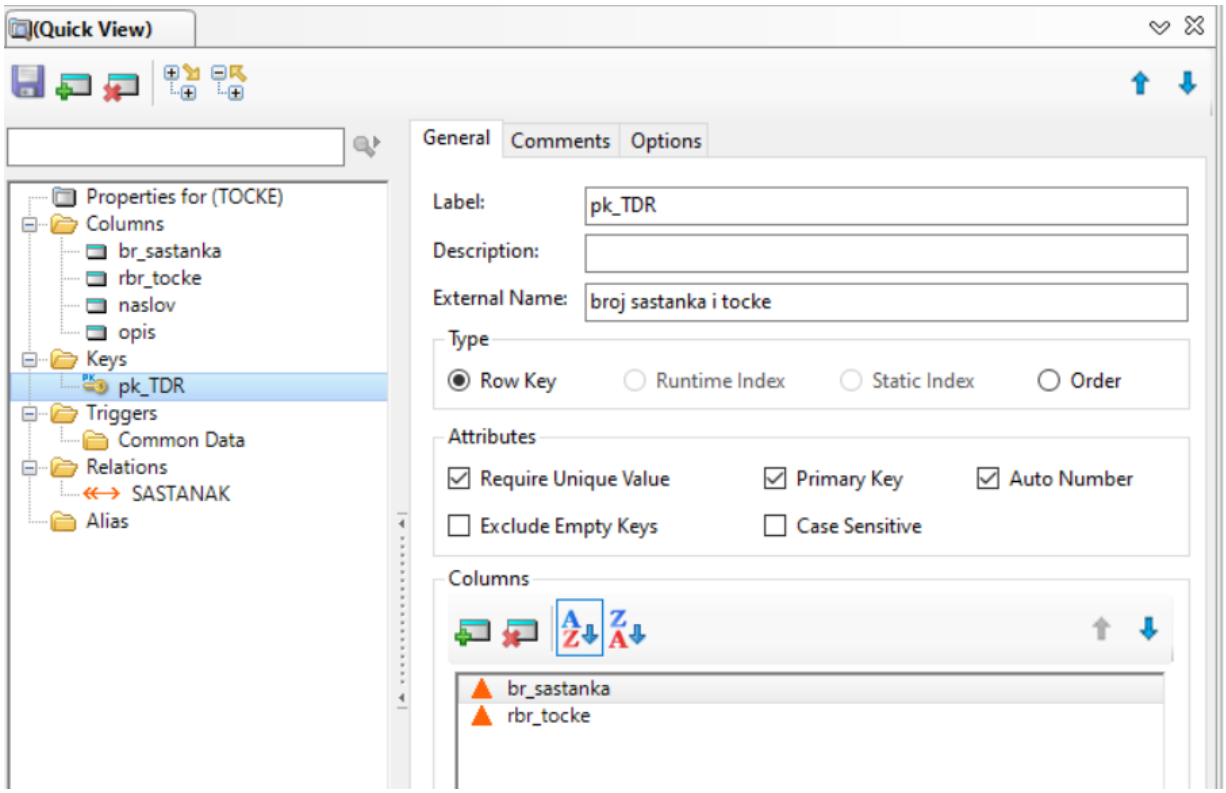
Slika 9 Vanjski ključ za radno mjesto u tablici zaposlenika



Slika 10 Primarni ključ u agregaciji

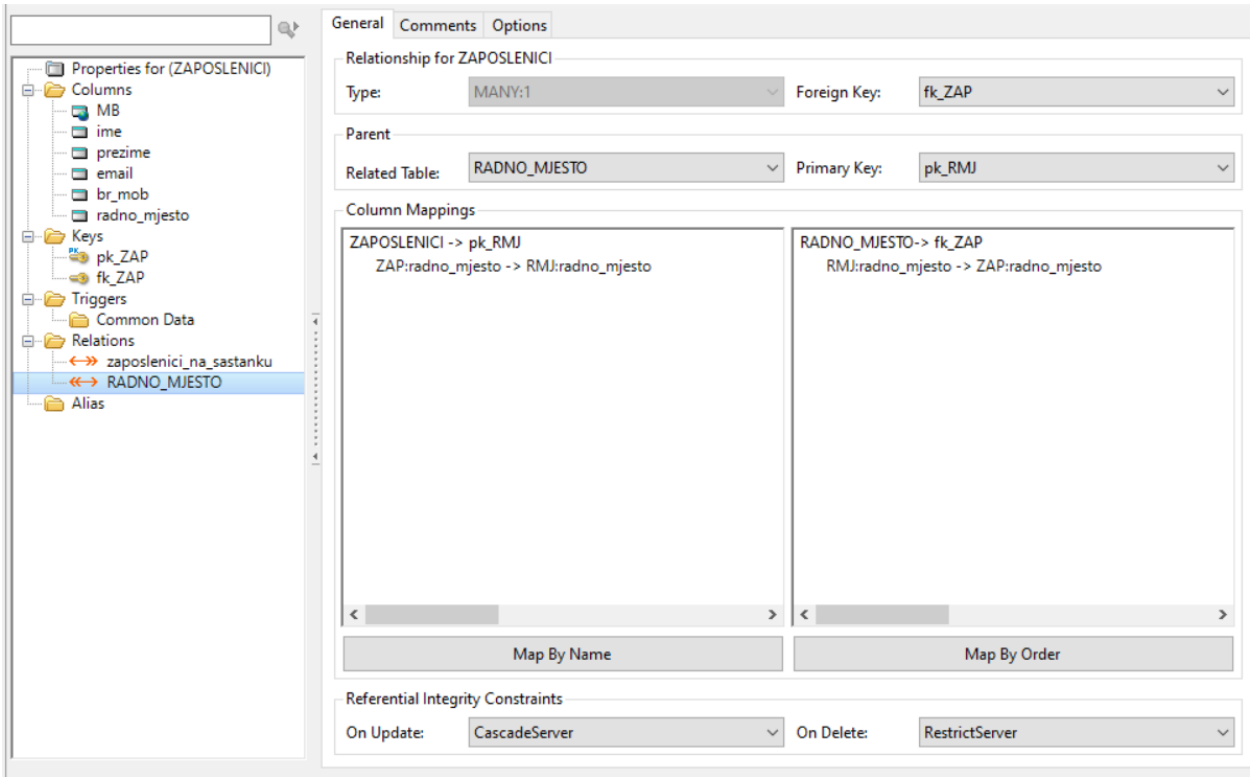


Slika 11 Rezervni ključ u agregaciji

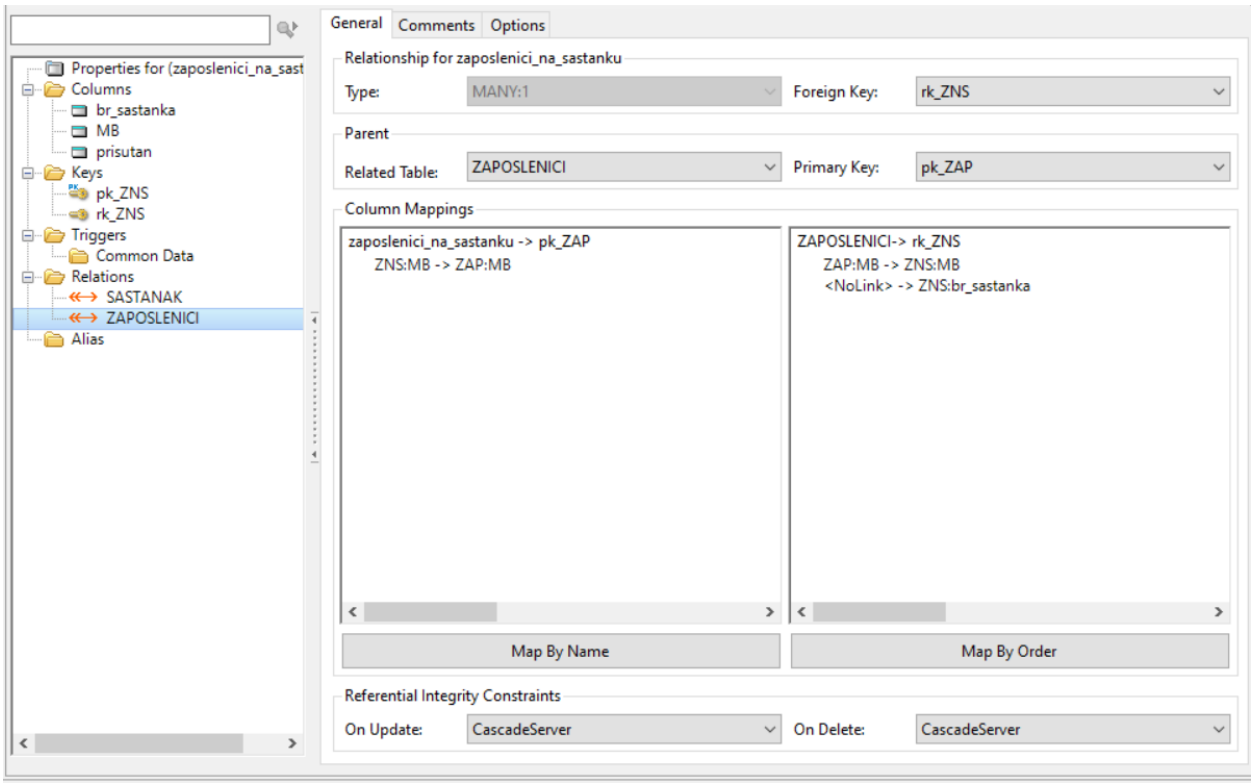


Slika 12Autonumber u primarnom ključu

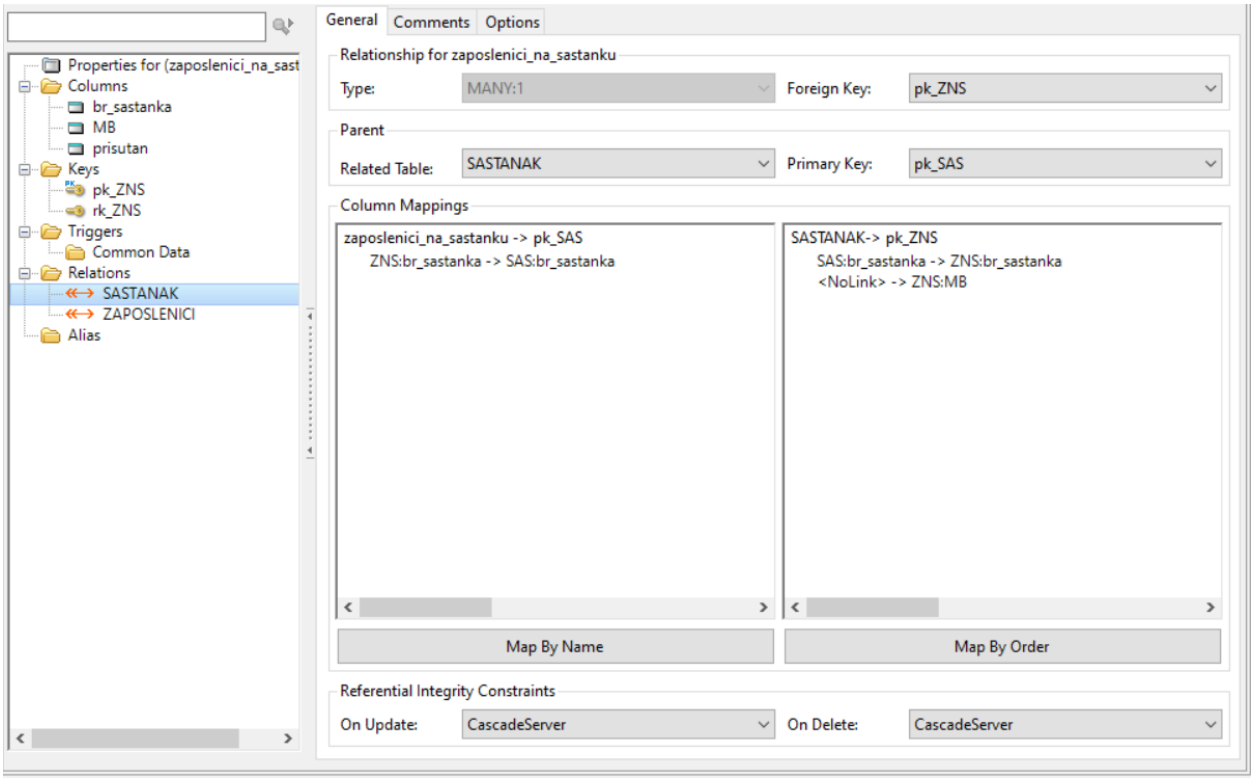
Sljedeće slike prikazuju izglede veza između tablica. Slike 14 i 15 pokazuju dvije veze koje su potrebne za agregaciju. Slika 16 pokazuje vezu u slabom entitetu. Dok se slika 13 odnosi na običnu vezu. Može se uočiti kako je kod agregacija i slabog entiteta uključena opcija „cascadeserver“ za brisanje podataka, dok kod obične veze je uključeno „restrictserver“. Kod obične veze kao što je npr. veza između tablica „zaposlenici“ i „radno mjesto“ opcija „restrictserver“ za brisanje uključena je kako bi se onemogućilo brisanje određenog radnog mjesta prilikom brisanja nekog zaposlenika iz tablice „zaposlenici“. Takva situacija nije povoljna za slabe entitete i agregacije jer npr. brisanje nekog sastanka iz tablice „sastanci“ se ne bi moglo izvršiti dok sve točke dnevnog reda, zaključci, zaposlenici i partnerovi zaposlenici koji su vezani na taj sastanak nisu izbrisani, što se rješava uključivanjem opcije „cascadeserver“.



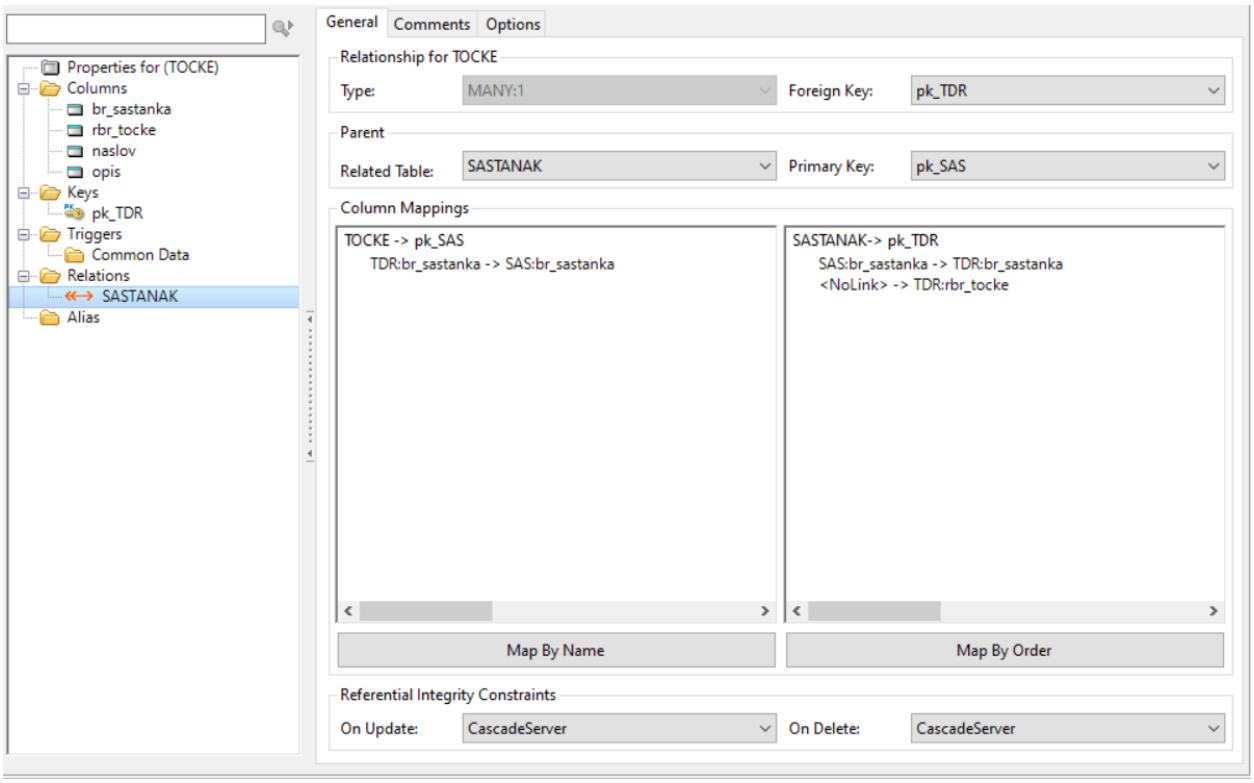
Slika 13 Veza između tablica „zaposlenici“ i „radno mjesto“



Slika 14 Veza u agregaciji „zaposlenici na sastanku“, pomoću rezervnog ključiča



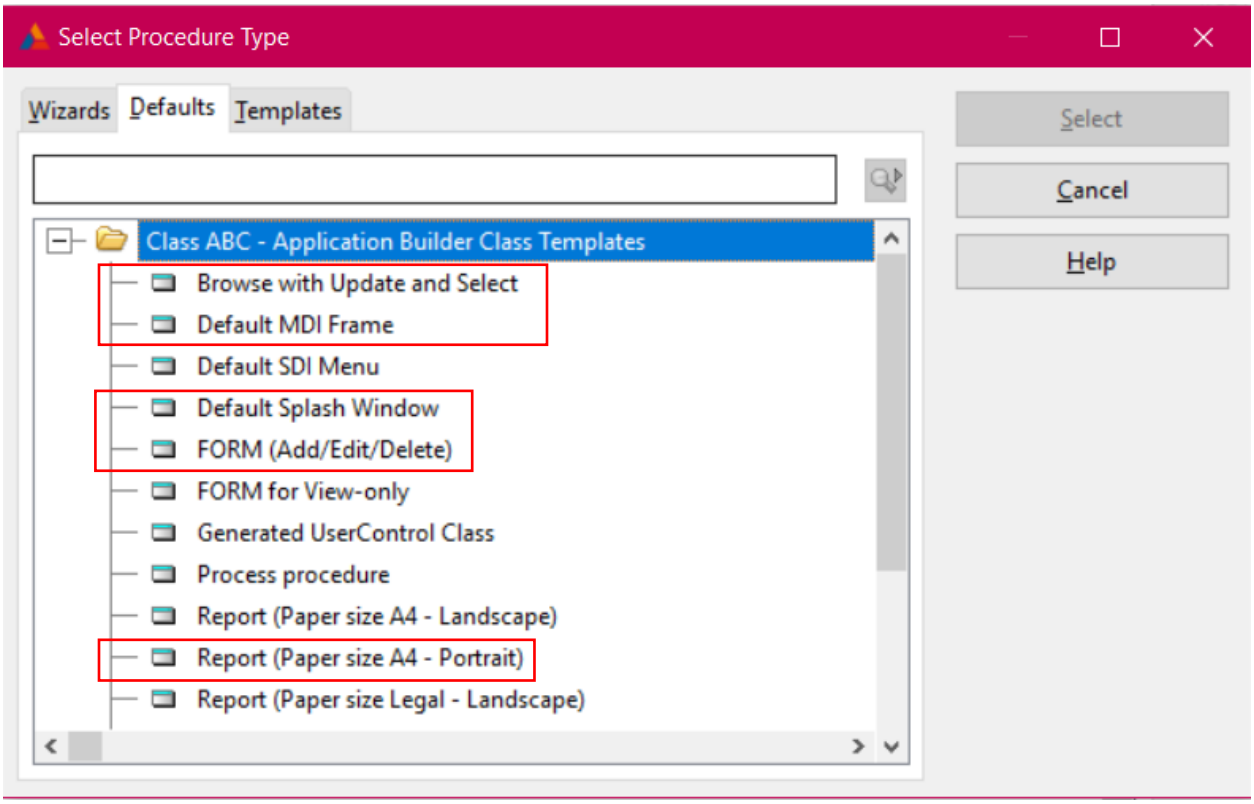
Slika 15 Veza u agregaciji pomoću primarnog ključa



Slika 16 Veza u slabom entitetu u tablici „točke dnevnog reda“

5.2 IZRADA APLIKACIJE

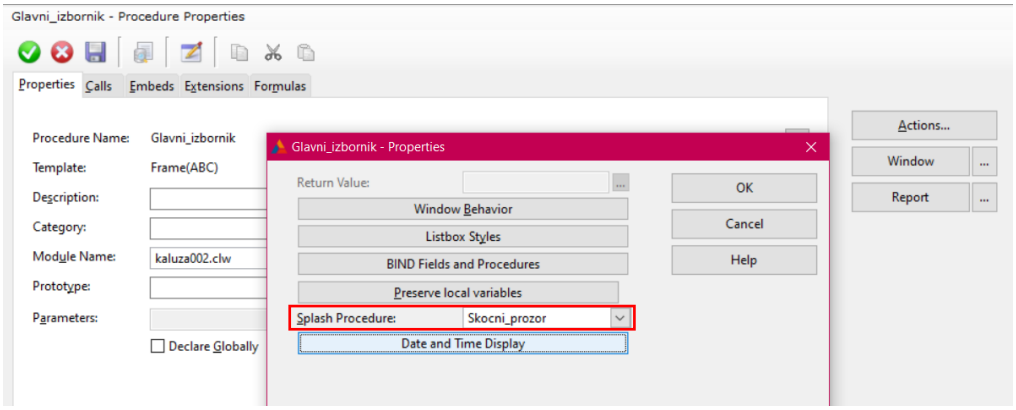
Sa gotovim rječnikom se krenulo u izradu aplikacije. Korišteni su defaultni templati: „Default MDI Frame“ za glavni izbornik, „Browse with Update and Select“ za popise, „FORM (Add/Edit/Delete)“ za ažuriranja, „Default Splash Window“ za skočni prozor te „Report (Paper size A4 - Portrait)“ za izvješća. Na sljedećoj slici su korišteni templatei označeni.



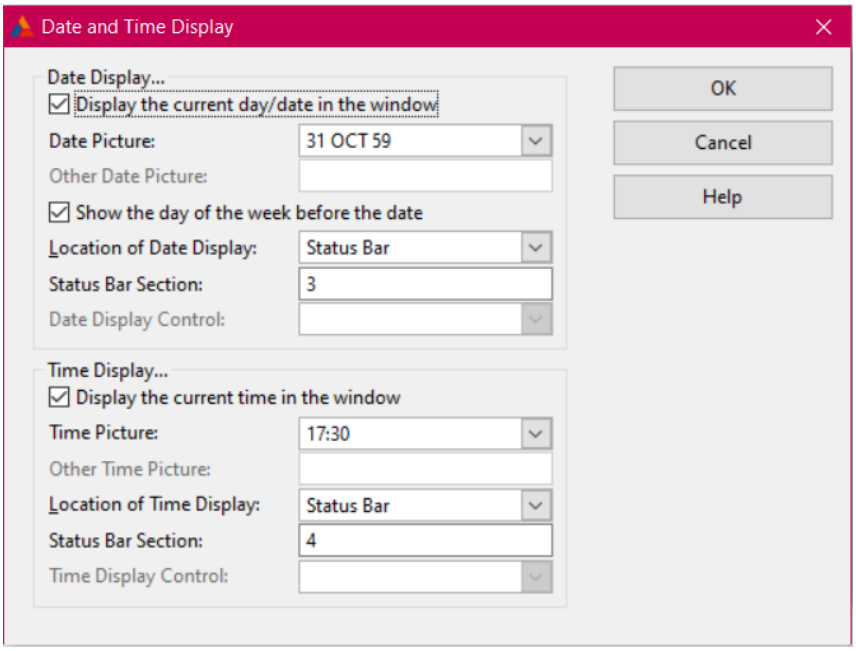
Slika 17 Templati u izradi aplikacije

5.2.1 Procedura za glavni izbornik

Glavna procedura je nazvana „Glavni_izbornik“, a korištena procedura je „Default MDI Frame“. U postavkama je podešeno postojanje skočnog prozora i dodan je prikaz datuma i vremena na glavnom prozoru. Slike 18 i 19 pokazuju te funkcije uključene.

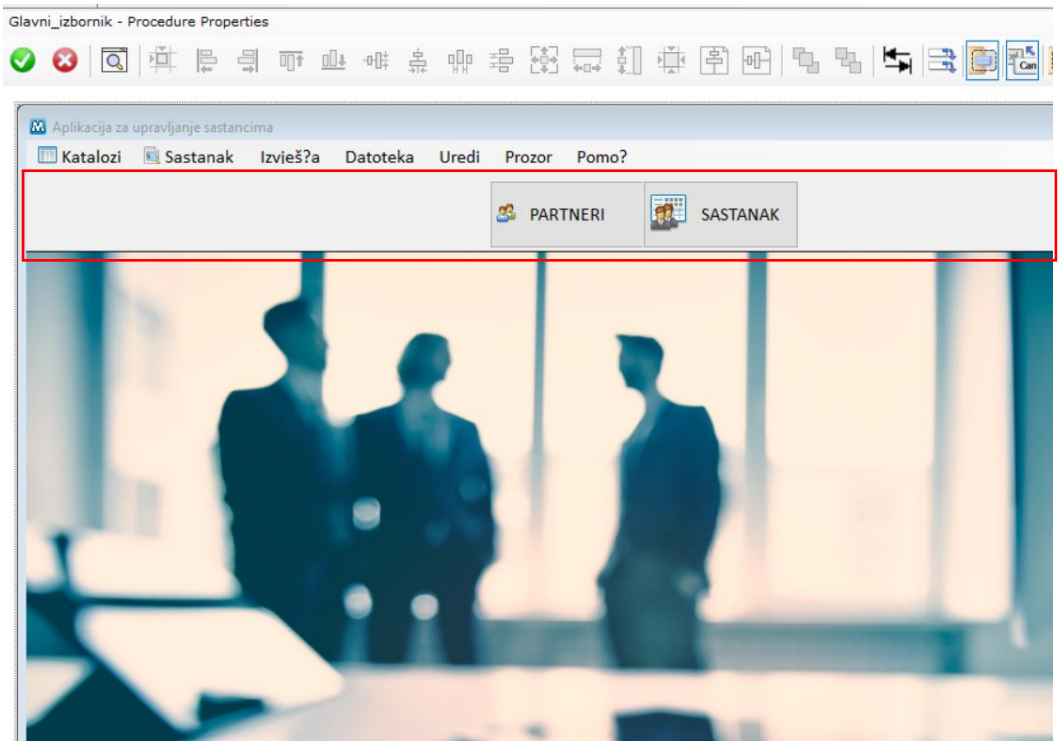


Slika 18 Definiran "Splash procedure"



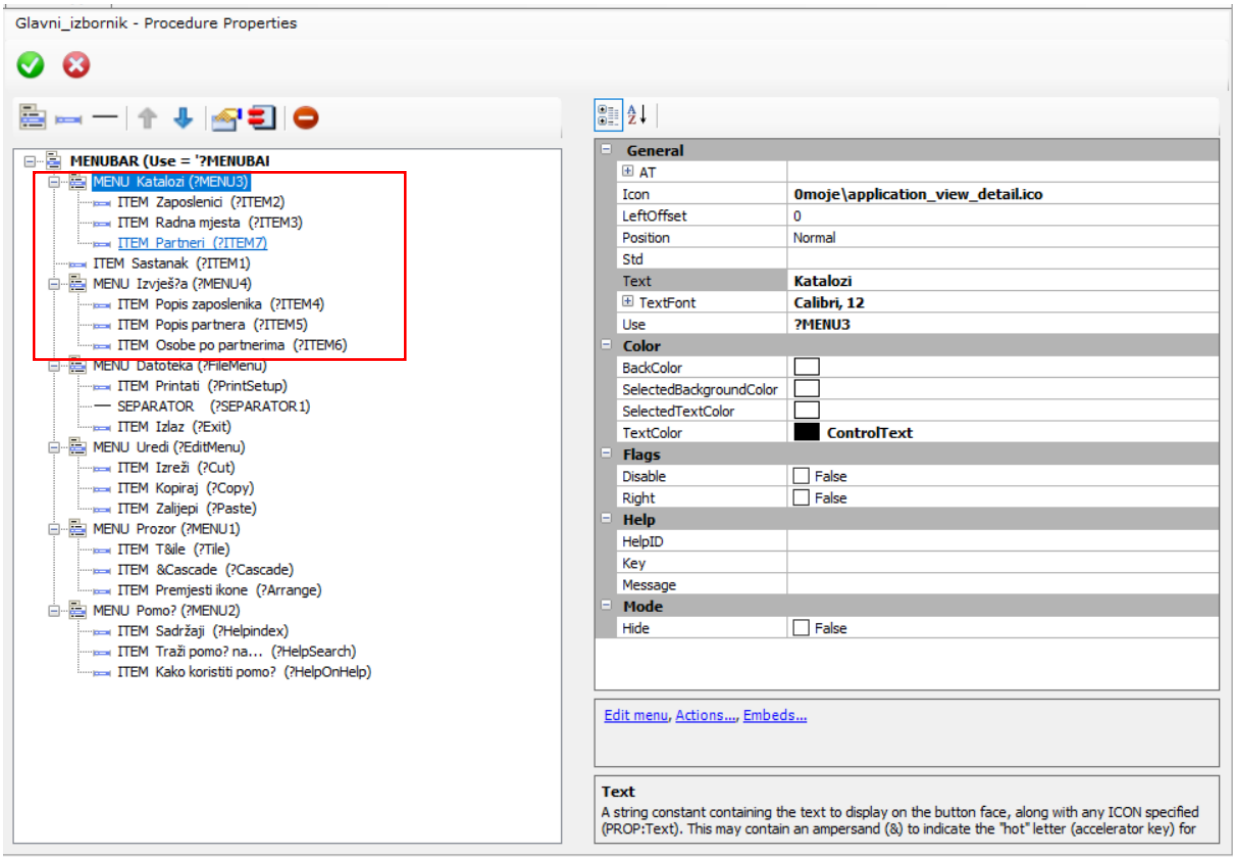
Slika 19 Podešeni prikazi datuma i vremena

Na prozoru je dodana alatna traka kao i gumbovi za otvaranje popisa, što je označeno na sljedećoj slici. Dodana je i pozadinska slika na glavnom prozoru.



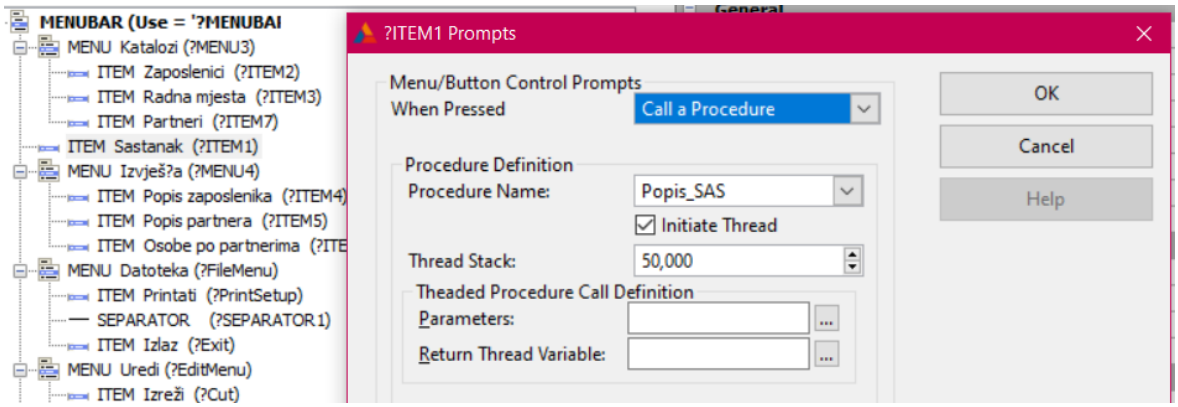
Slika 20 Izgled glavnog izbornika

U izborničkoj traci dodani su novi izbornici: „Katalozi“ i „Izvješća“. U katalozima su kao stavke stavljene poveznice za otvaranje popisa svih jakih entiteta iz modela: „Radna_mjesta“, „Zaposlenici“ i „Partneri“, dok je jaki entitet „Sastanak“ izvučen kao zasebna stavka. Dodani izbornici i stavke označeni su na slici 21. Kod defaultnih izbornika uključena je opcija right, kako bi oni bili na desnoj strani trake.



Slika 21 Menubar

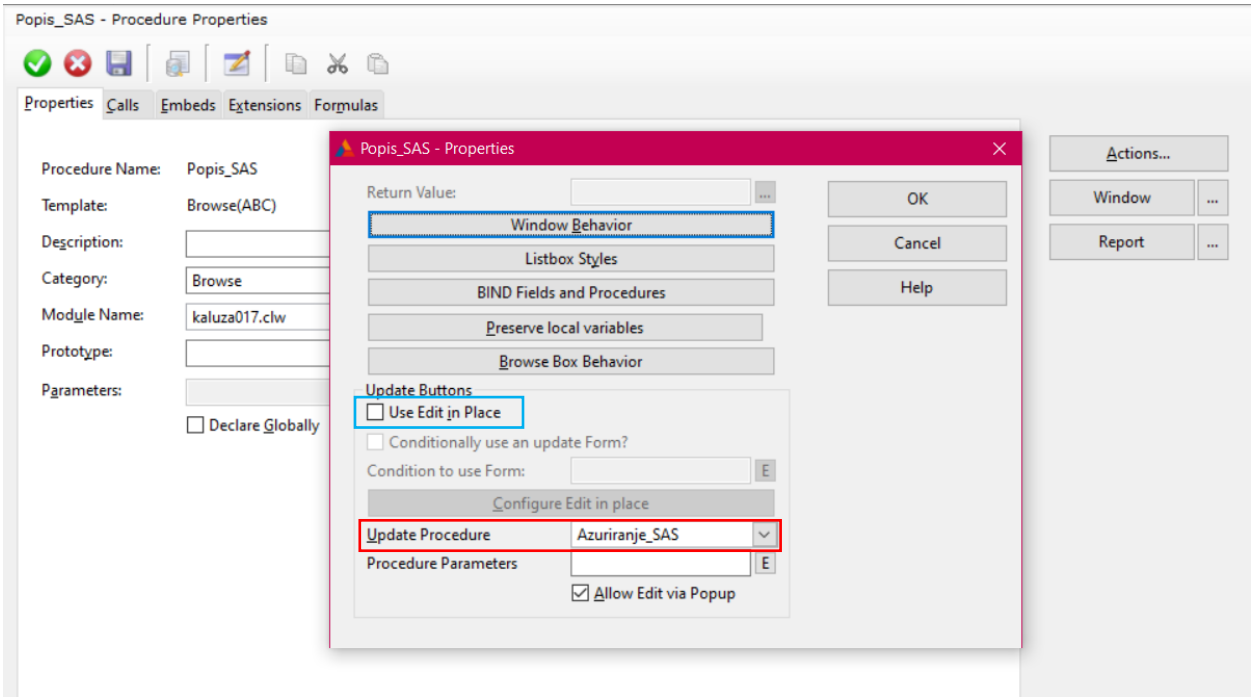
Na stavkama je pod akcije uključeno pozivanje procedure (slika 22), isto kao i na gumbovima na toolbaru, koji su ranije bili pokazani.



Slika 22 Pozivanje proceduri

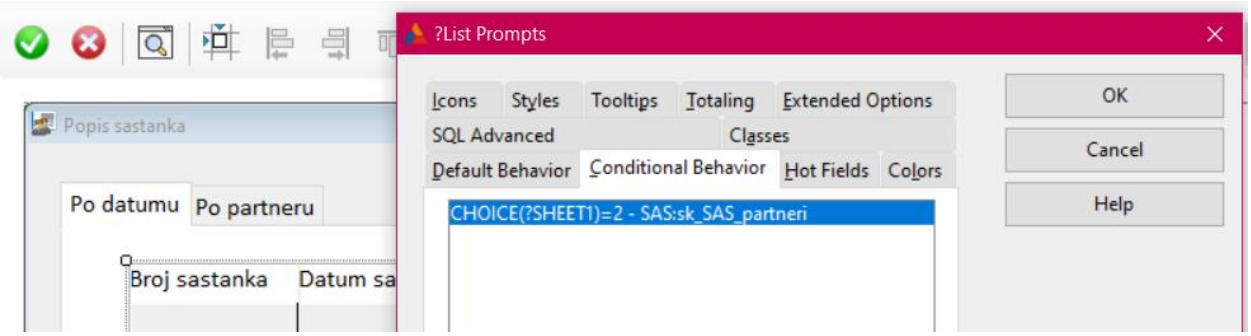
5.2.2 Procedura za popis

Popis procedure izrađene su pomoću „Browse with update and select“ templatea. Za sve popis procedure napravljeni su i procedure ažuriranja „update procedure“ označene crvenom bojom na slici 23. Za popis radnih mjesta proceduru uključena je opcija „use edit in place“ označena plavom bojom na slici 23. Kod procedura: popis zaposlenika, popis radnih mjesta i popis partnera uključena je opcija „enable sort header“ (Browse Box Behavior → Extended options) koja omogućuje sortiranje tablice po bilo kojem odabranom stupcu.

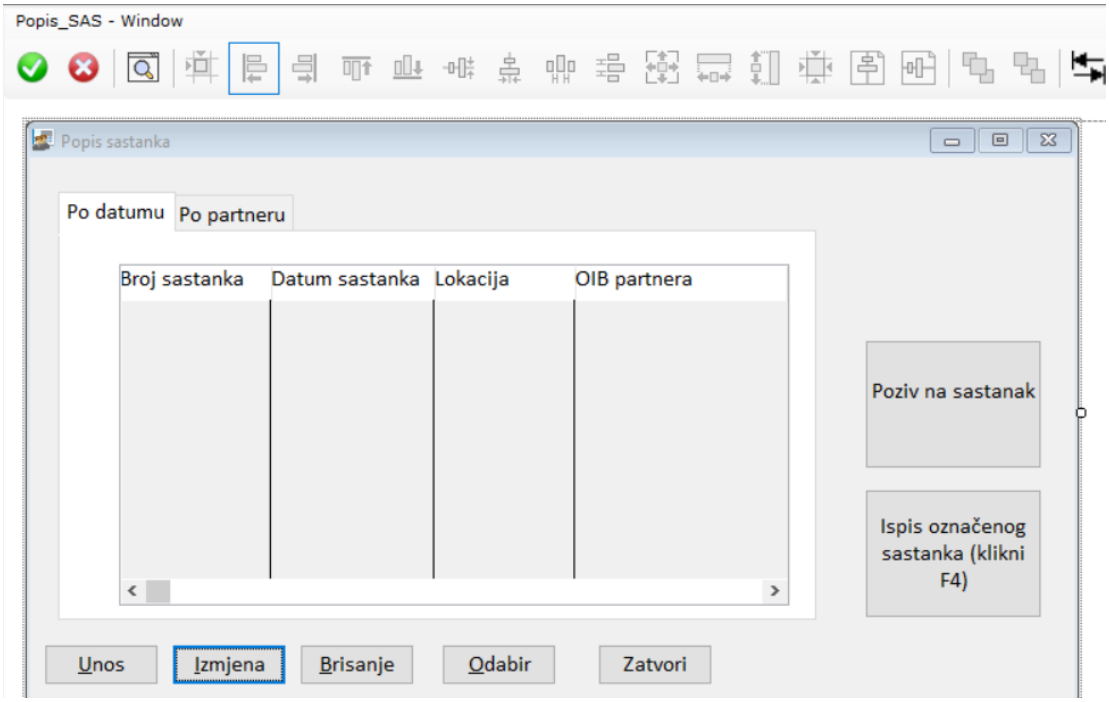


Slika 23 Postavke popis procedure

Kod popis sastanka sortiranje tablice omogućeno je pomoću sheeta i tabova, koji se odnose na stupce „datum“ i „OIB_partnera“. Ova opcije je pokazana na sljedećoj slici. Na prozoru popis sastanka dodani su i gumbovi za izvješće, gdje se za označeni sastanak može dobiti „Poziv na sastanak“ izvješće, te izvješće sastanka sa svim zaključcima i osobama koje su na sastanku sudjelovale. Prozor popis sastanka se vidi na slici 25.

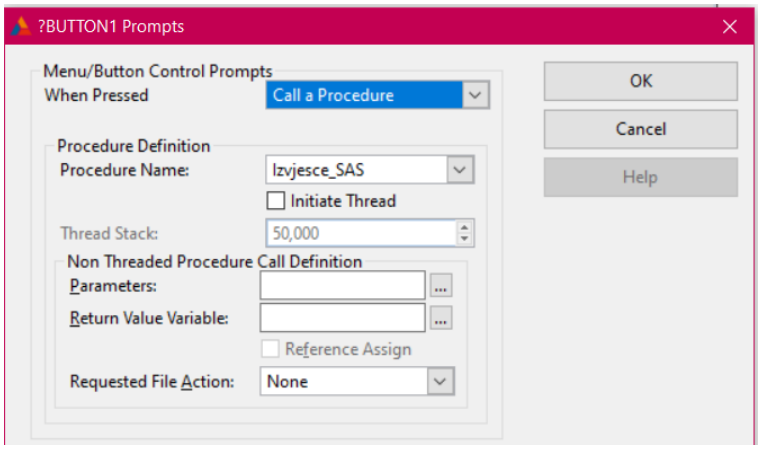


Slika 24 Soritranje pomoću sheet-a



Slika 25 Prozor popis sastanaka

Na sljedećoj slici je pokazana akcija za pozivanje izvješća.

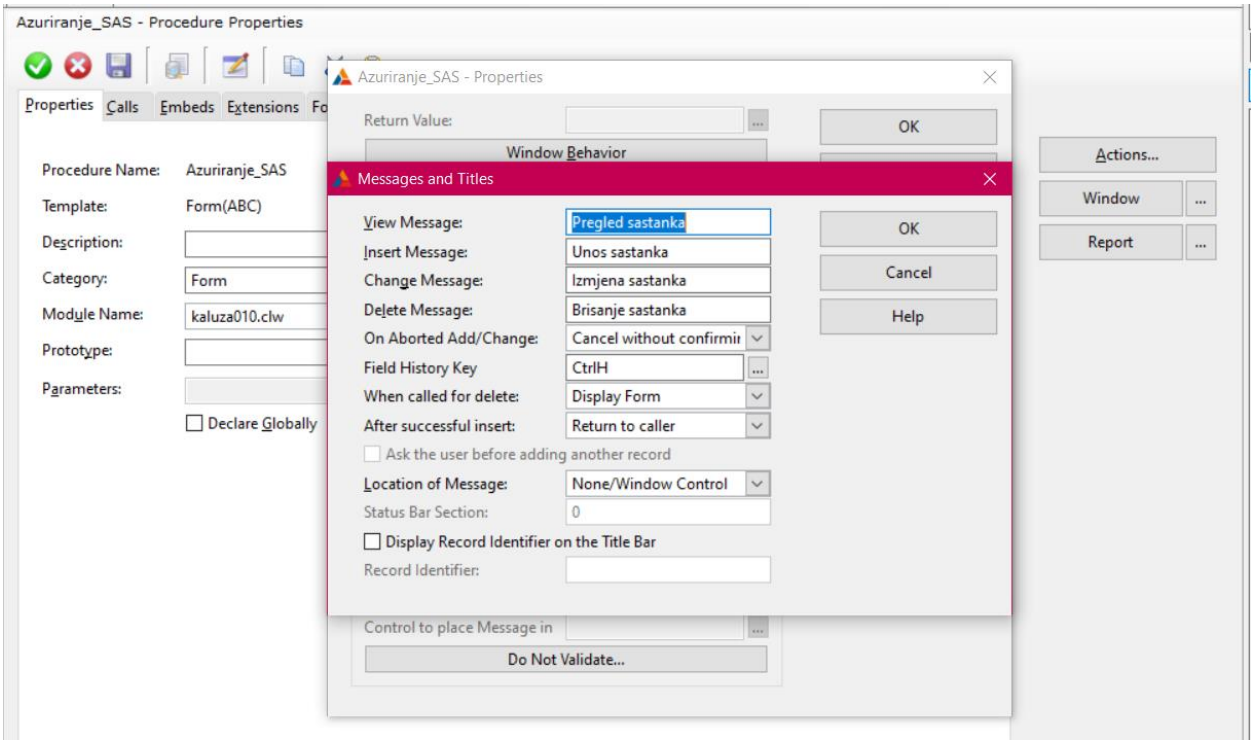


Slika 26 Gumb za pozivanje izvješća

Može se uočiti kako za otvaranje procedura iz glavnog izbornika je potrebno uključiti opciju „initiate thread“, dok se ista ne koristi za otvaranje procedura iz već postojećih prozora.

5.2.3 Procedura za ažuriranje

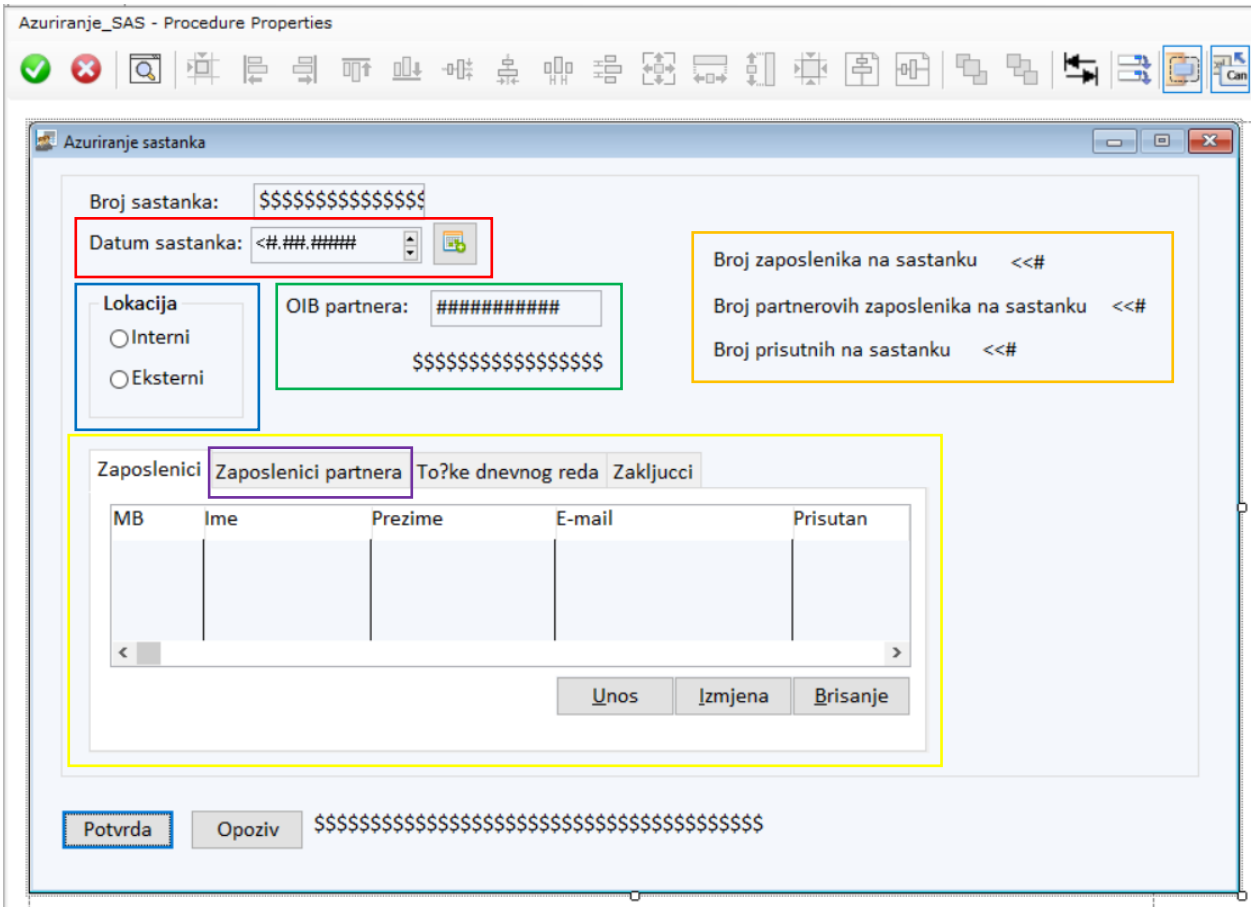
Procedure za ažuriranje napravljene su pomoću templatea „FORM (Add/Edit/Delete)“. Kod svake forme za ažuriranje je pod properties→messages and titles preveden izgled poruke o kojoj se akciji radi (unos, izmjena ili brisanje) što je vidljivo na sljedećoj slici



Slika 27 Poruke za ažuriranje

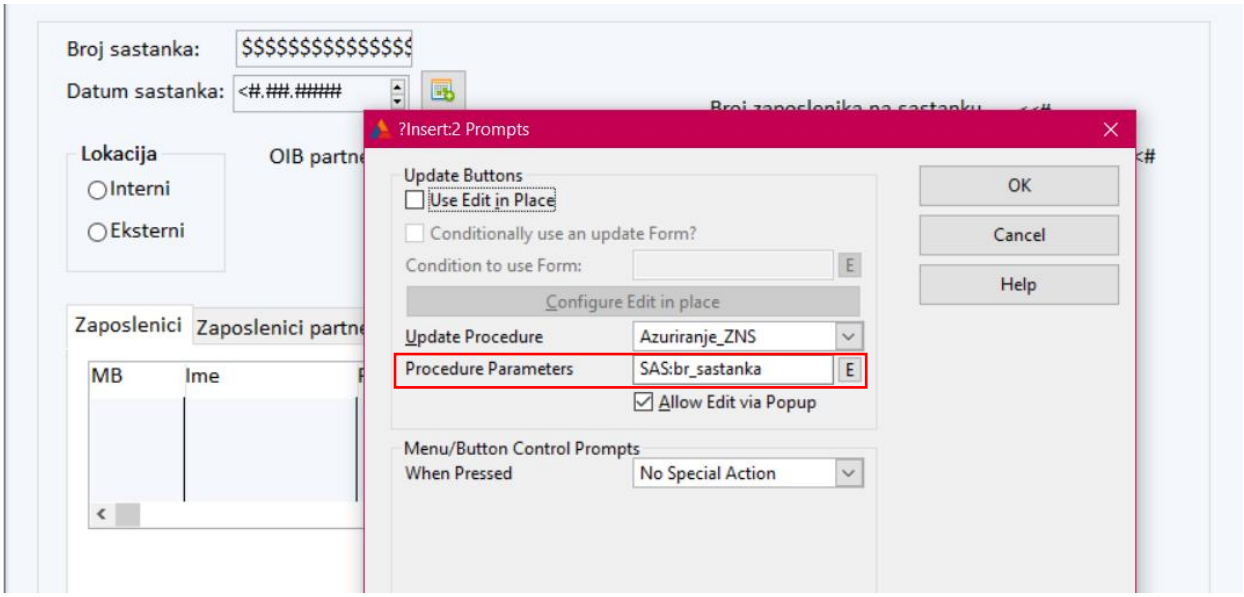
Kao primjer forme za ažuriranje je uzeto ažuriranje sastanaka pošto je to najkompleksnija forma. Za polje kalendar je dodan „Calendar button“ iz control template-a kako bi se lakše mogao odabrati datum održavanja sastanka (prikazano crvenom bojom). Ako se datum krivo odabere spin polje pomaže kako bi se datum mogao dodatno prepraviti. Lokacija se bira pomoću radio polja (prikazano plavom bojom) i ovisno o tome je li sastanak interni ili eksterni browse box za dodavanje partnerovih zaposlenika na sastanku se prikazuje (označeno ljubičastom bojom). Isto tako lokacija utječe na polje za biranje partnera s kojim se sastanak održava (označeno zelenom bojom). Radi boljeg pregled četiri browse boxa su postavljena u sheetove (označeno žutom bojom). Napravljena su lokalna polja za računanje zaposlenika koji su bili na sastanku (označeno narančastom bojom), ovisno o lokaciji sastanka prikazuje se ukupan broj partnerovih zaposlenika ili ne. Navedene funkcionalnosti su označene na slici 28.

Prilikom unosa novog sastanka moguće je unijeti samo podatke koji se odnose na tablicu sastanci (datum, lokacija, partner) i tek nakon spremanja tog zapisa se može izmjenom unijeti ostale podatke: točke dnevnog reda, zaključke, zaposlenike.

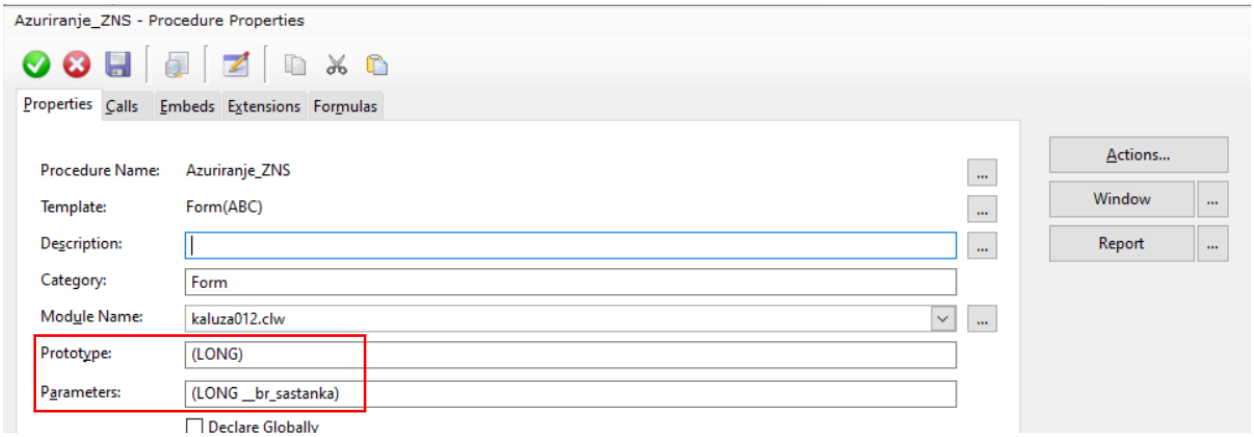


Slika 28 Forma za ažuriranje sastanka

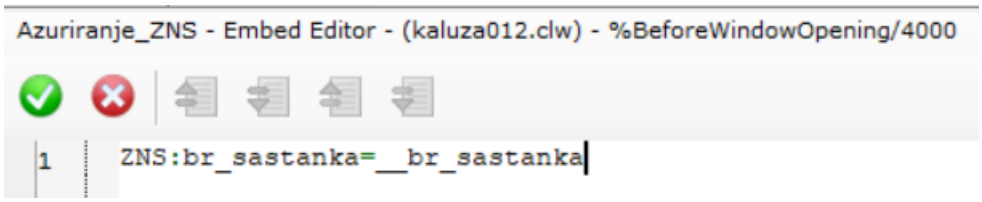
Klikom na gumb za ažuriranje nekog od dodanih browse box-ova, u tu proceduru se šalje broj sastanka (slika 29). Taj broj sastanka se nakon toga treba spremiti u neku varijablu (slika 30). Tip atributa prototipa mora biti jednak tipu atributa koji se šalje. I na kraju se poslani broj sastanka treba spremiti pod broj sastanka u slabom entitetu ili agregaciji (slika 31). Npr. ako se krenu unositi zaposlenici na sastanku šalje se broj sastanka koji ima prefiks tablice iz koje dolazi odnosno sastanak (SAS:br_sastanka). Taj podatak se treba spremiti u tablicu zaposlenici na sastanku (ZNS:br_sastanka). Ovim postupkom je to postignuto.



Slika 29 Slanje broj sastanka

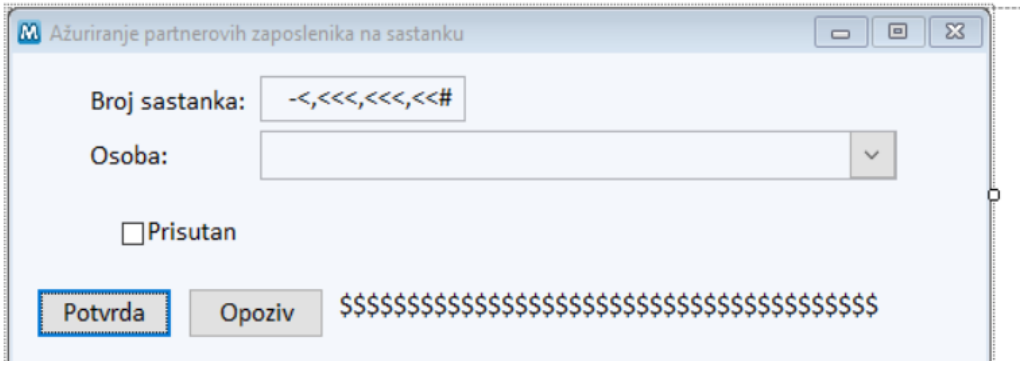


Slika 30 Pozivanje procedure sa poslanim brojem sastanka

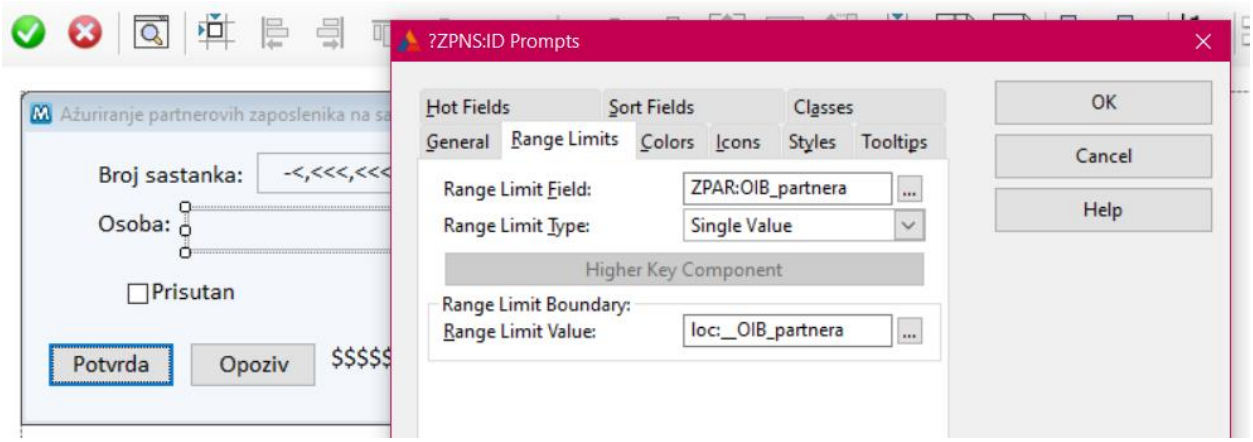


Slika 31 Postavljanje vrijednosti u slabom entitetu ili agregaciji

Na slici 32 je prikazana forma za ažuriranje partnerovih zaposlenika na sastanku. U tom ažuriranju odabir partnerovih zaposlenika koji su bili na sastanku je omogućeno pomoću list boxa (slika 33).

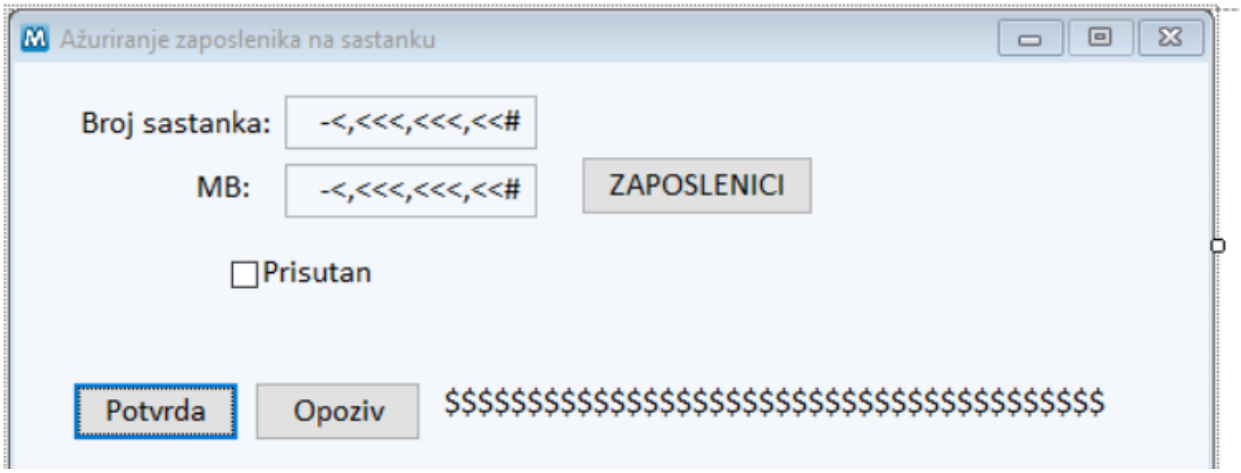


Slika 32 Ažuriranje partnerovih zaposlenika na sastanku

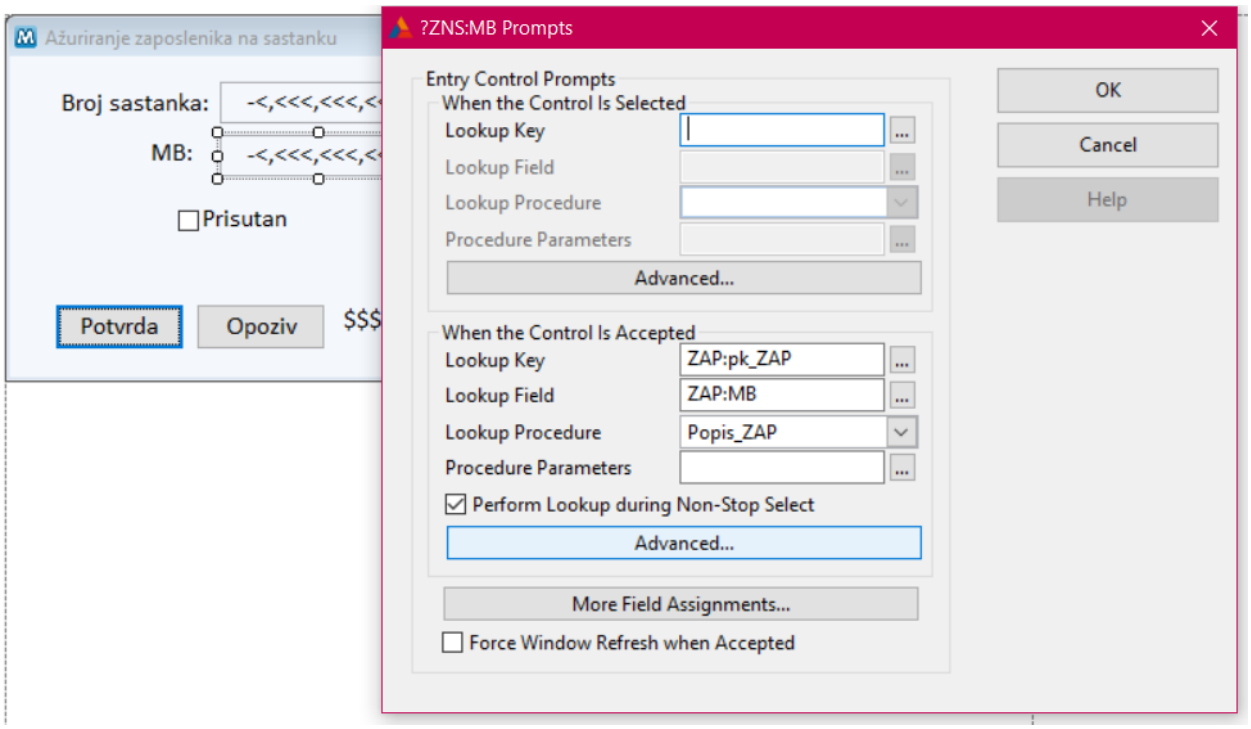


Slika 33 Akcija za biranje partnerovog zaposlenika

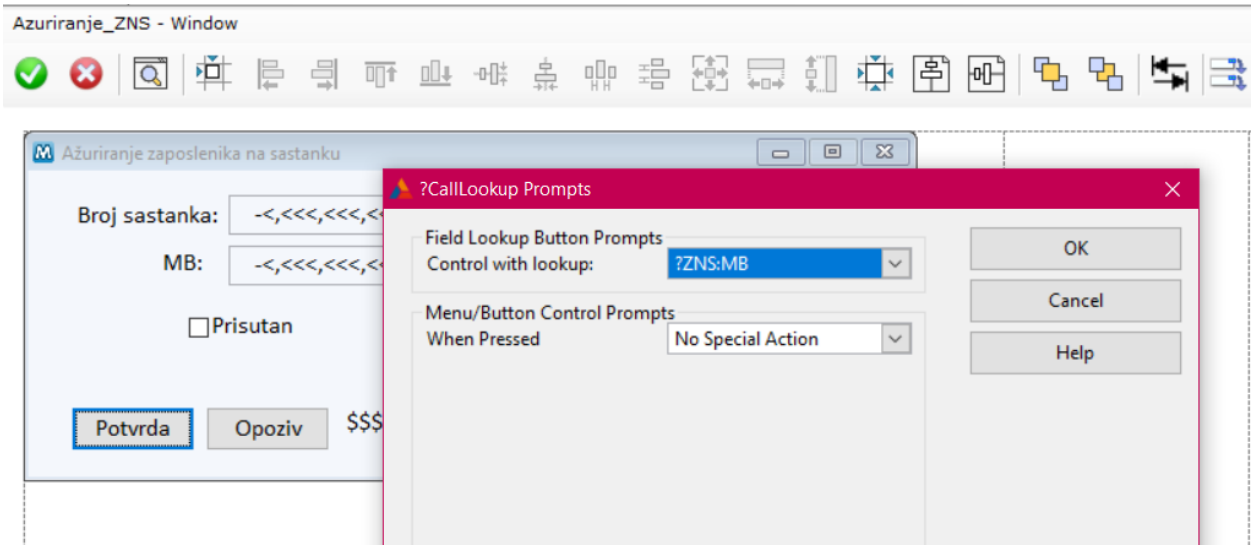
Kod procedure ažuriranje zaposlenika na sastanku odabir zaposlenika omogućen je pomoću lookup gumba (slika 34). Gumbom zaposlenici se otvara popis zaposlenika gdje je moguće odabrati već unesenog zaposlenika ili unijeti novoga. Na slikama 35 i 36 su pokazane potrebne akcije na entry polju MB i gumbu zaposlenici.



Slika 34 Ažuriranje zaposlenika na sastanku



Slika 35 Akcija na entry polju MB

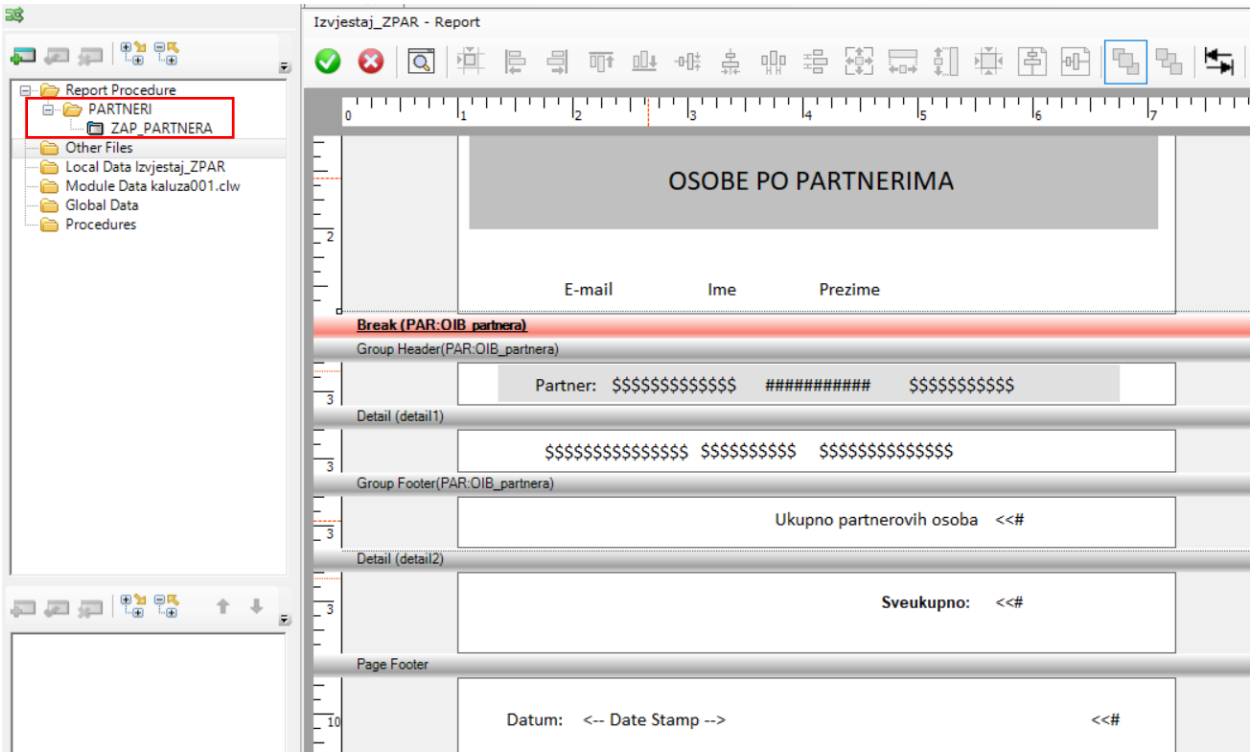


Slika 36 Akcija na gumbu zaposlenici

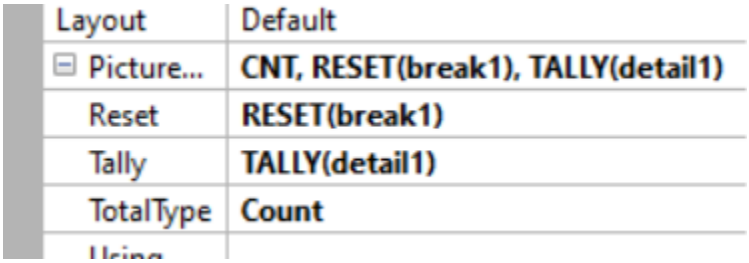
5.2.4 Procedura za izvješće

U aplikaciji su napravljena tri izvješća koja se odnose na sve podatke. Do njih se dolazi iz glavnog izbornika. Izvješće zaposlenika i izvješće partnera su jednostavna izvješća koja ispisuju podatke iz jedne tablice, dok je izvješće partnerovih zaposlenika složenije izvješće pošto koristi podatke iz dvije tablice.

Na sljedećoj slici je prikazan dizajn složenog izvješća. Crvenom bojom su označene tablice koje se koriste. Tablica partneri je nadređena tablici zaposlenici partnera. Napravljen je break po polju „OIB_partnera“ iz tablice partneri. U postavkama polja ukupno partnerovih osoba koje se nalazi u group footeru je uključeno brojanje zapisa kako bi se izračunalo koliko ima unesenih partnerovih zaposlenika (slika 38). Polje sveukupno broji sveukupan broj zapisa.

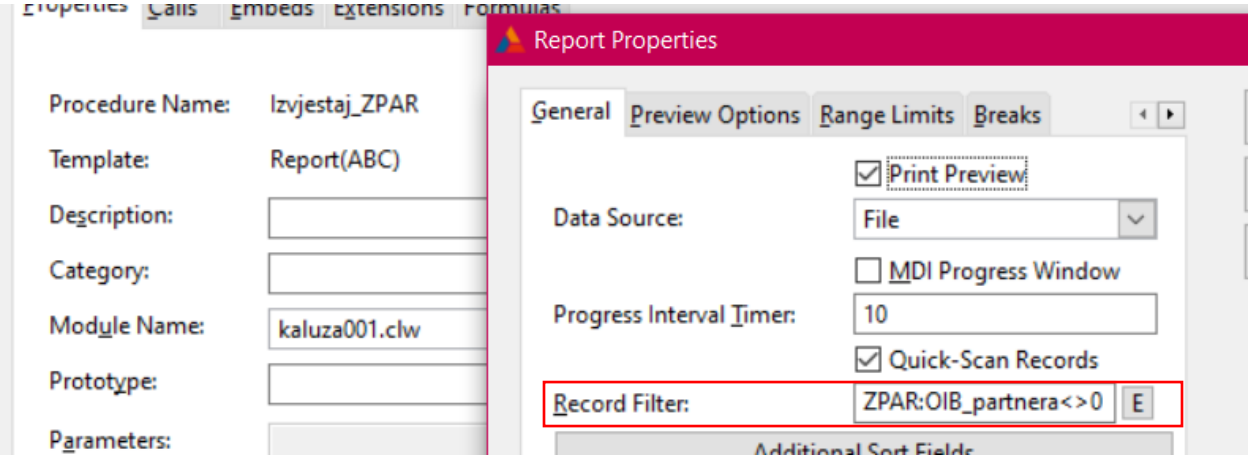


Slika 37 Izvješće partnerovih zaposlenika



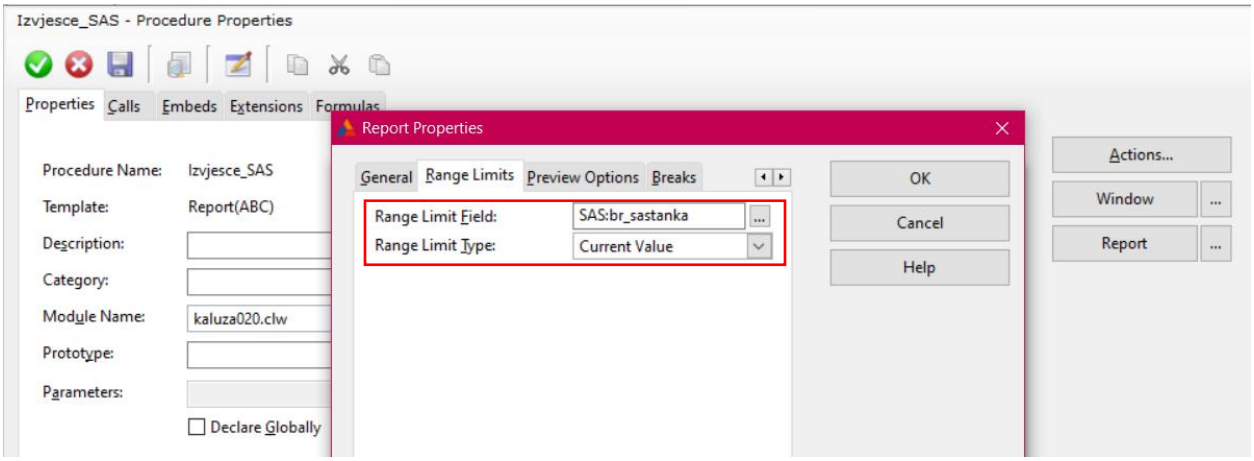
Slika 38 Broj zaposlenika po partneru

Kako se ne bi ispisao podatak o partneru za kojeg još nisu uneseni njegovi zaposlenici u postavkama izvješća (properties → actions →report properties) napisan je uvjet za mogućnost ispisa određenog partnera. Prikazano na sljedećoj slici.



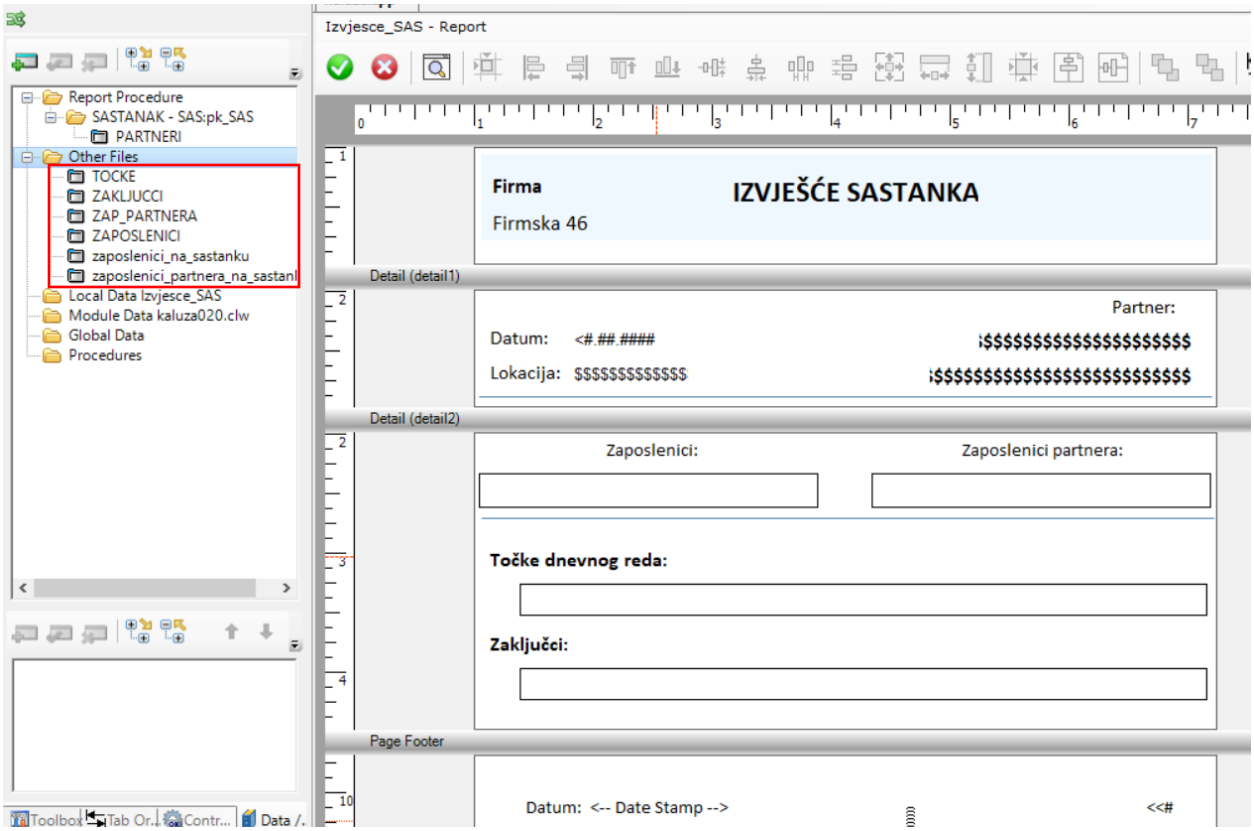
Slika 39 Filtriranje ispisa u izvješću zaposlenika partnera

Druga dva izvješća se pokreću iz popis procedure. U oba izvješća je pod properties →actions → report properties → range limits uključeno ograničenje ispisa za odabrani sastanak. Prikazano na slici 40.

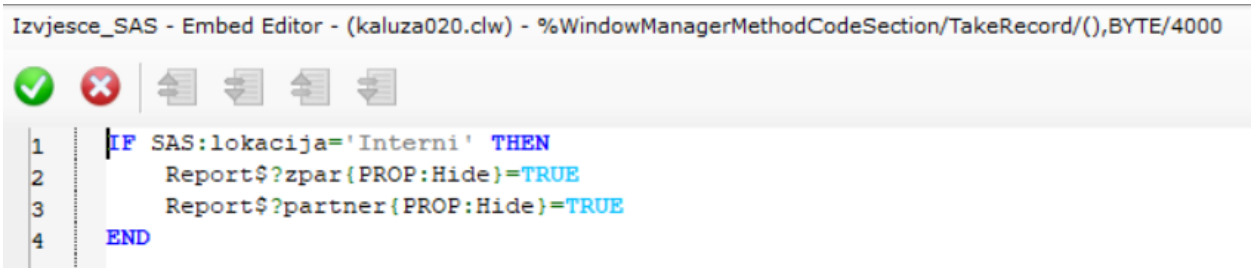


Slika 40 Ispis za samo jedan određeni sastanak

Na sljedećoj slici je prikazan dizajn izvješća sastanka. Tablice označene crvenom bojom su stavljene pod „other files“ jer se njihovi podaci ne prikazuju direktno na izvješću, nego pomoću koda kasnije pokazanog. Stringovi „Partner:“ i „Zaposlenici partnera“ se ne prikazuju ako se ne radi o internom sastanku (slika 42).



Slika 41 Izvješće sastanka



Slika 42 Uklanjanje stringova ako se radi o internom sastanku

Polja zaposlenici, zaposlenici partnera, točke dnevnog reda i zaključci dobivaju vrijednosti pomoću koda na sljedećoj slici. Prikazani kod se odnosi na polje partnerovih zaposlenika, dok je kod za ostala tri polja na isti način zapisan.

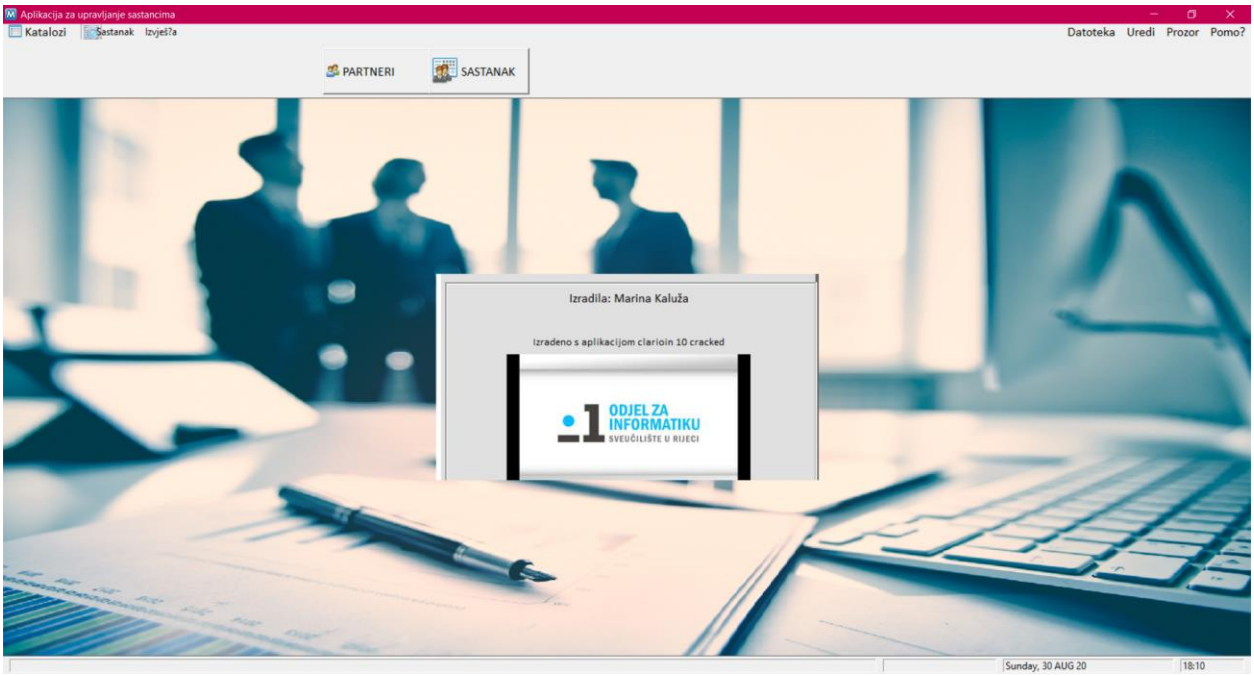
Izvjesce_SAS - Embed Editor - (kaluza020.clw) - %WindowManagerMethodCodeSection/TakeRecord/(),BYTE/4000

```
1   CLEAR(ZPNS:Record)
2   ZPNS:br_sastanka = SAS:br_sastanka
3   SET(ZPNS:pk_ZPNS, ZPNS:pk_ZPNS)
4
5   LOOP
6       NEXT(zaposlenici_partnera_na_sastanku)
7       IF ERRORCODE() THEN BREAK END
8       IF ZPNS:br_sastanka <> SAS:br_sastanka  THEN BREAK END
9
10      CLEAR(ZPAR:Record)
11      ZPAR:ID = ZPNS:ID
12      Access:ZAP_PARTNERA.TryFetch(ZPAR:pk_ZPAR)
13
14      ZPNS = ZPNS & ZPAR:ime & ' ' & ZPAR:prezime & CHR(13) & CHR(10)
15  END
```

Slika 43 Vrijednost polja partnerovih zaposlenika

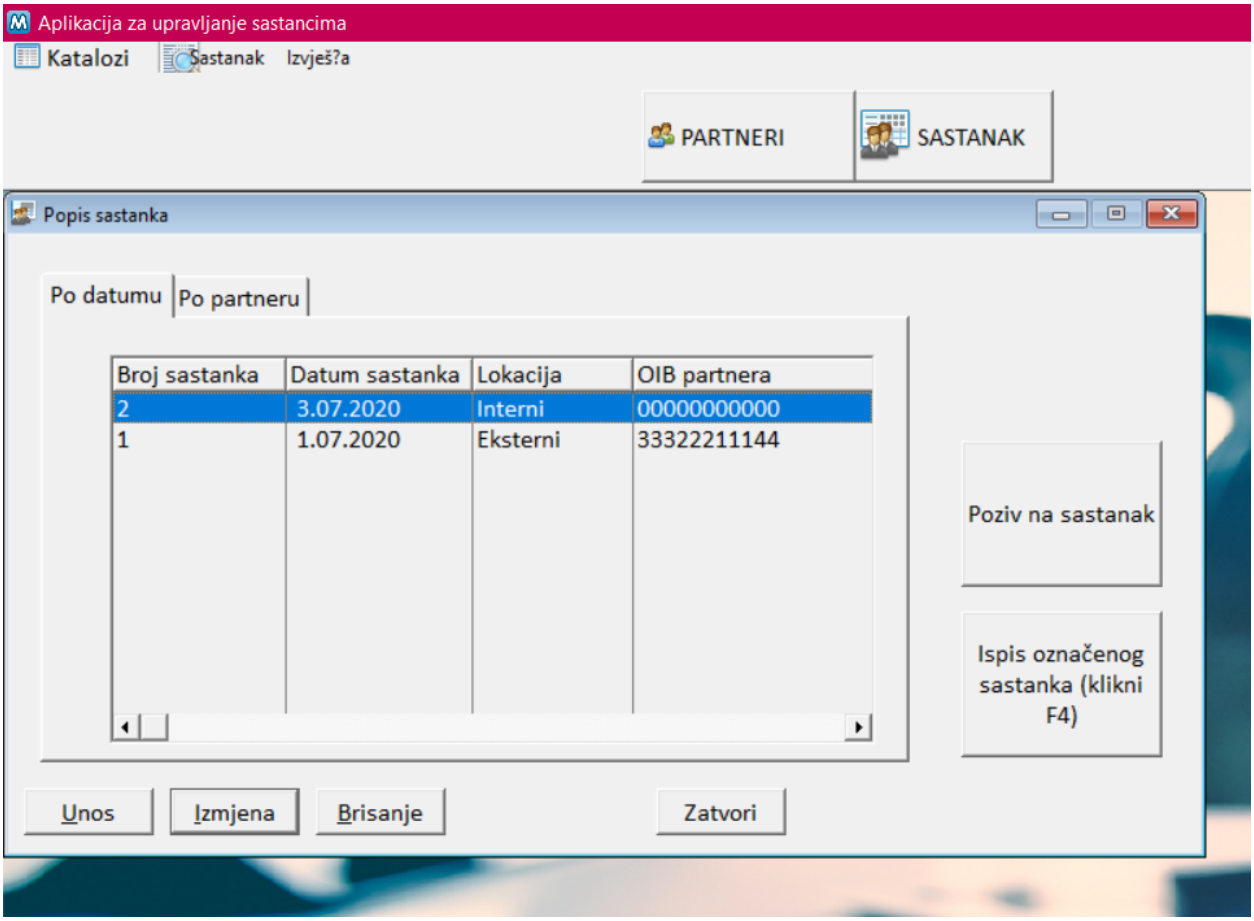
6 KORIŠTENJE APLIKACIJE

U ovom poglavlju će biti prikazan način za korištenje aplikacije. Na slici 44 vidi se izgled aplikacije odmah nakon pokretanja. Na sredini se nalazi skočni prozor.



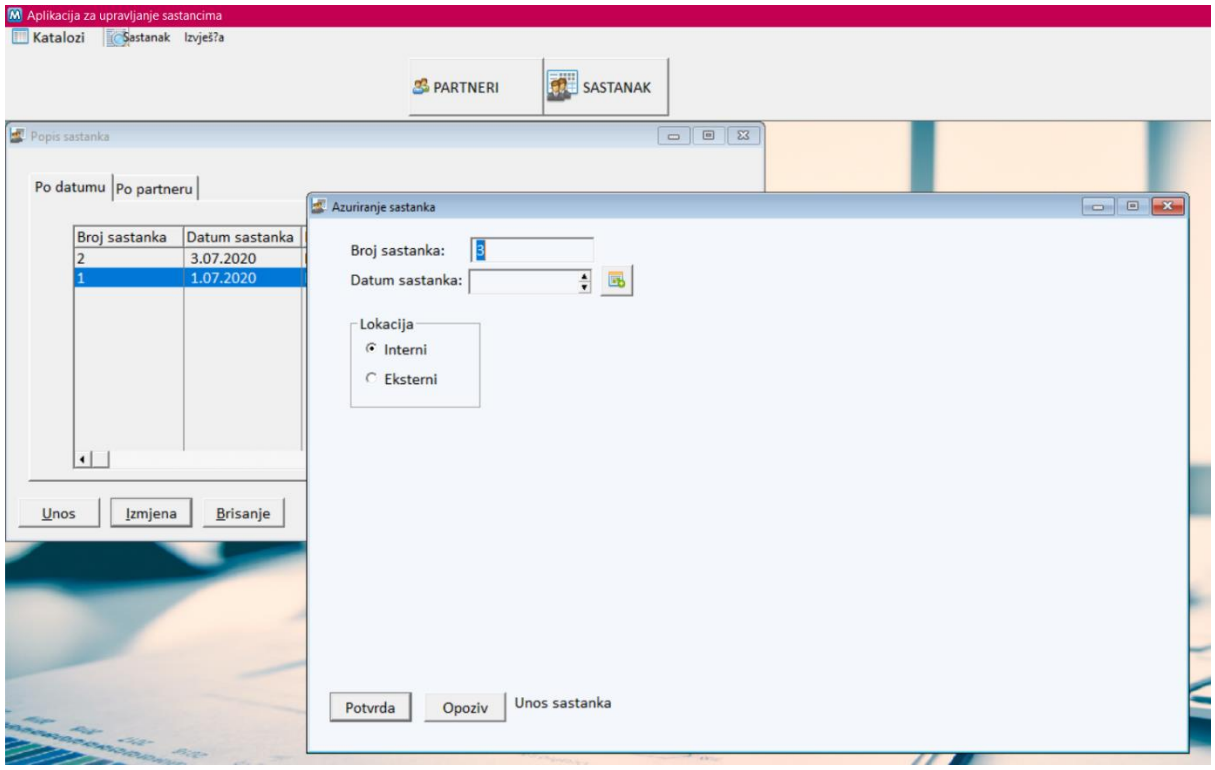
Slika 44 Početni prozor

Na sljedećoj slici je prikaz otvoren browse prozor sastanci. Nju se može uključiti pomoću druge kartice na alatnoj traci ili pomoću gumba „SASTANAK“.

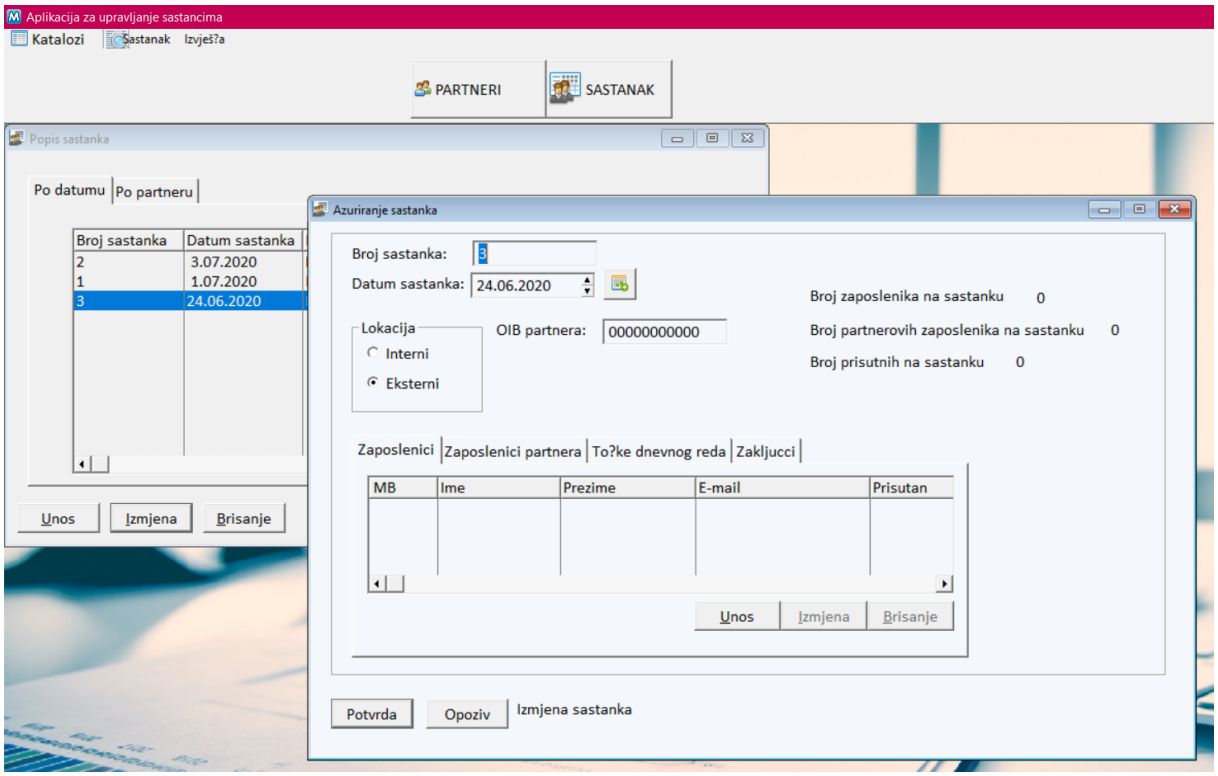


Slika 45 Browse prozor - Popis sastanka

Slike 45 i 46 pokazuju prozor ažuriranja sastanka. Prvo se unose podaci u polja koji su dio tablice „sastanci“, a to su: „br_sastanka“, „datum“ i „lokacija“. Nakon spremanja novog sastanka u browse prozoru se treba kliknuti na gumb „Izmjena“ i onda će se moći unijeti podaci o partneru ako je sastanak eksterni i podaci o zaposlenicima, partnerovim zaposlenicima, točkama dnevnog reda te zaključcima sastanka. Slika 45 se odnosi na unos, a slika 46 na izmjenu.

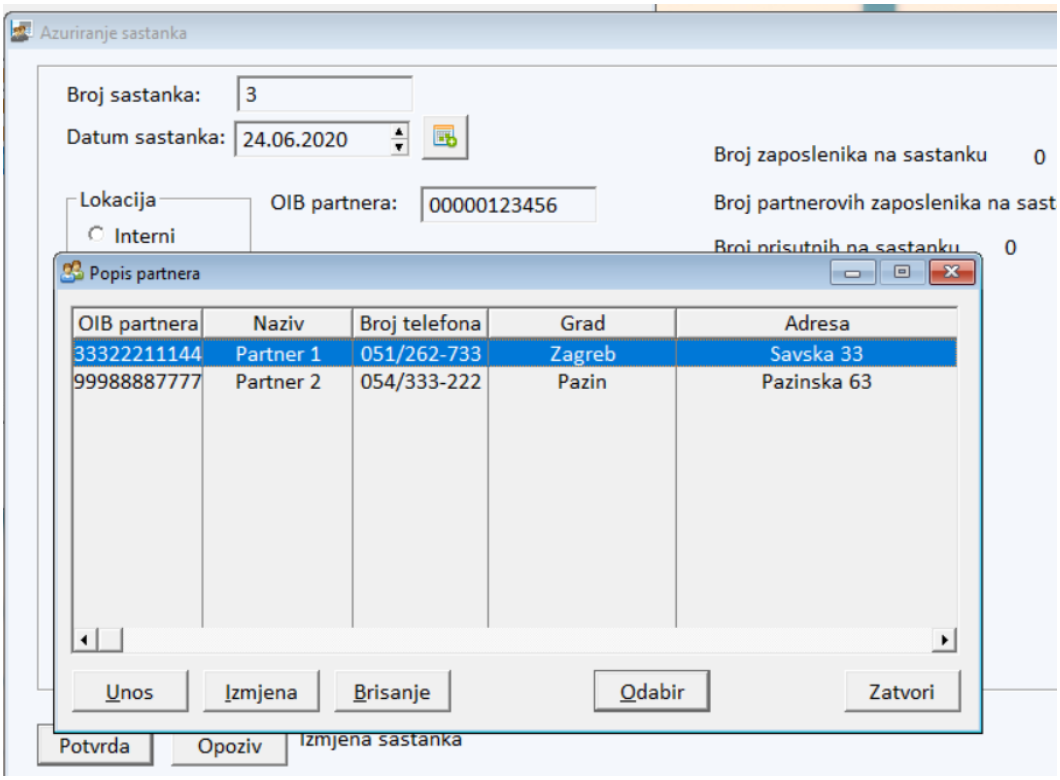


Slika 46 Ažuriranje prozor - unos sastanka

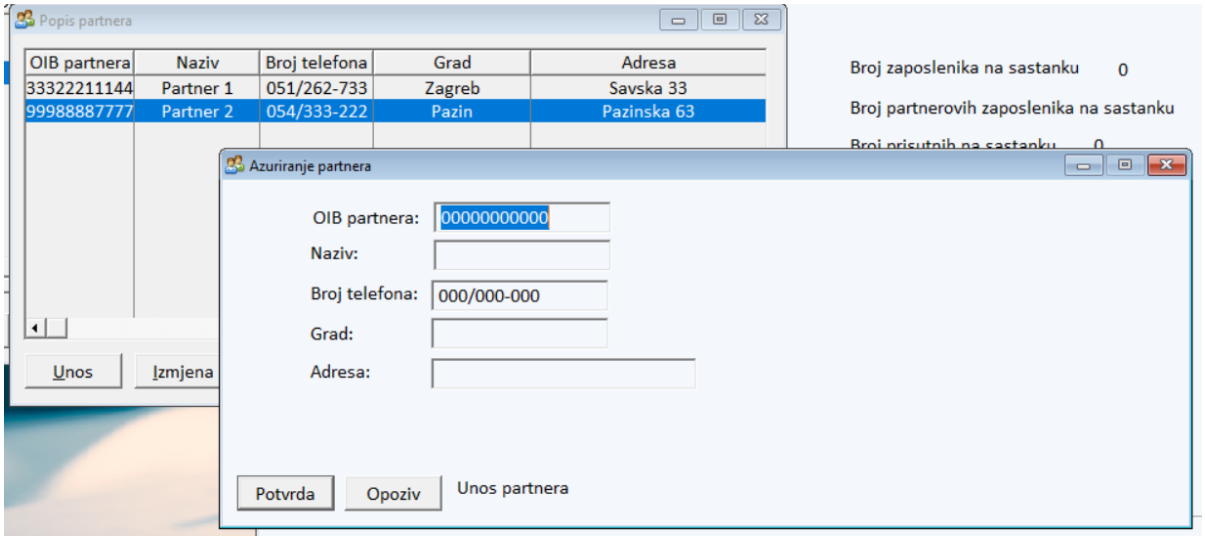


Slika 47 Ažuriranje prozor - izmjena sastanka

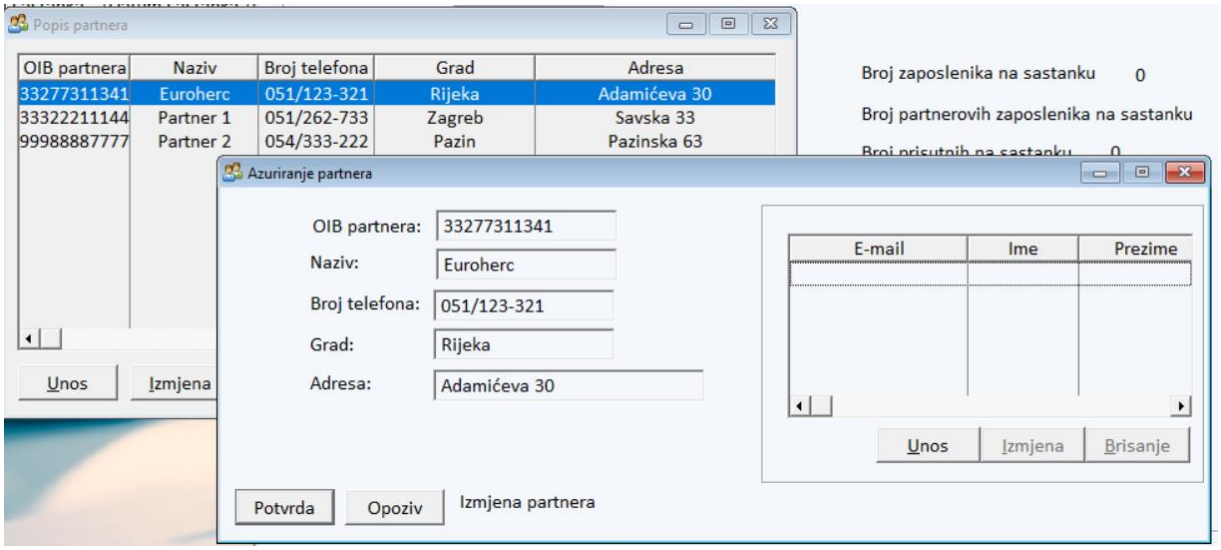
Na sljedećoj slici prikazan je browse prozor partneri koja se otvara za odabir partnera koji sudjeluju na sastanku. Taj prozor se otvara unosom OIB-a i klikom na tab. Moguće je odabrati nekog od već unesenih partnera ili ako se traženi partner ne nalazi na popisu onda se može klikom na „unos“ označeno crvenom unijeti novog partnera. Prozor za unos partnera prikazan je slici 49. Prilikom unosa partnera isto kao i kod unosa sastanka prvo je omogućen unos partnera, a kad se uključi izmjena onda je moguće unijeti i partnerove zaposlenike što je prikazano na slici 50.



Slika 48 Ažuriranje sastanka - odabir partnera

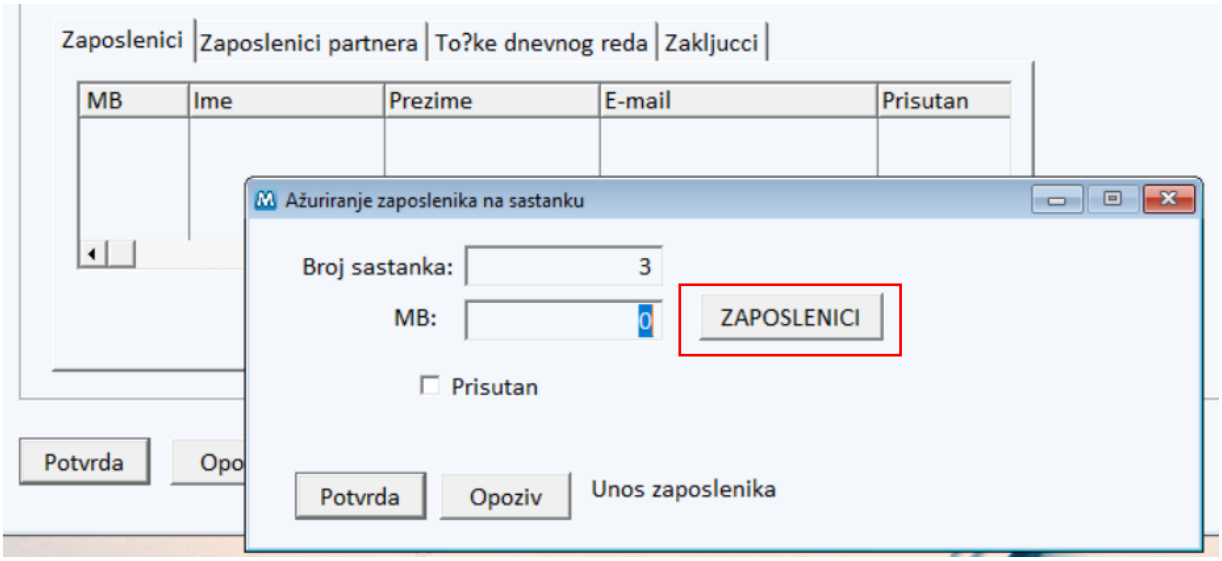


Slika 49 Ažuriranje sastanka - unos partnera



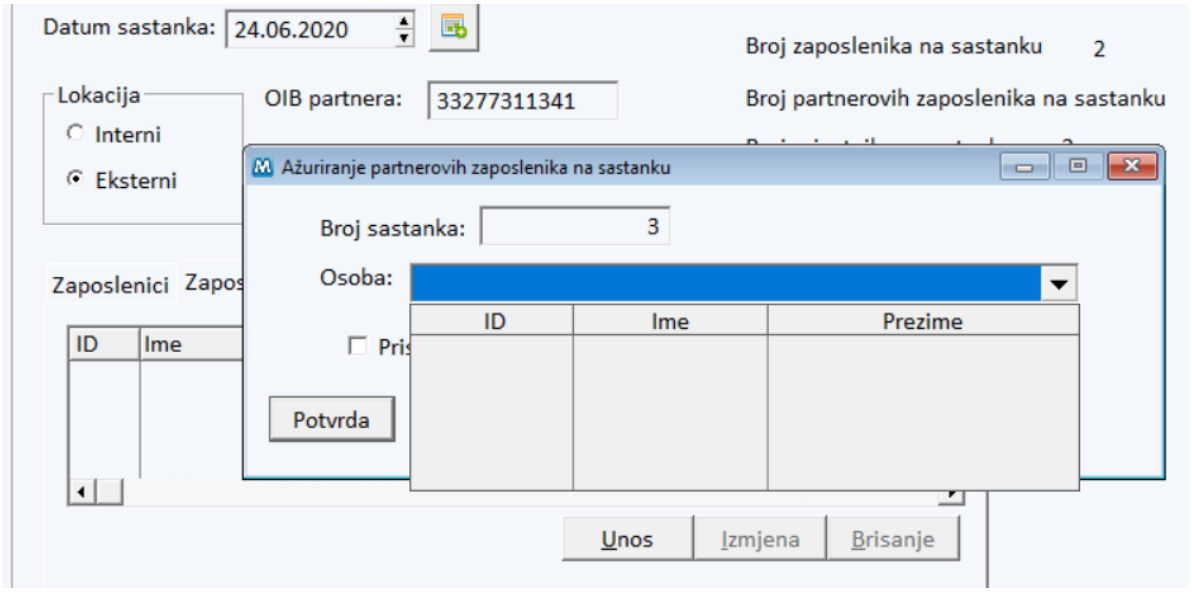
Slika 50 Ažuriranje partnera - izmjena partnera

Nakon što se unijelo koji partner sudjeluje na sastanku potrebno je unijeti koji su zaposlenici i partnerovi zaposlenici bili na sastanku. Klikom na „unos“ u tabu zaposlenici otvara se prozor za ažuriranje „zaposlenika_na_sastanku“, pokazan na slici 51. Klikom na označeni gumb se otvara popis svih zaposlenika u poduzeću gdje se po potrebi može unijeti i novi zaposlenik.

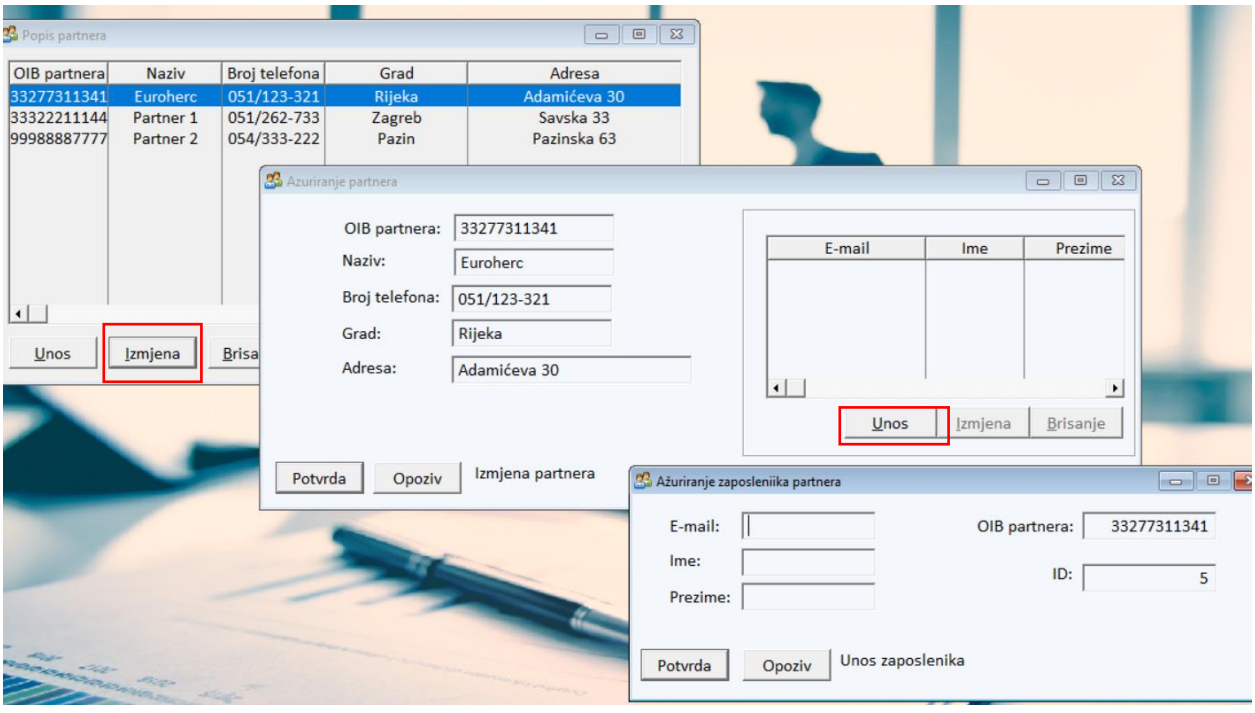


Slika 51 Ažuriranje sastanka - ažuriranje zaposlenika

Ažuriranje partnerovih zaposlenika je prikazano na sljedećoj slici. Pošto tablica „Zaposlenici partnera“ ima za primarni ključ polje „ID“, a za unique polja „OIB_partnera“ i „email“ jedini način na koji se moglo realizirati biranje partnerovog zaposlenika koji je bio na sastanku je pomoću drop liste. U slučaju da partnerovi zaposlenici nisi unijeti prije nego ih se htjelo dodati na sastanak niti jedan zaposlenik u listi neće biti ponuđen. Tada je potrebno upaliti prozor popis partnera pomoću gumba partneri ili na kartici Katalozi → Partneri i klikom na gumb za izmjenu za željenog partnera dodati zaposlenike (slika 53).



Slika 52 Ažuriranje sastanka - ažuriranje partnerovih zaposlenika



Slika 53 Ažuriranje partnera - unos partnerovih zaposlenika

Klikom na tabove „točke dnevnog reda“ i „zaključci“ otvaraju se obrasci za unos, izmjenu ili brisanje točke odnosno zaključka, ovisno o odabranoj akciji.

7 ZAKLJUČAK

U radu se prošlo kroz proces razvoja aplikacije. Prvo su se specificirali zahtjevi pomoću korisničkih priča. Korisnik je omogućio zapisnik sa sastanka pomoću kojeg su se izdvojili podaci (atributi) koje je potrebno spremiti u bazu. Sljedeći korak je bio izrada modela entiteti – veze. Ranije izdvojene atribute bilo je potrebno rasporediti po entitetima i ako je potrebno za neke entitete dodati još atributa sukladno s zahtjevima navedenim u korisničkim pričama. Nakon toga se izradio relacijskim model poštovanjem pravila pretvorbe. Zatim se izradila SQL skripta kojom je izrađen fizički model podataka odnosno baza podataka koja je implementirana na odabrani sustav za upravljanje bazama podataka.

Potom se krenulo u izradu aplikacije. Odabran je Clarion RAD alat. Prvi korak je bio izrada rječnika. U rječniku su dodana dodatna svojstva nad atributima koja nije moguće definirati na bazi, a potrebna su u izradi aplikacije. Drugi korak je izrada aplikacije. Izrađeni su potrebni prozori za pregled i ažuriranje podataka te su izrađeni izvještaji. Na kraju se napravio korisnički priručnik za pomoć pri korištenju aplikacije.

Clarion kao RAD alat za izradu softvera je vrlo efikasan. Budući da je Clarion jezik 4.generacije on uključuje automatizirane procese za generiranje programskog koda u Clarion programskom jeziku.

Izradom aplikacija u Clarionu zaključila sam da je Clarion jako dobar alat za brzu izradu aplikacija koji ima puno mogućnosti i gotovo sve, što je za neki sustav potrebno, se može na relativno jednostavan način ostvariti.

LITERATURA

[1] pokušaj pristupa 30.8.2020. https://en.wikipedia.org/wiki/User_story

[2] pokušaj pristupa 30.8.2020. https://www.agilealliance.org/agile101/

[3] Pavlić Mile: „Oblikovanje baza podataka“, Odjel za informatiku Sveučilišta u Rijeci, Rijeka, veljača 2011.

[4] pokušaj pristupa 30.8.2020. https://en.wikipedia.org/wiki/Data_model

[5] Pavlić Mile: „Informacijski sustavi“, školska knjiga, Zagreb, 2011.

[6] Materijali s kolegija „uvod u baze podataka“

[7] pokušaj pristupa 30.8.2020. https://en.wikipedia.org/wiki/Relational_database

[8] pokušaj pristupa 30.8.2020. http://barc-research.com/relational-databases/

[9] pokušaj pristupa 30.8.2020. https://blog.capterra.com/what-is-rapid-application-development/

[10] pokušaj pristupa 30.8.2020. https://www.icetips.com/clarion.php

[11] pokušaj pristupa 30.8.2020. https://www.mountaingoatsoftware.com/agile/user-stories

POPIS TABLICA

TABLICA 1 KORISNIČKE PRIČE 2

POPIS SLIKA

SLIKA 1 IZVJEŠTAJ ZAPISNIKA SASTANKA 3
SLIKA 2 MODEL ENTITETI-VEZE 5
SLIKA 3 LISTA SVIH TABLICA U RJEČNIKU I PODACI O TABLICI PARTNERI 9
SLIKA 4 RESTRIKCIJA NA ATRIBUTU TELEFON 10
SLIKA 5 POLJE TIPA BYTE 10
SLIKA 6 UNOS POMOĆU CHECKBOXA 11
SLIKA 7 UNOS POMOĆU LISTE OPCIJA 11
SLIKA 8 KLJUČ ZA SORTIRANJE U TABLICI SASTANAKA 12
SLIKA 9 VANJSKI KLJUČ ZA RADNO MJESTO U TABLICI ZAPOSLENIKA 13
SLIKA 10 PRIMARNI KLJUČ U AGREGACIJI 13
SLIKA 11 REZERVNI KLJUČ U AGREGACIJI 14
SLIKA 12AUTONUMBER U PRIMARNOM KLJUČU 15
SLIKA 13 VEZA IZMEĐU TABLICA „ZAPOSLENICI“ I „RADNO MJESTO“ 16

SLIKA 14 VEZA U AGREGACIJI „ZAPOSLENICI NA SASTANKU“, POMOĆU REZERVNOG KLJUIČA 16
SLIKA 15 VEZA U AGREGACIJI POMOĆU PRIMARNOG KLJUČA 17
SLIKA 16 VEZA U SLABOM ENTITETU U TABLICI „TOČKE DNEVNOG REDA“ 17
SLIKA 17 TEMPLATI U IZRADI APLIKACIJE 18
SLIKA 18 DEFINIRAN "SPLASH PROCEDURE" 19
SLIKA 19 PODEŠENI PRIKAZI DATUMA I VREMENA 19
SLIKA 20 IZGLED GLAVNOG IZBORNIKA 20
SLIKA 21 MENUBAR 21
SLIKA 22 POZIVANJE PROCEDURI 21
SLIKA 23 POSTAVKE POPIS PROCEDURE 22
SLIKA 24 SORITRANJE POMOĆU SHEET-A 22
SLIKA 25 PROZOR POPIS SASTANAKA 23
SLIKA 26 GUMB ZA POZIVANJE IZVJEŠĆA 23
SLIKA 27 PORUKE ZA AŽURIRANJE 24
SLIKA 28 FORMA ZA AŽURIRANJE SASTANKA 25
SLIKA 29 SLANJE BROJ SASTANKA 26
SLIKA 30 POZIVANJE PROCEDURE SA POSLANIM BROJEM SASTANKA 26
SLIKA 31 POSTAVLJANJE VRIJEDNOSTI U SLABOM ENTITETU ILI AGREGACIJI 26
SLIKA 32 AŽURIRANJE PARTNEROVIH ZAPOSLENIKA NA SASTANKU 27
SLIKA 33 AKCIJA ZA BIRANJE PARTNEROVOG ZAPOSLENIKA 27
SLIKA 34 AŽURIRANJE ZAPOSLENIKA NA SASTANKU 27
SLIKA 35 AKCIJA NA ENTRY POLJU MB 28
SLIKA 36 AKCIJA NA GUMBU ZAPOSLENICI 28
SLIKA 37 IZVJEŠĆE PARTNEROVIH ZAPOSLENIKA 29
SLIKA 38 BROJ ZAPOSLENIKA PO PARTNERU 29
SLIKA 39 FILTRIRANJE ISPISA U IZVJEŠĆU ZAPOSLENIKA PARTNERA 30
SLIKA 40 ISPIS ZA SAMO JEDAN ODREĐENI SASTANAK 30
SLIKA 41 IZVJEŠĆE SASTANKA 31
SLIKA 42 UKLANJANJE STRINGOVA AKO SE RADI O INTERNOM SASTANKU 31
SLIKA 43 VRIJEDNOST POLJA PARTNEROVIH ZAPOSLENIKA 32
SLIKA 44 POČETNI PROZOR 32
SLIKA 45 BROWSE PROZOR - POPIS SASTANKA 33
SLIKA 46 AŽURIRANJE PROZOR - UNOS SASTANKA 34
SLIKA 47 AŽURIRANJE PROZOR - IZMJENA SASTANKA 34
SLIKA 48 AŽURIRANJE SASTANKA - ODABIR PARTNERA 35
SLIKA 49 AŽURIRANJE SASTANKA - UNOS PARTNERA 35
SLIKA 50 AŽURIRANJE PARTNERA - IZMJENA PARTNERA 36
SLIKA 51 AŽURIRANJE SASTANKA - AŽURIRANJE ZAPOSLENIKA 36
SLIKA 52 AŽURIRANJE SASTANKA - AŽURIRANJE PARTNEROVIH ZAPOSLENIKA 37
SLIKA 53 AŽURIRANJE PARTNERA - UNOS PARTNEROVIH ZAPOSLENIKA 37