SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Nadja Kosojević

Josipa Tadić

**PROJEKTNA DOKUMENTACIJA – NadJos**

**PROJEKT IZ KOLEGIJA PROGRAMSKO INŽENJERSTVO**

Varaždin, 2015.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Nadja Kosojević, 39073/10-I

Josipa Tadić, 39086/10-I

**PROJEKTNA DOKUMENTACIJA – NadJos**

**PROJEKT IZ KOLEGIJA PROGRAMSKO INŽENJERSTVO**

**Nositelj kolegija:**

Prof. dr. sc. Vjeran Strahonja

**Mentor:**

Boris Tomaš, mag. inf.

Varaždin, 2015.

Sadržaj

[1. Opis dizajna sustava 1](#_Toc411424182)

[1.1. Dijagram slučajeva korištenja 1](#_Toc411424183)

[1.2. Dijagrami aktivnosti 3](#_Toc411424184)

[1.2.1. Upiši polaznika/zaposlenika + generiraj ugovor 3](#_Toc411424185)

[1.2.2. Ažuriraj polaznika/zaposlenika 5](#_Toc411424186)

[1.2.3. Upravljaj vozilima 7](#_Toc411424187)

[1.2.4. Evidentiraj aktivnosti polaznika 9](#_Toc411424188)

[1.2.5. Izdaj potvrdu o položenom vozačkom ispitu 11](#_Toc411424189)

[1.3. Dijagrami slijeda 13](#_Toc411424193)

[1.3.1. Upiši polaznika/zaposlenika 13](#_Toc411424194)

[1.3.2. Ažuriraj polaznika/zaposlenika 14](#_Toc411424195)

[1.3.3. Upravljaj vozilima 16](#_Toc411424196)

[1.3.4. Evidentiraj aktivnosti polaznika 18](#_Toc411424197)

[1.3.5. Izdaj potvrdu o položenom vozačkom ispitu 19](#_Toc411424198)

[1.4. Dijagram klasa 20](#_Toc411424199)

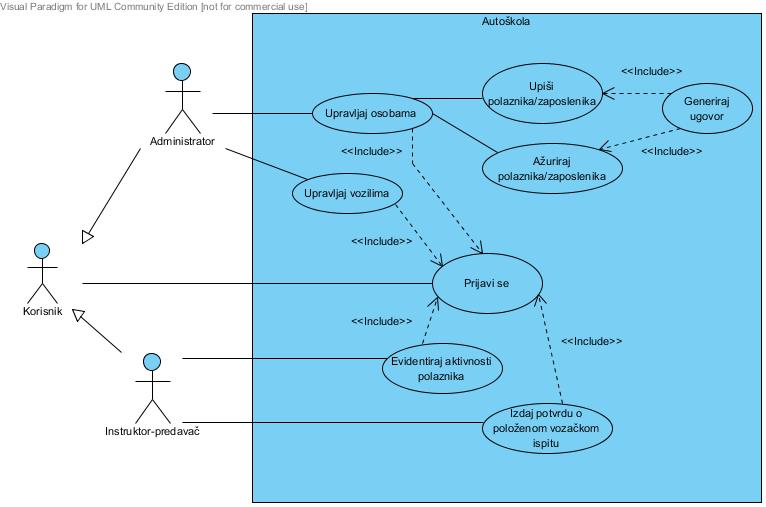
[1.5. ERA model 22](#_Toc411424200)

[2. Podaci o provedbi projekta 26](#_Toc411424201)

# Opis dizajna sustava

## Dijagram slučajeva korištenja

Dijagram slučajeva korištenja organiziran je na specifikacije slučajeva korištenja po korisnicima koji koriste aplikaciju. Dijagram slučajeva korištenja je dijagram ponašanja koji opisuje što sustav radi s motrišta vanjskog promatrača.



**Slika 1.1. Dijagram slučajeva korištenja**

Na dijagramu slučajeva korištenja prikazan je sustav Autoškola i njegovo ponašanje te funkcionalnosti. Dizajnira se u ranijoj fazi izrade projekta s ciljem prikazivanja interakcija različitih uloga korisnika i sustava. U poslovnom smislu, dijagram slučajeva korištenja služi za prikaz i specificiranje funkcionalnosti sustava iz korisničke perspektive te doprinosi mogućnosti pregovaranju s klijentom s ciljem poboljšavanja i modifikacije sustava i korisničkih zahtjeva.

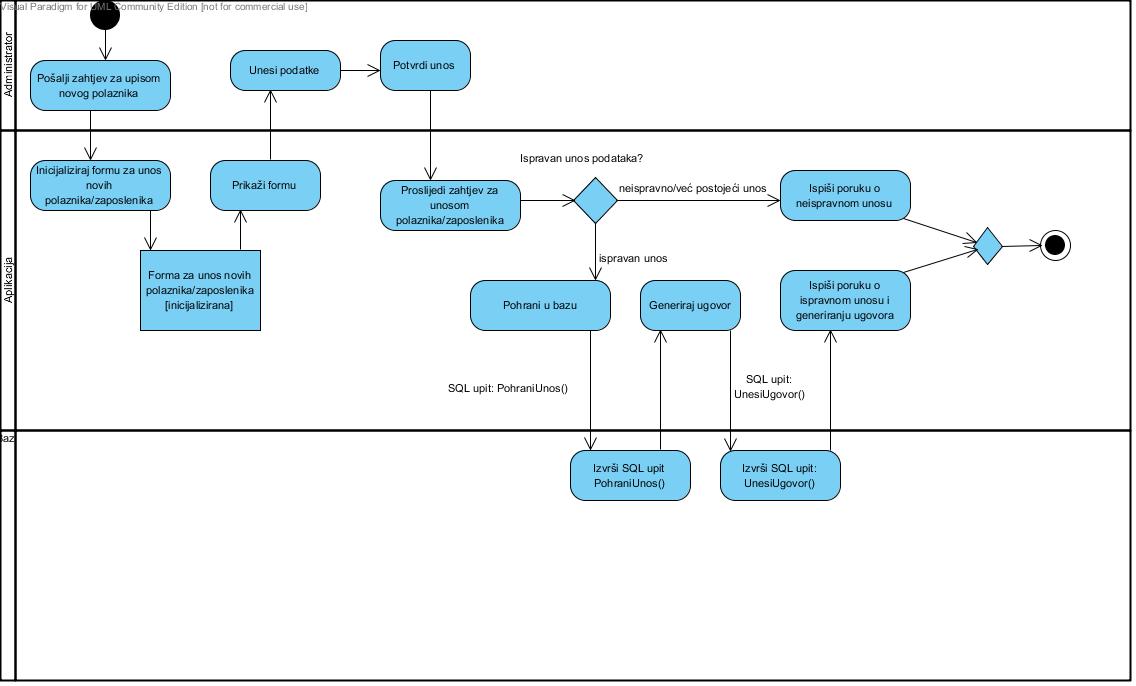
Uloga korisnika, koji s obzirom na vrstu aktivnosti koje obavlja, u sustavu može biti administrator ili instruktor/predavač.

Funkcionalnost prijave u sustav obavezno uključuje prijavu korisnika s pripadajućim jedinstvenim podacima – korisničkim imenom i lozinkom. Nabrojat ćemo sve funkcionalnosti administratora: *Upiši polaznika s mogućnošću generiranja ugovora* i *Ažuriraj polaznika s mogućnošću generiranja ugovora*. Tim su funkcionalnostima omogućene aktivnosti: dodavanja novog polaznika, uređivanje odnosno promjena podataka polaznika. Pomoću funkcionalnosti *Generiranje ugovora* se generira ugovor i upisuje se polaznik u registar autoškole.

Funkcionalnost kojom upravlja instruktor/predavač predstavlja *Evidentiraj aktivnosti polaznika* te *Izdaj potvrda o položenom vozačkom ispitu.* Prvom funkcionalnošću vodi se evidencija svakog polaznika posebno, za svaki odrađeni sat vožnje, dok druga funkcionalnost ostvaruje izdavanje potvrde u obliku MS Office Word dokumenta s pripadajućim generiranim podacima o polazniku koji je položio sve potrebne ispite te završio obuku u autoškoli.

## Dijagrami aktivnosti

* + 1. Upiši polaznika/zaposlenika + generiraj ugovor

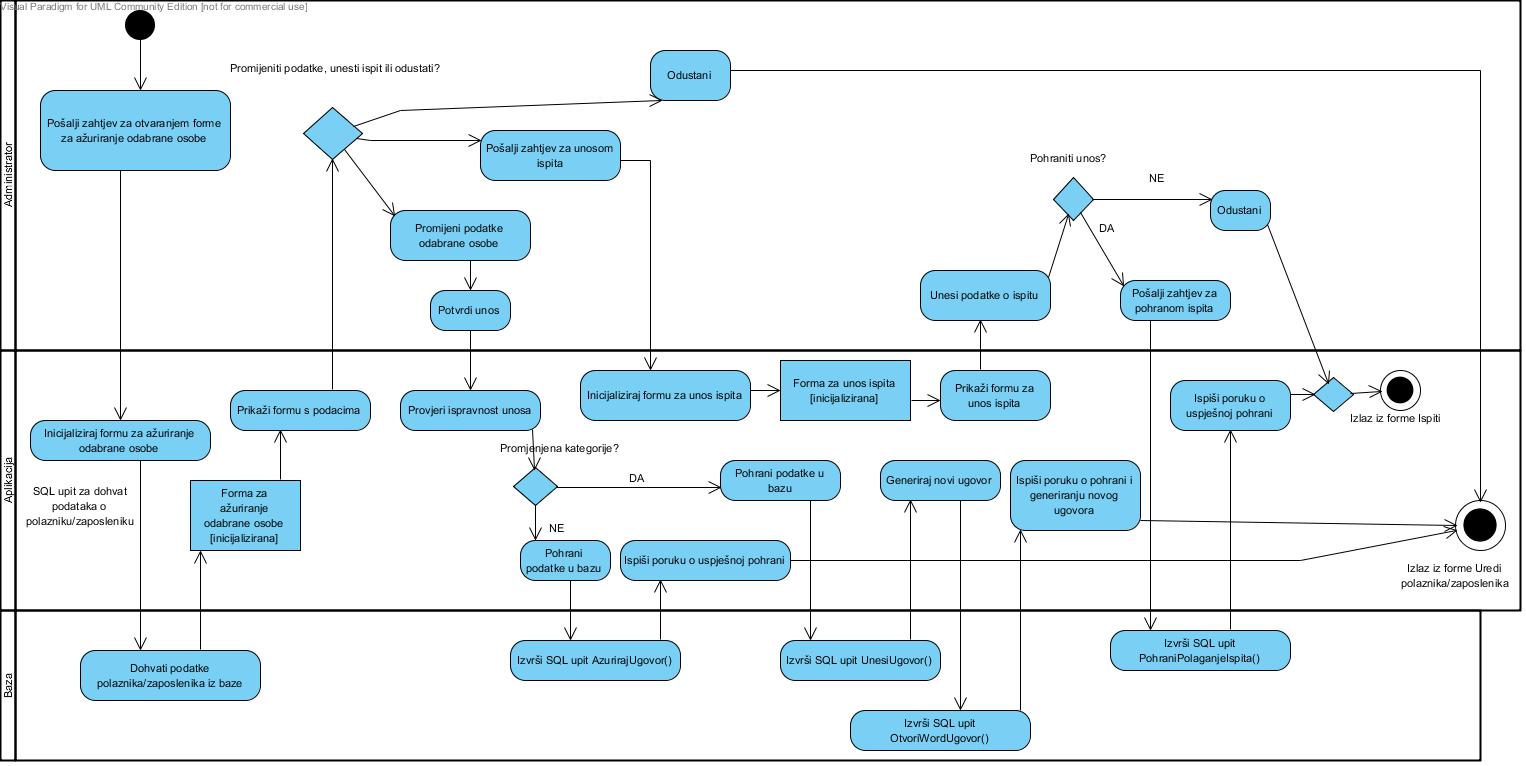


**Slika 1.2. Dijagram *Upiši polaznika/zaposlenika + generiraj ugovor***

Administrator prvenstveno šalje zahtjev za upisom novog polaznika u bazu podataka. Aplikacija prikazuje formu koju administrator ispunjava s potrebnim podacima kao što su ime, prezime, OIB, adresa, broj telefona itd. Nakon potvrde unosa aplikacija provjerava ispravnost podataka. Ukoliko su podaci ispravno uneseni, podaci se pohranjuju u bazu podataka te se generira novi ugovor za upisanog polaznika i nakon toga se završava s radom.

Ukoliko uneseni podaci već postoje ili su neispravni, tada aplikacija obavještava porukom administratora i zatvara se.

### Ažuriraj polaznika/zaposlenika

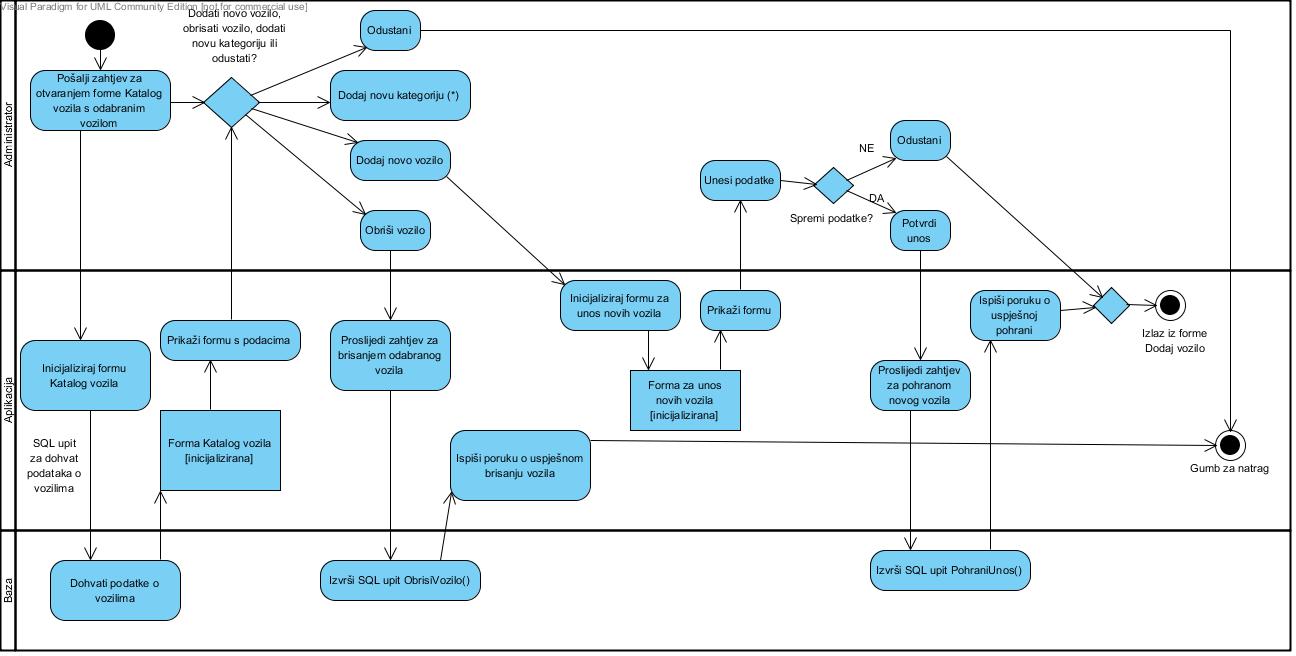


**Slika 1.3. Dijagram *Ažuriraj polaznika/zaposlenika***

Administrator odabire polaznika u listi polaznika koja je generirana kad se forma učitala te nakon toga odabirom ažuriranja polaznika, otvara se forma s postojećim podacima za odabranog polaznika te se podaci mogu mijenjati. Nakon promjene podataka poput imena, prezimena, adrese ili broja telefona, moguće je pohraniti promjene podataka u bazu, ako se kategorija koju polaznik polaže nije promijenila, samo se ispisuje poruka o promjeni podataka. Ako se promijenila kategorija koju polaznik polaže, generira se novi ugovor, ispisuje se poruka o uspješnoj pohrani te se izlazi iz trenutne forme. Time se osigurava ponovni upis polaznika za slučaj ako polaznik želi polagati drugu kategoriju vozila.

Na formi za uređivanje polaznika/zaposlenika, također postoji funkcionalnost unosa i evidencije ispita za odabranog polaznika. Evidentira se da li je polaznik položio ili nije položio ispit na koji je izašao te se navedeno također pohranjuje u bazi podataka, ispisuje se poruka o uspješnosti pohrane te se izlazi iz trenutne forme ili se u suprotnom može odlučiti na nepohranjivanje podataka, čijim završetkom se također izlazi iz trenutne forme bez ikakve pohrane podataka.

### Upravljaj vozilima

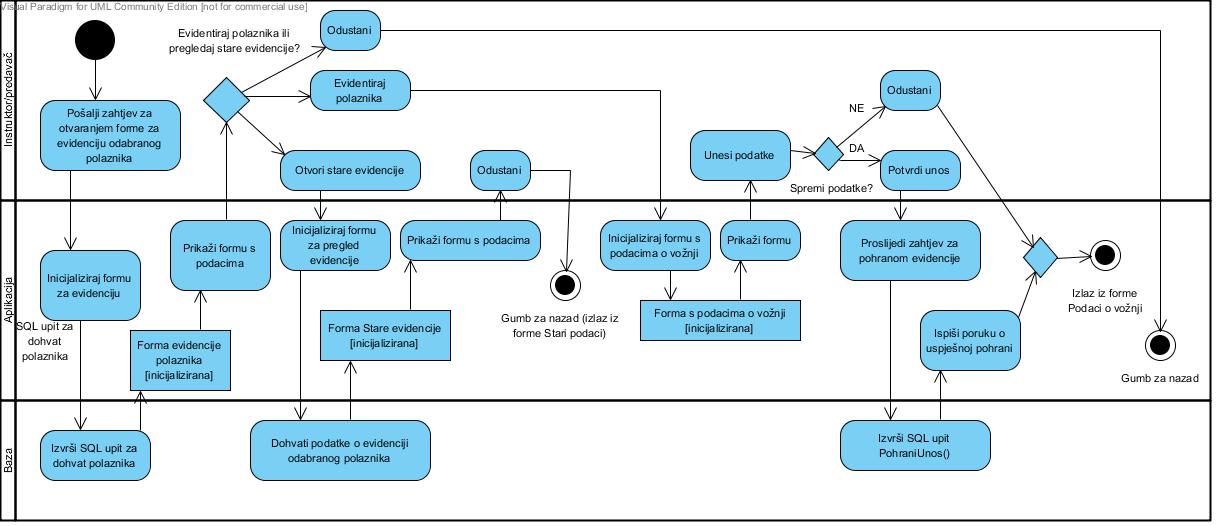


**Slika 1.4. Dijagram *Upravljaj vozilima***

Administrator otvara katalog vozila na formi gdje ima više funkcionalnosti koje može ostvariti. Administrator može dodati vozilo u katalog vozila, otvara formu za unos podataka o novom vozilu, unosi podatke te ih sprema. Nadalje, ispisuje se poruka o ispravnom unosu i pohrani podataka te se tada omogućuje još unosa novih vozila ili se odustajanjem zatvara trenutna forma i vraća se na prethodnu formu. Sljedeća funkcionalnost je brisanje vozila iz baze podataka, odabirom vozila koje se želi brisati u listi vozila te klikom na funkcionalnost brisanja, vozilo se uklanja iz baze podataka te iz liste, odnosno, kataloga vozila te se o navedenome ispisuje poruka.

I zadnja funkcionalnost koju dijagram na slici 1.4. prikazuje je dodavanje nove kategorije vozila u bazu. Klikom na navedeno, otvara se nova forma za unos podataka o novoj kategoriji, ispisuje se poruka o uspješnosti te se može dalje upisivati još novih kategorija ili se po završetku odustaje te se vraća na prethodnu formu, što je istovjetno akcijama koje se izvode za funkcionalnost dodaj novo vozilo te stoga nisu sve akcije prikazane u dijagramu (prikazano „(\*)“ oznakom iza akcije dodavanje nove kategorije vozila). Navedeni izostanak akcija za tu funkcionalnost je jednako implementirana u programskom kodu aplikacije kao funkcionalnost dodavanja vozila.

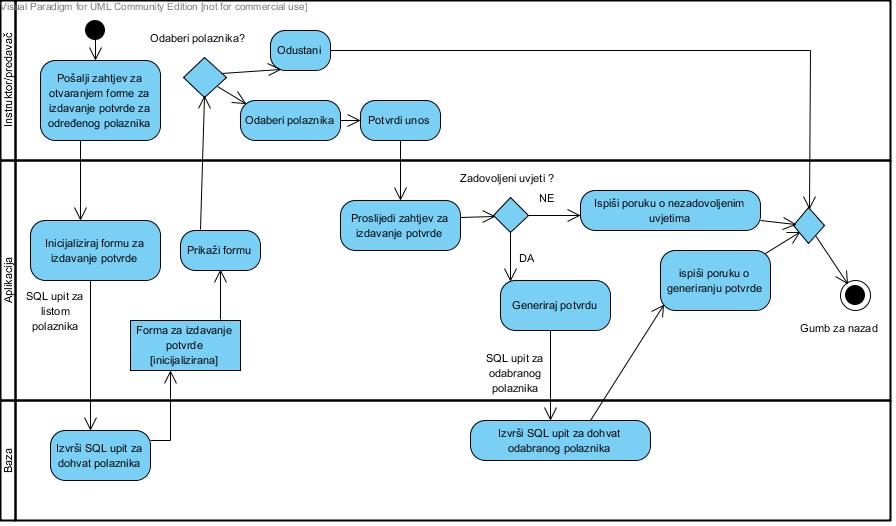
### Evidentiraj aktivnosti polaznika



**Slika 1.5. Dijagram *Evidentiraj aktivnosti polaznika***

Instruktor ili predavač otvara listu polaznika, odabire određenog polaznika te za njega unosi evidenciju podataka o svakom održanom satu vožnje. Tada se otvara nova forma u kojoj se odabire datum, vrijeme početka i kraja sata i ispisuje se vrijeme koliko je u tom danu polaznik odvozio sati, promjene se pohranjuju u bazu podataka te se ispisuje poruka o uspješnoj pohrani. Postupak se može ponoviti više puta, koliko je potrebno. Na kraju, za izlaz iz trenutne forme, odabire se odustajanje. Druga funkcionalnost predstavlja pregled starih evidencija za odabranog polaznika. Navedenim odabirom, otvara se forma s pregledom evidencije o polazniku, njegovom instruktoru i svih odrađenih satova vožnje, što uključuje i njihovu ukupnu sumu. S gumbom za natrag vraća se na početnu formu.

### Izdaj potvrdu o položenom vozačkom ispitu



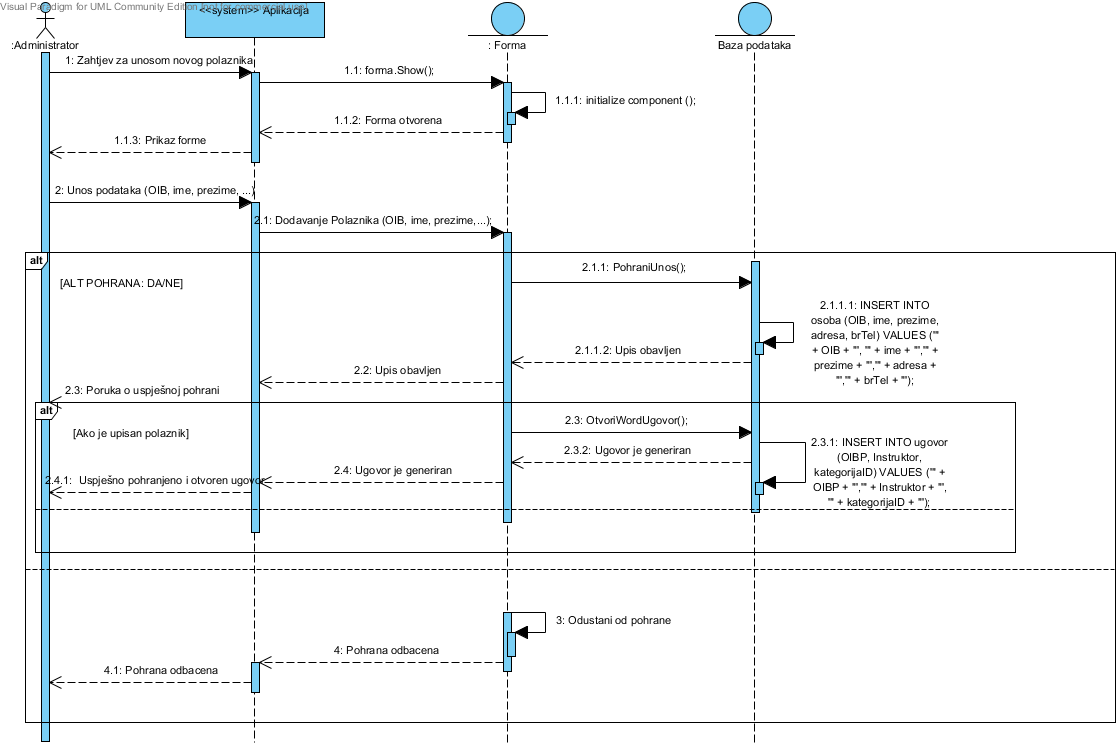
**Slika1. 6. Dijagram *Izdaj potvrdu o položenom vozačkom ispitu***

Za aktivnost Izdaj potvrdu o položenom vozačkom ispitu, instruktor ili predavač otvara popis polaznika gdje se odabire jedan polaznik te se za njega generira potvrda o položenom vozačkom ispitu, arhivira se u bazi podataka te se ispisuje poruka o uspješnosti istog. Ujedno se izvršava i provjera da li je polaznik zadovoljio sve uvjete, odnosno, je li položio sve ispite, ako nije, ispisuje se poruka i gumbom za nazad izlazi se iz forme.



## Dijagrami slijeda

### Upiši polaznika/zaposlenika



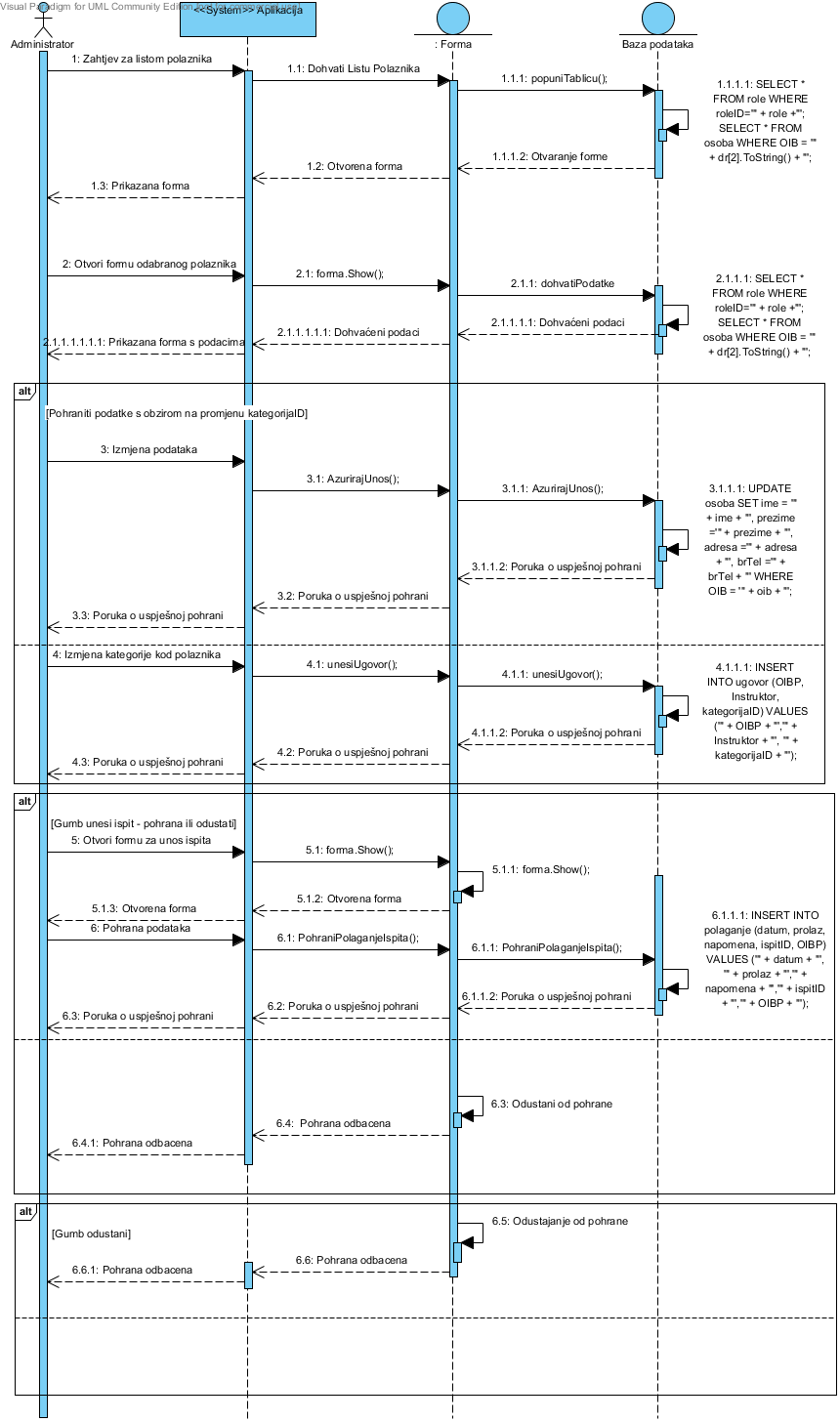
**Slika 1.7. Dijagram *Upiši polaznika/zaposlenika***

Dijagram Upiši polaznika/zaposlenika (PZ) predstavlja slijed realizacije funkcionalnosti za upisivanje podataka o novom PZ-u za koju je zadužen administrator.

Nakon što administrator pošalje zahtjev za unosom novog PZ-a, on se inicijalizira i forma se prikazuje administratoru.

U navedenom slučaju, po slanju zahtjeva za formom unosa i prikazu iste, unosi se novi PZ. Sustav prosljeđuje unesene podatke bazi, ako je unesen novi zaposlenik, a ako je unesen novi polaznik, generira se i otvara dokument ugovora za unesenog polaznika. Povratna poruka s unesenim podacima i obavijesti potom se prosljeđuju i prikazuju administratoru. Ako administrator odustane od pohrane, izlazi se iz navedene forme.

### Ažuriraj polaznika/zaposlenika



**Slika1. 8. Dijagram *Ažuriraj polaznika/zaposlenika***

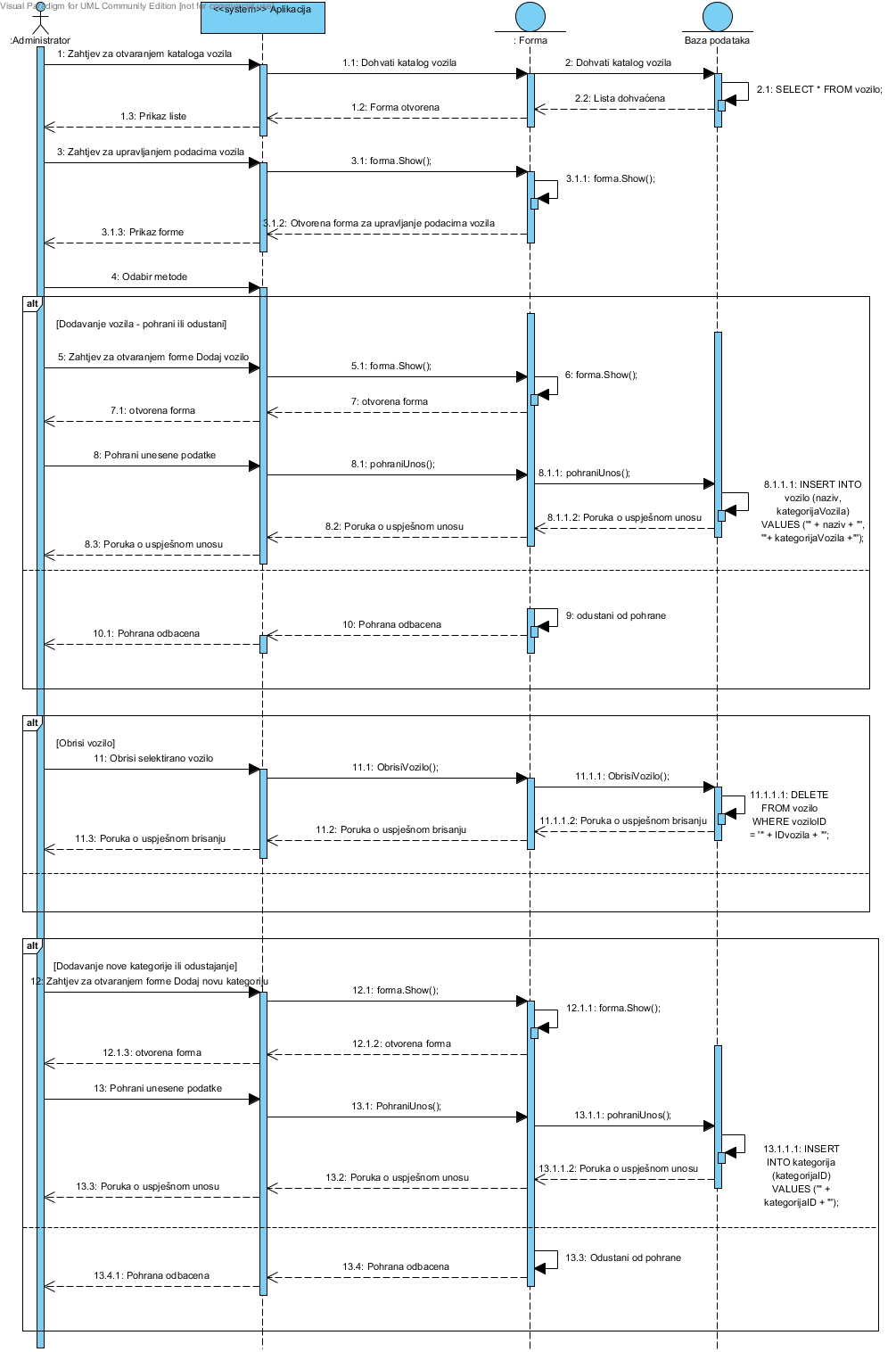
Dijagram Ažuriraj polaznika/zaposlenika (PZ) predstavlja slijed realizacije funkcionalnosti za ažuriranje podataka o postojećem PZ-u za koju je također zadužen administrator.

Administrator šalje zahtjev za listom PZ-a, inicijalizira se i prikazuje se forma administratoru, a prije toga se dohvaćaju potrebni podaci iz baze.

Zatim se odabire određeni PZ te se otvara forma za ažuriranje podataka o određenom PZ-u. Kada se podaci promijene, šalje se zahtjev za ažuriranjem podataka. Ako je promijenjen podatak o kategoriji, tada se generira i otvara novi ugovor za polaznika te se ispisuje poruka o uspješnoj pohrani i generiranju ugovora. Ako su promijenjeni podaci bez promjene podataka o kategoriji, tada se podaci samo ažuriraju, odnosno, pohranjuju u bazu te slijedi poruka o uspješnoj pohrani.

Ako se odabere funkcionalnost upisati podatke o ispitima, otvara se forma za unos podataka o ispitu te se pohranjuju podaci u bazi podataka i vraća se poruka o uspješnoj pohrani ili se odustaje bez pohrane podataka i vraća se na prethodnu formu.

### Upravljaj vozilima



**Slika 1.9. Dijagram *Upravljaj vozilima***

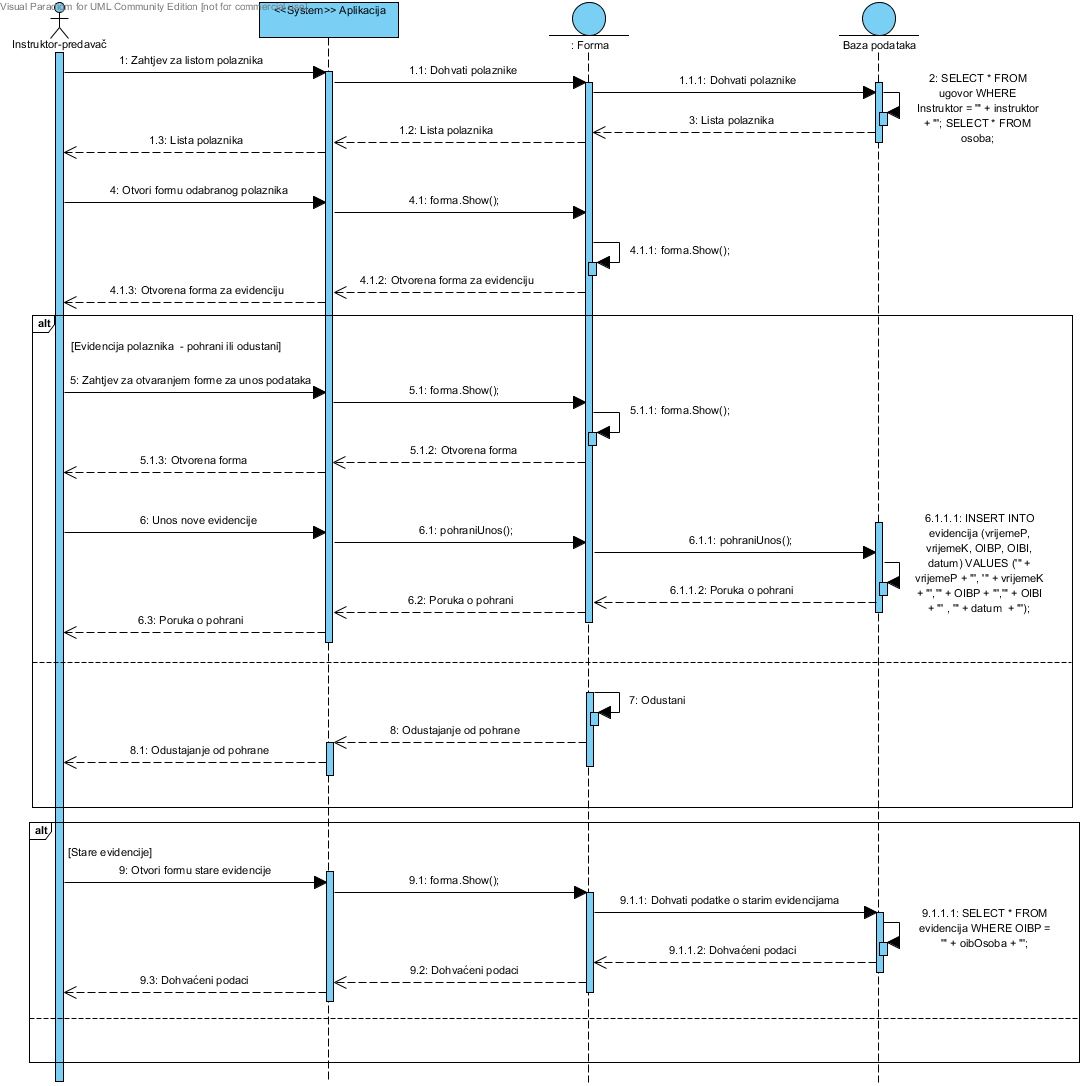
Dijagram Upravljaj vozilima koristi Administrator koji šalje zahtjev za otvaranjem forme za prikaz kataloga vozila. Nakon toga odabire se jedna od tri moguće alternative.

Prva alternativa omogućuje dodavanje novog vozila. Šalje se zahtjev za otvaranjem forme za unos novog vozila te se nakon unesenih podataka šalje zahtjev za pohranom novog vozila te se unos pohranjuje u bazu i ispisuje se poruka o uspješnoj pohrani ili se odustaje bez pohrane.

Na isti način kao i funkcionalnost dodavanja novog vozila, funkcionira i dodavanje nove kategorije, stoga nije potrebno sve korake ponovno objašnjavati.

Treća alternativa i funkcionalnost predstavlja brisanje odabranog vozila gdje se odabire određeno vozilo te se poziva navedena funkcionalnost prikazana gumbom „Obriši“. Poziva se funkcija brisanja vozila te se ispisuje poruka o brisanju vozila i lista vozila se ažurira.

### Evidentiraj aktivnosti polaznika

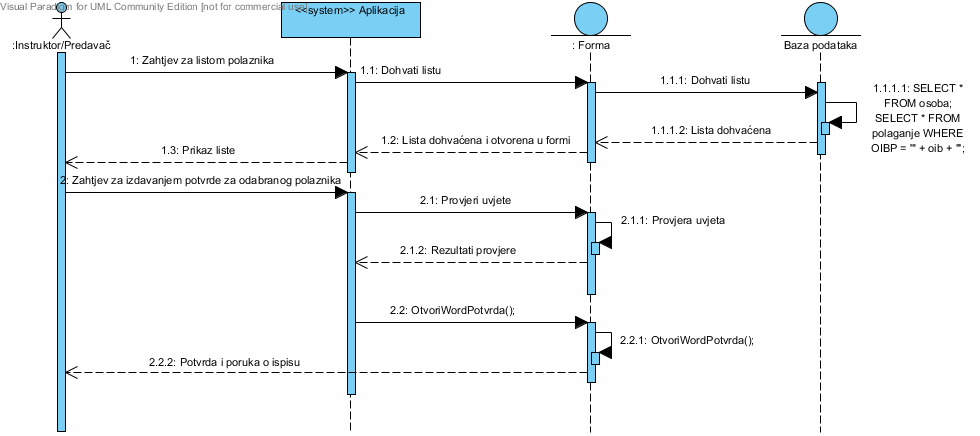


**Slika 1.10. Dijagram *Evidentiraj aktivnosti polaznika***

Evidentiraj aktivnosti polaznika predstavlja dijagram slijeda koji koristi instruktor ili predavač. Funkcionalnosti koje ostvaruje su evidencija polaznika i prikaz stare evidencije.

Za odabir evidencije polaznika, otvara se nova forma u koju se unose podaci o evidenciji određenog polaznika, isti se pohranjuju te se vraća poruka o pohrani evidencije. Ili se odustaje od unosa evidencije te se izlazi iz forme. Za pregled stare evidencije otvara se forma, dohvaćaju se podaci o starim evidencijama za određenog polaznika, prikazuje se evidencija u listi, odnosno, tablici te se u istome prikazuje ukupan broj odvoženih sati.

### Izdaj potvrdu o položenom vozačkom ispitu



**Slika1. 11. Dijagram *Izdaj potvrdu o položenom vozačkom ispitu***

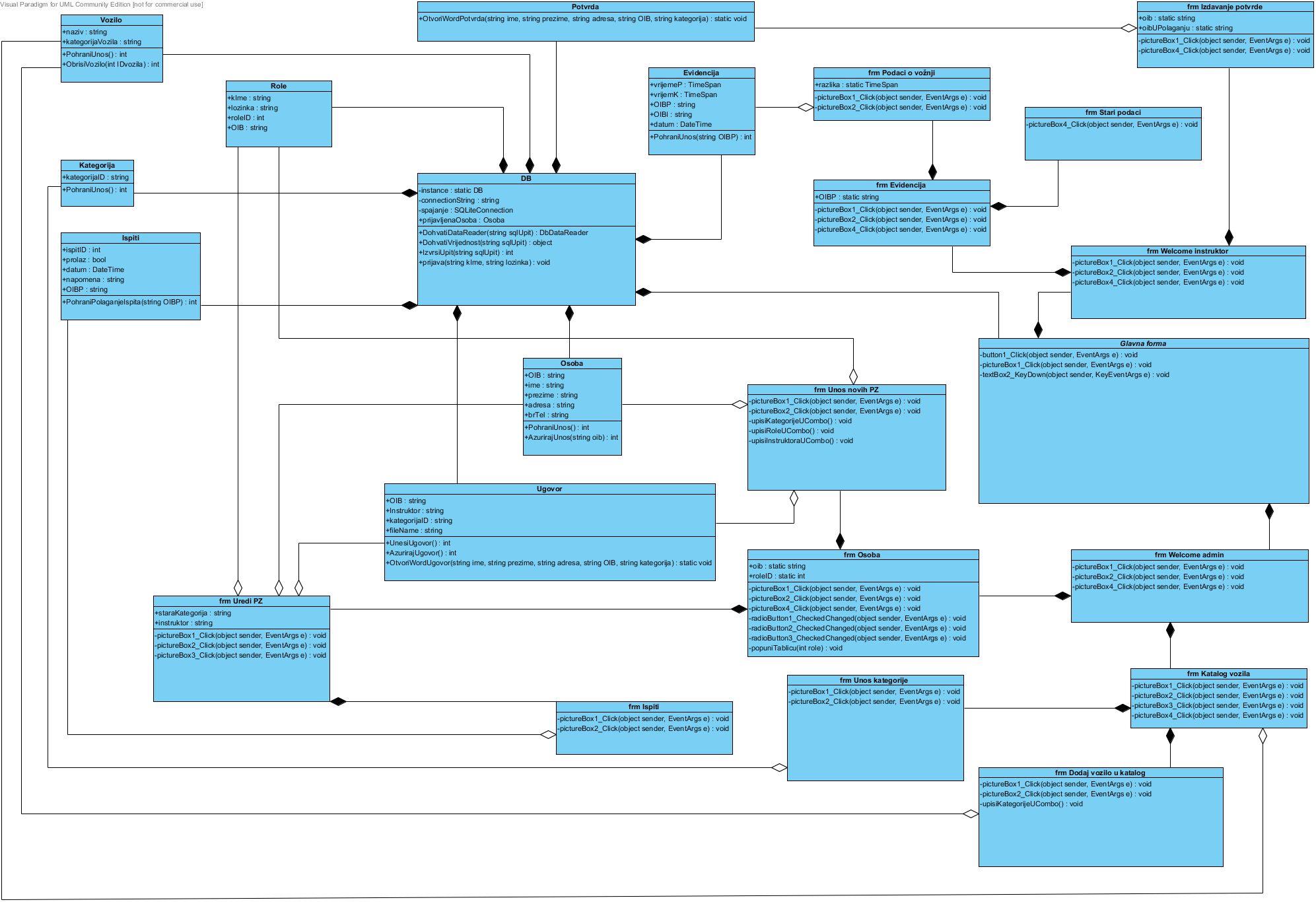
U dijagramu slijeda Izdaj potvrdu o položenom vozačkom ispitu prikazane su slijedne funkcionalnosti kojima upravlja Instruktor ili predavač.

Instruktor ili predavač šalje zahtjev za otvaranjem forme te se dohvaćaju podaci o osobama koje imaju položen ispit vožnje te se prikazuju u listi, odnosno, tablici. Nakon toga odabire se jedan polaznik za kojeg se provjeravaju uvjeti da li je položio sve ispite te tako osigurao izdavanje potvrde koja se generira i otvara te se prikazuje poruka o ispisu potvrde.

## 1.4. Dijagram klasa

Dijagram klasa je strukturni statički dijagram koji prikazuje klase, atribute i operacije te uz to i veze između klasa: asocijacije, kompozicije i agregacije te generalizacije.

Klasa predstavlja skup objekata koji posjeduju iste atribute, metode, odnose, odnosno strukturu, ponašanje i semantiku. Nadalje, dijagram klasa u objektno orijentiranom razvoju služi za vizualno modeliranje strukture klasa te ćemo modeliranjem dijagrama klasa nastojati obuhvatiti sve klase s pripadajućim atributima te operacijama koje ćemo realizirati u izradi aplikacije.



**Slika 1.12. Dijagram klasa**

## 1.5. ERA model

ERA model je grafički prikaz logičkog modela podataka. Osnovni elementi ERA modela su entiteti, veze i atributi. Svaki entitet može biti jak ili slab (neovisan ili ovisan). Jaki entitet je nezavisan od ostalih entiteta, dok slabi entitet egzistencijalno i/ili identifikacijski ovisi o jakom.. Za lakše praćenje relacija u nastavku ćemo navesti relacijsku shemu.

CREATE TABLE [osoba] (

[OIB] CHAR(11) NOT NULL ON CONFLICT ABORT,

[ime] VARCHAR(30) NOT NULL,

[prezime] VARCHAR(40) NOT NULL,

[adresa] VARCHAR(100) NOT NULL,

[brTel] CHAR(10),

CONSTRAINT [] PRIMARY KEY ([OIB]));

CREATE TABLE [evidencija] (

[vrijemeP] TIME NOT NULL,

[vrijemeK] TIME NOT NULL,

[OIBP] CHAR(11) CONSTRAINT [OIBP] REFERENCES [osoba]([OIB]),

[OIBI] CHAR(11) CONSTRAINT [OIBI] REFERENCES [osoba]([OIB]),

[datum] DATETEXT NOT NULL);

CREATE TABLE [ispiti] (

[ispitID] INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

[naziv] VARCHAR(30) NOT NULL);

CREATE TABLE [kategorija] (

[kategorijaID] CHAR NOT NULL CONSTRAINT [unq\_kategorijaID] UNIQUE,

CONSTRAINT [sqlite\_autoindex\_kategorija\_1] PRIMARY KEY ([kategorijaID]));

CREATE TABLE [kreiranjeRola] (

[roleID] INTEGER NOT NULL,

[naziv] VARCHAR NOT NULL);

CREATE TABLE [polaganje] (

[datum] DATETEXT NOT NULL,

[prolaz] BOOL NOT NULL,

[napomena] TEXT,

[ispitID] INTEGER CONSTRAINT [ispitID] REFERENCES [ispiti]([ispitID]),

[OIBP] CHAR(11) CONSTRAINT [OIBP] REFERENCES [osoba]([OIB]));

CREATE TABLE [role] (

[kIme] VARCHAR2(255),

[lozinka] CHAR(4),

[OIB] CHAR(11) NOT NULL CONSTRAINT [OIB] REFERENCES [osoba]([OIB]),

[roleID] INTEGER NOT NULL);

CREATE TABLE [ugovor] (

[ugovorID] INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

[datumUpisa] DATE NOT NULL DEFAULT CURRENT\_DATE,

[OIBP] CHAR(11) CONSTRAINT [OIBP] REFERENCES [osoba]([OIB]),

[Instruktor] VARCHAR,

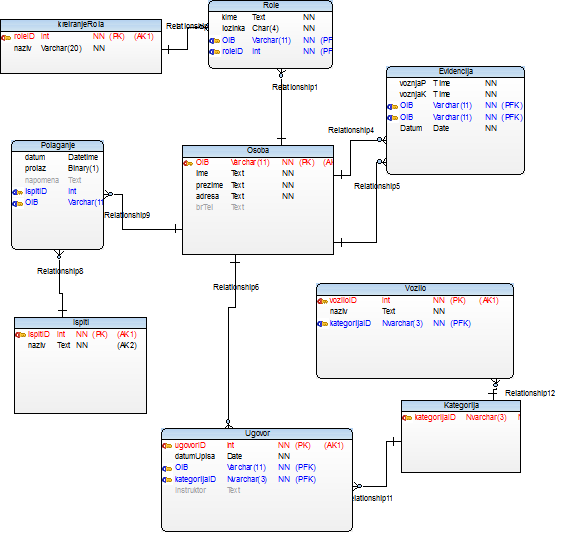
[kategorijaID] CHAR CONSTRAINT [kategorijaID] REFERENCES [kategorija]([kategorijaID]));

CREATE TABLE [vozilo] (

[voziloID] INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

[naziv] VARCHAR(30) NOT NULL,

[kategorijaVozila] CHAR(3) CONSTRAINT [kategorijaVozila] REFERENCES [kategorija]([kategorijaID]));



**Slika 1.13. ERA model**

## 2. Podaci o provedbi projekta

Josipa Tadić kao voditelj tima snosi odgovornost za izradu i razvijanje projektnog plana. Ugovara i organizira sastanke na kojim dodjeljuje zadatke ostalim članovima. Također, dužna je održati sklad u timu i eventualne probleme riješiti. U početnoj fazi izrađuje terminski plan projekta kako bi timu na pregledan način prikazala korake u izradi projekta s istaknutim rokovima.

Nadja Kosojević prati učinkovitost i djelotvornost programa kao i sveobuhvatnu analizu. Pomaže prilikom planiranja te izradi slučajeva korištenja i ERA modela. Odgovorna je za izradu specifikacija zahtjeva prema IEEE standardu.

Svi članovi međusobno surađuju i nadopunjuju se u radu kako bi stekli što više iskustva u izradi projekta. Opisu se pridružuje ranije navedena Tablica 3 - *Planiranje ljudskih resursa* u projektnoj dokumentaciji.