SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Hrvoje Kurilić, 39121/10-R

Tihana Lapaš, 39060/10-R

Tina Mehanović, 39072/10-R

Branimir Štivić, 39078/10-R

Davor Vučina, 39138/10-R

Projekt iz kolegija Programsko inženjerstvo

Kino.NET – Aplikacija za prodaju kino ulaznica

Projektna dokumentacija

Nositelj kolegija:

**Prof. dr. sc. Vjeran Strahonja**

Asistenti:

**Ivan Švogor, mag. inf.**

**Marko Mijač, mag. inf.**

**Zlatko Stapić, dipl. inf.**

**Boris Tomaš, dipl. inf**.

Varaždin, svibanj 2013.

Sadržaj

[1 Uvod 1](#_Toc359967029)

[2 Korištene tehnologije i alati 2](#_Toc359967030)

[3 Dijagrami 4](#_Toc359967031)

[3.1 UML dijagrami 4](#_Toc359967032)

[3.1.1 Dijagram slučajeva korištenja 4](#_Toc359967033)

[3.1.2 Dijagram slijeda 10](#_Toc359967034)

[3.1.3 Dijagram aktivnosti 17](#_Toc359967035)

[3.2 Dijagram klasa 23](#_Toc359967036)

[3.3 ERA model 25](#_Toc359967037)

[3.4 Relacijski model podatka 27](#_Toc359967038)

[4 Baza podataka 28](#_Toc359967039)

[4.1 SQL kod 28](#_Toc359967040)

[4.2 Unos podataka 32](#_Toc359967041)

[Slike 35](#_Toc359967042)

# Uvod

U ovom dokumentu prikazat ćemo tehničku dokumentaciju vezanu za programski proizvod koji, kako smo to već i u projektnoj dokumentaciji naveli, pomaže pri radu kina.

Ukratko da ponovimo, naš programski proizvod ***Kino.NET*** poslovanju kina pomaže na način da omogućuje lakšu i bržu prodaju ulaznica i izdavanje računa, zaposlenicima (administratoru) olakšava administriranje podataka o zaposlenicima, filmovima i dvoranama, te daje mogućnost vođenja statistike o posjećenosti pojedine projekcije na dnevnoj/tjednoj/mjesečnoj bazi.

U prvom dijelu ovog rada navest ćemo i ukratko nešto reći o samim **tehnologijama** koje smo koristili pri izradi ovog projektnog zadatka.

Drugi dio tehničke dokumentacije odnosi se na dijagrame. U ovom dijelu ćemo prvo prikazati **UML dijagrame:** dijagrami slučajeva korištenja - *Use case dijagrami*, dijagrami slijeda - *Sequence dijagrami*, dijagrami aktivnosti - *Activity dijagrami*, dijagram klasa– *Class dijagram,* a nakon njih ćemo prikazati **ERA model.**

Konačno, u trećem dijelu tehničke dokumentacije prikazat ćemo generiranu **SQL skriptu** koja omogućuje izradu baze podataka. ERA model i SQL skriptu smo izradili pomoću alata ***Toad Data Modeler v3.6.6.*** Za unos podataka u bazu i njen izvoz koristili smo alat ***SQLite Expert Personal v3.4.61.***

.

# Korištene tehnologije i alati

Pri izradi projekta koristili smo nekoliko programskih alata. U nastavku se nalazi popis korištenih alata, te kratko objašnjenje za što smo koji od alata konkretno upotrebljavali.

1. Microsoft Project 2007
2. Microsoft Office 2010
3. Visual Paradigm for UML CE 10.1
4. Visual Studio 2012
5. MS .NET (aplikacija)
6. Toad Data Modeler v3.6.6
7. SQLite Expert Personal v3.4.61

**Microsoft Project 2007**

Microsoft Project 2007 koristi se za planiranje zadataka, određivanje i dodjeljivanje resursa potrebnih za izvršenje zadataka i definiranje vremena trajanja aktivnosti i zadataka. Microsoft Project 2007 koristili smo kako bi jednostavnije i preglednije pratili razvoj informacijskog sustava za prodaju kino ulaznica.

**Microsoft Office 2010**

Microsoft Office 2010, točnije alat koji se nalazi u sklopu njega, Microsoft Word koristili smo za izradu projektne dokumentacije.

**Visual Paradigm for UML CE 10.1**

Alat Visual Paradigm for UML CE 10.1 koristili smo za izradu UML dijagrama (dijagram slučajeva korištenja, dijagram slijeda, dijagram aktivnosti i dijagram klasa).

**Visual Studio 2012**

Microsoft Visual Studio 2012 koristili smo za izradu aplikacije Kino.NET. Aplikacija je napisana u programskom jeziku Visual C# koji omogućuje pisanje aplikacija koje se izvršavaju u .NET Frameworku.

**MS .NET (aplikacija)**

Kao što smo već ranije spomenuli, aplikacija Kino.NET je napisana i izvršava se u .NET razvojnom okruženju. .NET Framework je biblioteka klasa koja pruža pristup mnogim servisima operacijskog sustava i drugim klasama koje pripomažu i ubrzavaju proces razvoja aplikacije.

**Toad Data Modeler**

Toad Data Modeler smo koristili za kreiranje ERA modela i generiranje SQL skripte za kreiranje baze podataka na temelju ERA modela. SQL skriptu smo generirali za bazu podataka.

**SQLite Expert Personal**

Pomoću generirane SQL skripte u alatu SQLite Expert Personal smo kreirali SQLite bazu podataka, unesli podatke i izvezli ju u **.db3** formatu.

# Dijagrami

## UML dijagrami

### Dijagram slučajeva korištenja

Dijagrami slučajeva korištenja su prvi dijagrami koje smo morali izraditi, kako bismo definirali koje sve učesnike imamo u našem sustavu, te, što se sve vezuje uz njih. Dijagrami slučajeva korištenja su nam veoma bitni jer predstavljaju temelj za sve ostale dijagrame, a to je stoga što oni zapravo prikazuju što to ustvari sustav radi.

U našem slučaju imamo tri tipa učesnika, a to su **administrator**, **prodavač** i **baza podataka**. **Administrator**  je u ovom slučaju osoba koja se brine za podatke koji su potrebni za rad same aplikacije, a to su podaci o zaposlenicima, filmovima, artiklima, dvoranama i projekcijama. **Prodavač** je osoba koja radi na kasi, a njegova uloga je prodaja ulaznica i artikala kao što su kokice, sokovi itd. **Baza podataka** je sustav u kom se pohranjuju podaci koje administrator administrira putem aplikacije. Ona je vanjski sustav na koji se spaja aplikacija Kino.NET. Za te učesnike smo načinili dva dijagrama slučajeva korištenja. Prvi dijagram prikazuje slučaj korištenja za administratora, a drugi slučaj korisnika za prodavača.

**Slučaj korištenja za administratora:**

* Logiranje u sustav
* Autentifikacija
* Administriranje podataka
  + Administriranje dvorana
  + Administriranje filmova
  + Administriranje projekcija
  + Administriranje zaposlenika
  + Administriranje artikala
* Kreiranje izvještaja

**Logiranje u sustav**

Administrator se logira u sustav na način da upiše svoje korisničko ime i lozinku. Ovaj slučaj korištenja uključuje i slučaj korištenja **Autentifikacija.** Taj slučaj korištenja provjerava u bazi podataka točnost unesenih podataka i na temelju onoga što provjera vrati, administratoru se omogućuje, ili u slučaju neispravno unesenih podataka onemogućuje, pristup određenim funkcionalnostima aplikacije.

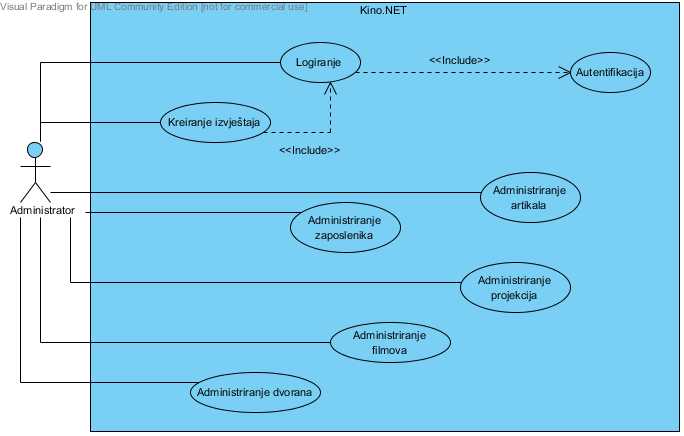
**Administriranje Podataka**

Slučaj korištenja **Administriranje podataka** odnosi se na unos, mijenjanje, brisanje i pregled podataka o zaposlenicima, filmovima, dvoranama, projekcijama i artiklima putem aplikacije. Također, ovaj slučaj nužno zahtjeva, odnosno uključuje slučaj korištenja **Logiranje u sustav.**

**Kreiranje izvještaja**

Slučaj **Kreiranje izvještaja**, kao i prethodni slučaj, uključuje slučaj korištenja **Logiranje u sustav**. Kreiranje izvještaja se odnosi na izradu dnevnih/tjednih/mjesečnih (ovisno o potrebi) izvještaja o posjećenosti kina, odnosno posjećenosti pojedine projekcije.

**Slučaj korištenja za administratora**



Slika 3.1: Slučaj korištenja za administratora

**Slučaj korištenja za prodavača:**

* Logiranje u sustav
* Autentifikacija
* Korištenje aplikacije
* Prodaja ulaznica
* Odabir projekcije
* Pregled popisa projekcija
* Pregled raspoloživosti ulaznica
* Ispis ulaznica
* Prodaja artikala
* Odabir artikala
* Izdavanje računa

**Logiranje u sustav**

Prodavač se logira u sustav na način da upiše svoje korisničko ime i lozinku. Ovaj slučaj korištenja uključuje i slučaj korištenja **Autentifikacija.** Taj slučaj korištenja provjerava u bazi podataka točnost unesenih podataka i na temelju onoga što provjera vrati, prodavaču se omogućuje, ili u slučaju neispravno unesenih podataka onemogućuje, pristup određenim funkcionalnostima aplikacije.

**Korištenje aplikacije**

**Korištenje aplikacije** predstavlja slučaj korištenja koji uključuje slučaj korištenja **Logiranje u sustav.**  Ovaj slučaj korištenja ima i dva proširenja. Ta dva proširenja predstavljeni su slučajevima korištenja **Prodaja ulaznica**  i **Prodaja artikala.**

**Prodaja ulaznica**

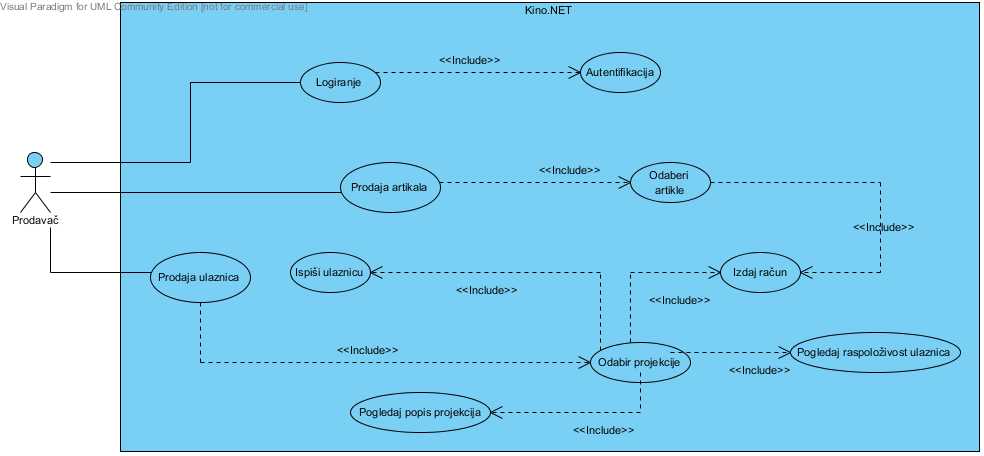
**Prodaja ulaznica** je slučaj korištenja koji uključuje još nekoliko drugih slučajeva korištenja, a to su:

* **Odabir projekcije**, taj slučaj korištenja se odnosi na to da prodavač odabire određenu projekciju koja je dostupna određenog dana u određeno vrijeme, u određenoj dvorani. Za odabranu projekciju mora vrijediti da još uvijek ima raspoloživih karata. Kako bi se taj uvjet mogao ostvariti, ovaj slučaj korištenja uključuje još i slučajeve **Pogledaj popis projekcija** koji uključuje odabir određene projekcije (film, dvorana, vrijeme), te slučaj korištenja **Pogledaj raspoloživost ulaznica** u kom se provjerava ima li još slobodnih mjesta u dvorani za tu projekciju.
* **Ispiši ulaznicu** je slučaj korištenja koji je također uključen u slučaj korištenja **Prodaja ulaznica**. Taj slučaj korištenja predstavlja ispis kreirane ulaznice.
* **Izdaj račun** je slučaj korištenja u kom se izdaje račun za prodane ulaznice.

**Prodaja artikala**

**Prodaja artikala** je slučaj korištenja koji u sebi uključuje slučaj **Odaberi artikle**. Slučaj korištenja **Odaberi artikle** odnosi se na izbor tipa i količine artikla. Taj slučaj uključuje slučaj korištenja **Izdaj račun.** **Izdaj račun** je slučaj korištenja u kom se izdaje račun za prodane artikle.

**Slučaj korištenja za prodavača**



Slika 3.2: Slučaj korištenja za prodavača

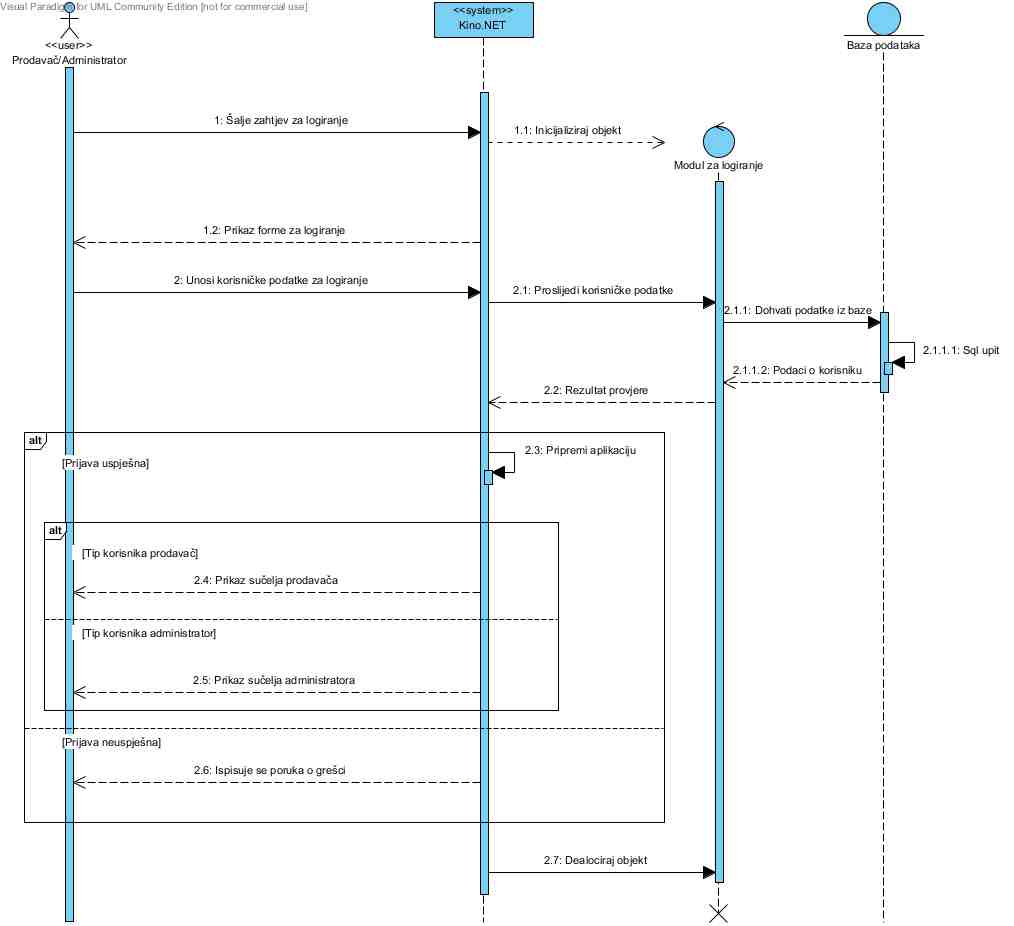
### Dijagram slijeda

Kako bismo što detaljnije opisali slučajeve korištenja, izradili smo i dijagrame slijeda koji prikazuju vremenski uređenu interakciju između prethodno navedenih učesnika.

U nastavku su opisni i prikazani dijagrami slijeda za ove slučajeve korištenja:

* Logiranje u sustav
* Prodaja ulaznica
* Prodaja artikala
* Administriranje podataka
* Kreiranje izvještaja

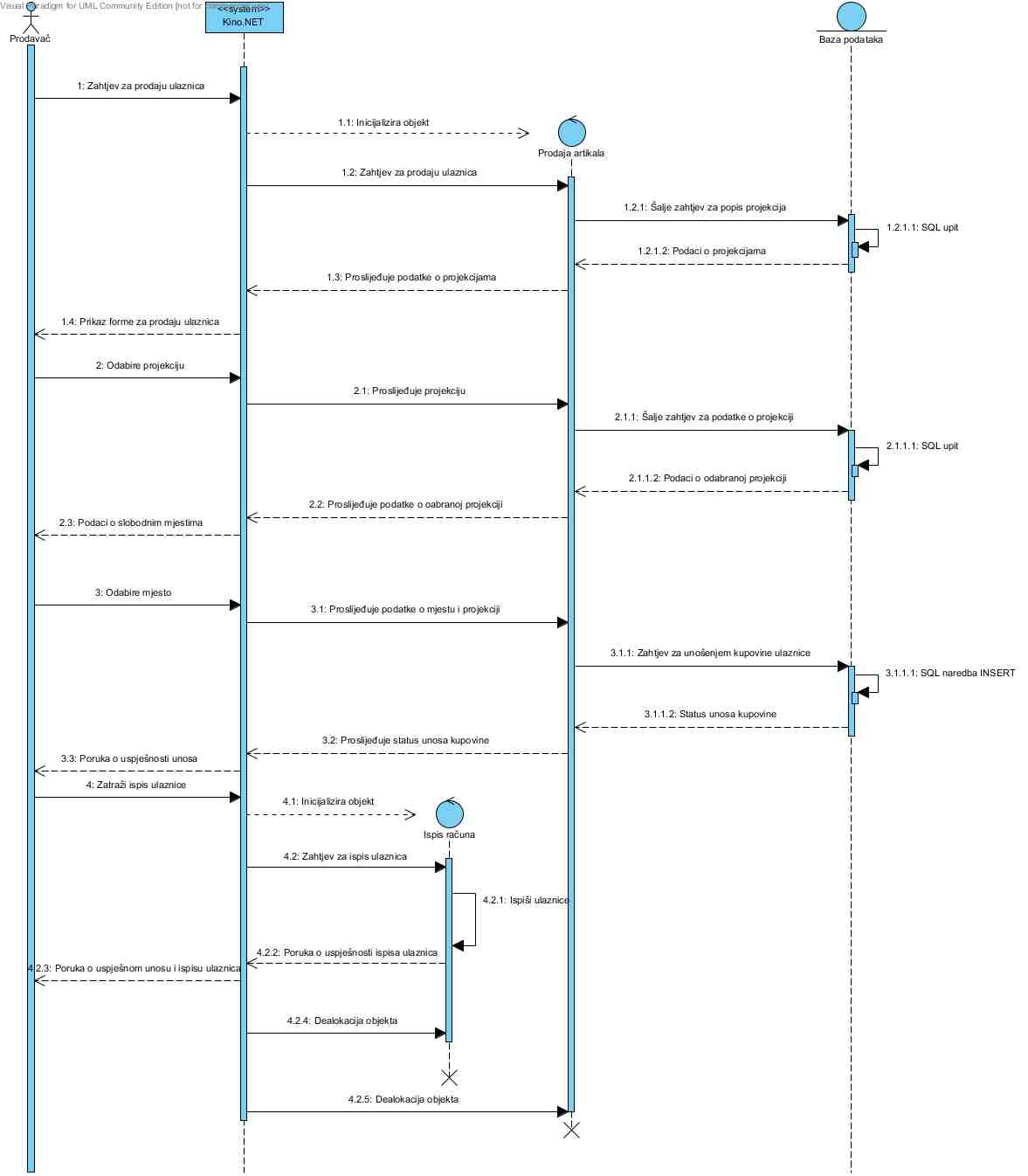
#### Logiranje u sustav



Slika 3.3: Dijagram slijeda - Logiranje u sustavLogiranje u sustav

Na početku korištenja aplikacije korisnik se prvo mora prijaviti u sustav kako bi mu se dodijelila određena prava za korištenje sustava kina. Nakon što korisnik unese vlastito korisničko ime i lozinku sustav prosljeđuje korisničke podatke modulu za logiranje, a nakon toga se izvršava upit nad tablicom korisnika u bazi podataka. Nakon što se izvrši upit rezultati se vraćaju nazad do sustava gdje se, ovisno o vraćenom rezultatu upita, prikazuje poruka o neuspješnom logiranju u sustav ukoliko korisnik ne postoji ili je pogrešno unesena lozinka. U slučaju uspješnog logiranja u sustav korisniku se prikazuje sučelje, a ovisno o tipu korisnika sustav prikazuje korisniku sučelje za prodavača ili administratora.

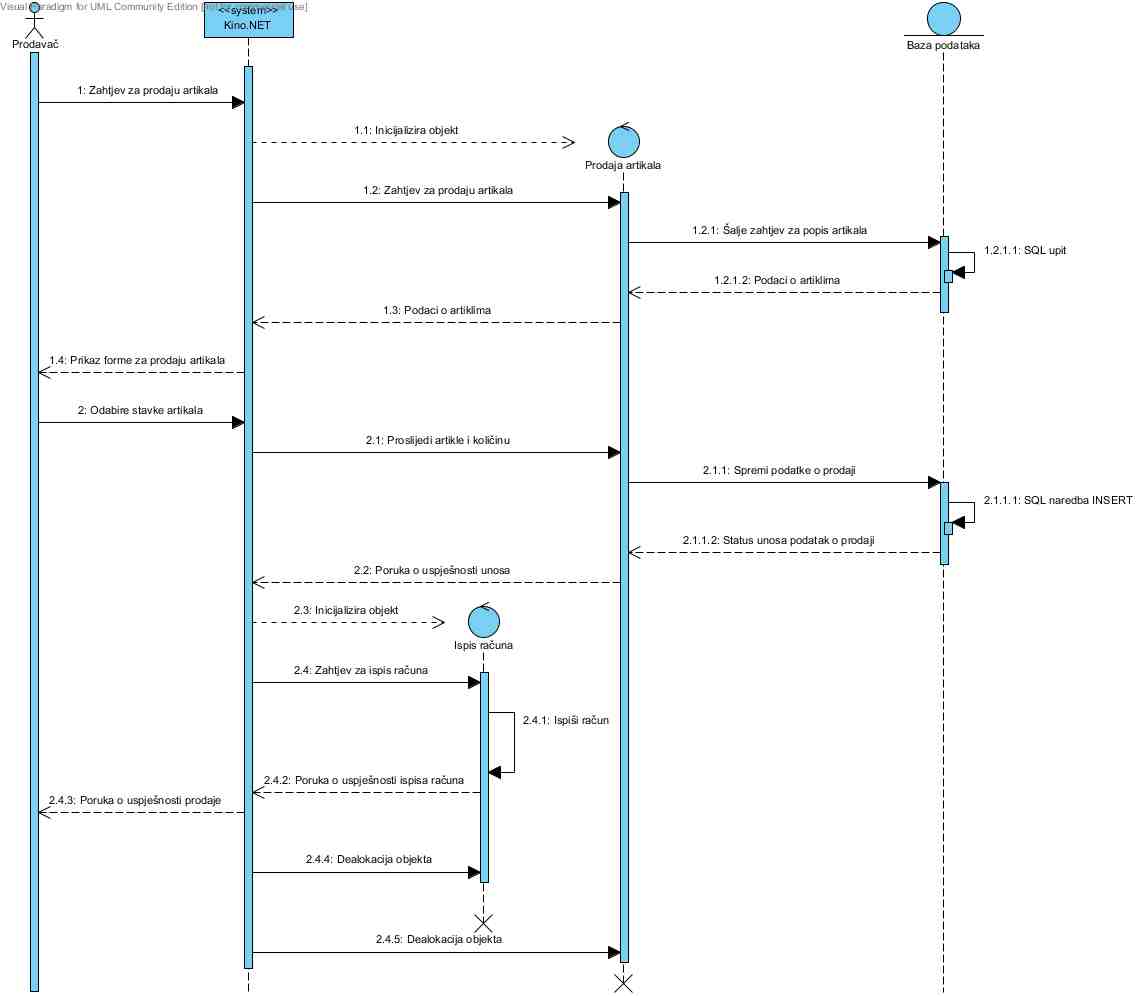
#### Prodaja ulaznica



Slika 3.4. Dijagram slijeda - Prodaja ulaznica

Ako je korisnik prijavljen u sustav sa privilegijama prodavača prvotno mu je prikazana forma sa izborom mogućnosti prodaje ulaznica ili artikala. Ukoliko prodavač odabere prodaju ulaznica sustav zatraži popis projekcija za određeni datum te se prikazuje forma sa mogućnosti odabira projekcija. Nakon odabira projekcija iz baze podatak se izvlače podaci o pojedinoj projekciji. Nakon toga prodavač je u mogućnosti odabrati sjedala koja su slobodna za određenu projekciju. Nakon odabira sjedala za određenu projekciju podaci se zapisuju u bazu podataka te se ispisuje ulaznica.

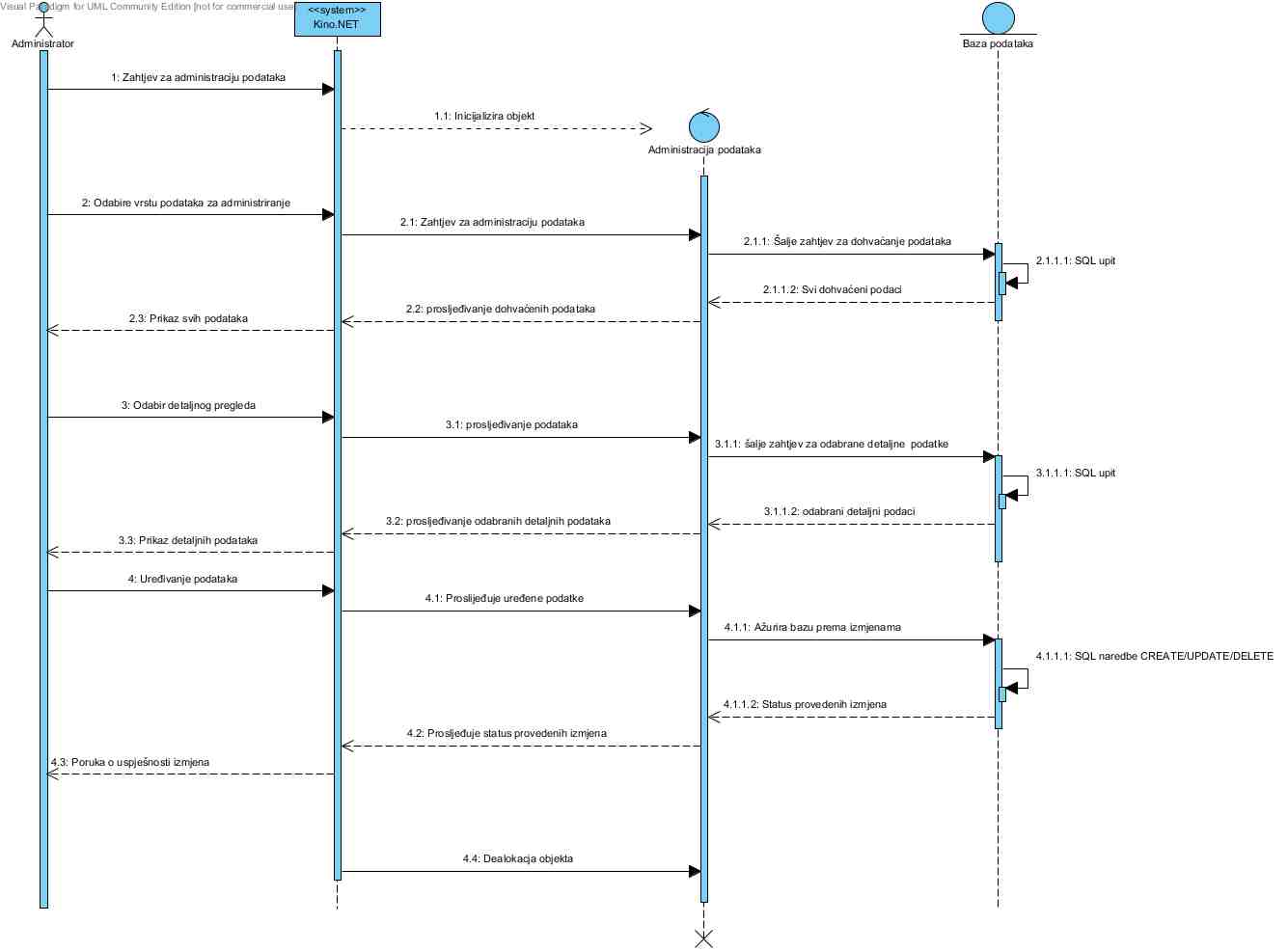
#### Prodaja artikala



Slika 3.5. Dijagram slijeda - Prodaja artikala

Ukoliko je korisnik prijavljen u sustav kao prodavač i odabere na sučelju opciju prodaje artikala sustav šalje zahtjev za popisom artikala iz baze podataka. Nakon dohvaćanja artikala iz baze podataka prikazuje se forma za odabir stavki artikala. Nakon odabira stavki artikala sustav prosljeđuje podatke koji se nakon toga zapisuju u bazi podataka. Nakon toga ispisuje se račun te sustav prodavaču šalje poruku o uspješnom unosu i ispisu kupovine artikala.

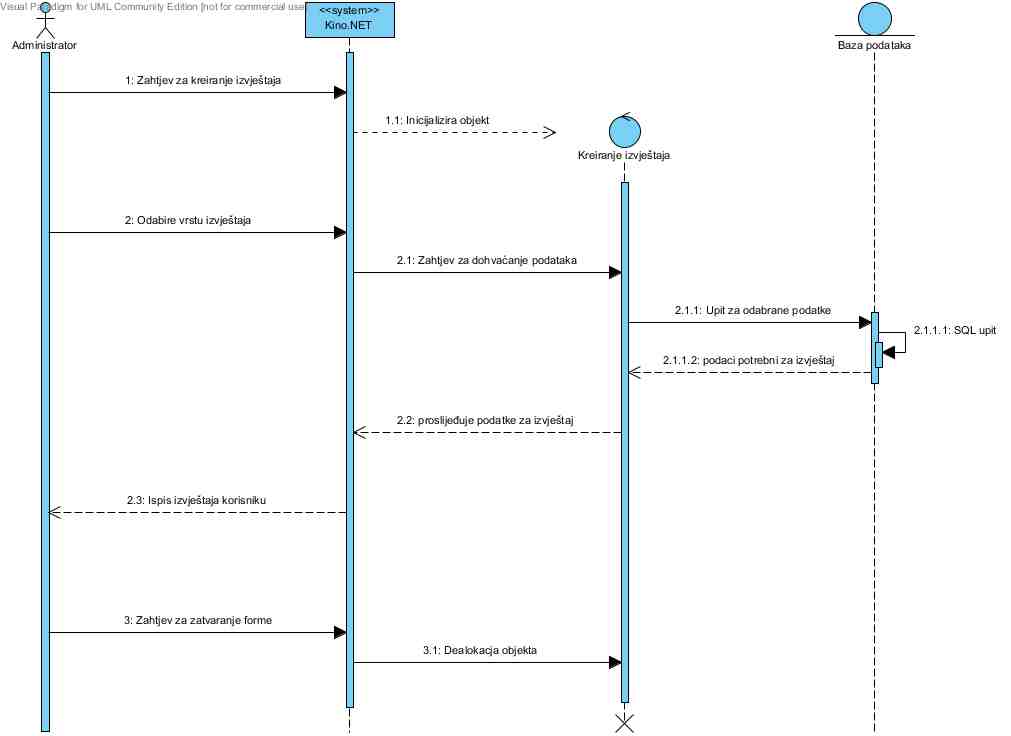
#### Administriranje podataka



Slika 3.6. Dijagram slijeda - Administriranje podataka

Ukoliko je korisnik prijavljen u aplikaciju sa administratorskim pravima omogućene su mu funkcionalnosti administracije podataka o zaposlenicima, projekcijama i filmovima. Nakon što odabere vrstu podataka koju želi uređivati iz baze podataka se dohvaćaju svi zaposlenici/filmovi/projekcije te mu je omogućen odabir pojedinog zaposlenika/filma/projekcije. Nakon toga se iz baze podataka dohvaćaju detaljni podaci koji se mogu uređivati ili je moguće dodavati nove zaposlenike/filmove/projekcije.

#### Kreiranje izvještaja



Slika 3.7. Dijagram slijeda - Kreiranje izvještaja

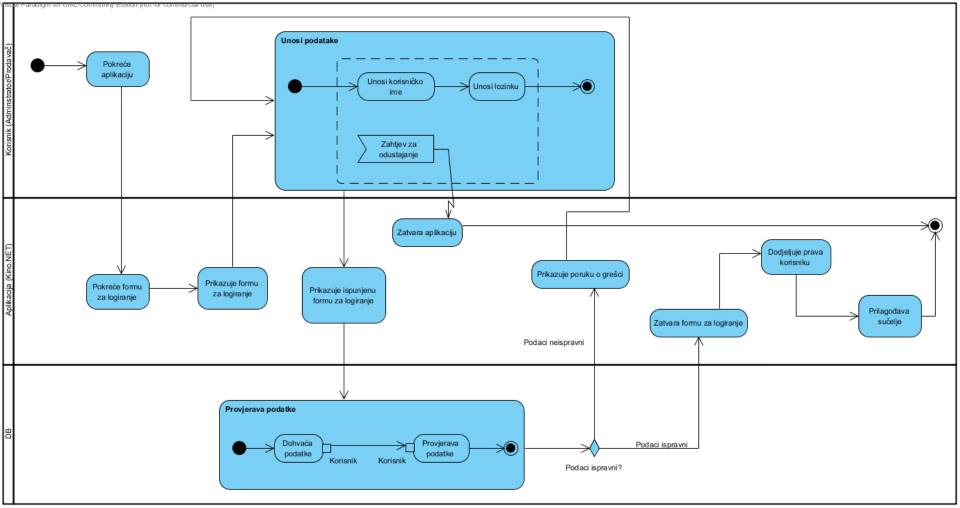
Administratoru je omogućena funkcionalnost kreiranja izvještaja o projekcijama i artiklima. Nakon što administrator odabere vrstu izvještaja sustav prosljeđuje zahtjev za kreiranje izvještaja nakon čega se pretražuje baza podataka. Nakon dohvaćanja svih bitnih entiteta iz baze podataka podaci se prosljeđuju na daljnju obradu. Nakon izrade izvještaja sustav korisniku prikazuje izvještaj. Nakon što korisnik ne želi više kreirati izvještaje zatvara formu sa zahtjevom za zatvaranje.

### Dijagram aktivnosti

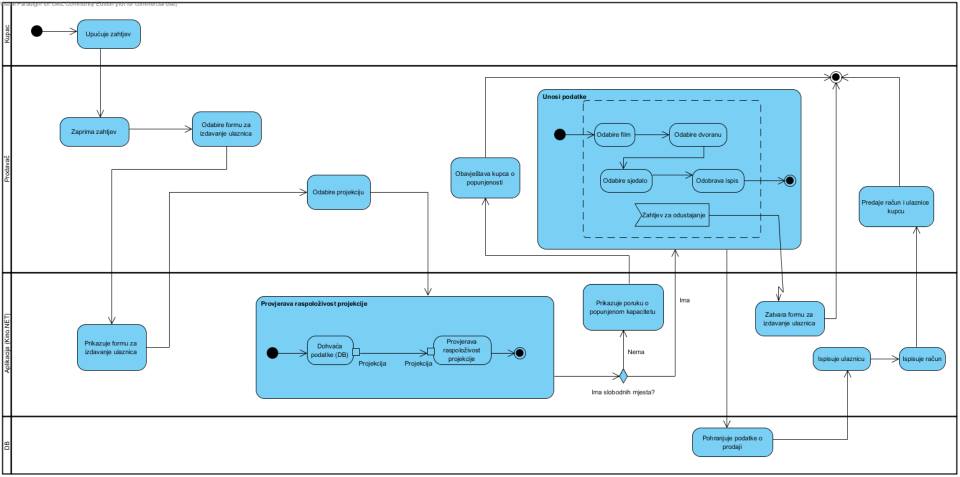
Sljedeći tip dijagrama su dijagrami aktivnosti. Pomoću ovih dijagrama detaljnije smo prikazali tijek aktivnosti unutar slučajeva korištenja koji su važni za naš projekt.

Ti slučajevi korištenja su:

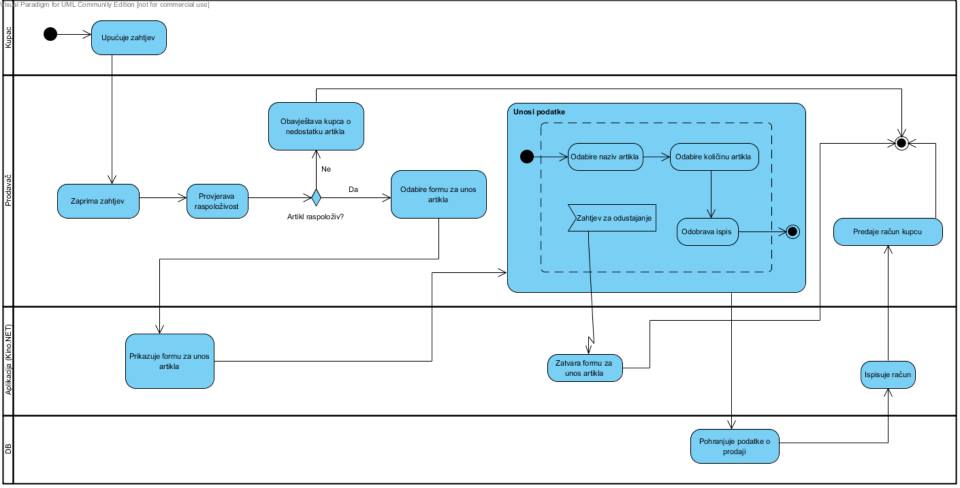
* Logiranje u sustav
* Prodaja ulaznica
* Prodaja artikala
* Administriranje podataka
* Kreiranje izvještaja



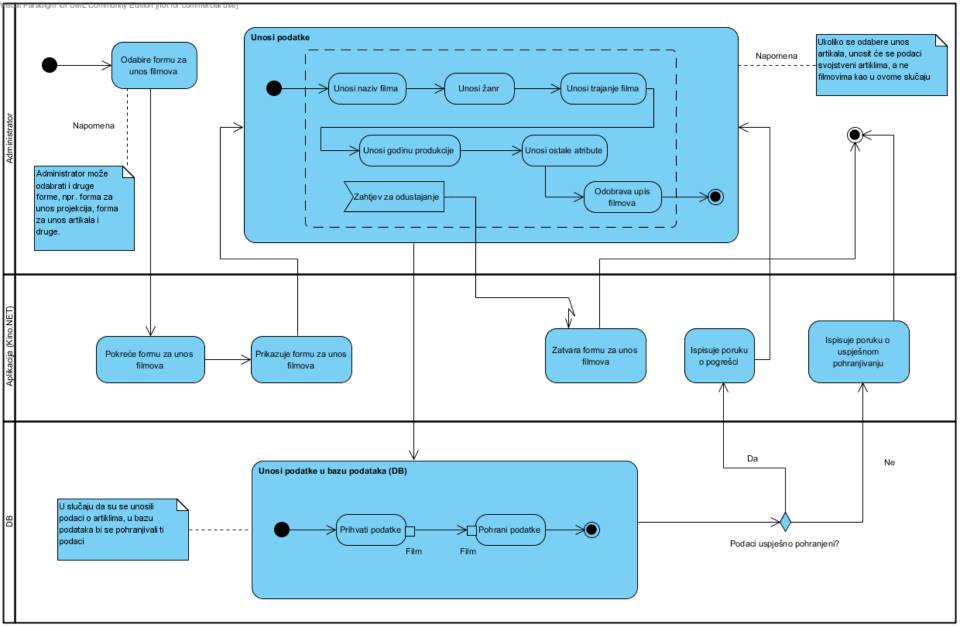
Slika 3.8: Dijagram aktivnosti - Logiranje u sustav



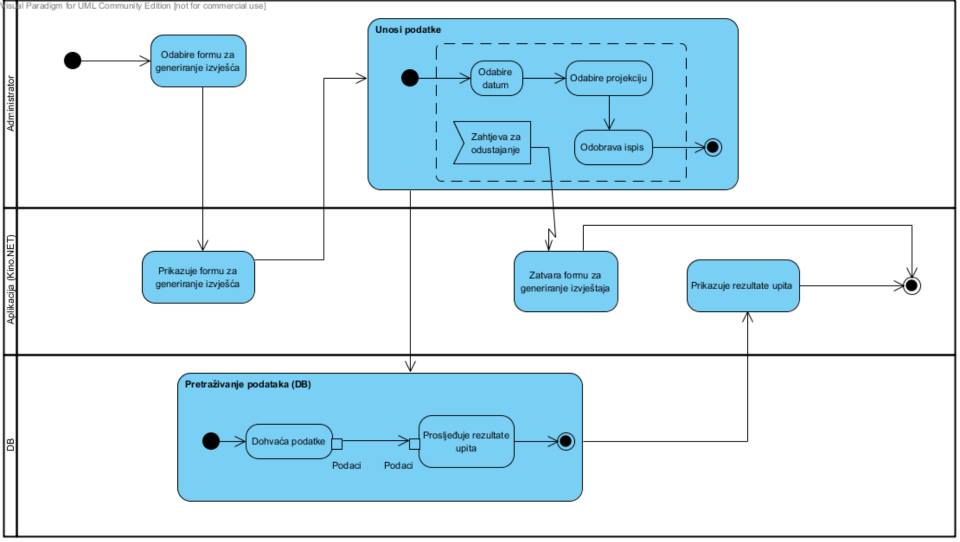
Slika 3.9: Dijagram aktivnosti - Prodaja karata



Slika 3.10: Dijagram aktivnosti - Prodaja artikala



Slika 3.11: Dijagram aktivnosti - Administriranje podataka

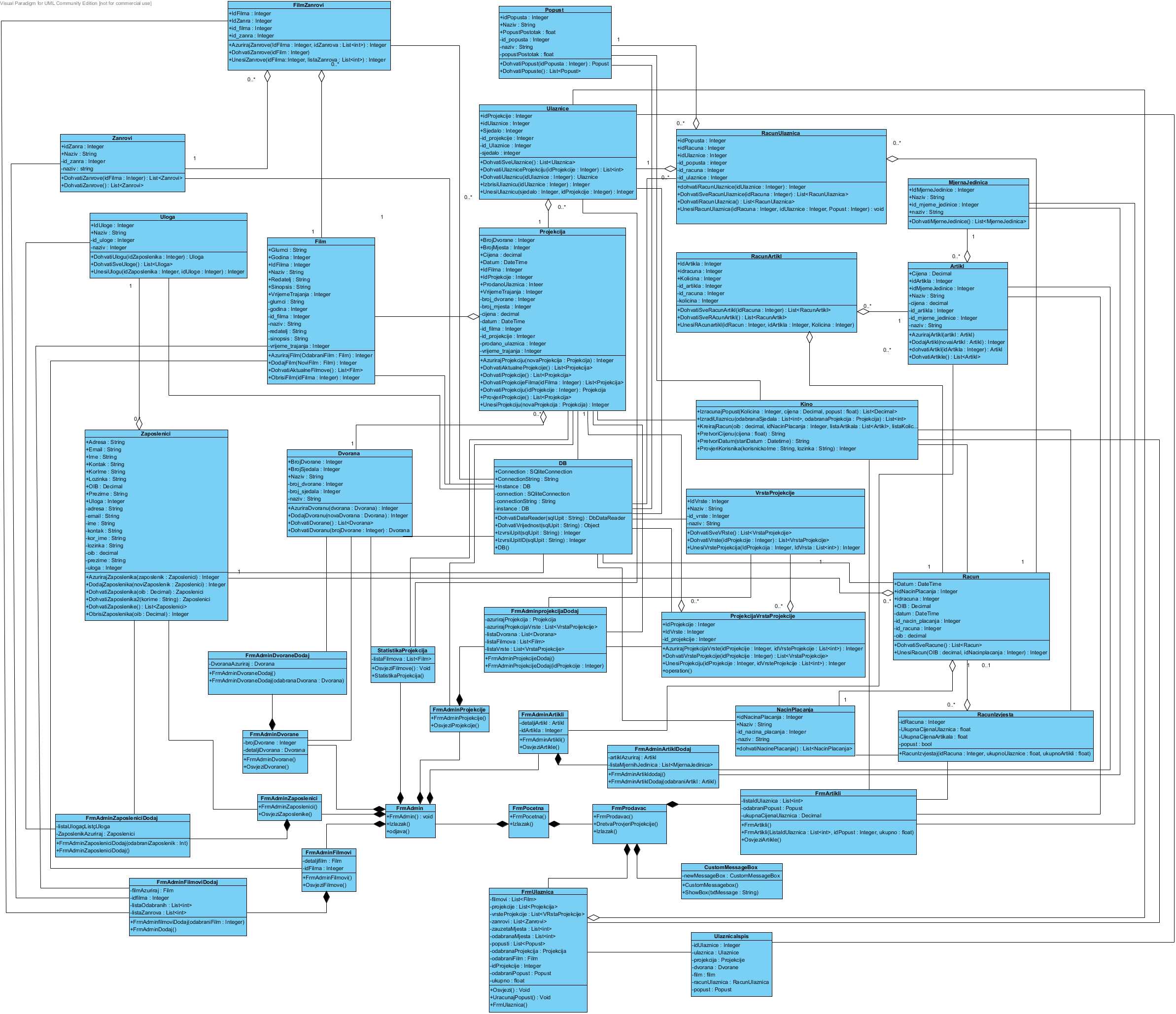


Slika 3.12: Dijagram aktivnosti - Kreiranje izvještaja

## Dijagram klasa

Dijagramom klasa ćemo prikazati strukturu našeg sustava uz pomoć atributa, operacija, klasa i odnosa među njima. Početna klasa je **frmGlavna** koja je povezana sa klasama **frmLogin** za logiranje korisnika u sustav. Ovisno o tipu korisnika prodavač/administrator, a nakon uspješnog logiranja u sustav, otvara se forma **frmAdmin** ili forma **frmProdavac.** Na klasu **frmAdmin** povezane su klase **frmIzvjestaji** i **frmAdministracija** podataka koje koriste operacije drugih klasa za administraciju podataka (filmovi, projekcije, zaposlenici i dvorane) i operacije kreiranja raznih izvještaja o posjećenosti projekcija.

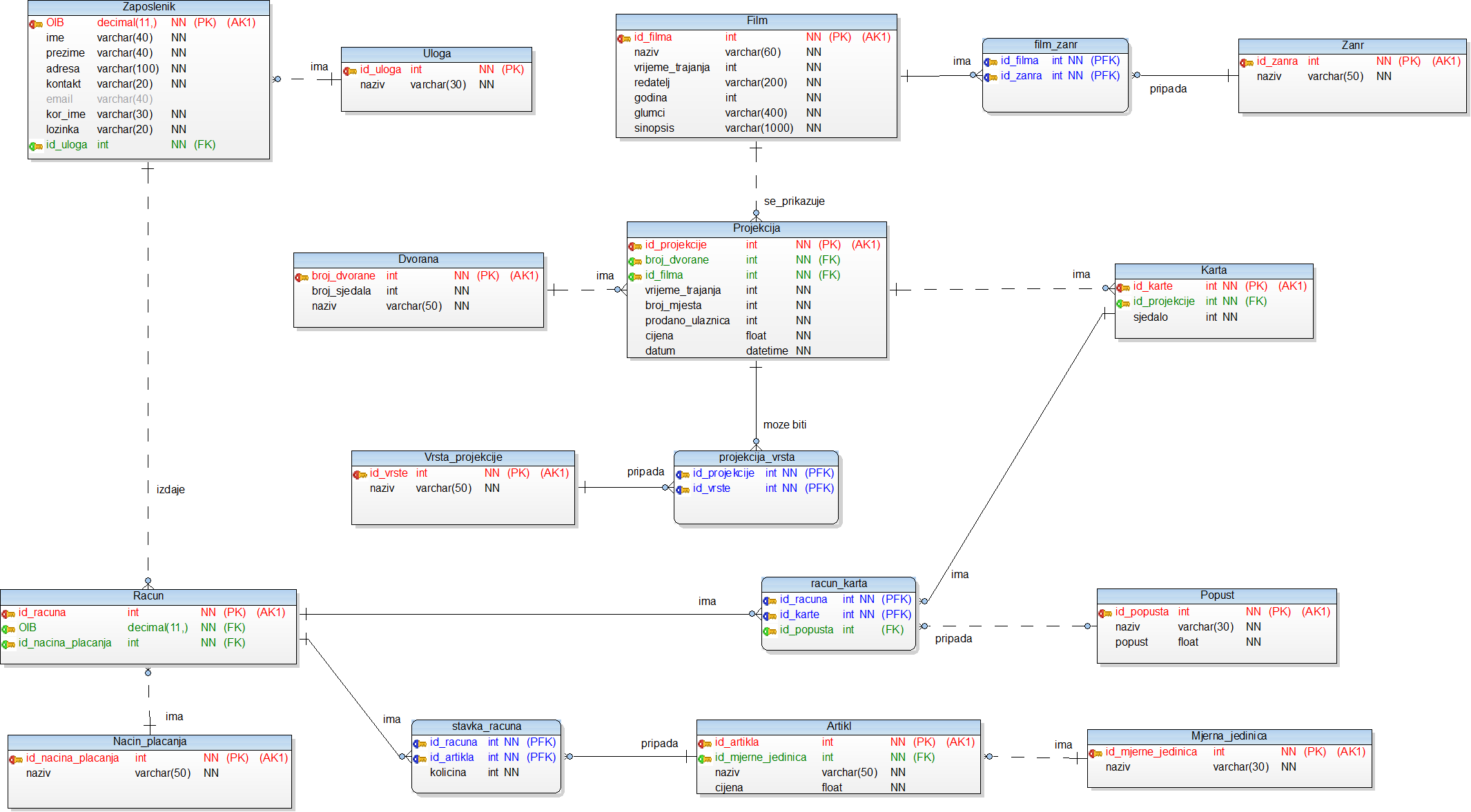
Klase **frmUlaznica** i **frmArtikli** su povezane sa klasom **frmProdavac** i omogućuju prodaju ulaznica i artikala. Ostale klase su **Artikl**, **Racun**, **Projekcija**, **Film**, **Dvorana**, **Ulaznica**, **Stavke** **racuna**, **Popust** te pripadne enumeracije **Zanr**, **VrstaProjekcije**, **MjernaJedinica** i **NacinPlacanja**.



Slika 3.13: Dijagram klasa

## ERA model

Kao što smo već ranije spomenuli, ERA model smo kreirali u alatu ***Toad Data Modeler v3.6.6*** iz kojeg smo na temelju modela izgenerirali i SQL skriptu. ERA model je statički model podataka koji služi za logičko modeliranje podataka. Logičko modeliranje podataka važan je dio projekta iz razloga što će aplikacija biti povezana na bazu podataka, te ukoliko podatke dobro povežemo u ERA modelu i kasnije u bazi podataka, rad na aplikaciji će biti dosta lakši. Osnovni elementi ERA modela su entiteti, njihovi atributi i veze, odnosno relacije



Slika 3.14: ERA model

## Relacijski model podatka

Prikazat ćemo i relacijski model podataka koji se radi na temelju ERA modela. Relacijski model prikazuje entitete, njihove atributi, primarne i vanjske ključeve koji označavaju veze, odnosno relacije. Primarne ključeve smo podcrtali, a vanjske ključeve pobojali.

Zaposlenik (OIB, ime, prezime, adresa, kontakt, email, kor\_ime, lozinka);

Filmovi (id\_filma, naziv, vrijeme\_trajanja);

Zanr (id\_zanra, naziv);

film\_zanr (id\_zanra, id\_filma);

Dvorana (broj\_dvorane, broj\_sjedala, naziv);

projekcija (broj\_dvorane, id\_filma, vrijeme\_trajanja, broj\_mjesta, prodano\_ulaznica, cijena, datum);

Karta (id\_karte, sjedalo, broj\_dvorane, id\_filma);

Vrsta\_projekcije (id\_vrste, naziv);

projekcija\_vrsta (broj\_dvorane, id\_filma, id\_vrste);

Artikl (id\_artikla, naziv, cijena, id\_mjerne\_jedinice);

Racun (id\_racuna, OIB, id\_nacina\_placanja);

stavka\_racuna (id\_racuna, id\_artikla, kolicina);

racun\_karta (id\_racuna, id\_karte, id\_popusta);

Popust (id\_popusta, naziv, popust);

Nacin\_placanja (id\_nacina\_placanja, naziv);

Mjerna\_jedinica (id\_mjerne\_jedinice, naziv);

# Baza podataka

## SQL kod

Nakon što smo izradili ERA dijagram u alatu ***Toad Data Modeler v3.6.6.*** na temelju njega smo izgenerirali **SQL skriptu**. Pomoću SQL skripte smo u alatu ***SQLite Expert Personal v3.4.61.*** kreirali relacije, njihove atribute i veze između njih, odnosno samu bazu podataka. Nakon toga smo u istom alatu unosili podatke. U nastavku slijedi SQL skripta.

CREATE TABLE Zaposlenik

(

OIB decimal(11) NOT NULL,

ime varchar(40) NOT NULL,

prezime varchar(40) NOT NULL,

adresa varchar(100) NOT NULL,

kontakt varchar(20) NOT NULL,

email varchar(40),

kor\_ime varchar(30) NOT NULL,

lozinka varchar(20) NOT NULL,

CONSTRAINT Key1 PRIMARY KEY (OIB),

CONSTRAINT OIB UNIQUE (OIB)

);

CREATE TABLE Filmovi

(

id\_filma integer NOT NULL

CONSTRAINT Key2 PRIMARY KEY AUTOINCREMENT,

naziv varchar(60) NOT NULL,

vrijeme\_trajanja time NOT NULL,

CONSTRAINT id\_filma UNIQUE (id\_filma)

);

CREATE TABLE Zanr

(

id\_zanra integer NOT NULL

CONSTRAINT Key3 PRIMARY KEY AUTOINCREMENT,

naziv varchar(50) NOT NULL,

CONSTRAINT id\_zanra UNIQUE (id\_zanra)

);

CREATE TABLE film\_zanr

(

id\_zanra integer NOT NULL,

id\_filma integer NOT NULL,

CONSTRAINT Key4 PRIMARY KEY (id\_filma,id\_zanra),

CONSTRAINT zanr\_pripada FOREIGN KEY (id\_zanra) REFERENCES Zanr (id\_zanra),

CONSTRAINT film\_ima FOREIGN KEY (id\_filma) REFERENCES Filmovi (id\_filma)

);

CREATE TABLE Dvorana

(

broj\_dvorane integer NOT NULL,

broj\_sjedala integer NOT NULL,

naziv varchar(50) NOT NULL,

CONSTRAINT Key5 PRIMARY KEY (broj\_dvorane),

CONSTRAINT broj\_dvorane UNIQUE (broj\_dvorane)

);

CREATE TABLE projekcija

(

broj\_dvorane integer NOT NULL,

id\_filma integer NOT NULL,

vrijeme\_trajanja time NOT NULL,

broj\_mjesta integer NOT NULL,

prodano\_ulaznica integer NOT NULL DEFAULT 0,

cijena float NOT NULL,

datum date NOT NULL,

CONSTRAINT Key6 PRIMARY KEY (broj\_dvorane,id\_filma),

CONSTRAINT ima1 FOREIGN KEY (broj\_dvorane) REFERENCES Dvorana (broj\_dvorane),

CONSTRAINT se\_prikazuje FOREIGN KEY (id\_filma) REFERENCES Filmovi (id\_filma)

);

CREATE TABLE Karta

(

id\_karte integer NOT NULL

CONSTRAINT Key7 PRIMARY KEY AUTOINCREMENT,

sjedalo integer NOT NULL,

broj\_dvorane integer NOT NULL,

id\_filma integer NOT NULL,

CONSTRAINT id\_karte UNIQUE (id\_karte),

CONSTRAINT sjedalo UNIQUE (sjedalo),

CONSTRAINT ima2 FOREIGN KEY (broj\_dvorane, id\_filma) REFERENCES projekcija (broj\_dvorane, id\_filma)

);

CREATE TABLE Vrsta\_projekcije

(

id\_vrste integer NOT NULL

CONSTRAINT Key8 PRIMARY KEY AUTOINCREMENT,

naziv varchar(50) NOT NULL,

CONSTRAINT id\_vrste UNIQUE (id\_vrste)

);

CREATE TABLE projekcija\_vrsta

(

broj\_dvorane integer NOT NULL,

id\_filma integer NOT NULL,

id\_vrste integer NOT NULL,

CONSTRAINT Key9 PRIMARY KEY (broj\_dvorane,id\_filma),

CONSTRAINT moze\_biti FOREIGN KEY (broj\_dvorane, id\_filma) REFERENCES projekcija (broj\_dvorane, id\_filma),

CONSTRAINT pripada1 FOREIGN KEY (id\_vrste) REFERENCES Vrsta\_projekcije (id\_vrste)

);

CREATE TABLE Artikl

(

id\_artikla integer NOT NULL

CONSTRAINT Key12 PRIMARY KEY AUTOINCREMENT,

naziv varchar(50) NOT NULL,

cijena float NOT NULL,

id\_mjerne\_jedinice integer NOT NULL,

CONSTRAINT id\_artikla UNIQUE (id\_artikla),

CONSTRAINT ima6 FOREIGN KEY (id\_mjerne\_jedinice) REFERENCES Mjerna\_jedinica (id\_mjerne\_jedinice)

);

CREATE TABLE Racun

(

id\_racuna integer NOT NULL

CONSTRAINT Key13 PRIMARY KEY AUTOINCREMENT,

OIB decimal(11) NOT NULL,

id\_nacina\_placanja integer NOT NULL,

CONSTRAINT id\_racuna UNIQUE (id\_racuna),

CONSTRAINT izdaje FOREIGN KEY (OIB) REFERENCES Zaposlenik (OIB),

CONSTRAINT ima5 FOREIGN KEY (id\_nacina\_placanja) REFERENCES Nacin\_placanja (id\_nacina\_placanja)

);

CREATE TABLE stavka\_racuna

(

id\_racuna integer NOT NULL,

id\_artikla integer NOT NULL,

kolicina integer NOT NULL,

CONSTRAINT Key15 PRIMARY KEY (id\_racuna,id\_artikla),

CONSTRAINT ima3 FOREIGN KEY (id\_racuna) REFERENCES Racun (id\_racuna),

CONSTRAINT pripada2 FOREIGN KEY (id\_artikla) REFERENCES Artikl (id\_artikla)

);

CREATE TABLE racun\_karta

(

id\_racuna integer NOT NULL,

id\_karte integer NOT NULL,

id\_popusta int,

CONSTRAINT Key16 PRIMARY KEY (id\_racuna,id\_karte),

CONSTRAINT ima4 FOREIGN KEY (id\_racuna) REFERENCES Racun (id\_racuna),

CONSTRAINT pripada3 FOREIGN KEY (id\_karte) REFERENCES Karta (id\_karte),

CONSTRAINT pripada4 FOREIGN KEY (id\_popusta) REFERENCES Popust (id\_popusta)

);

CREATE TABLE Popust

(

id\_popusta integer NOT NULL

CONSTRAINT Key17 PRIMARY KEY AUTOINCREMENT,

naziv varchar(30) NOT NULL,

popust float NOT NULL,

CONSTRAINT id\_popusta UNIQUE (id\_popusta)

);

CREATE TABLE Nacin\_placanja

(

id\_nacina\_placanja integer NOT NULL

CONSTRAINT Key18 PRIMARY KEY AUTOINCREMENT,

naziv varchar(50) NOT NULL,

CONSTRAINT id\_nacina\_placanja UNIQUE (id\_nacina\_placanja)

);

CREATE TABLE Mjerna\_jedinica

(

id\_mjerne\_jedinice integer NOT NULL

CONSTRAINT Key19 PRIMARY KEY AUTOINCREMENT,

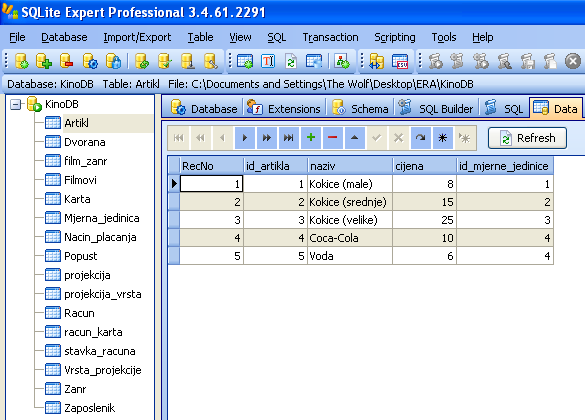
naziv varchar(30) NOT NULL,

CONSTRAINT id\_mjerne\_jedinice UNIQUE (id\_mjerne\_jedinice)

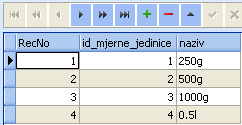
);

## Unos podataka

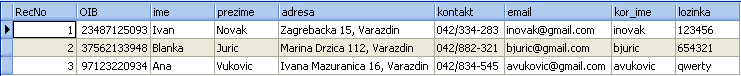
Kao što smo već spomenuli, podatke smo unosili u alatu ***SQLite Expert Personal v3.4.61.***, te smo iz njega izvukli bazu podataka u **.db3** formatu. U nastavku slijedi nekoliko screenshotova na kojima su prikazani uneseni podaci.



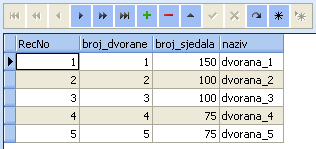
Slika 4.1: Podaci o artiklima

****

Slika 4.2: Mjerne jedinice

****

Slika 4.3: Korisnici

****

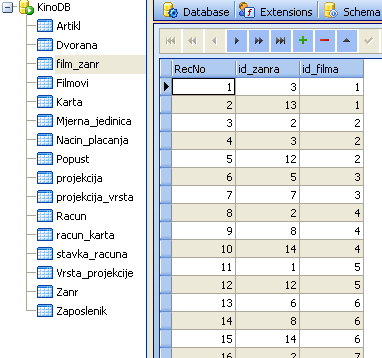
Slika 4.4: Dvorane

****

Slika 4.5: Filmovi

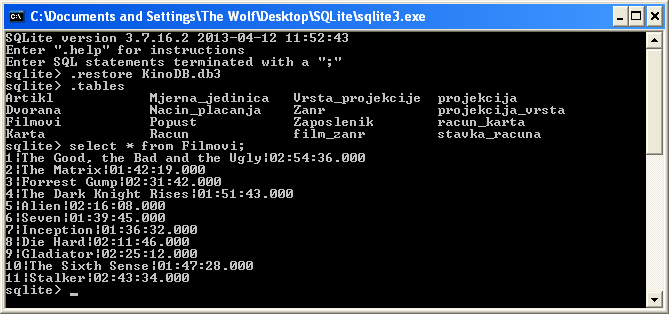
****

Slika 4.6: Žanrovi

****

Slika 4.7: film\_zanr – spajanje tablica Film i Zanr

Nakon što smo unijeli podatke i izvezli bazu podataka u **.db3** formatu, bazu smo učitali u **SQLite Command Shell** kako bi ju testirali. Na sljedećoj slici možemo vidjeti da je baza uspješno učitana i da radi.



Slika 4.8: SQLite3 Command Shell – testiranje baze KinoDB.db3

Slike

[Slika 3.1: Slučaj korištenja za administratora 6](#_Toc359967043)

[Slika 3.2: Slučaj korištenja za prodavača 9](#_Toc359967044)

[Slika 3.3: Dijagram slijeda - Logiranje u sustavLogiranje u sustav 11](#_Toc359967045)

[Slika 3.4. Dijagram slijeda - Prodaja ulaznica 12](#_Toc359967046)

[Slika 3.5. Dijagram slijeda - Prodaja artikala 14](#_Toc359967047)

[Slika 3.6. Dijagram slijeda - Administriranje podataka 15](#_Toc359967048)

[Slika 3.7. Dijagram slijeda - Kreiranje izvještaja 16](#_Toc359967049)

[Slika 3.8: Dijagram aktivnosti - Logiranje u sustav 18](#_Toc359967050)

[Slika 3.9: Dijagram aktivnosti - Prodaja karata 19](#_Toc359967051)

[Slika 3.10: Dijagram aktivnosti - Prodaja artikala 20](#_Toc359967052)

[Slika 3.11: Dijagram aktivnosti - Administriranje podataka 21](#_Toc359967053)

[Slika 3.12: Dijagram aktivnosti - Kreiranje izvještaja 22](#_Toc359967054)

[Slika 3.13: Dijagram klasa 24](#_Toc359967055)

[Slika 3.14: ERA model 26](#_Toc359967056)

[Slika 4.1: Podaci o artiklima 32](#_Toc359967057)

[Slika 4.2: Mjerne jedinice 32](#_Toc359967058)

[Slika 4.3: Korisnici 32](#_Toc359967059)

[Slika 4.4: Dvorane 33](#_Toc359967060)

[Slika 4.5: Filmovi 33](#_Toc359967061)

[Slika 4.6: Žanrovi 33](#_Toc359967062)

[Slika 4.7: film\_zanr – spajanje tablica Film i Zanr 34](#_Toc359967063)

[Slika 4.8: SQLite3 Command Shell – testiranje baze KinoDB.db3 34](#_Toc359967064)