

# Izrada poslovne aplikacije u ClarionAnyScreen za upravljanje teretanom

---

**Višić, Lovro**

**Undergraduate thesis / Završni rad**

**2023**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:195:083679>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-17**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Prijediplomski studij Informatike

Lovro Višić

Izrada poslovne aplikacije u  
ClarionAnyScreen za upravljanje  
teretanom

Završni rad

Mentor: doc. dr. sc. Martina Ašenbrener Katić

Rijeka, Rujan 2023.

Rijeka, 12. lipnja 2023.

## Zadatak za završni rad

Pristupnik: Višić Lovro

Naziv završnog rada: Izrada poslovne aplikacije u Clarion AnyScreen za upravljanje teretanom

Naziv završnog rada na engleskom jeziku: Development of a business application in Clarion AnyScreen for gym management

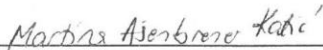
Sadržaj zadatka:

Zadatak završnog rada je opisati postupak planiranja i stvaranja aplikacije za upravljanje teretanom. Aplikacija će imati različite funkcionalnosti kao što su evidencije podataka (evidentirat će se informacije o članovima teretane, njihovim članarinama, inventaru i zaposlenicima, podatci o treningu kao što su vrijeme treninga, datum treninga i slično).

Procesu izrade aplikacije prethodit će analiza sustava, definiranje korisničkih zahtjeva, oblikovanje modela podataka i dizajn sučelja. Potrebno je aplikaciju izraditi koristeći Clarion AnyScreen te opisati postupak izrade aplikacije.


Mentor

Doc. dr. sc. Martina Ašenbrener Katić

  
\_\_\_\_\_

Voditelj za završne radove

Doc. dr. sc. Miran Pobar

  
\_\_\_\_\_

Zadatak preuzet: 19. lipnja 2023.

  
\_\_\_\_\_

(potpis pristupnika)

## Sažetak

Tema ovog završnog rada je izrada poslovne aplikacije koja će omogućavati upravljanje poslovanjem teretane. Rad obuhvaća cjelokupni postupak izrade aplikacije, definiranje persona, korisničkih priča, dizajn sučelja te sam razvoj aplikacije.

Ključne riječi: Clarion, tip entiteta, slabi tip entiteta, procedura, izvještaj

## Sadržaj

Uvod.....	1
Persone.....	2
Korisničke priče .....	4
Model podataka.....	5
Relacijska shema .....	7
Opis tipova entiteta i atributa .....	8
Dizajn aplikacije .....	10
Izrada aplikacije .....	12
Izrada rječnika.....	12
Generator aplikacija .....	16
Izrada procedura ažuriranja za složene tablice .....	23
Pregled slabih tablica.....	25
Automatski izračuni u polju preko definirane formule .....	27
Dodatne opcije.....	29
Izrada izvještaja.....	31
Jednostavni izvještaj.....	32
Izvještaj o jednome zapisu.....	34
Složeni izvještaj.....	35
Postavljanje aplikacije na server .....	40
Zaključak.....	42
Literatura.....	43
Popis slika.....	43

## Uvod


U ovom radu opisuje se postupak razvoja poslovne aplikacije za upravljanje teretanom. Ova aplikacija, dizajnirana za korištenje od strane djelatnika teretane u njihovom svakodnevnom radu, osmišljena je kako bi pojednostavila i ubrzala administrativne zadatke. Omogućit će brz i jednostavan pristup relevantnim informacijama, unos novih podataka, ažuriranje postojećih te brisanje nepotrebnih zapisa. Kako bih razvio ovu aplikaciju, koristi se razvojno okruženje Clarion.

Clarion je 4GL jezik i okruženje za generiranje koda za izgradnju pojedinačnih i višekorisničkih programa. Možemo reći da je Clarion jedan od RAD (Rapid Application Development) alata (Jakšić,2020).

## Persone

Kako bih stekao bolju ideju o korisnicima moje aplikacije, kreirat ću dvije persone. Prva persona predstavlja Marka Marinovića, zaposlenika teretane koji obavlja poslove poput učlanjivanja novih korisnika, praćenja članarina, provođenja treninga i slično. Personu možemo vidjeti na sljedećoj slici.

# Marko Marinović



### Ciljevi

- Napredovati u svojoj karijeri kao privatni trener
- Jednog dana otvoriti svoju teretanu
- Promovirati zdravi način života
- Motivirati članove da napreduju u vježbanju te da budu redoviti

### Važnost kod aplikacije

Jednostavno korištenje	<div style="width: 90%;"></div>
Dizajn	<div style="width: 60%;"></div>
Interaktivnost	<div style="width: 80%;"></div>

"Želim motivirati ljude da uzmu svoje zdravlje u svoje ruke, te da to postignu zdravom prehranom i fitnessom, to me inspirira svaki dan"

Dob: **25**  
Posao: **Trener**  
Obitelj: **Slobodan**  
Mjesto: **Zadar, Hrvatska**  
Prihod: **Srednji**

### Frustracije

- Članovi teretane koji izgube motivaciju i prestanu dolaziti na njegove treninge.
- Ne motivirani klijenti
- Klijenti koji ne prate njegove savjete

### Biografija

Marko ima 25 godina. Od malena je volio razne sportove i natjecao se u njima. Kroz svoju mladost trenirao je nogomet, košarku i plivanje. Zbog svoje ljubavi prema sportu, odlučio je upisati fakultet kineziologije u Zagrebu. Tijekom studija, fokusirao se na proučavanje predmeta koji se bave vježbanjem u teretani, s ciljem da nakon diplomiranja može raditi kao privatni trener. Diplomirao je s 24 godine i nakon toga se vratio u Zadar, gdje je pronašao posao u teretani. Osim upravljanja teretanom, radi i kao privatni trener za polaznike teretane. U budućnosti ima planove za otvaranje vlastite teretane.


### Scenariji korištenja aplikacije

- Želi dodati novog korisnika koji se učlanio u sustav
- Želi provjeriti stanje članarine korisnika
- Želi provjeriti informacije o treninzima

Slika 1- Persona Marko Marinović

Sljedeća persona predstavlja Anu Janjić, vlasnicu teretane za čiju teretanu razvijamo sustav. Ona će također obavljati administrativne poslove poput provjera članarina, organiziranja treninga i slično. Personu Ana Janjić možemo vidjeti na sljedećoj slici.

# Ana Janjić



*"Slijedite svoje snove i vjerujte u sebe"*

Dob: **45**  
Posao: **Vlasnica teretane**  
Obitelj: **U braku**  
Mjesto: **Zadar, Hrvatska**  
Prihod: **Visoki**

## Ciljevi

- Povećati broj članova teretane
- Imati zadovoljne korisnike
- Proširiti poslovanje, otvoriti više teretana
- Osigurati da klijenti imaju ugodno iskustvo pri korištenju teretane

## Frustracije

- Slab broj povećanja članarina
- Klijenti ne obnavljaju članarine
- Ne adekvatni zaposlenici
- Zaposlenici koji nisu motivirani pri radu s klijentima

## Biografija

Ana ima 45 godina. Uvijek je voljela biti aktivna, te se voljela baviti nekim oblikom vježbe. Svoju karijeru započela je rano, nije upisala fakultet, već se odlučila odmah po završetku škole tražiti posao. Tijekom tog vremena, redovito je vježbala u teretani i istovremeno se odlučila natjecati u powerliftingu. Njena predanost donijela joj je uspjeh, te je tako stvorila ime za sebe unutar powerlifting zajednice.

Kroz svoje iskustvo u treniranju i natjecanju, Ana je odlučila stjecati znanje i vještine, te je stekla licencu kao privatni trener, te se zapošljava kao privatni trener. U svojim tridesetima se odlučila umiroviti od natjecanja, te nastavlja raditi kao osobna trenerica.

S 40 godina ostvaruje svoj san otvaranja teretane. Trenutno vodi teretanu na jednoj lokaciji, no ima planove za otvaranje na novim lokacijama u budućnosti.

## Važnost kod aplikacije

Jednostavno korištenje

Dizajn

Interaktivnost

## Scenariji korištenja aplikacije

- Želi dodati novog korisnika koji se učlanio u sustav
- Želi dodati novog zaposlenika u sustav
- Želi organizirati raspored treninga
- Želi pratiti zaposlenike te njihove odgovornosti

Slika 2- Persona Ana Janjić



## Korisničke priče

Korisnička priča je informativno, općenito objašnjenje značajke softvera napisano iz perspektive krajnjeg korisnika. Njegova je svrha artikulirati kako će značajka softvera pružiti vrijednost korisniku (Rehkopf, 2023).

Korisnička priča je kratak, jednostavan opis značajke ispričan iz perspektive osobe koja želi novu mogućnost, obično korisnika ili kupca sustava (Cohn, 2023)

Svrha korisničke priče je artikulirati kako će projekt vratiti određenu vrijednost kupcu (Rehkopf, 2023).

Pomoću korisničkih priča, dobivamo ideju o tome kakve funkcionalnosti bi naša aplikacija trebala sadržavati, te uvjete korisnika koje želimo zadovoljiti.

U nastavku su korisničke priče za kreirane persone Anu i Marka.

Kao korisnica aplikacije, želim imati mogućnost dodavanja informacija o novom klijentu u sustav kako bih uspješno provela postupak učlanjivanja u teretanu. Ovo će mi omogućiti učinkovito upravljanje informacijama o klijentima, praćenje njihovih članarina te osiguravanje pristupa teretani.

Kao korisnica aplikacije, želim imati mogućnost dodavanja informacija o nadolazećim treninzima kako bi svi zaposlenici teretane mogli pregledati raspored nadolazećih treninga, te znati za koji trening su zaduženi. Ovo mi omogućava lakšu koordinaciju zaposlenika teretane, te bolje upravljanje treninzima.

Kao korisnik aplikacije, želim jednostavno i intuitivno sučelje aplikacije, koje mi omogućava laganu navigaciju kroz aplikaciju pri odrađivanju potrebnih poslova unutar aplikacije. Ovo je važno jer želim svoje administracijske poslove obavljati brzo i učinkovito, jednostavno navigirajući kroz sučelje aplikacije.

Kao korisnik aplikacije, želim jednostavno pregledavati i ažurirati nadolazeće treninge. Ovo mi je važno kako bi znao za koje treninge sam zadužen, kada je termin treninga, te koju vrstu treninga provodim, što mi omogućava bolje planiranje treninga.

Kao korisnik aplikacije, želim imati mogućnost pretraživanja podataka unutar aplikacije kako bih lakše mogao pronalaziti potrebne podatke koje moram ažurirati. Ova opcija će mi omogućiti brže i učinkovitije obavljanje poslova jer ću moći lako locirati tražene podatke bez nepotrebnog gubljenja vremena.

Kao korisnik aplikacije, želim imati opciju generiranja izvještaja o inventaru opreme kako bih mogao pratiti stanje opreme i, ako je potrebno, lako naručiti novu opremu. Ova mogućnost će mi omogućiti bolju kontrolu nad opremom te efikasno planiranje nabavke nove opreme kad god je to potrebno.

Kriteriji prihvatljivosti:

- Aplikacija se pokreće

- Omogućen unos novog zapisa u tablicu klijent
- Omogućen unos podataka u potrebna polja kao što su ime, prezime...
- Mogućnost pohrane informacija u sustav
- Informacije o klijentu su uspješno dodane u sustav
- Mogućnost evidencije klijenta
- Mogućnost pregleda informacija o članarinama klijenata
- Mogućnost pregleda unesenih podataka
- Mogućnost ažuriranja unesenih podataka, unos novih
- Generiranje izvješća

## Model podataka

Model podataka (engl. data model) je skup međusobno povezanih podataka koji opisuju entitete, veze i attribute poslovnoga sustava. On je reprezentacija skupa podataka koji se modelom interpretiraju preko aspekta: strukture, ograničenja i operatora (Pavlič, 2011).

Za izradu modela podataka koristi se model entiteti-veze. Korištena je metodologija MIRIS.

Specijalizirana metodologija MIRIS (skraćeno od hrvatskog Metodologija za Razvoj Informacijskog Sustava) je skup metoda i uputa čiji je ukupni cilj projektirati i izgraditi informacijski sustav. Poznata je i kao metodika MIRIS. Ta specijalizirana metodologija propisuje faze razvoja i aktivnosti pojedine faze do potrebne razine detalja informacijskih sustava (Pavlič, 2011).

Metoda entiteti - veze (skraćeno EV) je grafički prikaz međusobno povezanih grupa podataka promatranoga sustava (Pavlič, 2011).

Za model podataka teretane definirao sam pet tipova entiteta, četiri slaba tipa entiteta, te veze između njih, kao i pripadajuće ključeve i attribute.

Entitet (engl. entity) je pojam u poslovanju o kome se zahtijeva čuvanje nekih podataka (Pavlič, 2011).

Tip entiteta (engl. entity type) je skup sličnih pojedinačnih entiteta dobiven procesom klasifikacijske apstrakcije. To je skup entiteta istoga tipa (Pavlič, 2011).

Agregacija (mješoviti tip entiteta, agregirani tip entiteta) je apstrakcija u kojoj se tip veze između dvaju ili više tipova entiteta tretira kao novi tip entiteta (Pavlič, 2011).

Slab tip entiteta (engl. weak entity) je tip entiteta koji je na neki način ovisan o nekom drugome tipu entiteta, a ta se ovisnost prikazuje specijalnim tipom veze među jakim i slabim tipom entiteta (Pavlič, 2011).

Veza predstavlja odnos koji postoji među entitetima bilo u stvarnosti, bilo u mislima. Entiteti se nalaze u različitim odnosima, a te odnose nazivamo veze (Pavlič, 2011).

Tip veze (engl. relationship type) je skup veza između istih tipova entiteta (Pavlič, 2011).

Brojnost tipa veze je broj (upisuje se eksplicitno na DEV) koji kaže koliko entiteta pojedinoga tipa entiteta E1 sudjeluje (pojavljuje se) u tipu veze V s entitetom iz tipa entiteta E (Pavlič, 2011)

Atribut entiteta (engl. attribute) je neko svojstvo entiteta. Na primjer: ime radnika, datum prispjeća narudžbe, faktor uvećanja, brzina vozila, šifra općine i dr. (Pavlič, 2011).

Za svaki tip entiteta definiram i njegov primarni ključ.

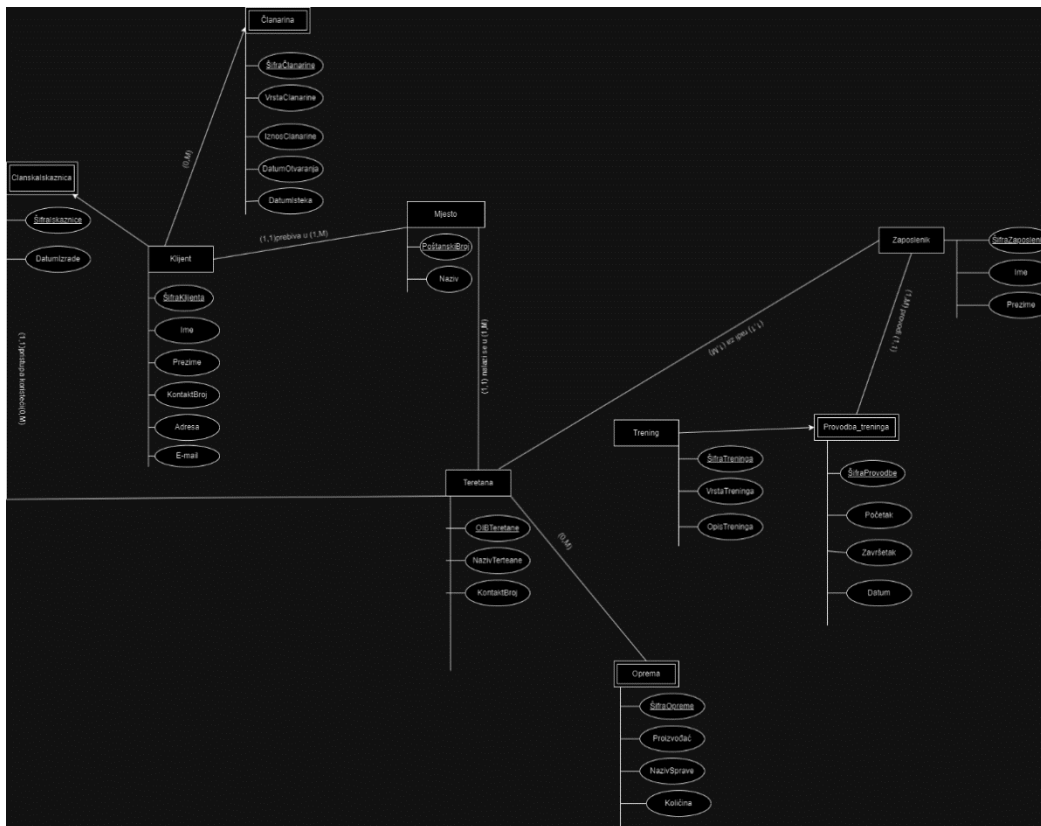
Ključ tipa entiteta (identifikator entiteta, engl. entity type key) je takav skup atributa koji neovisno o vremenu zadovoljava uvjet jedinstvenosti i uvjet neredundantnosti (Pavlič, 2011).

Primarni ključ (glavni ključ, engl. primary key) je jedan od ključeva tipa entiteta koji je odabran zbog nekoga značajnijeg razloga za zamjenika entiteta toga tipa (za ključ). Jedan od razloga je njegovo često korištenje. U EV metodi uvodimo samo jedan ključ za svaki tip entiteta i to primarni (glavni) ključ. Ovaj će ključni atribut postati osnova za fizičko zapisivanje podataka u bazu podataka (Pavlič, 2011).

Osim primarnih ključeva, neke tablice također sadržavaju vanjske ključeve.

Ako neki skup atributa u promatranoj relaciji nije ključ, ali je ključ u nekoj drugoj relaciji u modelu, onda se on naziva vanjski ključ (engl. foreign key) (Pavlič, 2011).

Na sljedećoj slici prikazani su jaki i slabi tipovi entiteta zajedno s njihovim atributima i vezama.



Slika 3- Model EV

## Relacijska shema

Definirani model podataka prevodi se u relacijsku shemu, koja će se kasnije koristiti pri izradi rječnika podataka. Kako bi preveli model podataka u relacijsku shemu, prate se sljedeća osnovna pravila (Pavlič, 2011):

- Atribut u DEV-u (na primjer Ime radnika) odgovara konceptu Atribut u RM-u
- Svaki tip entiteta iz DEV-a postaje relacija (tablica) u RM-u i to tako da atributi tipa entiteta postaju atributi relacije, primarni (glavni) ključ tipa entiteta postaje primarni (glavni) ključ relacije
- Svaka agregacija postaje relacija u RM-u, ključ relacije je složen od ključeva tipova entiteta koji sudjeluju u vezi. Atributi agregacije postaju atributi relacije
- Svaki tip veze s brojnostima (1,1):(0,M) (tip veze gdje je strana s brojnošću po gornjoj granici 1 totalna) ne prevodi se u relaciju, već se ključ tipa entiteta koji ulazi u vezu sa strane s brojnošću M umeće kao atribut (vanjski ključ) u relaciju dobivenu prevođenjem tipa entiteta s brojnošću (1,1). Tip veze ne smije imati attribute jer, ako ih ima, to nije tip veze već agregirani tip entiteta
- Tip slabog entiteta prevodi se u relaciju. Atributi tipa slabog entiteta postaju atributi relacije. Ključ relacije je složen i od ključa jakoga tipa entiteta i ključa slaboga tipa entiteta tako da jedinstveno identificira n-torke relacije u okviru relacijskoga modela

Prateći navedena pravila, dobivamo sljedeću relacijsku shemu.

**Klijent**(ŠifraKlijenta, Ime, Prezime, KontaktBroj, Adresa,E-mail,#*PoštanskiBroj*)

**Mjesto**(PoštanskiBroj, Naziv)

**ČlanskaIskaznica**(ŠifraKlijenta, ŠifraIskaznice, DatumIzrade,#*OIBTeretane* )

**Članarina**(ŠifraKlijenta, ŠifraČlanarine, VrstaČlanarine, IznosČlanarine, DatumOtvaranja, DatumIstecka)

**Teretana**(OIBTeretane, NazivTeretane, KontaktBroj, #*PoštanskiBroj*)

**Zaposlenik**(ŠifraZaposlenika, Ime, Prezime, #*OIBTeretane*)

**Trening**(ŠifraTreninga, VrstaTreninga, OpisTreninga)

**ProvdbaTreninga**(ŠifraTreninga, ŠifraProvodbe, Početak, Završetak, Datum, #*ŠifraZaposlenika*)

**Oprema** (ŠifraOpreme, OIBTeretane, Proizvođač, NazivSprave, Količina)

### Opis tipova entiteta i atributa

Tip entiteta KLIJENT koristi se kako bi predstavljao ljude koji vježbaju u teretani, te koji su učlanjeni u teretani. Sastoji se od sljedećih atributa:

- ŠifraKlijenta – ovaj atribut koristim kako bi jednoznačno mogao označiti svakog klijenta
- Ime – prvo ime klijenta
- Prezime – prezime klijenta
- KontaktBroj – telefonski broj klijenta
- E-mail – e-mail adresa klijenta
- Adresa – kućna adresa klijenta

Slabi tip entiteta Članarina predstavlja članarinu koju korisnik mora kupiti kako bi trenirao u teretani. Članarine se kupuju mjesečno, te je moguć odabir više vrsta članarine. Sastoji se od sljedećih atributa:

- ŠifraČlanarine – ovaj atribut koristim kako bi jednoznačno mogao označiti kupovinu svake članarine
- VrstaČlanarine – ovaj atribut koristim kako bi se specificiralo koju vrstu članarine je klijent odabrao. Neke od vrsti članarina koje će klijent moći odabrati su: privatni trening, grupni trening, 3-puta tjedno trening, no-limit trening, jutarnja članarina, studentska članarina
- IznosČlanarine – cijena odabrane članarine u eurima, cijene variraju ovisno o odabranoj vrsti članarine
- DatumOtvaranja – datum kojeg je korisnik kupio ili produžio svoju članarinu
- DatumIstecka – datum kada trenutna članarina istječe te je treba obnoviti

Tip entiteta Mjesto predstavlja jedan grad. U ovom slučaju to će predstavljati grad u kojem se nalazi teretana, te gradove u kojima klijenti imaju prebivalište. Sastoji se od sljedećih atributa:

- PoštanskiBroj – jedinstveno identificira svaki grad u Hrvatskoj
- Naziv – naziv grada kojem pripada definirani poštanski broj

Slabi tip entiteta ČlanskaIskaznica predstavlja člansku iskaznicu koju klijent dobije pri prvom učlanjivanju u teretanu, te je klijent nadalje koristi kako bi mogao dokazati da je učlanjen u ovu teretanu, te tako pristupa teretani. Sastoji se od sljedećih atributa:

- ŠifraIskaznice – šifra koja jedinstveno označuje svaku iskaznicu, a tim i svakog klijenta
- DatumIzrade – datum kada je klijentu izrađena njihova iskaznica

Tip entiteta Teretana, koristi se kako bi predstavljale informacije o teretani. Sastoji se od sljedećih atributa:

- OIBTeretane – koristi se kako bi jedinstveno identificirali teretanu. Koristim OIB jer svako poslovanje ima svoj OIB, te teretana kao registrirano poslovanje također ima svoj OIB.
- NazivTeretane
- KontaktBroj – broj telefona kojim se može nazvati teretanu

Tip entiteta Zaposlenik predstavlja zaposlenike teretane. Zaposlenici su ujedno i treneri koji su zaduženi za provođenje privatnih i grupnih treninga. Sastoji se od sljedećih atributa:

- ŠifraZaposlenika – šifra koju koristim kako bi mogao identificirati svakog zaposlenika
- Ime
- Prezime

Tip entiteta Trening sadrži informacije o treninzima koje teretana nudi. Ovo se ne odnosi na treninge koje klijenti sami obavljaju, već na ponuđene privatne i grupne treninge. Sastoji se od sljedećih atributa:

- ŠifraTreninga – koristim kako bi mogao identificirati svaku vrstu treninga
- VrstaTreninga – npr grupni trening
- Opis treninga – opis što će se raditi na tom treningu, npr. trening za noge

Slabi tip entiteta Oprema predstavlja popis opreme koju teretana posjeduje. To su sprave za vježbanje kao npr. utezi 10kg, 12 kg...30kg, šipka 10kg, 20kg, te ostale razne sprave. Sastoji se od sljedećih atributa:

- ŠifraOpreme – koristimo kako bi mogli identificirati svaku vrstu opreme
- Proizvođač – proizvođač sprave
- NazivSprave – pripadajući naziv sprave, npr. Squat Rack
- Količina – količina odabrane opreme koja se trenutno nalazi u teretani

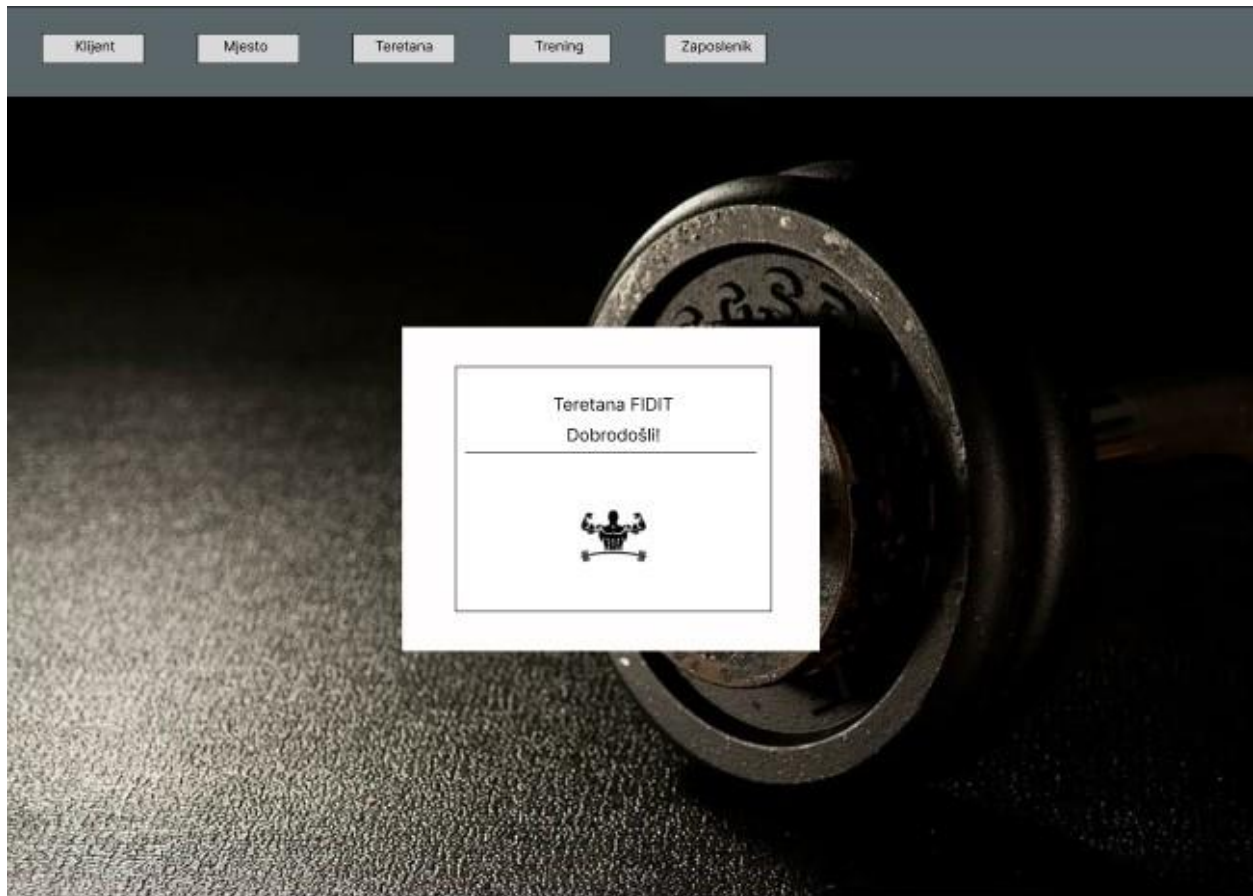
Slabi tip entiteta Provodba\_Treninga nam predstavlja dodatne informacije o pojedinom treningu. Uz dodatne informacije o provedbi treninga također ima vezu na entitet Zaposlenik, te nam time pruža informaciju o tome koji zaposlenik je zadužen za provedbu pojedinog treninga. Sastoji se od sljedećih atributa:

- ŠifraProvodbe – koristimo kako bi pojedinačno označili svaku provedbu treninga
- Početak – vrijeme početka treninga
- Završetak – vrijeme završetka treninga

- Datum – datum provedbe treninga

## Dizajn aplikacije

Prije početka razvoja aplikacije, radim skice za 3 ekrana. Skice su izrađene za desktop izgled početnog ekrana, ekrana za procedure popisa, te za ekran ažuriranja podataka. Skice su prikazane na sljedećim slikama.



*Slika 4 Skica početnog ekrana*

Popis Zaposlenika

Pretraga:

Sifra Zaposlenika	Ime	Prezime	Naziv teretane

Slika 5 Skica ekrana za popis

Ažuriranje podataka o zaposleniku

Sifra zaposlenika

Ime

Prezime

Oib teretane

68 x 19

Slika 6 Skica ekrana za ažuriranje

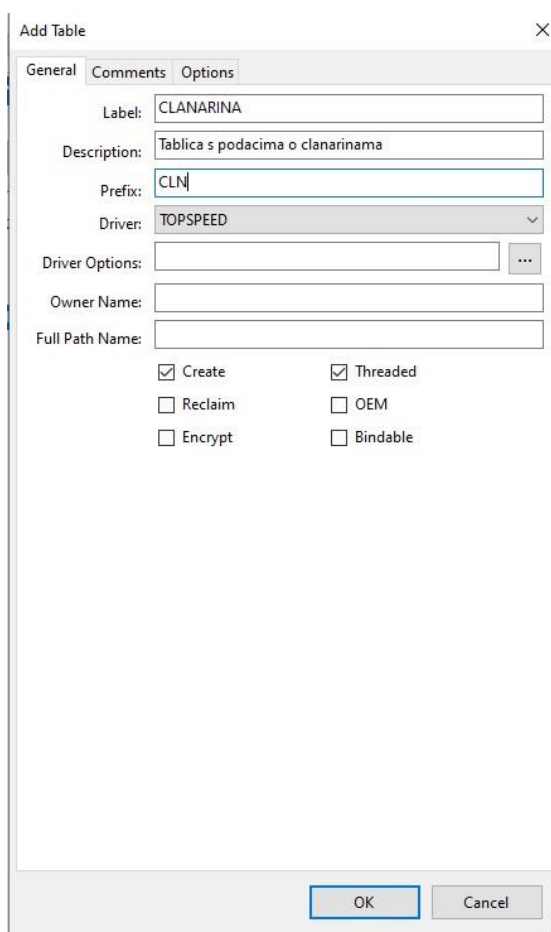


## Izrada aplikacije

### Izrada rječnika

Prvi korak u izradi aplikacije je izrada rječnika. Rječnik izrađujemo na temelju definiranog modela podataka, u kojem definiramo tablice i njihove informacije, kao što su primarni ključevi, vanjski ključevi, veze te svojstva podataka.

Prvi korak u definiranju rječnika je kreiranje pripadajućih tablica. U ovom slučaju, koristimo relacijski model podataka te kreiramo ukupno 9 tablica. Na sljedećoj slici možemo vidjeti primjer kreiranja tablice Članarina. Sljedeći isti postupak, kreiramo i ostale tablice: Članska\_iskaznica, Klijent, Mjesto, Oprema, Provodba\_treninga, Teretana, Trening i Zaposlenik.

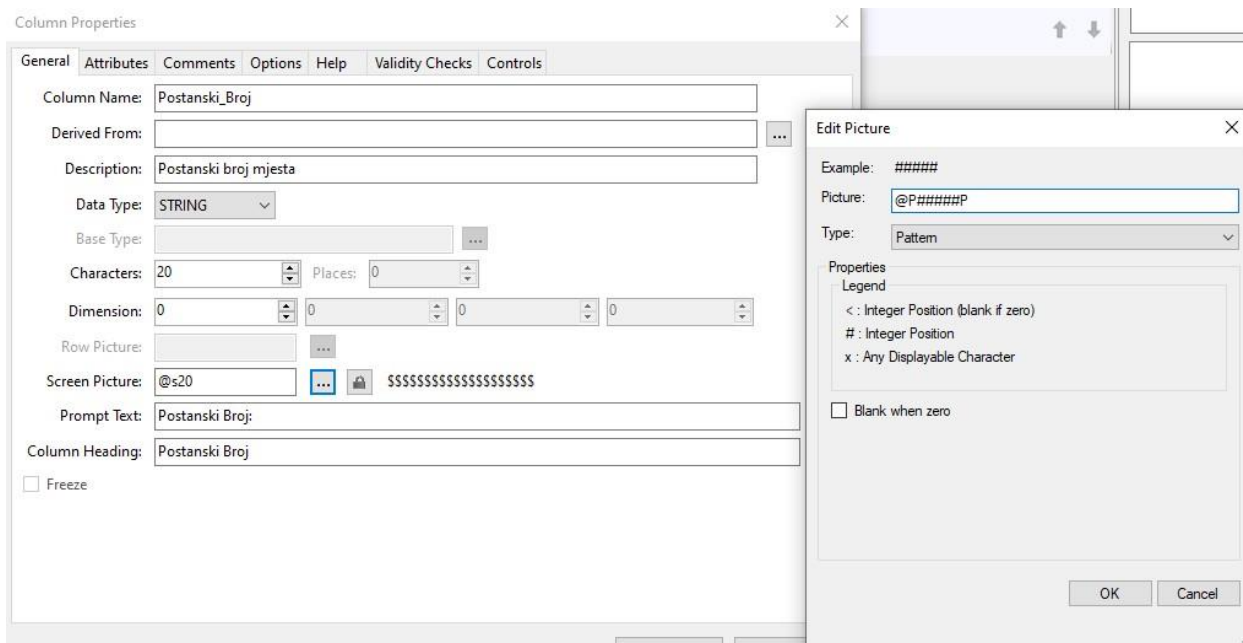


The image shows a 'Add Table' dialog box with the following fields and options:

- Label: CLANARINA
- Description: Tablica s podacima o clanarinama
- Prefix: CLN
- Driver: TOPSPEED
- Driver Options: (empty field with a menu icon)
- Owner Name: (empty field)
- Full Path Name: (empty field)
- Checkboxes:  Create,  Threaded,  Reclaim,  OEM,  Encrypt,  Bindable

Slika 7 Kreiranje tablice Članarina

Nakon što kreiramo tablice rječnika, sljedeći korak je definiranje stupaca za svaku tablicu. Prilikom kreiranja stupaca također definiramo tip podataka koji će sadržavati, te po potrebi određujemo i maksimalni broj znakova koje podatak može sadržavati. Podatci mogu biti tipa STRING, CSTRING, DATE i slično. U slučaju da je potrebno da podatak sadržava točan broj znamenki ili znakova, koristi se opcija „Pattern“. Primjer takvog stupca prikazan je na sljedećoj slici.



Slika 8 Kreiranje stupca tablice

Nakon definiranja osnovnih parametara, također je moguće dodatno specificirati opcije kao što su pravila za unos podataka te atributi kao što je početna vrijednost polja i slično. Kada je stupac kojeg definiramo vanjski ključ druge povezane tablice, koristi se opcija "Derived From", kojom se odabire odgovarajući pripadajući stupac.

Nakon što kreiramo sve potrebne stupce, sljedeći korak u izradi rječnika je definiranje primarnih i vanjskih ključeva stupaca. Također postoji mogućnost definiranja sortirnog ključa, koji se kasnije koristi u aplikaciji za sortiranje podataka po definiranom stupcu. Primjer kreiranja primarnog ključa tablice *Klijent* prikazan je na sljedećoj slici.

The image shows a configuration window for a primary key in a database management system. The window has three tabs: 'General', 'Comments', and 'Options', with 'General' selected. The 'Label' field contains 'PK\_KLIJENT\_SIFRA\_KLIJENTA'. The 'Description' field contains 'primarni kljuc tablice klijent'. The 'External Name' field is empty. Under the 'Type' section, 'Row Key' is selected with a radio button, while 'Runtime Index', 'Static Index', and 'Order' are not. The 'Attributes' section has checkboxes for 'Require Unique Value' (checked), 'Primary Key' (checked), 'Auto Number' (unchecked), 'Exclude Empty Keys' (unchecked), and 'Case Sensitive' (unchecked). The 'Columns' section shows a list of columns with 'Sifra\_Klijenta' selected. The list has a header bar with a red triangle icon and the text 'Sifra\_Klijenta'. Above the list are icons for adding, deleting, and sorting columns.

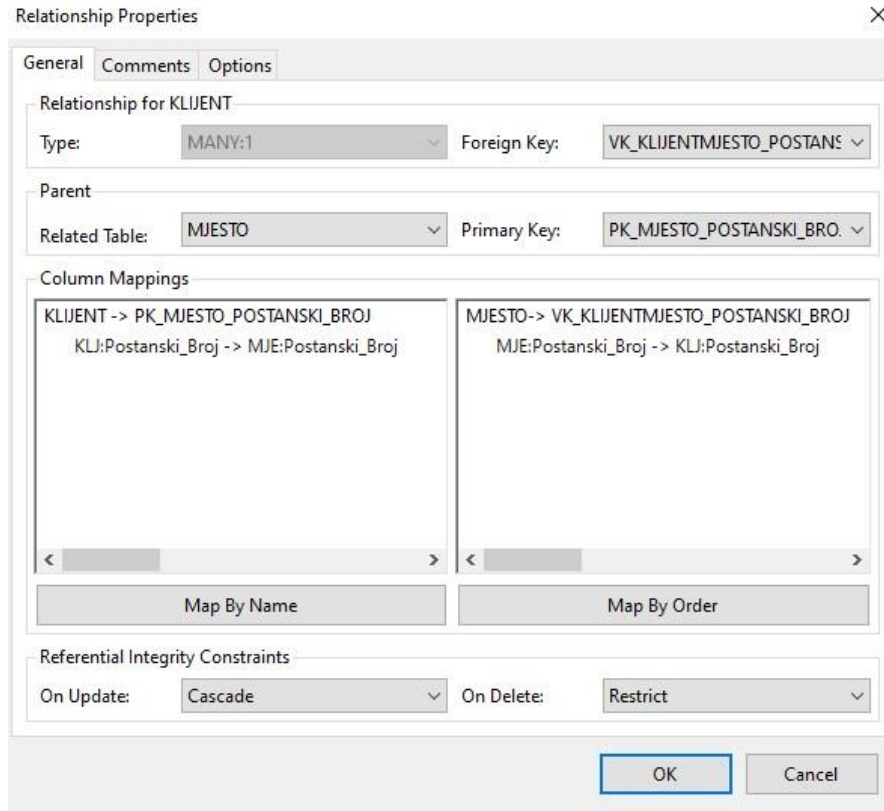
Slika 9 Definiranje primarnog ključa

Pri definiranju ključeva definiramo njegova osnovna svojstva. To uključuje naziv ključa, opis vrste podataka koje ključ pohranjuje te vrstu ključa koju definiramo. U našem rječniku odabire se opcija „Row Key“. Zatim se definiraju atributi ključa, atributi koji se za ovu aplikaciju najviše koriste su:

- „Require Unique Value“- ova opcija osigurava da stupac ne može sadržavati dvostruke vrijednosti
- „Primary Key“ - ova opcija označava polje kao primarni ključ koji jedinstveno identificira svaki zapis.

Zadnji korak u definiranju ključa tablice je odabir stupaca koji će biti dio ključa.

Nakon što smo za sve tablice definirali potrebne ključeve, prelazimo na definiranje veza između tablica. Na sljedećoj slici prikazano je kreiranje veze između tablica Klijent i Mjesto.



Slika 10 Definiranje veze između tablica

Prva opcija koju biramo pri kreiranju veze je tip veze. Ovdje možemo birati između dvije opcije:

- 1:MANY – ova opcija znači da jedan klijent može imati prebivalište u više mjesta
- MANY:1 – ova opcija znači da više klijenata može imati prebivalište u jednom mjestu

Nakon odabira tipa veze, sljedeća opcija koju biramo je ključ koji se koristi kako bi se tablica Klijent povezala s tablicom Mjesto. U primjeru na slici to je vanjski ključ klijenta koji sadržava informaciju o poštanskom ključu.

U sljedećem bloku biramo tablicu s kojom želimo povezati tablicu za koju kreiramo vezu. U primjeru prikazanom na prethodnoj slici, biramo tablicu Mjesto. Nadalje, u istom bloku također biramo ključ iz odabrane tablice koji će se koristiti za povezivanje s tablicom Klijent.

Nakon što su sve opcije definirane, izvršavamo povezivanje tablica klikom na „Map By Name“ ili „Map By Order“ (u ovom primjeru korištena je opcija „Map By Name“).

Posljednje opcije koje biramo su „On Update“ i „On Delete“. U opciji „On Update“ biramo što se događa sa zapisima jedne tablice ako se podatak iz druge povezane tablice izmjeni.

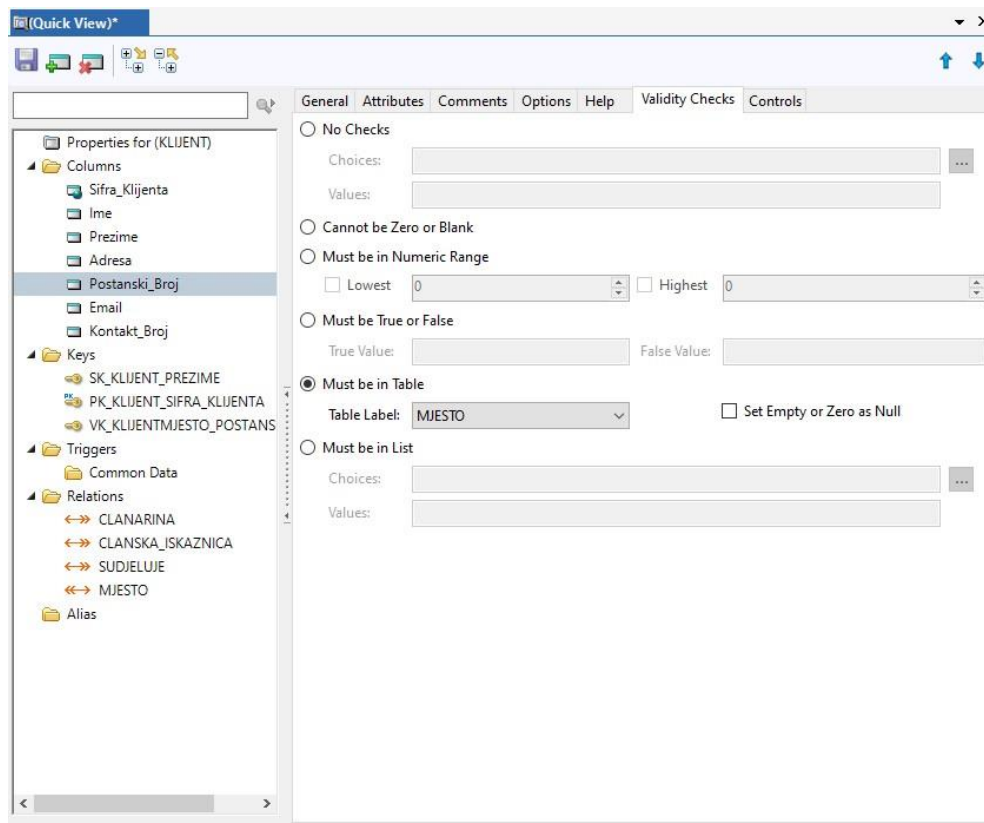
U opciji „On Delete“ biramo što se događa sa zapisima jedne tablice ako se podatak iz druge povezane tablice izbriše.

Prateći ovaj postupak kreiramo sve potrebne veze definirane u modelu podataka.

Nakon što smo definirali sve potrebne veze, slijedi postavljanje referencijalnog integriteta.

Postavljanje referencijalnog integriteta u rječniku podataka predstavlja definiranje relacijsko zavisnih polja(Jakšić,2020).

Referencijalni integritet postavimo tako da u kartici „Validity Checks“ odaberemo opciju „Must be in Table“.



Slika 11- Definiranje referencijalnog integriteta

Ovo ograničenje postavljamo za(Jakšić,2020):

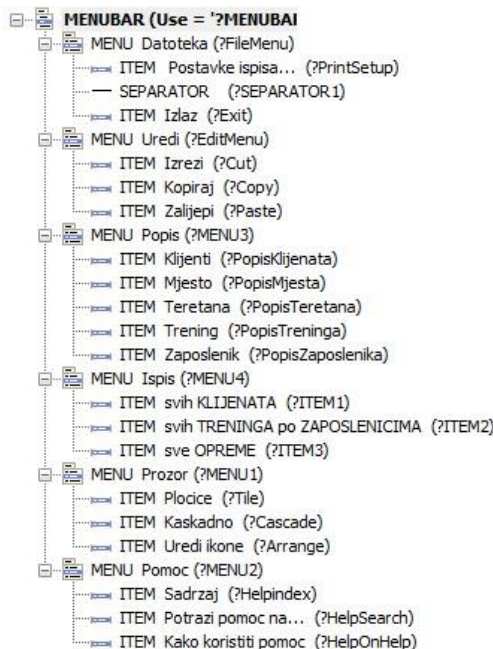
- sva polja koja se nalaze u vanjskim ključevima
- sva polja koja se nalaze u složenim primarnim ključevima agregacija
- sva polja koja se nalaze u složenim ključevima slabog tipa entiteta (slabe tablice), a da su „posuđena“ od jake tablice.

Ovo je zadnji korak u definiranju rječnika, te sada imamo izrađen rječnik i nastavljamo dalje s izradom aplikacije.

## Generator aplikacija

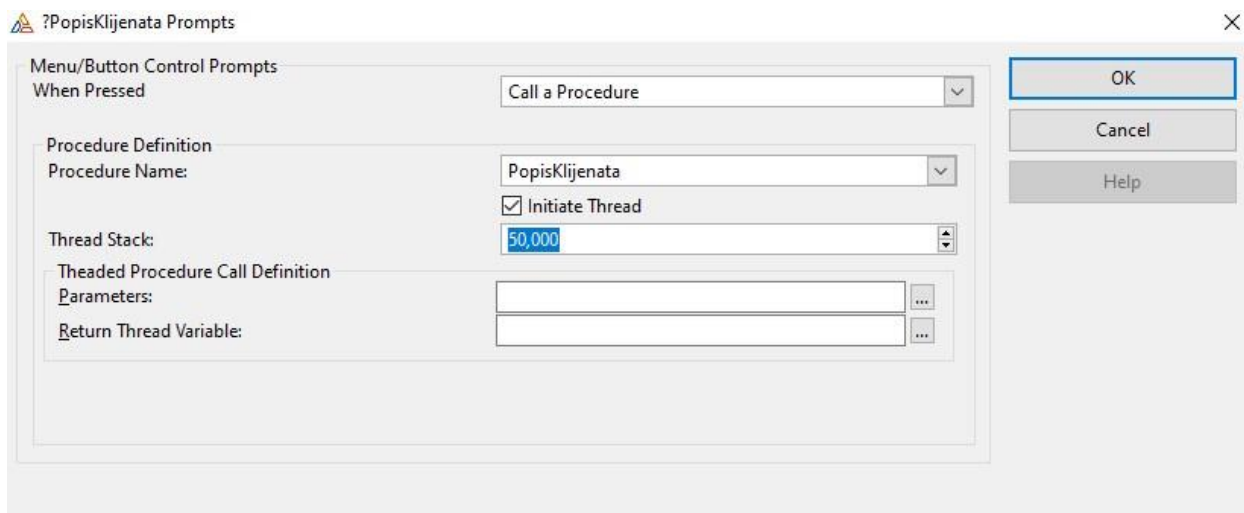
Prvi korak u radu s generatorom aplikacija je izrada aplikacijskog projekta. Nakon što kreiramo aplikacijski projekt, prelazimo na uređivanje glavnog izbornika. Tu prevodimo postojeći izbornik s engleskog jezika na hrvatski jezik, te po potrebi dodajemo još izbornika na izborničku traku. U ovom primjeru uz već postojeće izbornike, dodajemo još jedan koji koristimo za pristup tablicama

aplikacije. Da bismo to postigli, odaberemo izborničku traku i opciju „Edit menu“. Time se otvara uređivač izbornika, u kojem radimo potrebne promjene. Za potrebe naše aplikacije uz već definirane izbornike dodajemo izbornike „Popis“, u kojemu omogućujemo pristup svim tablicama modela podataka, osim slabim tipovima entiteta i agregacijama. Također, dodajemo izbornik „Ispis“, preko kojeg pristupamo izvješćima koja ćemo kasnije kreirati. Definirani glavni izbornik možete vidjeti na sljedećoj slici.



Slika 12 Definiranje izbornika aplikacije

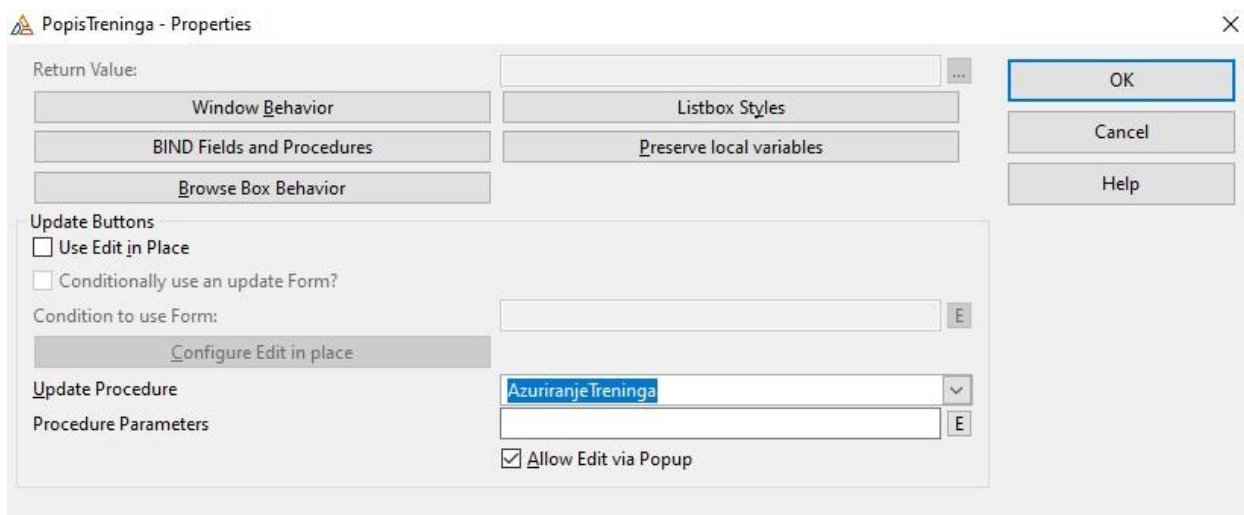
Također moramo definirati koja akcija se izvršava kada kliknemo na element izbornika. Da bi to definirali odaberemo željeni element, te desnim klikom odaberemo opciju „Actions“. Time se otvara novi prozor u kojem stvaramo proceduru koja će se izvršiti kada korisnik klikne na taj element. Definiranje procedure za „ITEM Klijenti“ iz izbornika prikazan je na sljedećoj slici.



Slika 13 Definiranje Popis procedure

Ovaj postupak ponavljamo i za preostale elemente iz izbornika „Popis“.

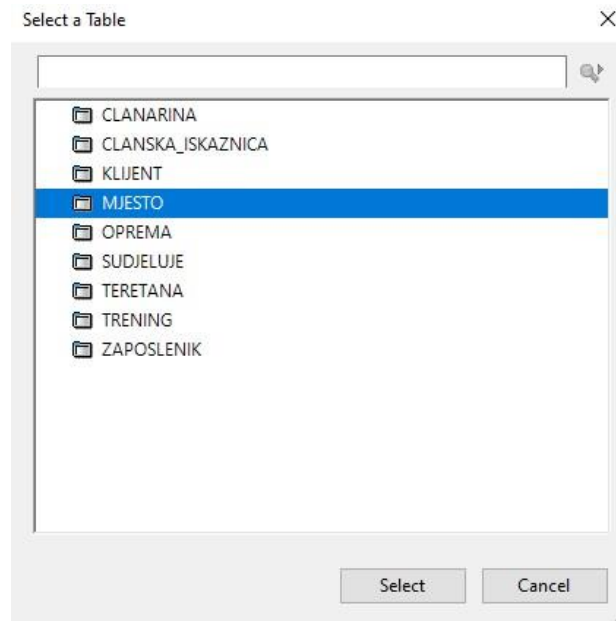
Nakon definiranja procedura izbornika, definiramo procedure za pregled podataka iz tablica. Kako bi ovo napravili, odaberemo proceduru za koju kreiramo pregled podataka, te za nju biramo predložak „Browse with Update and Select“. Pri kreiranju ove procedure omogućavamo automatsku prilagodbu veličine ekrana i elemenata na ekranu. Također, za određene ekrane omogućujemo opciju „Enable Sort Header“, koja omogućuje sortiranje podataka po zaglavlju stupca u tablici. Kako bismo dovršili kreiranje ove procedure, dodjeljujemo joj ime, što je prikazano na sljedećoj slici.



Slika 14 Procedura ažuriranja

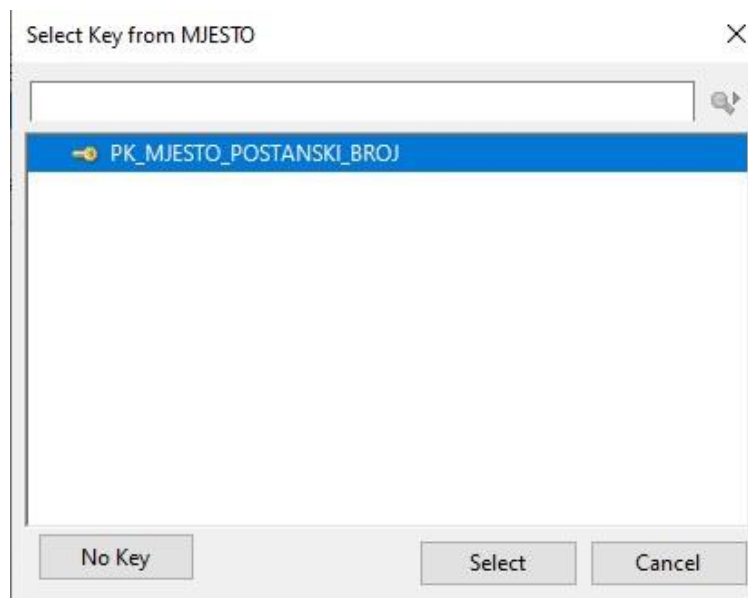
Prateći ovaj postupak, za svaku „Popis“ proceduru definiram i proceduru ažuriranja. Nakon definiranja svih potrebnih procedura, prelazimo na uređivanje ekrana koji ispisuju podatke tablica, kao što je primjerice ekran s popisima klijenata. Kada otvorimo prozor za uređivanje ekrana, prvo

u bloku „Data/Tables“ povezujemo pripadajuću tablicu iz rječnika s procedurom. Na sljedećoj slici prikazan je odabir odgovarajuće tablice za proceduru.



*Slika 15 Odabir tablice*

Procedura za koju radimo ispis je tipa „Browse with Update and Select“, te je svakoj proceduri tog tipa potrebno pridružiti uz tablicu iz rječnika podataka i njen primarni ključ. Na sljedećoj slici možemo vidjeti odabir odgovarajućeg primarnog ključa za proceduru.



*Slika 16 Odabir primarnog ključa*



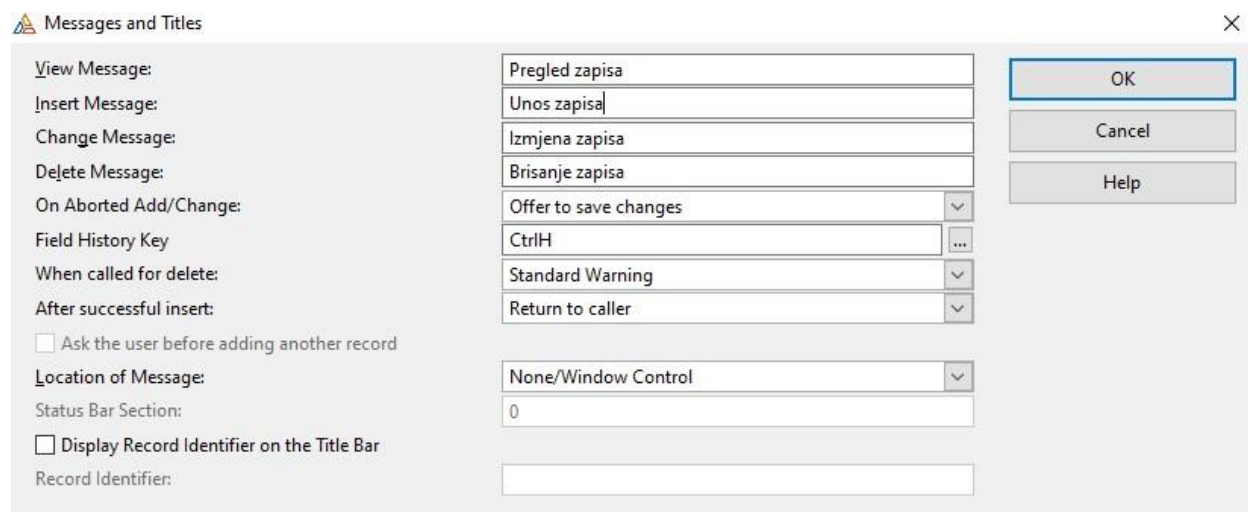
Sljedeći korak je omogućavanje ispisa podataka na ekranu aplikacije. Kako bismo to omogućili, kliknemo „Listbox“ element te otvorimo karticu „List Box format“. Otvara se okvir u kojem odabiremo stupce čije podatke želimo ispisivati.



Slika 17 Dodavanje stupaca u listbox

Također moguće je grupirati stupce, kao što je prikazano u primjeru na gornjoj slici grupiramo stupce Adresa, Naziv, Kontakt broj i E-mail u zajedničko zaglavlje naziva „Kontakt“.

Sljedeći korak je uređivanje ekrana za unos podataka i definiranje procedure za ažuriranje podataka u tablici. Odaberemo jednu od procedura ažuriranja, te joj za predložak odaberemo „Form(Add/Edit/Delete)“. Za svaku ovu proceduru, potrebno je prevesti zadane sistemske poruke koje će se prikazivati na ekranu.



Slika 18 Postavljanje sistemskih poruka

Kada se postave potrebne opcije, prelazimo na uređivanje ekrana ažuriranja. Kao i kod procedure za popis, procedurama ažuriranja na ovom ekranu prvo je potrebno pridružiti odgovarajuću tablicu iz rječnika podataka. No, procedurama ovog tipa nije potrebno pridruživati i primarni ključ.

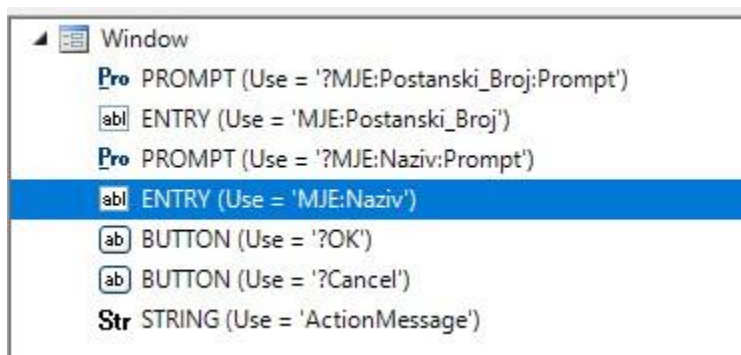
Sada, nakon što je proceduri pridružena odgovarajuća tablica, moramo omogućiti unos i izmjenu podataka te tablice na ekranu ažuriranja. Polja za unos i izmjenu možemo jednostavno odabirom povući i ispustiti u ekran. Sada nam ekran za proceduru ažuriranja mjesto izgleda ovako.



Slika 19 Ekran ažuriranja podataka

Primjećujemo da za svaki stupac iz tablice koji smo dodali na ekran, dobivamo tekst koji objašnjava polje i dinamičko polje za unos ili izmjenu vrijednosti.

Kada smo dodali sva potrebna polja na ekran, moramo postaviti redoslijed kojim će se pokazivač kretati nakon što korisnik pritisne tipku TAB. Primjer postavljenog redoslijeda možete vidjeti na slici koja je prikazana.

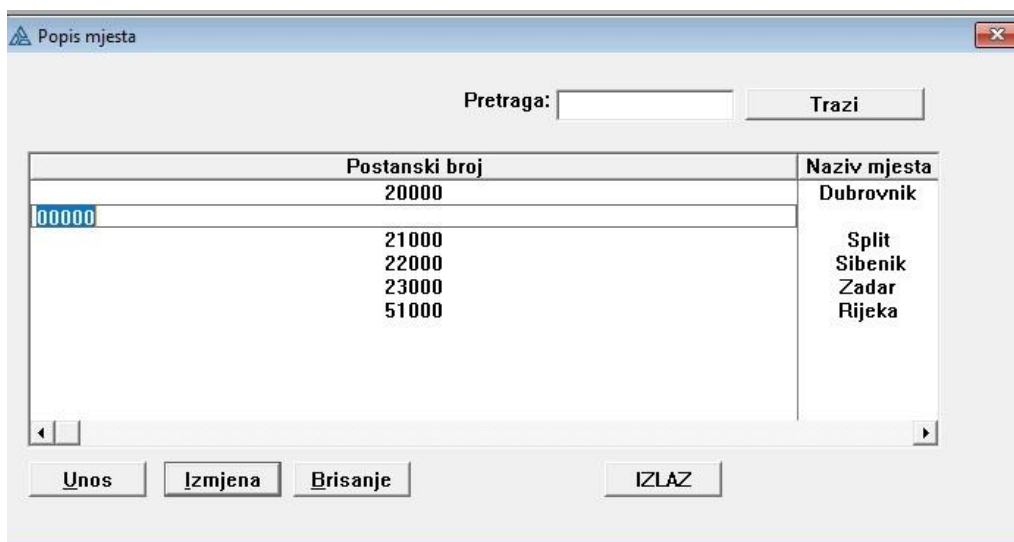


Slika 20 Postavljanje redoslijeda tipke TAB

Možemo vidjeti da u ovom redoslijedu, prvo ide tekst unosa, a zatim polje za unos. Ovaj redoslijed se ponavlja za svaki par teksta i polja za unos. Na kraju dolaze gumbovi.

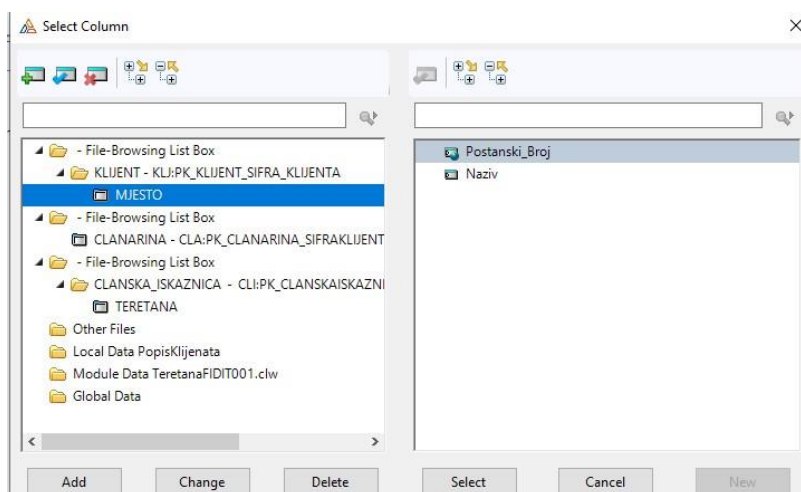
Primjećujemo da se u ovom definiranom rasporedu kretanja tipkom TAB prvo pojavljuje tekstualni opis, a zatim polje za unos podataka. Ovaj uzorak se ponavlja za svaki par teksta i odgovarajućeg polja za unos, dok se na kraju nalaze gumbi.

U proceduri PopisMjesta koristimo opciju „Use Edit in Place“, koja nam omogućava unos podataka na ekranu za popis. Ovdje nije potrebno otvarati dodatni ekran za ažuriranje podataka. Ovu opciju postavljamo samo za proceduru PopisMjesta, jer se ova opcija koristi samo kod jednostavnih tablica s jednim primarnim ključem i bez vanjskih ključeva, te također moraju imati samo „Entry“ vrstu polja za unos. Izgled unosa podataka s ovom opcijom prikazan je na sljedećoj slici.



Slika 21 Opcija Use Edit in Place

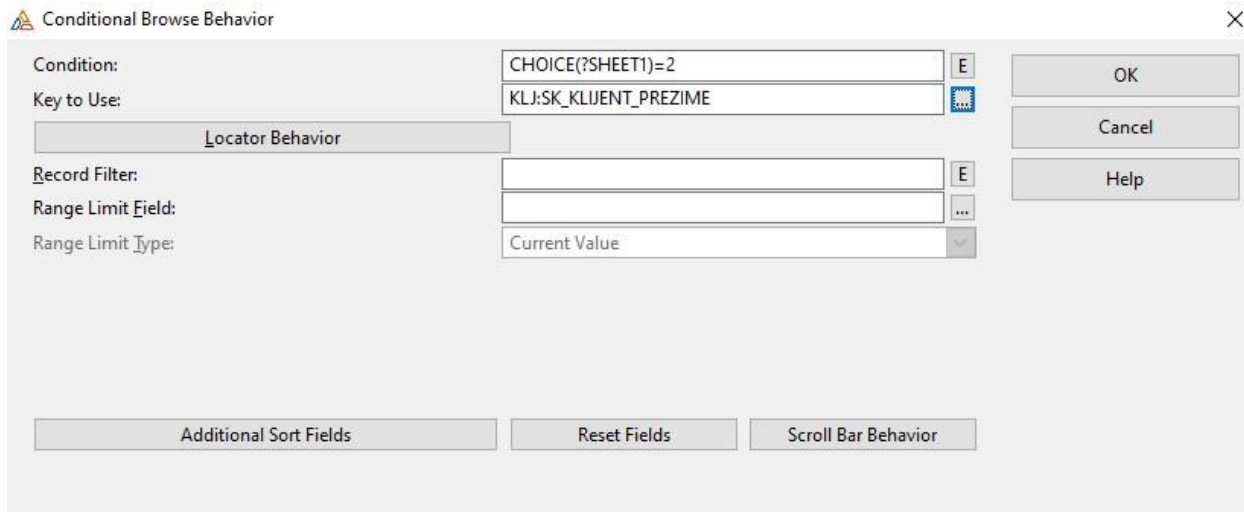
Neke tablice imaju vanjske ključeve, te je iz tog razloga potrebno dodavati stupce iz povezanih tablica u ekran za prikaz podataka kako da bi taj podatak bio bolje opisan. Na primjer, u našoj aplikaciji, tablica Klijent ima vanjski ključ koji sadrži poštanski broj. Međutim, bolje bi tu informaciju opisivao sam naziv mjesta koji pripada odgovarajućem poštanskom broju, umjesto samo poštanskog broja. Stupac dodajemo tako da prvo dodajemo povezanu tablicu kojoj taj stupac pripada. Kada dodamo potrebnu tablicu, možemo odabrati stupac čiji podatak želimo prikazati u povezanoj tablici. Na sljedećoj slici možemo vidjeti da je tablici Klijent dodana povezana tablica Mjesto te je odabran stupac „Naziv“ koji će se prikazivati umjesto stupca „poštanski broj“.



Slika 22 Odabir stupca Naziv

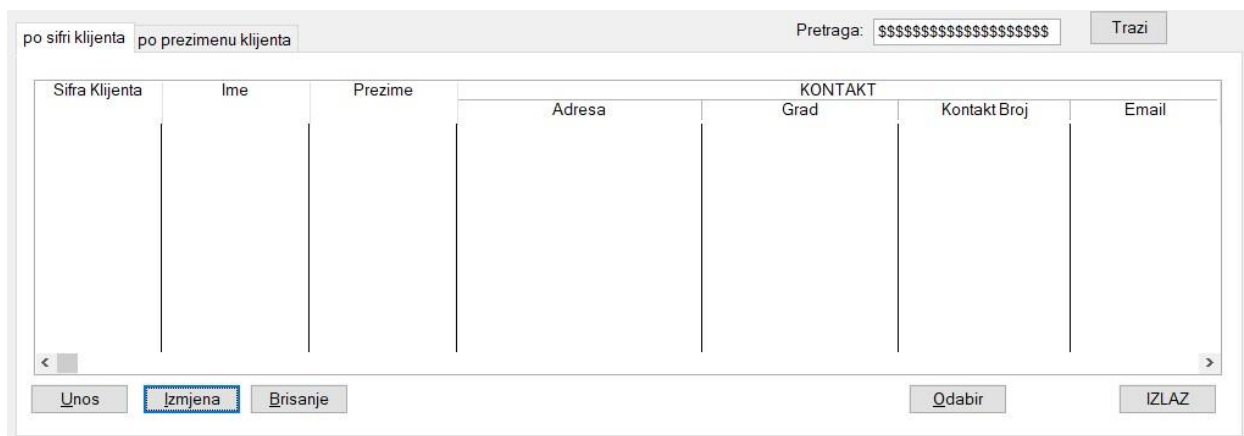
Prilikom izrade rječnika, uz potrebne primarne i vanjske ključeve definiramo i jedan ključ za sortiranje u tablici Klijent. Taj ključ omogućava sortiranje po prezimenu klijenta. Kako bi u aplikaciji omogućio sortiranje po prezimenu klijenta, na ekranu za popis klijenata moram omogućiti sortiranje pomoću kartica za sortiranje. Da bismo dodali kartice za sortiranje, na ekran povlačimo element „SHEET“ iznad željenog „listbox-a“. Element „SHEET“ donosi i prvu karticu

za sortiranje koja sortira podatke po primarnom ključu tablice. Kako bi se podatci zaista sortirali, definiramo akciju koja će se izvršavati pri odabiru kartice. Kako bismo omogućili sortiranje po prezimenu klijenta otvaramo karticu „Conditional Behavior“ u kojoj postavljamo uvjet sortiranja. Postavljanje uvjeta za sortiranje po prezimenu klijenta prikazano je na sljedećoj slici.



Slika 23 Uvjet sortiranja

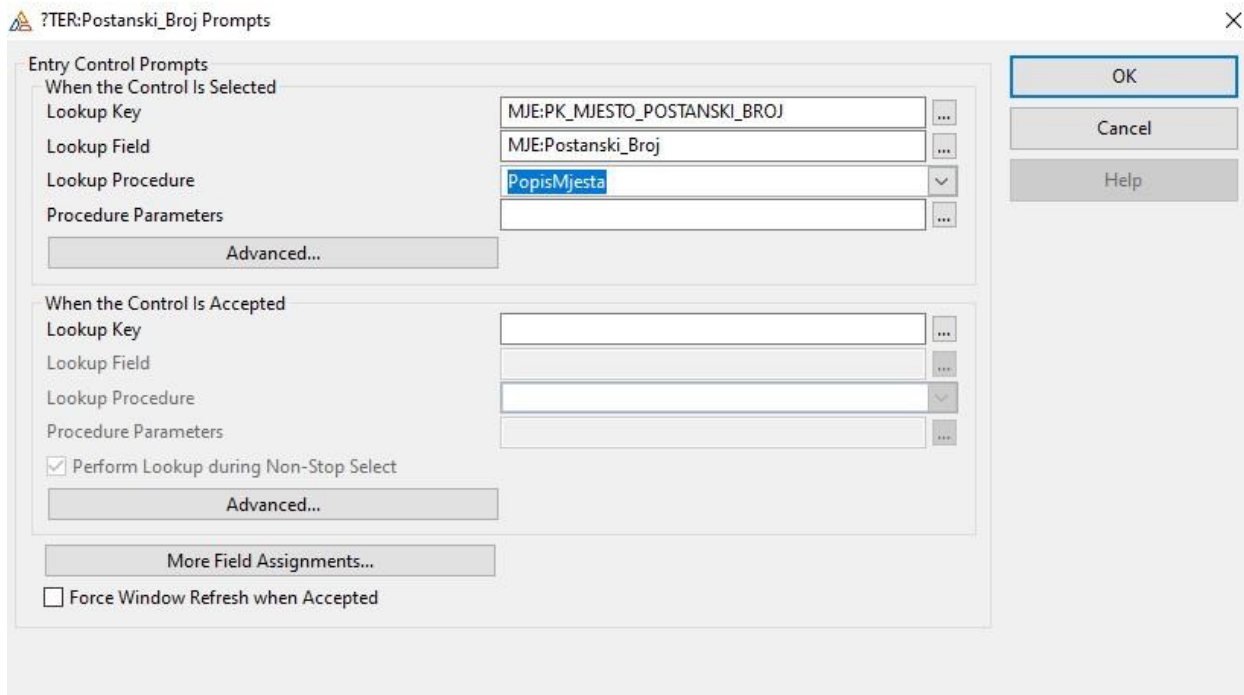
Ekran za popis klijenata s dodanim karticama za sortiranje prikazan je na sljedećoj slici.



Slika 24 Kartice za sortiranje

### Izrada procedura ažuriranja za složene tablice

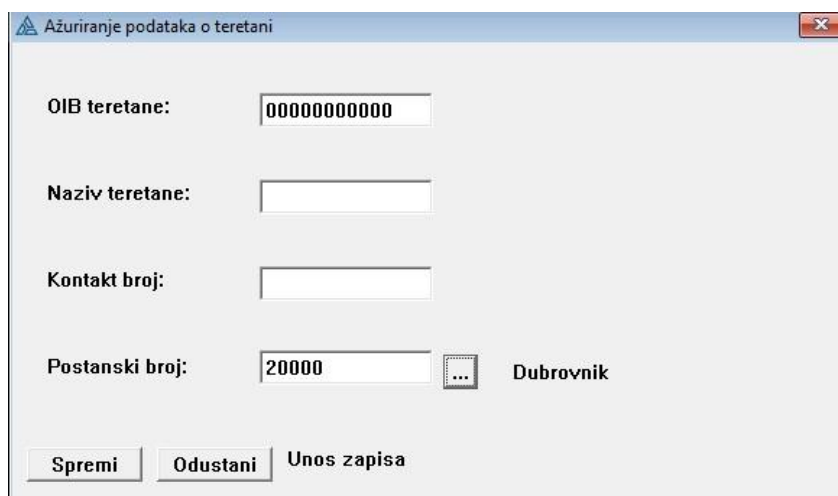
Kada izrađujemo procedure ažuriranja za tablice koje sadrže vanjske ključeve potrebno je postaviti opciju „Lookup“ za odabir podataka iz povezane tablice. Na ovaj način sprječavamo korisnika da unese proizvoljne podatke koji nisu točni i relevantni za našu povezanu tablicu. Na primjer, ovu opciju moramo postaviti u aplikaciji za ažuriranje podataka u teretani, gdje imamo polje za unos poštanskog broja. Postupak definiranja „Lookup“ polja vidimo na sljedećoj slici.



Slika 25 Postavljanje Lookup opcije

U polju „Lookup key“, bismo primarni ključ tablice iz koje želimo birati vrijednosti. U polju „Lookup field“, bismo polje iz primarnog ključa koje odgovara polju za koje postavljamo „Lookup“ opciju.

Uz polje za „Lookup“ opciju također dodajemo i dinamički tekst s ispisom vrijednosti. Na primjer, kada odaberemo željeni poštanski broj u „Lookup“ polju, pored polja za unos poštanskog broja ispisuje se naziv mjesta koje pripada izabranom poštanskom broju. Primjer ovoga možemo vidjeti na sljedećoj slici.



Slika 26 Ispis dinamičkog teksta

Primjećujemo da u polju za unos poštanskog broja odabran je poštanski broj 20000. Pored tog unosa, vidimo dinamički tekst koji ispisuje naziv grada čiji smo poštanski broj odabrali. Ovaj postupak ponavljamo u svim tablicama koje imaju vanjske ključeve, u agregacijama za polja koja čine primarni ključ agregacije, te na poljima koja predstavljaju primarni ključ slabog tipa entiteta.

### Pregled slabih tablica

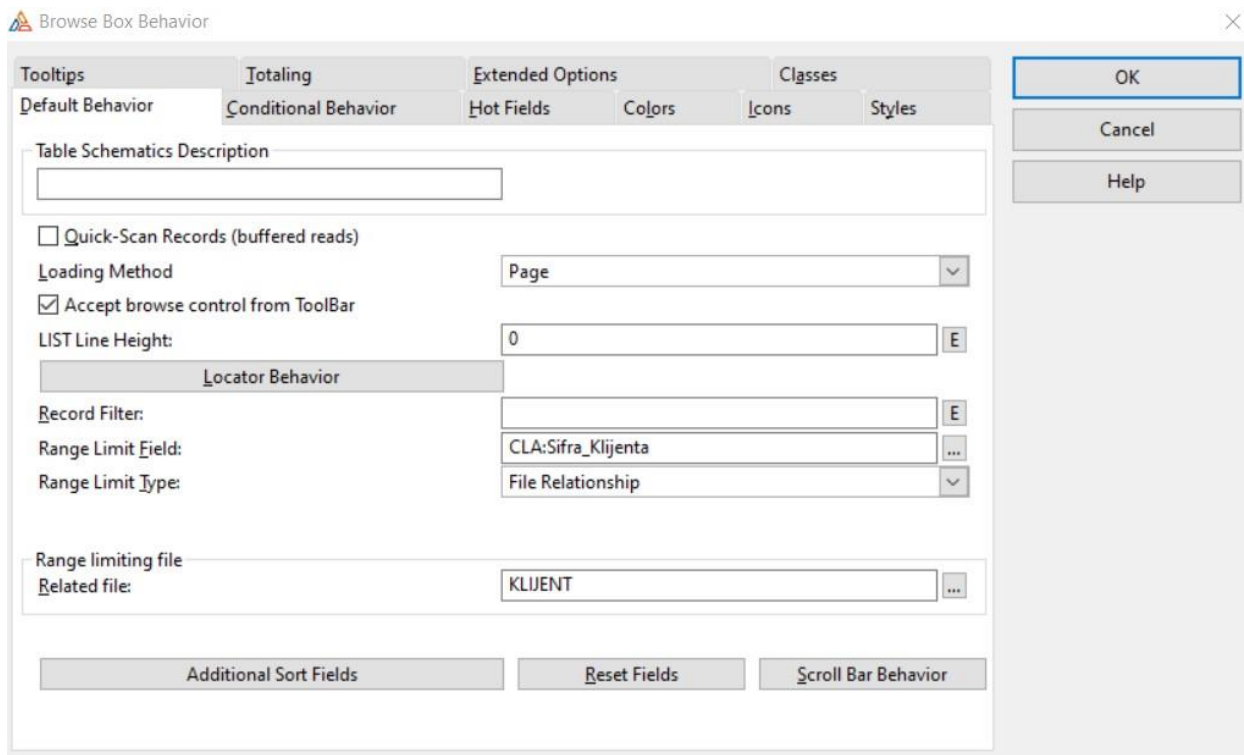
Nakon definiranja potrebnih procedura za jake tablice, nastavljamo s procedurama za pregled i ažuriranje slabih tipova entiteta. U našem definiranom modelu imamo četiri slabe tablice: ČlanskaIskaznica, Oprema, Članarina i Provodba\_treninga.

Na primjer, tablica Klijent ima dvije povezane slabe tablice, ČlanskaIskaznica i Članarina. Pregledu i ažuriranju slabih tablica ne može se pristupiti preko glavnog izbornika, nego im pristupamo preko ekrana njihove jake tablice. Konkretno, podatke ČlanskaIskaznica i Članarina pregledavamo i ažuriramo putem ekrana procedure PopisKlijenata.

### Procedura PopisOpreme

Procedure za popis i ažuriranje slabih tablica definiramo prateći prethodno opisane postupke za pregled jakih tablica. Jedna od takvih procedura je procedura koja nam omogućuje pregled članarina kupca. Popis podataka o članarinama nalazit će se na ekranu procedure PopisKlijenata. Korisnik će u tablici klijenata klikom odabrati jednog klijenta, te će se njegovi odgovarajući podatci o članarini prikazati u tablici članarina.

Prvo moramo dodati novi „listbox“ na ekran procedure PopisKlijenata. To postizemo tako da iz bloka „Control Templates“ povučemo element „BrowseBox“ na ekran. Nakon što smo dodali element „BrowseBox“, biramo tablicu čije podatke želimo prikazati te njen primarni ključ. Zatim dodajemo u "listbox" sve potrebne stupce čije podatke želimo prikazati. Dodatno, na ovaj "listbox" trebamo postaviti ograničenje kako bi se prikazivali samo podatci odabranog klijenta, umjesto svih podataka slabe tablice. Ove dvije tablice međusobno su povezane preko stupca ŠifraKlijenta. To znači da kada u tablici Klijent odaberemo jednog klijenta, prikazivat će se samo podatci o članarini koji imaju istu vrijednost šifre klijenta. Postupak postavljanja ograničenja ranga prikazan je na sljedećoj slici.



Slika 27 Ograničavanje ispisa

Nakon što smo definirali potrebne postavke za novi "listbox" element, moramo dodati i gumbе za ažuriranje podataka. To postizemo tako da iz bloka „Control Templates“ povučemo element „BrowseUpdateButtons“ na ekran. Nakon dodavanja ovih gumbova, odaberemo koja se procedura poziva njihovim klikom, što će biti odgovarajuća procedura ažuriranja. Ekran PopisKlijenata s dodanim promjenama je prikazan na sljedećoj slici.

po sifri klijenta | po prezimenu klijenta | Pretraga:  Trazi

Sifra Klijenta	Ime	Prezime	KONTAKT			
			Adresa	Grad	Kontakt Broj	Email
123	Lovro	Visic	Cvijete Z 11	Zadar	0981988287	visic@gmail.com
345	Duje	Dujic	Puntamika 15	Rijeka	0915321435	dujic@gmail.com
432	Mate	Matic	Augusta Senoe 8	Dubrovnik	0921435653	aticmate@gmail.co

Unos | Izmjena | Brisanje | IZLAZ

Informacije o članarini:

Sifra Članarine	Vrsta Članarine	Iznos Članarine	Datum Otvaranja	Datum Isteka
1	STUDENTSKA	35.00	5/07/2023	5/08/2023

Unos | Izmjena | Brisanje

Slika 28 Ekran PopisKlijenata

Prateći ovaj postupak, omogućen je pregled i ažuriranje podataka slabih tablica na njihovim odgovarajućim ekranima. Na ekranu PopisKlijenata uz opisan pregled Članarina također je omogućen i pregled članskih iskaznica.

#### Automatski izračuni u polju preko definirane formule

U izradi aplikacije naišao sam na funkcionalnosti koje sam morao ostvariti koristeći Clarionov-u „Embeds“ opciju. Ova opcija nam omogućava da pišemo vlastiti kod koji će se izvršiti u definiranom polju.

Ovu opciju koristim u ažuriranju članarine, gdje omogućujem da kada se odabere vrsta članarine, klikom na polje šifra članarine automatski bude odabrana i šifra članarine. Npr. ako u polju vrsta članarine odaberemo studentska članarina, šifra članarine će biti 1. To postizemo koristeći kod prikazan na slici putem opcije „Embeds“ nad poljem šifra članarine.



```

1  IF CLA:Vrsta_Clanarine = 'STUDENTSKA' THEN
2      CLA:Sifra_Clanarine = 1
3
4  ELSIF CLA:Vrsta_Clanarine = 'NO LIMIT' THEN
5      CLA:Sifra_Clanarine = 2
6  ELSIF CLA:Vrsta_Clanarine = '3 PUTA TJEDNO' THEN
7      CLA:Sifra_Clanarine = 3
8  ELSIF CLA:Vrsta_Clanarine = 'PRIVATNI TRENING' THEN
9      CLA:Sifra_Clanarine = 4
10 ELSE CLA:Sifra_Clanarine = 5
11
12
13 END

```

Slika 29 Automatski izračun šifre članarine

Drugi primjer korištenja koda je kod automatskog izračunavanja iznosa cijene članarine. Npr. ako korisnik odabere studentsku vrstu članarine, onda će klikom na polje za unos cijene članarine automatski će se upisati odgovarajuća cijena. Korišteni kod prikazan je na sljedećoj slici.

```

IF CLA:Vrsta_Clanarine = 'STUDENTSKA' THEN
    CLA:Iznos_Clanarine = 35

ELSIF CLA:Vrsta_Clanarine = 'NO LIMIT' THEN
    CLA:Iznos_Clanarine = 50
ELSIF CLA:Vrsta_Clanarine = '3 PUTA TJEDNO' THEN
    CLA:Iznos_Clanarine = 40
ELSIF CLA:Vrsta_Clanarine = 'PRIVATNI TRENING' THEN
    CLA:Iznos_Clanarine = 80
ELSE CLA:Iznos_Clanarine = 70

END

```

Slika 30 Automatski izračun cijene članarine

Još jedan primjer korištenja koda za automatski izračun je u ažuriranju članarine, nad poljem datuma isteka. Želimo da se to polje automatski popuni datumom koji je mjesec dana nakon datuma unesenog u polje datuma otvaranja. Korišteni kod prikazan je na sljedećoj slici.

```

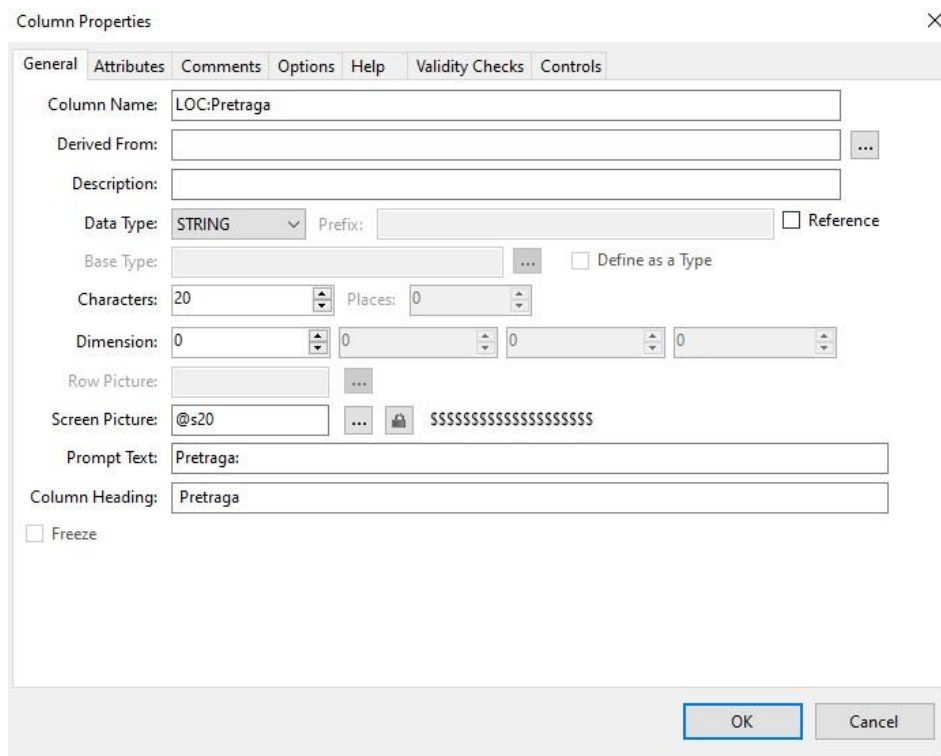
IF MONTH(CLA:Datum_Otvaranja) = 4 OR MONTH(CLA:Datum_Otvaranja) = 6 OR MONTH(CLA:Datum_Otvaranja) = 9 OR MONTH(CLA:Datum_Otvaranja) = 11 THEN
    CLA:Datum_Isteka = CLA:Datum_Otvaranja + 30
ELSIF (YEAR(CLA:Datum_Otvaranja) % 4 = 0 AND YEAR(CLA:Datum_Otvaranja) % 100 <> 0) OR YEAR(CLA:Datum_Otvaranja) % 400 = 0 THEN
    CLA:Datum_Isteka = CLA:Datum_Otvaranja + 29
ELSIF MONTH(CLA:Datum_Otvaranja) = 1 OR MONTH(CLA:Datum_Otvaranja) = 3 OR MONTH(CLA:Datum_Otvaranja) = 5 OR MONTH(CLA:Datum_Otvaranja) = 7
    OR MONTH(CLA:Datum_Otvaranja) = 8 OR MONTH(CLA:Datum_Otvaranja) = 10 OR MONTH(CLA:Datum_Otvaranja) = 12 THEN
    CLA:Datum_Isteka = CLA:Datum_Otvaranja + 31
ELSE
    CLA:Datum_Isteka = CLA:Datum_Otvaranja + 28
END

```

Slika 31 Automatski izračun datuma isteka

## Dodatne opcije

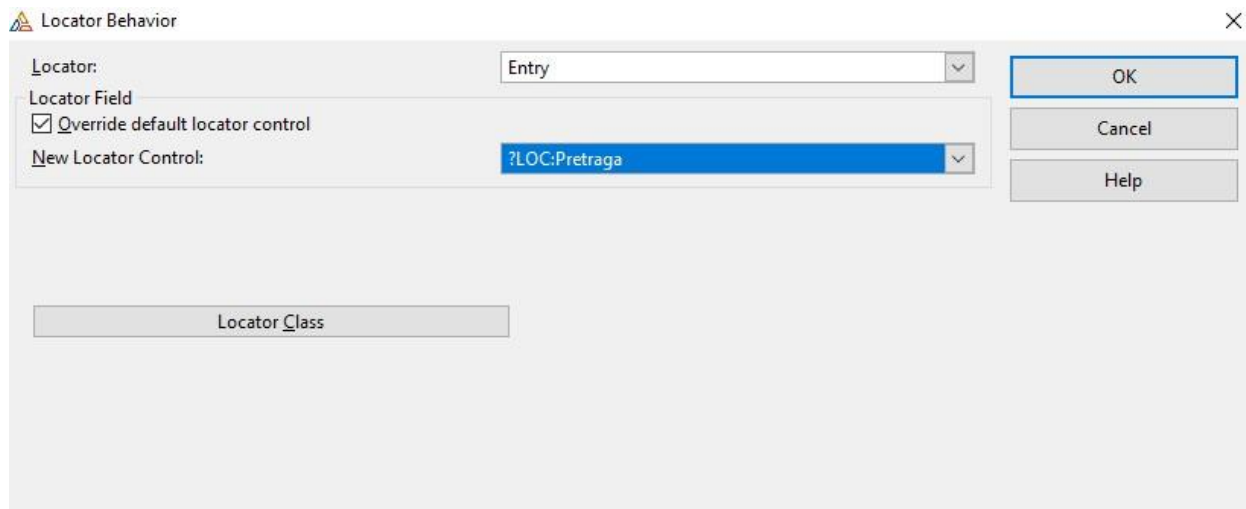
Na ekranima za prikaz podataka u aplikaciji, također omogućujemo i pretraživanje podataka. Pretraživanje se omogućuje po primarnom ključu ili, ako ga tablica ima, po sortirnom ključu. Kako bismo kreirali polje za pretragu, prvo kreiramo dodatno polje u rječniku podataka. Kreiranje novog polja prikazano je na sljedećoj slici.



Slika 32 Definiranje polja za pretragu

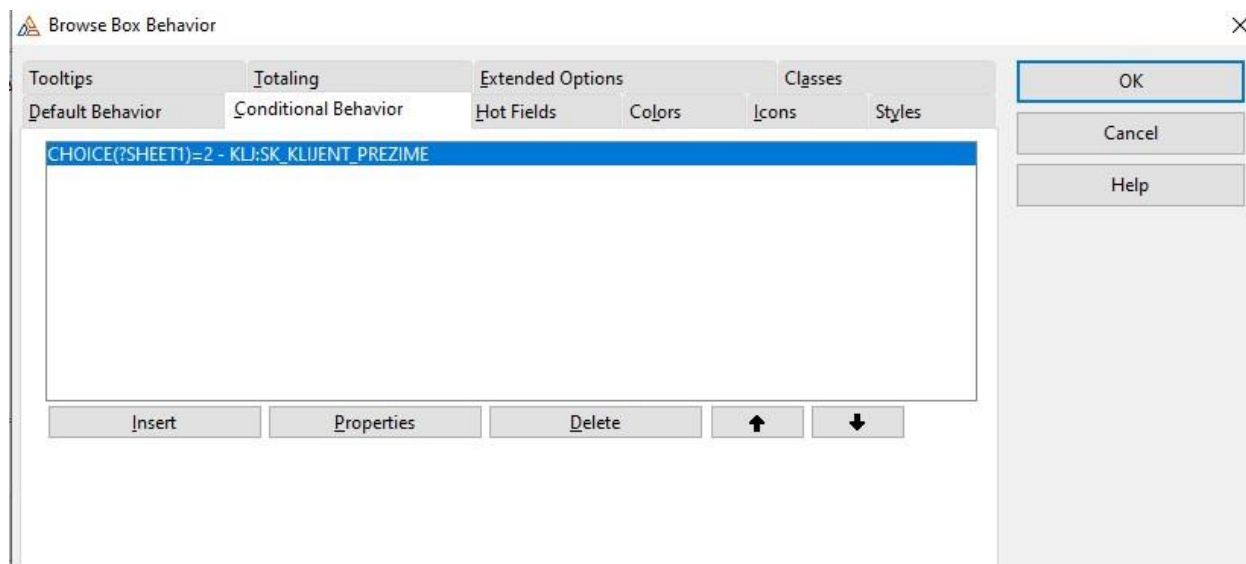
Nakon što kreiramo polje za pretragu, možemo ga dodati na ekran. Također, kada ga dodamo na ekran, moramo prilagoditi redoslijed u „Tab Order Assisstant“, tako da polje za pretragu bude postavljeno prije „listbox-a“ čije podatke pretražuje. Posljednji korak u postavljanju pretraživanja je postavljanje akcije za pretragu po svakoj kartici za sortiranje. To znači da trebamo povezati unesenu vrijednost u polju za pretragu s vrijednošću ključa prema kojem se sortira. Drugim riječima, pojedinačno povezujemo svaku karticu za sortiranje s poljem za pretragu.

Na primjer, „listbox“ na ekranu procedure PopisKlijenata ima dvije kartice za sortiranje. Postavljanje pretrage za prvu karticu, koja sortira po primarnom ključu prikazana je na sljedećoj slici.



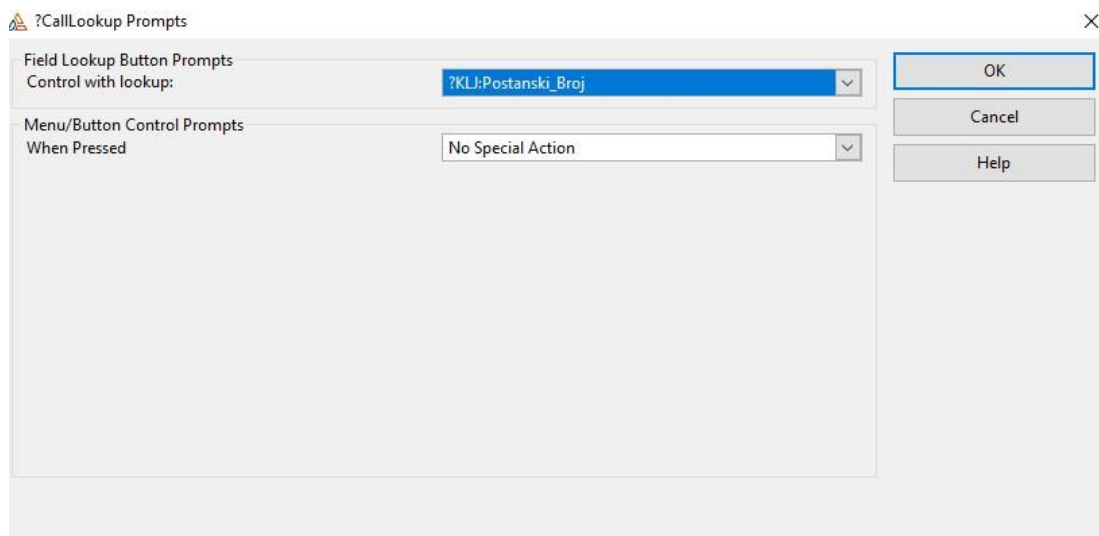
Slika 33 Definiranje prvog polja za pretragu

Sljedeća slika prikazuje postavljanje uvjeta za pretragu za drugu karticu koja pretražuje po sortirnom ključu.



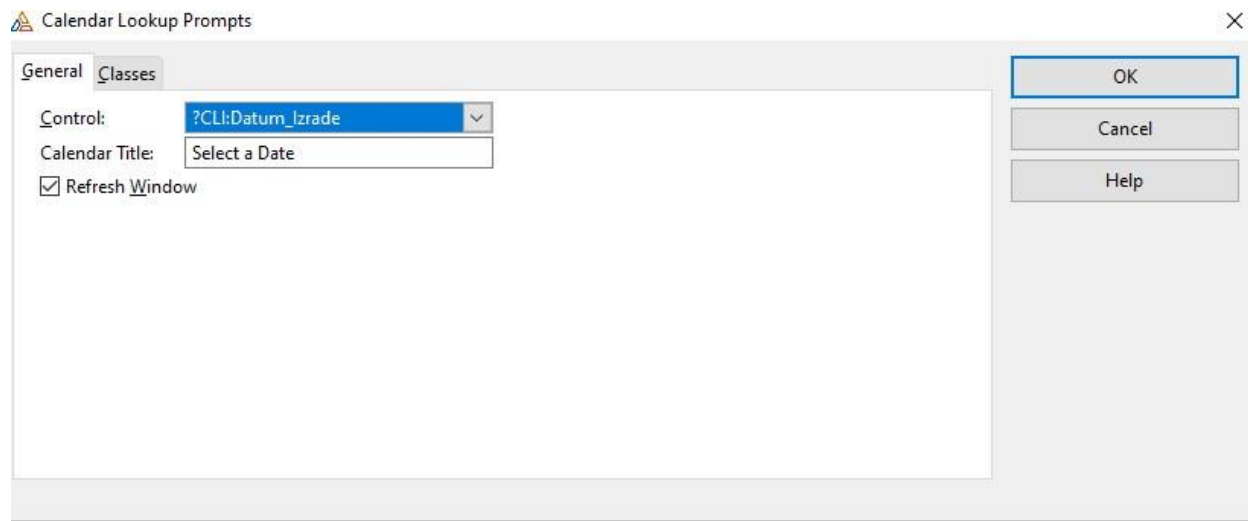
Slika 34 Postavljanje drugog polja za pretragu

Osim polja za pretragu, uz polja koja imaju definiranu „Lookup“ opciju, omogućujemo korištenje „Lookup“ opcije klikom na gumb koji se nalazi pored polja za unos. Kako bi dodali „Lookup“ gumb iz bloka „Control Templates“ koristimo opciju „FieldLookupButton“ te je povlačimo na ekran. Posljednji korak u postavljanju „Lookup“ gumb opcije je pridruživanje akcije koja se izvršava kada se klikne na gumb.



Slika 35 Definiranje Lookup gumba

Osim „Lookup“ gumba, uz polja za unos koji su tipa „Date“, dodajemo gumb za kalendar. Klikom na ovaj gumb otvara se kalendar te na njemu odabiremo datum koji želimo unijeti. Kako bismo dodali gumb za kalendar na ekran, iz bloka „ControlTemplates“ odabiremo opciju „CalendarButton“ i smještamo ga na ekran. Nakon dodavanja gumba za kalendar, potrebno je definirati u koje polje će se spremati datum kojeg odaberemo. Postupak definiranja polja u kojeg spremamo odabrani datum prikazan je na sljedećoj slici.

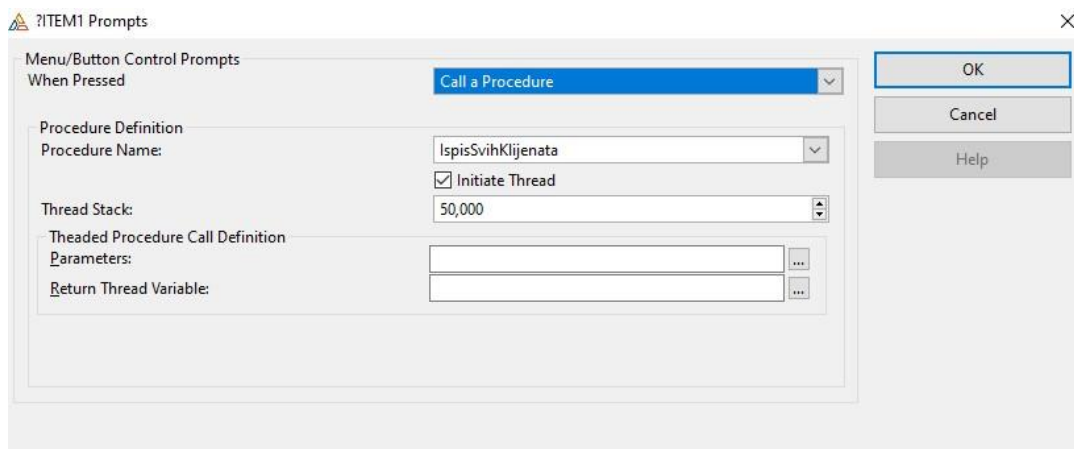


Slika 36 Definiranje gumba za kalendar

## Izrada izvještaja

Izvještaje koristimo kako bismo ispisali željene podatke iz aplikacije. U ovoj aplikaciji izradili smo dva jednostavna izvještaja, jedan izvještaj o jednome zapisu, te jedan složeni izvještaj. Jednostavnim i složenim izvještajima pristupamo putem izbornika aplikacije, dokle izvještaju o jednome zapisu pristupamo putem ekrana čiji redak ispisujemo klikom na gumb. Prije nego izradimo izvještaje, potrebno je izraditi izbornik preko kojeg ćemo pristupati izvještajima. Na

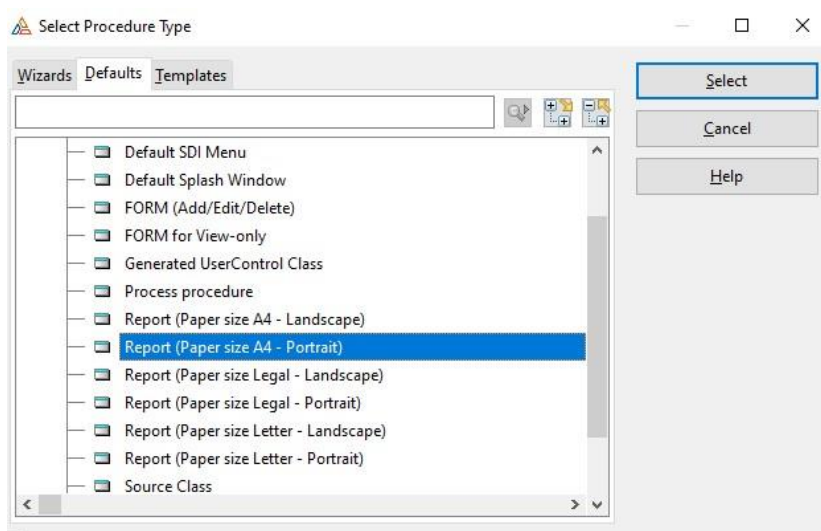
traku izbornika dodajemo novi izbornik „Ispis“ te mu dodajemo potrebne elemente. Nakon kreiranja novih elemenata, postavljamo im procedure koje se izvršavaju njihovim klikom. Primjer definirane procedure prikazan je na sljedećoj slici.



Slika 37 Definiranje procedure izvještaja

### Jednostavni izvještaj

Jednostavni izvještaj koristimo kada želimo ispisati sve retke jedne tablice. U ovom slučaju, koristimo ga kako bi ispisali sve retke tablice Oprema. Kako bi kreirali ovaj izvještaj, izaberemo odgovarajuću proceduru te za nju i željeni predložak. U ovoj aplikaciji za izvještaje koristimo predloške „Paper size A4 – Portrait“ i „Paper size A4 – Landscape“.



Slika 38 Odabir procedure za izvještaj

Zatim, klikom na gumb „Report“ ulazimo u uređivač izvještaja. U uređivanju izvješća u „Header“ dio dodajemo potrebne elemente, kao što su datum i vrijeme učitavanja izvješća te naslov. U dijelu „Detail“ dodajemo podatke iz tablice čiji izvještaj izrađujemo. Kako bismo mogli dodavati podatke tablice u izvještaj, prvo moramo dodati odgovarajuću tablicu u „Data/Tables“ bloku, te također nakon dodavanja tablice dodajemo joj i njen primarni ključ. Nakon što smo dodali potrebne

podatke i ostale elemente, završavamo izradu jednostavnog izvještaja za popis opreme. Izgled izvještaja je prikazan na sljedećoj slici.

Page Header

# Popis opreme

Datum: <-- Date Stamp -->  
Vrijeme: <-- Time Stamp -->

Detail (Detail)

Sifra opreme: \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$      Naziv sprave: \$  
Proizvodac: \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$      Kolicina: -<#

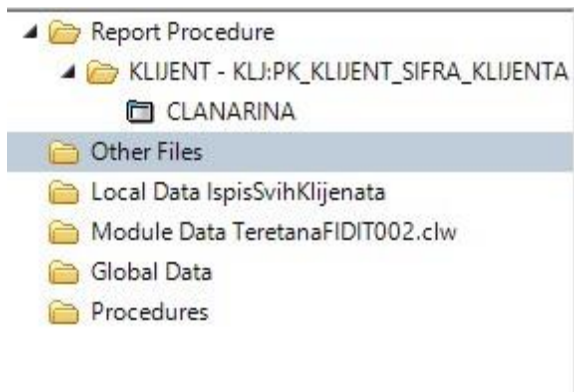
Page Footer

Stranica: <<#

Report Form

Slika 39 Izvještaj popisa opreme

Također, izrađujemo i drugi jednostavni izvještaj, ali u ovom izvještaju dodajemo podatke iz druge povezane tablice. Ispisujemo podatke tablice Klijent zajedno sa podacima njene slabe tablice Članarina. Za ovaj izvještaj pratimo isti postupak kao i za prethodni izvještaj, ali u ovom slučaju moramo u „Data/Tables“ bloku osim glavne tablice dodati i željene povezane tablice. Na dodane tablice ne postavljamo primarni ključ, već samo glavna tablica ima pridruženi primarni ključ. Hijerarhija tablica za ovaj izvještaj prikazana je na sljedećoj slici.



Slika 40 Hijerarhija tablica izvještaja klijenata

Nakon dodavanja potrebnih podataka te ostalih elemenata, dobijemo izvještaj prikazan na sljedećoj slici.

## Popis svih klijenata

Datum: <-- Date Stamp -->  
Vrijeme: <-- Time Stamp -->


---

Detail (Detail)

Sifra klijenta:	<<#	E-mail:	\$
Ime:	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	Telefon:	\$\$\$\$\$\$\$\$
Prezime:	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	Vrsta članarine:	\$
Adresa:	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	Datum otvaranja:	<##/####
Postanski broj:	####	Datum isteka:	<##/####

---

Page Footer

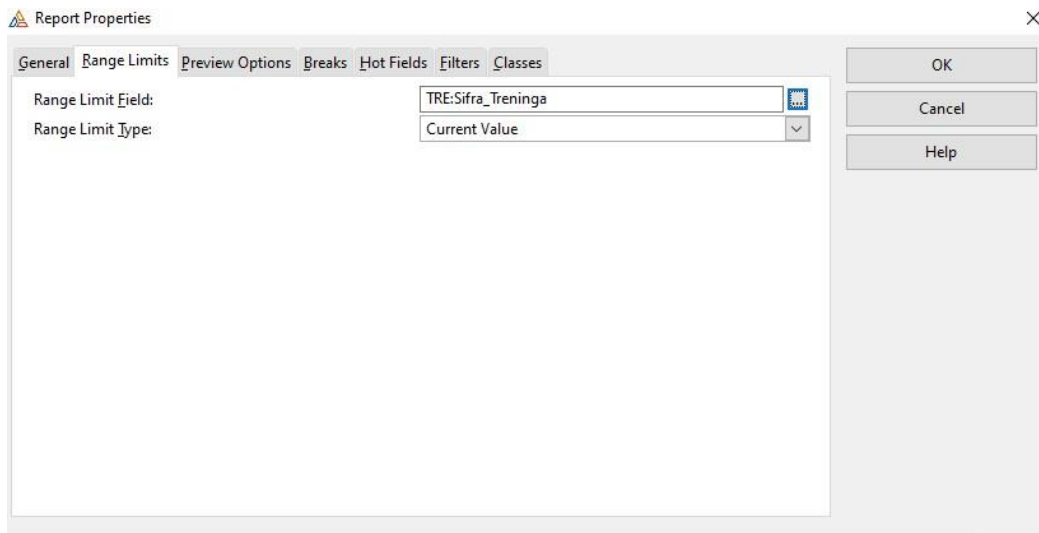


Slika 41 Izvještaj o klijentima

### Izvještaj o jednome zapisu

Izvještaj o jednome zapisu koristi se kako bi se ispisao jedan redak odabrane tablice za koju smo izradili izvještaj. U našoj aplikaciji, ovaj izvještaj je izrađen za tablicu Treening i koristi se za ispisivanje podataka pojedinog odabranog treeninga. Na ekranu procedure PopisTreeninga dodajemo gumb koji će omogućiti pristup izvještaju odabranog treeninga. Za dodavanje potrebnog gumba koristimo opciju „BrowsePrintButton“ iz bloka „Control Templates“. Novo dodanom gumbu definiramo i odgovarajuću proceduru koja se izvršava njegovim klikom.

Izvještaj ovog zapisa uredimo prateći prethodno opisane korake kreiranja i uređivanja izvještaja. Kod ovog izvještaja, želimo da se ispisuje samo odabrani treening iz tablice, umjesto svih treeninga koji su upisani u tablicu. Kako bismo to postigli, postavljamo ograničenje ispisa podataka u ovome izvještaju, tako da se ispiše samo označeni redak tablice. Postavljeno ograničenje prikazano je na sljedećoj slici.



Slika 42 Ograničenje ispisa podataka

Ovime završavamo izradu o jednome zapisu, te njegov izgled možemo vidjeti na sljedećoj slici.

## Informacije o treningu

Datum: 28.08.2023

Vrijeme: 16:10

**Sifra treninga:** 1      **Vrsta treninga** GRUPNI

**Opis treninga:** Grupni kardio trening

Slika 43 Izvještaj o jednome zapisu

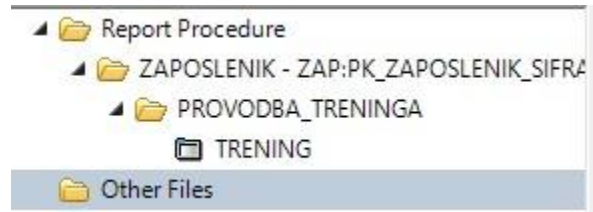
### Složeni izvještaj

Složeni izvještaj koristimo kako bismo ispisali podatke triju hijerarhijski povezanih tablica. U složenom izvještaju podatci u dijelu „Detail“ ovise o vrijednosti primarnog ključa druge povezane tablice. Drugim riječima, kada se taj podatak promijeni, promijenit će se i podatci u „Detail“ sekciji.

U ovoj aplikaciji, složeni izvještaj ispisuje podatke o treningu. Prikazani su odabrani podatci tablice `Trening` i `Provodba_treninga`, te se ti odabrani podatci grupiraju po primarnom ključu tablice `Zaposlenik`. Ovime ćemo dobiti informacije o svim treninzima koje provode zaposlenici teretane, te ćemo moći pregledati za koje treninge je svaki zaposlenik zadužen.

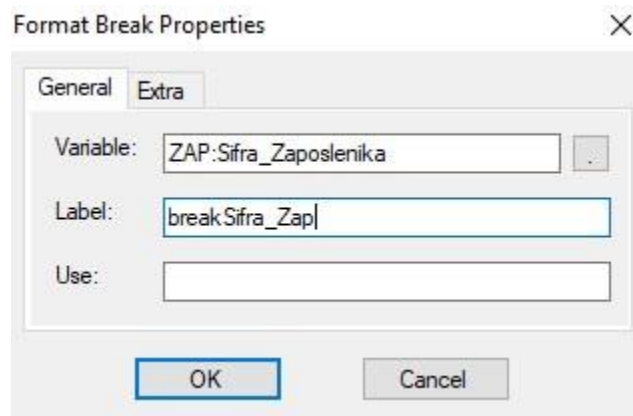


Da bismo izradili ovaj izvještaj, pratimo prethodno opisane korake te definiramo potrebnu proceduru za ispis te definiramo hijerarhiju tablica u izvještaju. U blok "Data/Tables", prva tablica koju dodajemo je ona po čijim podacima grupiramo ostale podatke. Hijerarhija tablica prikazana je na sljedećoj slici.



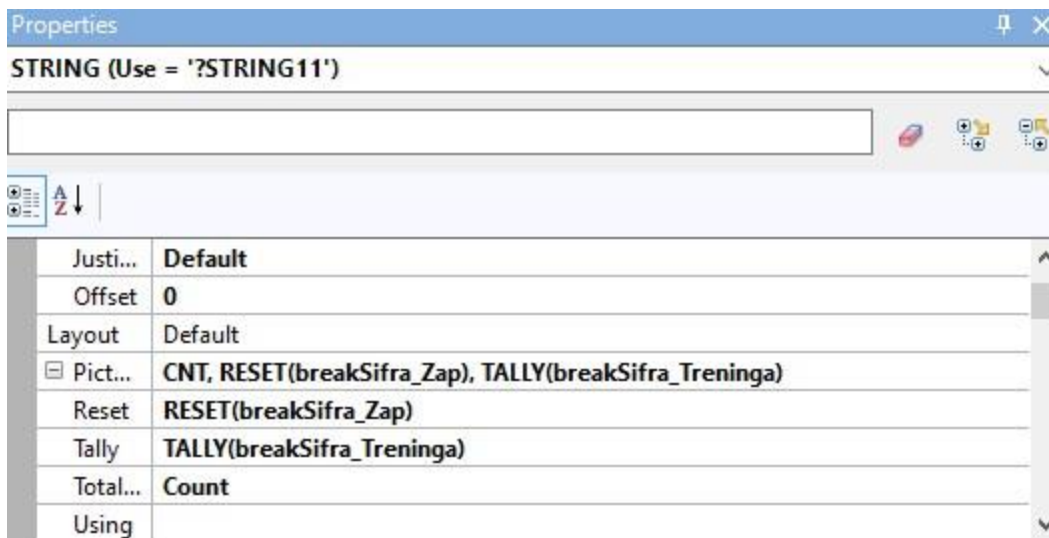
Slika 44 Hijerarhija tablica složenog izvještaja

Kod složenog izvještaja, važno je napraviti prijelomne stranice. Prijelomne stranice koristimo zbog ovisnosti podataka o primarnom ključu glavne tablice. U našem slučaju, za svakog trenera se ispisuju podatci o treninzima koje provodi, te o njegovoj šifri ovisi i mijenjanje podataka, stoga se prijelom stranica postavlja na temelju tog polja. Postavljanje prijelomne stranice prikazano je na sljedećoj slici.



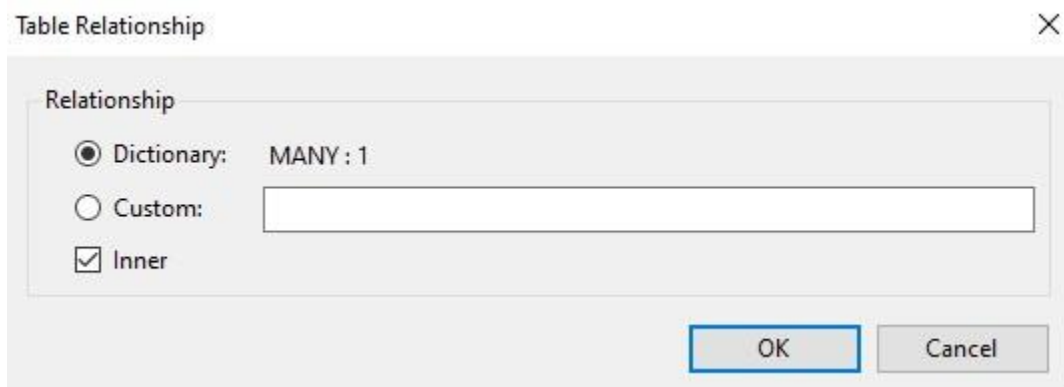
Slika 45 Prijelom stranice

Nakon dodavanja ovog prijeloma, slijedi dodavanje grupnih zaglavlja i podnožja. Dodajemo grupno zaglavlje i podnožje za tablicu Zaposlenik po prijelomu ZAP:Sifra\_Zaposlenika. U grupnom zaglavlju tablice Zaposlenik ispisuju se odabrani podatci iz tablice. U „Detail“ dijelu izvještaja ispisuju se odabrani podaci tablica Trening i Provodba\_treninga, koji se mijenjaju ovisno o šifri zaposlenika. U grupnom podnožju tablice Zaposlenik ispisivat će se ukupan broj nadolazećih treninga za koje je zaposlenik zadužen. Kako bismo ispisali ukupan broj treninga za koje je zaposlenik zadužen, u bloku „Properties“ dodanog „string“ elementa moramo postaviti odgovarajuće opcije, koje su prikazane na sljedećoj slici.



Slika 46 Postavljanje prebrojavanja

Kod složenog izvještaja također trebamo spriječiti ispis nepostojećih podataka. Na primjer, u našem slučaju ne želimo da se ispišu podaci o zaposlenicima koji nisu zaduženi za provođenje nadolazećih treninga. Postavljanje odgovarajuće opcije prikazano je na sljedećoj slici.



Slika 47 Opcija za sprječavanje ispisa nepostojećih podataka

Ovime smo postavili sve potrebne opcije složenog izvještaja, te možemo vidjeti njegov ispis prikazan na sljedećim slikama.

# Popis treninga po zaposleniku

Zaposlenik:

Sifra: 12

Ime: Pero

Prezime: Peric

---

Sifra treninga: 1

Vrsta treninga GRUPNI

Pocetak: 20:00

Zavrsetak: 21:00

Datum: 3/09/2023

Opis treninga: Grupni kardio trening

Sifra treninga: 123

Vrsta treninga GRUPNI

Pocetak: 21:00

Zavrsetak: 22:00

Datum: 20/09/2023

Opis treninga: Trening za ruke



Slika 48 Složeni izvještaj 1

# Popis treninga po zaposleniku

Sifra treninga: 72

Vrsta treninga GRUPNI

Pocetak: 8:00

Zavrsetak: 10:00

Datum: 29/09/2023

Opis treninga: Pilates

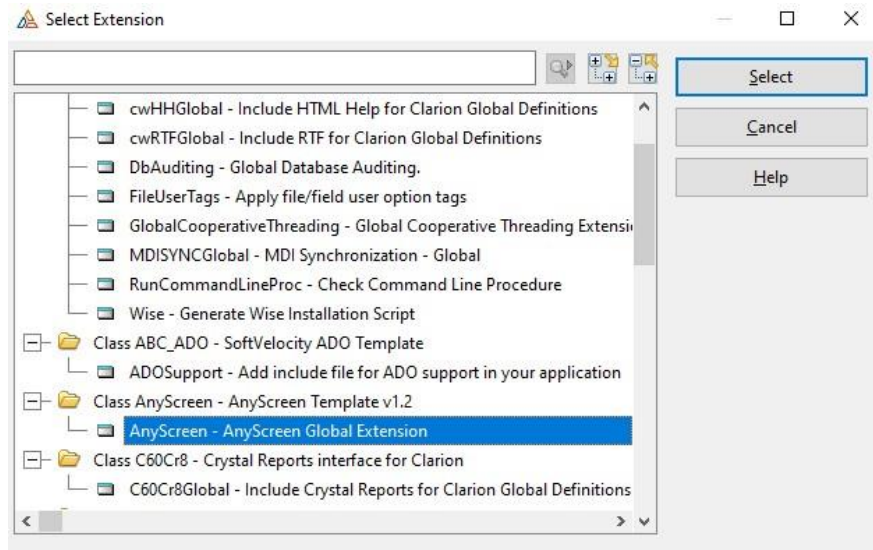


Ukupno treninga: 3

*Slika 49 Složeni izvještaj 2*

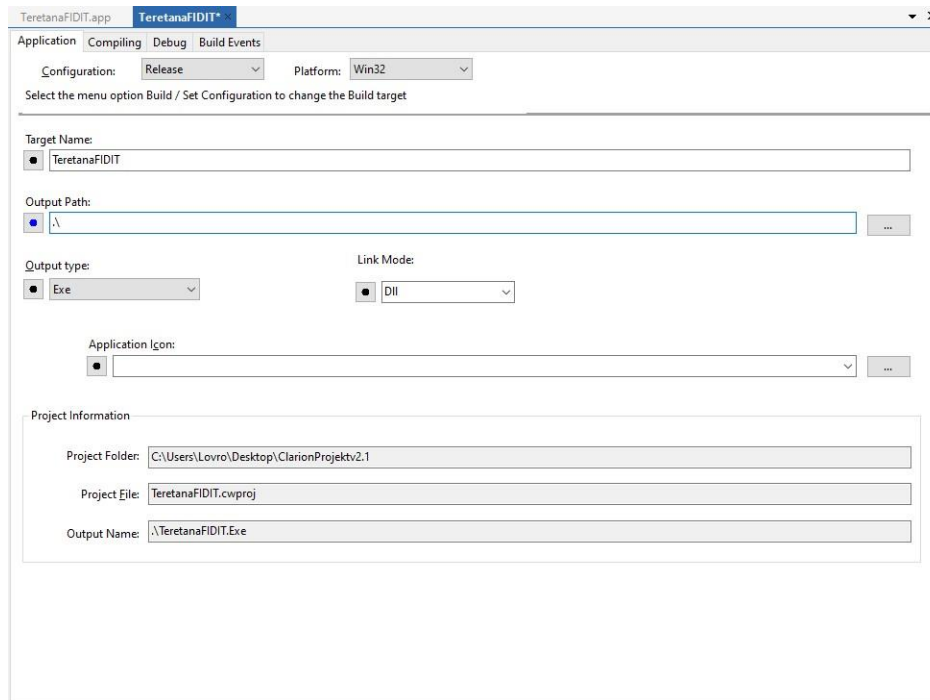
## Postavljanje aplikacije na server

Kako bi postavili aplikaciju na server te je pokretali u web pregledniku prvo moramo postaviti odgovarajuće opcije u Clarion aplikaciji. U Clarion-u je potrebno odabrati karticu „Global Extensions“, te potom opciju „Insert“. Otvara nam se prozor u kojem odabiremo odgovarajuću sliku, postupak možemo vidjeti na sljedećoj slici.



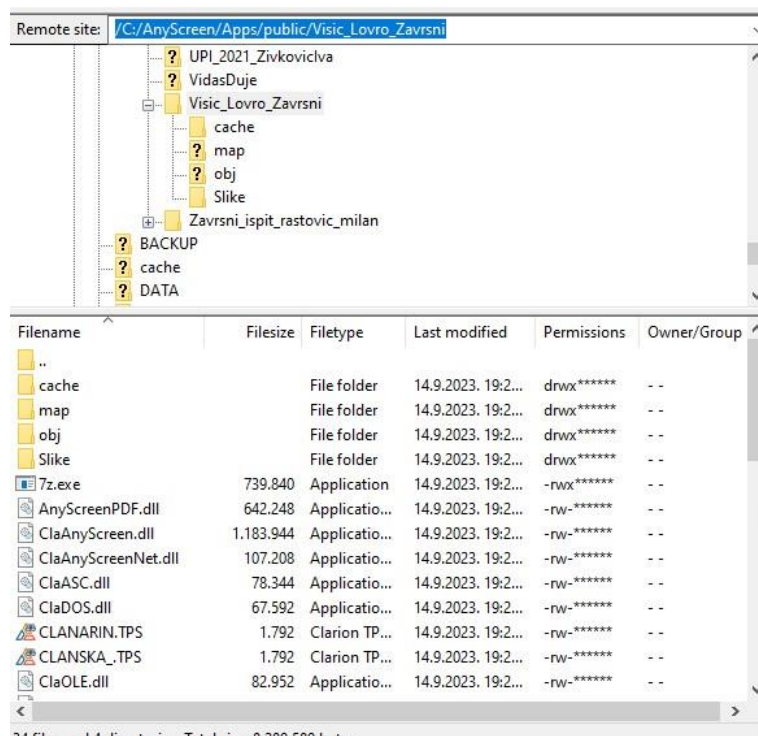
Slika 50 Opcija AnyScreen

Nakon odabira odgovarajuće opcije, odabiremo karticu „Solution Explorer“, u kojoj postavljamo „OutputPath“ na „\“. Postupak postavljanja opcije možemo vidjeti na sljedećoj slici.



Slika 51 Postavljanje opcije OutputPath

Nadalje, kako bi postavili aplikaciju na server koristimo program FileZilla. U programu FileZilla, prvo je potrebno unijeti pripadajuće podatke, kao što su „Host“, korisničko ime i lozinka, kako bi se spojili na server. Nakon spajanja na server, u odgovarajućem direktoriju kreiramo mapu, u koju zatim kopiramo sve datoteke i direktorije kreirane aplikacije. Postavljene datoteke aplikacije u programu FileZilla možemo vidjeti na sljedećoj slici.



Slika 52 FileZilla datoteke

Aplikaciji pristupamo putem sljedećeg linka:

[http://anahita.uniri.hr/public/Visic\\_Lovro\\_Zavrzni/TeretanaFIDIT](http://anahita.uniri.hr/public/Visic_Lovro_Zavrzni/TeretanaFIDIT)

## Zaključak

Pri razvoju aplikacije za upravljanje teretanom prošlo se kroz razne faze izrade aplikacije. Prvi korak bio je definiranje funkcionalnosti aplikacije, koristeći persone i korisničke priče. Potom se nastavlja s izradom modela podataka, te relacijske sheme na temelju koje se kasnije izrađuje rječnik aplikacije. Pri izradi aplikacije upoznao sam se s raznim funkcionalnostima alata Clarion, koje su mi omogućile razvijanje značajki aplikacije. Clarion ima razne mogućnosti unutar svog okruženja, te mislim da je koristan i efektivan alat za razvijanje administrativnih aplikacija, gdje je fokus rad s podacima.

## Literatura

- Cohn, M. (Pristupljeno: 10. 07 2023). *Mountain Goat Software*. Dohvaćeno iz User Stories: <https://www.mountaingoatsoftware.com/agile/user-stories>
- Čandrić, S. (2023). *Materijali s kolegija Dizajn korisničkog sučelja i iskustva*.
- Jakšić, D., & Pošćić, P. (2020). *Clarion II - Generator aplikacija*. Rijeka.
- Jakšić, D., & Pošćić, P. (2020). *Clarion II - Uređivač rječnika*. Rijeka.
- Pavlić, M. (2011). *Oblikovavne baza podataka*. Rijeka.
- Rehkopf, M. (Pristupljeno: 31. 08 2023). *Atlassian*. Dohvaćeno iz User stories with examples and a template: <https://www.atlassian.com/agile/project-management/user-stories>

## Popis slika

Slika 1- Persona Marko Marinović .....	2
Slika 2- Persona Ana Janjić.....	3
Slika 3- Model EV .....	7
Slika 4 Skica početnog ekrana .....	10
Slika 5 Skica ekrana za popis.....	11
Slika 6 Skica ekrana za ažuriranje .....	11
Slika 7 Kreiranje tablice Članarina .....	12
Slika 8 Kreiranje stupca tablice .....	13
Slika 9 Definiranje primarnog ključa.....	14
Slika 10 Definiranje veze između tablica .....	15
Slika 11- Definiranje referencijalnog integriteta .....	16
Slika 12 Definiranje izbornika aplikacije .....	17
Slika 13 Definiranje Popis procedure .....	18
Slika 14 Procedura ažuriranja .....	18
Slika 15 Odabir tablice.....	19
Slika 16 Odabir primarnog ključa.....	19
Slika 17 Dodavanje stupaca u listbox .....	20
Slika 18 Postavljanje sistemskih poruka.....	20
Slika 19 Ekran ažuriranja podataka .....	21
Slika 20 Postavljanje redosljeda tipke TAB.....	21
Slika 21 Opcija Use Edit in Place .....	22
Slika 22 Odabir stupca Naziv .....	22
Slika 23 Uvjet sortiranja .....	23
Slika 24 Kartice za sortiranje.....	23
Slika 25 Postavljanje Lookup opcije.....	24
Slika 26 Ispis dinamičkog teksta.....	24



Slika 27 Ograničavanje ispisa.....	26
Slika 28 Ekran PopisKlijenata .....	27
Slika 29 Automatski izračun šifre članarine .....	28
Slika 30 Automatski izračun cijene članarine.....	28
Slika 31 Automatski izračun datuma isteka .....	28
Slika 32 Definiranje polja za pretragu .....	29
Slika 33 Definiranje prvog polja za pretragu.....	30
Slika 34 Postavljanje drugog polja za pretragu.....	30
Slika 35 Definiranje Lookup gumba.....	31
Slika 36 Definiranje gumba za kalendar .....	31
Slika 37 Definiranje procedure izvještaja .....	32
Slika 38 Odabir procedure za izvještaj .....	32
Slika 39 Izvještaj popisa opreme .....	33
Slika 40 Hijerarhija tablica izvještaja klijenata .....	33
Slika 41 Izvještaj o klijentima.....	34
Slika 42 Ograničenje ispisa podataka .....	35
Slika 43 Izvještaj o jednome zapisu.....	35
Slika 44 Hijerarhija tablica složenog izvještaja .....	36
Slika 45 Prijelom stranice .....	36
Slika 46 Postavljanje prebrojavanja.....	37
Slika 47 Opcija za sprječavanje ispisa nepostojećih podataka .....	37
Slika 48 Složeni izvještaj 1 .....	38
Slika 49 Složeni izvještaj 2 .....	39
Slika 50 Opcija AnyScreen .....	40
Slika 51 Postavljanje opcije OutputPath.....	40
Slika 52 FileZilla datoteke .....	41