

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Andrea Međan

PRIMJER BAZE PODATAKA U SUSTAVU SQLITE

ZAVRŠNI RAD

Varaždin, 2014.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Andrea Međan

Izvanredni student

Broj indeksa: K-37061/08-I

Smjer: Primjena informacijske tehnologije u poslovanju

PRIMJER BAZE PODATAKA U SUSTAVU SQLITE

ZAVRŠNI RAD

Mentor:

Doc. dr. sc. Markus Schatten

Varaždin, 2014.

Sadržaj

1. UVOD.....	1
2. SQLITE	2
2.1. Kreiranje baze podataka	2
2.2. Kreiranje tablica	4
2.3. Upravljanje bazom podataka	7
3. MODEL BAZE PODATAKA	10
3.1. Era model baze podataka.....	10
4. OPIS APLIKACIJSKE DOMENE.....	13
4.1. Uvod u PHP	13
4.2. Spajanje na bazu podataka	14
4.3. Ispis podataka iz baze podataka	15
4.3. Unos novih podataka u bazu podataka.....	21
4.4. Brisanje podataka iz baze podataka.....	22
5. OPIS IMPLEMENTACIJE	24
5.1. Implementacija uloga.....	24
5.2. Kontrola zapisa prijavljenih i odjavljenih uloga	32
6. PRIMJERI KORIŠTENJA.....	36
7. ZAKLJUČAK	38
Literatura	39

Prilog A: CD sa popisom programskih kodova po datotekama

1. UVOD

U današnje vrijeme ne postoji institucija, ustanova ili bilo koje drugo poduzeće koje si poslovanje nije olakšalo korištenjem baze podataka. Cilj ovog rada je prikazati izradu baze podataka jedne autoškole u SQLite sustavu za upravljanje relacijskom bazom podataka (Relation Database Managment System – RDBMS). Izradom takve baze podataka cilj je skratiti vrijeme pretrage polaznika, instruktora i ispitivača, imati jasan pregled nad time koji instruktor raspolaže kojim vozilom, koji ispitivač dolazi na polaganje pojedinog polaznika, lakša evidencija nad uplatama ili eventualnim dugovanjima, unos novih podataka, brisanje starih i nepotrebnih ili pak izmjena postojećih podataka. Kroz pet poglavlja prikazat će se način izrade same baze podataka i aplikacije za korištenje baze podataka. Prikazat će se kreiranje baze podataka, kreiranje tablica, te njihovo međusobno povezivanje. Objasnit će se osnovne naredbe za upravljanje bazom podataka, koje će se konkretno koristiti kod unosa podataka u tablicu, brisanje podataka iz tablice po zadanom uvjetu, promjena nekih podataka ili pak brisanje cijele tablice. Izradom ERA modela prikazat će se veze i odnosi među tablicama. Definirat će se tipovi podataka koji se unose u tablice i njihova ograničenja. Prikazat će se i izrada aplikacije u PHP programskom jeziku, koja će biti korisničko sučelje za korištenje i upravljanje nad bazom podataka unutar prave autoškole, na način da će se kreirati forme za unos podataka, dodavati opcije za izmjenu podataka ili njihovo brisanje, s ciljem da aplikacija bude stilski dotjerana, jednostavna za korištenje, funkcionalna i upotrebljiva u poslovanju jedne autoškole. Napravit će se implementacija aplikacije na način da će se dodati određene uloge koje će imati pristup aplikaciji korištenjem korisničkog imena i lozinke. Kreirat će se zapisnik koji će bilježiti vrijeme prijave i odjave uloga sa pristupom sustavu, te će ukratko biti opisani primjeri korištenja aplikacije.

2. SQLITE

“SQLite je javna domena paketa software-a koja osigurava upravljanje sustavom relacijskih baza podataka” [Kreibich, 2010, str 1.]. Za potrebe izrade baze podataka „Autoškola“, najprije će se instalirati SQLite program koji prima SQL naredbe iz interaktivnih upita te propušta naredbe i obrađuje ih. „SQL se sastoji od određenog broja naredbi, s time da se svaka naredba sastoji od određenih dijelova“ [Rabuzin, 2011, str 2.]. SQLite program omogućuje jednostavnu liniju zapovijedanja pomoću sqlite3.exe datoteke za Windows operacijski sustav, koja je sredstvo interakcije programa u kojem će se izdavati naredbe za izradu baze podataka ali i tablica, uzastopnim linijama naredbi [SQLite, dostupno 15.09.2014. na <http://www.sqlite.org>].

2.1. Kreiranje baze podataka

Za kreiranje baze podataka koristit će se Command Prompt, sučelje za unošenje naredbi, odnosno za unošenje SQLite sintakse¹ koja se temelji na maloj C programskoj biblioteci. Za pokretanje SQLite3 programa upisat će se „sqlite3“ i naziv baze podataka koja se želi kreirati. Na isti se način baza podataka i otvara ukoliko već postoji baza podataka koja se želi nadopuniti, izmijeniti, obrisati neke tablice, redove i sl. U slici ispod (*Slika 2.1. Kreiranje baze podataka*) prikazat će se kreiranje baze podataka.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Andrea>cd desktop
C:\Users\Andrea\Desktop>cd baza
C:\Users\Andrea\Desktop\baza>sqlite3 autoskola.db_
```

Slika 2.1. Kreiranje baze podataka

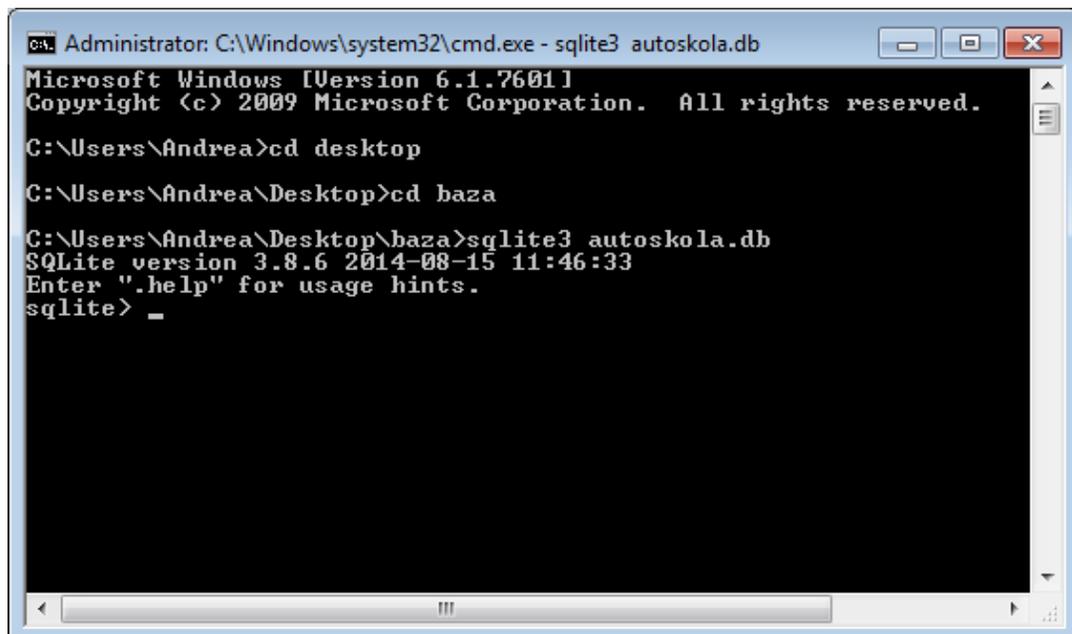
¹ Poruke prenešene preko komunikacijske veze ili računalnog programa koji se prevodi.

C:\Users\Andrea>cd desktop je linija naredbe koja predstavlja, odnosno govori lokaciju na koju će se spremi baza podataka, ili ukoliko baza podataka već postoji pa se samo želi urediti, na kojoj se lokaciji baza podataka nalazi, u ovom slučaju to će biti desktop računala.

C:\Users\Andrea\Desktop>cd baza je linija naredbe koja označava naziv mape za spremanje baze podataka ili njezino otvaranje radi mogućih izmjena.

C:\Users\Andrea\Desktop\baza>sqlite3 autoskola.db je linija naredbe koja u mapi imena "baza" putem SQLite3 programa kreira bazu podataka "autoskola.db".

Nakon što će se baza podataka kreirati, pojavit će se Dot-Command² (točka naredba) `.help` koja prikazuje sve instrukcije SQLite-a sa kratkim pojašnjenjem (*Slika 2.3. Dot-Command .help*) i oznaka `sqlite>` govori da je SQLite3 spreman primiti naredbu (*Slika 2.2. Kreiranje baze podataka 2*).



```
Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Andrea>cd desktop

C:\Users\Andrea\Desktop>cd baza

C:\Users\Andrea\Desktop\baza>sqlite3 autoskola.db
SQLite version 3.8.6 2014-08-15 11:46:33
Enter ".help" for usage hints.
sqlite> _
```

Slika 2.2. Kreiranje baze podataka 2

² Naredbe koje kontroliraju izlazno formatiranje i pružaju brojne korisne značajke i moraju biti napisane unutar jedne linije.

```

Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
SQLite version 3.8.6 2014-08-15 11:46:33
Enter ".help" for usage hints.
sqlite> .help
.backup ?DB? FILE          Backup DB (default "main") to FILE
.bail on|off              Stop after hitting an error. Default OFF
.clone NEWDB              Clone data into NEWDB from the existing database
.databases                List names and files of attached databases
.dump ?TABLE? ...        Dump the database in an SQL text format
                          If TABLE specified, only dump tables matching
                          LIKE pattern TABLE.
.echo on|off              Turn command echo on or off
.exit on|off              Enable or disable automatic EXPLAIN QUERY PLAN
.exit                    Exit this program
.explain ?on|off?         Turn output mode suitable for EXPLAIN on or off.
                          With no args, it turns EXPLAIN on.
.fullschema               Show schema and the content of sqlite_stat tables
.headers on|off           Turn display of headers on or off
.help                     Show this message
.import FILE TABLE       Import data from FILE into TABLE
.indices ?TABLE?          Show names of all indices
                          If TABLE specified, only show indices for tables
                          matching LIKE pattern TABLE.
.load FILE ?ENTRY?        Load an extension library
.log FILE?on|off          Turn logging on or off. FILE can be stderr/stdout
.mode MODE ?TABLE?        Set output mode where MODE is one of:
                          csv          Comma-separated values
                          column       Left-aligned columns. (See .width)
                          html         HTML <table> code
                          insert       SQL insert statements for TABLE
                          line         One value per line
                          list         Values delimited by .separator string
                          tabs         Tab-separated values

```

Slika 2.3. Dot-Command .help

2.2. Kreiranje tablica

Ključna naredba za kreiranje tablice u SQLite sustavu je CREATE TABLE. Kreiranje tablice uključuje imenovanje tablice i definiranje stupaca, te će se svakom stupcu dodijeliti vrsta podataka. Ovim radom predstaviti će se 12 tablica za koje se smatra da ih jedna autoškola treba imati i koje su međusobno povezane. To su tablice „Polaznik“, „Instruktor“, „Ispitivač“, „Vozilo“, „Autokuća“, „Autoservis“, „Dodatni sati“, „Polaganje“, „Propisi“, „Vožnja“, „Plaćanje“ i „Uplate“ Kao što je prikazano na slici ispod (*Slika 2.4. Kreiranje tablice „Autokuća“*), za tablicu „Autokuća“ odredit će se stupac ID koji će biti primarni ključ³, ključnom riječi PRIMARY KEY AUTOINCREMENT, koja će definirati da se primarni ključ sam automatski dodjeljuje prilikom unosa nove autokuće. Važno je napomenuti kako se AUTOINCREMENT može koristiti samo kod integer (cjelobrojnog) tipa podataka [SQLite SQL Database Engine, dostupno 15.9.2014. na <http://www.tutorialspoint.com/sqlite/>], kako je ujedno i definirano prilikom kreiranja tablice. Definirati će se polje Naziv kojem će se odrediti ograničenje TEXT NOT NULL, koje pokazuje kako polje Naziv ne može biti nula. Polju Adresa odredit će se dužina na 50 znakova (CHAR (50)), a za polje Telefon postaviti će se ograničenje TEXT NOT NULL.

³ Skup polja koji jedinstveno identificiraju svaki zapis koji je spremljen u tablici, to je jedinstveni identifikator za svaki redak.

```
Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Andrea>cd desktop
C:\Users\Andrea\Desktop>cd baza
C:\Users\Andrea\Desktop\baza>sqlite3 autoskola.db
SQLite version 3.8.6 2014-08-15 11:46:33
Enter ".help" for usage hints.
sqlite> create table AUTOKUCA
...> <ID INT PRIMARY KEY NOT NULL,
...> Naziv TEXT NOT NULL,
...> Adresa CHAR(50),
...> Telefon TEXT NOT NULL);
sqlite> insert into autokuca (id, naziv, adresa, telefon)
...> values (1, "Opel Knež", "Laniste 17, Koprivnica", "048/789-987");
```

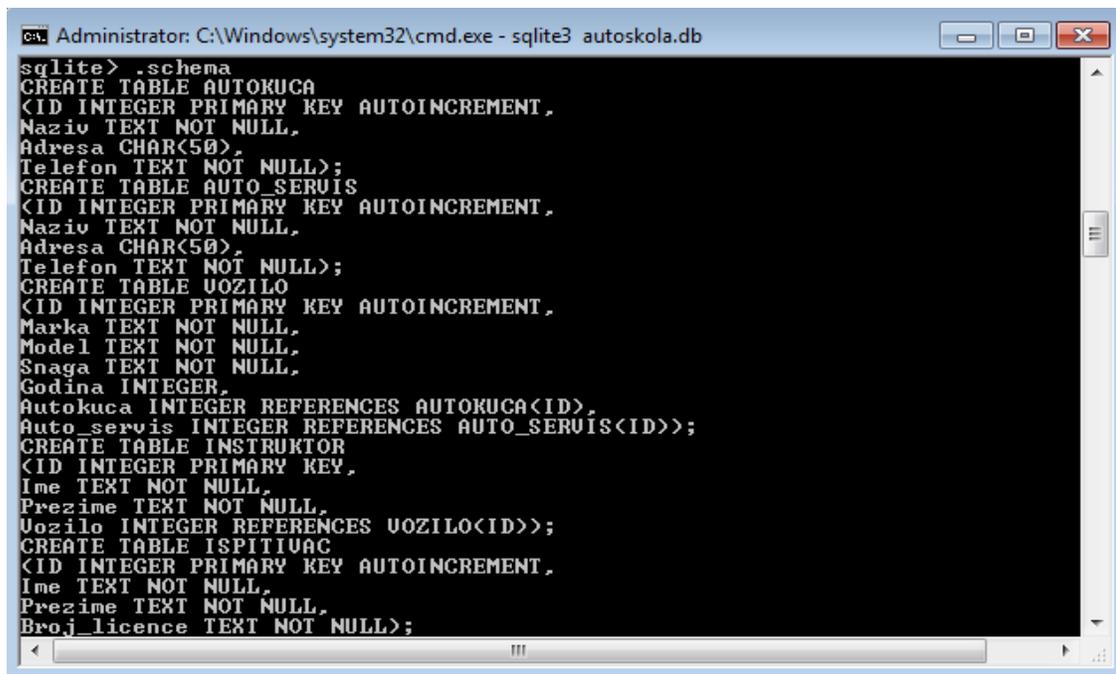
Slika 2.4. Kreiranje tablice „Autokuća“

Kao što je prikazano na slici ispod (Slika 2.5. Unos podataka u tablicu „Vozilo“), kreirati će se tablica „Vozilo“, kojoj će stupac ID biti primarni ključ i po tipu podataka je integer, postavljen da se automatski postavlja kao što je već ranije spomenuto pomoću riječi AUTOINCREMENT, stupcu Marka dodijelit će se ograničenje TEXT NOT NULL, kao i za stupac Model i Snaga. To ograničenje „sprječava izvršavanje naredbe kojom bi vrijednost stupca postala NULL“ [Rabuzin, 2011, str 26.]. Stupac Godina će po tipu podataka biti integer. Stupac Autokuća će biti povezan sa tablicom „Autokuća“ preko primarnog ključa tablice „Autokuća“, isto kao i stupac Auto_servis sa tablicom „Auto_servis“. Unos podataka u bazu podataka napraviti će se pomoću ključne riječi (naredbe) INSERT. Sintaksa je takva da će se pomoću ključne riječi INSERT upisati tablica kojoj će se dodati nove vrijednosti, odabrat će se polja (marka, model, snaga, godina, autokuća, auto_servis), te će se upisati vrijednosti koje se žele upisati u tablicu, na način da se poštuje redoslijed stupaca prilikom upisivanja vrijednosti.

```
Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
sqlite> create table UOZILO
...> <ID INTEGER PRIMARY KEY AUTOINCREMENT,
...> Marka TEXT NOT NULL,
...> Model TEXT NOT NULL,
...> Snaga TEXT NOT NULL,
...> Godina INTEGER,
...> Autokuca INTEGER REFERENCES AUTOKUCA(ID),
...> Auto_servis INTEGER REFERENCES AUTO_SERVIS(ID));
sqlite> insert into Uozilo (marka, model, snaga, godina, autokuca, auto_servis)
...> values ("Opel Corsa", "1,2", "74kW", 2012, 1, 2)
sqlite> insert into Uozilo (marka, model, snaga, godina, autokuca, auto_servis)
...> values ("Peugeot", "206", "55kW", 2010, 2, 3)
sqlite> insert into Uozilo (marka, model, snaga, godina, autokuca, auto_servis)
...> values ("Opel Astra", "1,4", "103kW", 2010, 1, 4)
sqlite> insert into Uozilo (marka, model, snaga, godina, autokuca, auto_servis)
...> values ("Mazda2", "1,3", "55kW", 2011, 3, 1)
sqlite> insert into Uozilo (marka, model, snaga, godina, autokuca, auto_servis)
...> values ("VW Golf", "2,0 TSI", "155kW", 2012, 3, 3)
sqlite> insert into Uozilo (marka, model, snaga, godina, autokuca, auto_servis)
```

Slika 2.5. Unos podataka u tablicu „Vozilo“

Pomoću Dot-Command (naredbe) `.schema` (Slika 2.6. *Dot-Command .schema*), dobit će se pregled nad svim tablicama koje su se izradile, prikazat će kako su tablice nazvane, kakva su ograničenja zadana, a samim time i na koji način su tablice povezane. Prikazat će se da je kreirana tablica „Auto_servis“ sa stupcem ID koji je ujedno i primarni ključ, po tipu podataka integer, postavljen da se sam automatski dodjeljuje pomoću naredbe `AUTOINCREMENT`, stupac Naziv imat će ograničenje od 50 znakova, a stupcu Telefon postaviti će se ograničenje da tekst nije nula. Kreirana tablica „Vozilo“ već je spominjana, dok će se u tablici „Instruktor“ uz stupac ID koji je zapravo OIB instruktora, ujedno i primarni ključ, postaviti na način da ga je potrebno ručno unositi i da je po tipu podataka integer, stupcima Ime i Prezime postaviti će se ograničenje da tekst nije nula, dok će se u stupac Vozilo zapravo povezati tablica „Vozilo“ putem primarnog ključa pojedinog vozila.

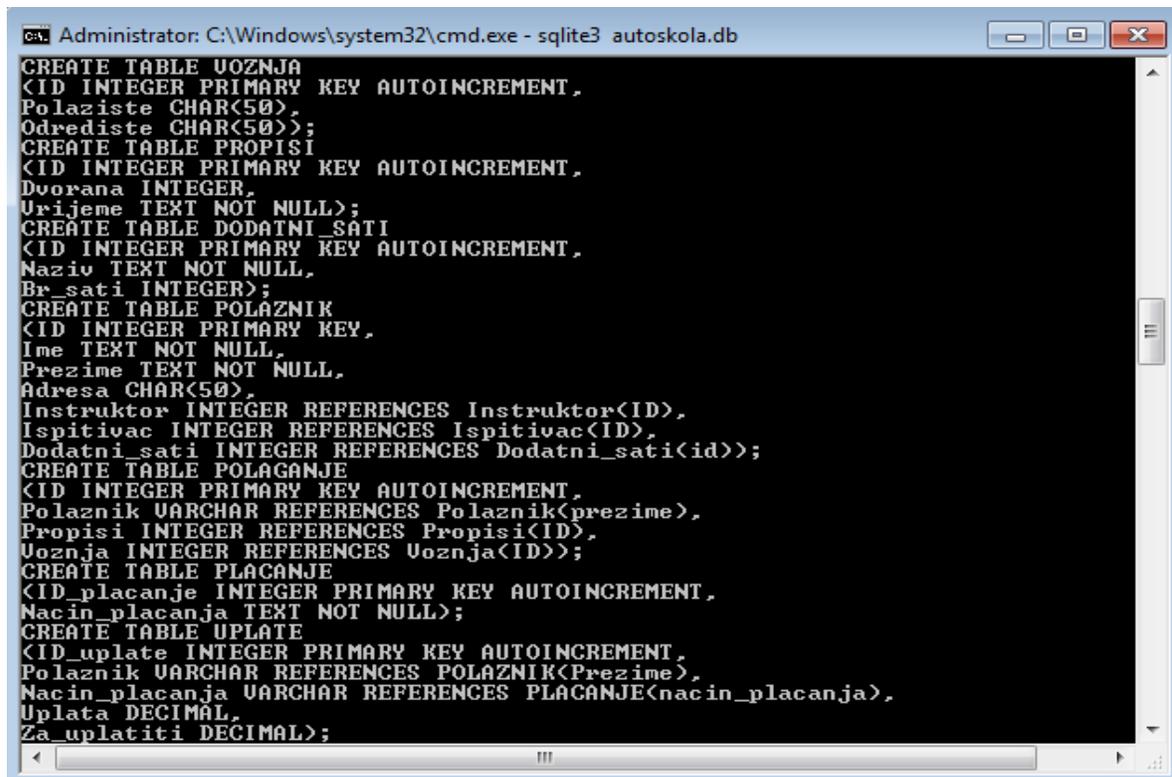


```
Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
sqlite> .schema
CREATE TABLE AUTOKUCA
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Naziv TEXT NOT NULL,
Adresa CHAR(50),
Telefon TEXT NOT NULL);
CREATE TABLE AUTO_SERVIS
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Naziv TEXT NOT NULL,
Adresa CHAR(50),
Telefon TEXT NOT NULL);
CREATE TABLE UOZILLO
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Marka TEXT NOT NULL,
Model TEXT NOT NULL,
Snaga TEXT NOT NULL,
Godina INTEGER,
Autokuca INTEGER REFERENCES AUTOKUCA(ID),
Auto_servis INTEGER REFERENCES AUTO_SERVIS(ID));
CREATE TABLE INSTRUKTOR
(ID INTEGER PRIMARY KEY,
Ime TEXT NOT NULL,
Prezime TEXT NOT NULL,
Vozilo INTEGER REFERENCES UOZILLO(ID));
CREATE TABLE ISPITIVAC
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Ime TEXT NOT NULL,
Prezime TEXT NOT NULL,
Broj_licence TEXT NOT NULL);
```

Slika 2.6. *Dot-Command .schema*

Kreirat će se nova tablica „Vožnja“, sa stupcem ID koji će biti primarni ključ, postavljen preko `AUTOINCREMENT` naredbe, integer po tipu podataka i stupcima Polazište i Odredište kojima će se ograničenje postaviti na 50 znakova. Tablici „Propisi“ će uz stupac ID, kao i kod ostalih tablica biti primarni ključ sa naredbom `AUTOINCREMENT`, također će se dodijeliti stupci Dvorana koja će biti integer po tipu podataka i stupac Vrijeme sa ograničenjem da tekst nije nula. Tablici

„Dodatni_sati“ primarni ključ biti će u stupcu ID, također postavljen preko AUTOINCREMENT naredbe, stupac Naziv će imati ograničenje da nije nula, dok je broj sati po tipu podataka integer. Tablica „Polaznik“ napraviti će se ručnim unošenjem primarnog ključa, gdje će stupac ID označavati OIB polaznika, stupcima Ime i Prezime ograničenje će se postaviti na način da tekst nije nula, a stupac Adresa će imati ograničenje na 50 znakova. U stupce Instruktor, Ispitivač i Dodatni sati povezati će se preko primarnog ključa polja iz tablica „Instruktor“, „Ispitivač“ i „Dodatni_sati“ (Slika 2.7. Dot-Command .schema 2).



```
Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
CREATE TABLE UOZNJA
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Polaziste CHAR(50),
Odrediste CHAR(50));
CREATE TABLE PROPISI
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Dvorana INTEGER,
Urijeme TEXT NOT NULL);
CREATE TABLE DODATNI_SATI
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Naziv TEXT NOT NULL,
Br_sati INTEGER);
CREATE TABLE POLAZNIK
(ID INTEGER PRIMARY KEY,
Ime TEXT NOT NULL,
Prezime TEXT NOT NULL,
Adresa CHAR(50),
Instruktor INTEGER REFERENCES Instruktor(ID),
Ispitivac INTEGER REFERENCES Ispitivac(ID),
Dodatni_sati INTEGER REFERENCES Dodatni_sati(id));
CREATE TABLE POLAGANJE
(ID INTEGER PRIMARY KEY AUTOINCREMENT,
Polaznik VARCHAR REFERENCES Polaznik(prezime),
Propisi INTEGER REFERENCES Propisi(ID),
Uoznja INTEGER REFERENCES Uoznja(ID));
CREATE TABLE PLACANJE
(ID_placanje INTEGER PRIMARY KEY AUTOINCREMENT,
Nacin_placanja TEXT NOT NULL);
CREATE TABLE UPLATE
(ID_uplate INTEGER PRIMARY KEY AUTOINCREMENT,
Polaznik VARCHAR REFERENCES POLAZNIK(Prezime),
Nacin_placanja VARCHAR REFERENCES PLACANJE(nacin_placanja),
Uplata DECIMAL,
Za_uplatiti DECIMAL);
```

Slika 2.7. Dot-Command .schema 2

2.3. Upravljanje bazom podataka

Mnoge Dot-Command (naredbe) uvelike pomažu kod upravljanja bazom podataka, a samim time bržu i lakšu pretragu podataka. Izdvojit će se neke koje će se koristiti prilikom izrade baze podataka „Autoškola“. Da bi se dobio pregled nad svim tablicama koje su kreirane, naredba `.tables` ispisat će sve kreirane tablice (Slika 2.8. Naredba `.tables`).

```
Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
sqlite> .tables
AUTOKUCA      INSTRUKTOR    POLAGANJE     UPLATE
AUTO_SERVIS   ISPITIVAC     POLAZNIK      UOZILO
DODATNI_SATI  PLACANJE     PROPISI       UOZNJA
sqlite> _
```

Slika 2.8. Naredba .tables

SQLite naredba SELECT će iz baze podataka dohvatiti podatke koji se žele dohvatiti, bilo da se radi o cijeloj tablici ili samo o nekim poljima koja zadovoljavaju uvjet koji je zadan i vratiti podatke u obliku rezultata tablice. Hoće li se ispisivati svi podaci (stupci) iz tablice, ili samo neki, ovisi o potrebama korisnika. Sintaksa je takva da se naredbi SELECT odredi koji podaci (stupci) će se ispisivati, na način da se upiše naziv jednog ili više stupaca, zatim se doda naredba FROM i upiše naziv tablice koja će se naredbom SELECT pretražiti. Ukoliko se žele dohvatiti svi stupci iz jedne tablice, umjesto ispisivanja naziva svih stupaca upiše se zvijezdica (*), što će prikazivati slika ispod (Slika 2.9. Naredba SELECT).

```
Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Andrea>cd desktop
C:\Users\Andrea\Desktop>cd baza
C:\Users\Andrea\Desktop\baza>sqlite3 autoskola.db
SQLite version 3.8.6 2014-08-15 11:46:33
Enter ".help" for usage hints.
sqlite> select * from vozilo;
1|Opel Corsa|1,2|74kW|2012|1|2
2|Peugeot|206|55kW|2010|2|3
3|Opel Astra|1,4|103kW|2010|1|4
4|Mazda2|1,3|55kW|2011|3|1
5|VW Golf|2,0 TSI|155kW|2012|3|3
6|Renault Clio|1,5 dCi|66kW|2013|4|1
sqlite> _
```

Slika 2.9. Naredba SELECT

Radi lakše preglednosti i lakšeg snalaženja te lakšeg samog čitanja podataka, pomoću Dot-Command `.header on` i `.mode column` dobit će se oku ugodniji prikaz i izgled tablice. Zbog toga što će `.header on` naredba uključiti da se prikažu nazivi stupaca, a `.mode column` će formatirati izlazni ispis (*Slika 2.10. Izgled tabličnog prikaza*)

```

Administrator: C:\Windows\system32\cmd.exe - sqlite3 autoskola.db
sqlite> .header on
sqlite> .mode column
sqlite> SELECT * FROM UOZILO;
ID      Marka      Model      Snaga      Godina      Autokuca      Auto_ser
vis
-----
1      Opel Corsa  1,2      74kW      2012      1      2
2      Peugeot   206      55kW      2010      2      3
3      Opel Astra 1,4      103kW     2010      1      4
4      Mazda2    1,3      55kW      2011      3      1
5      VW Golf   2,0 TSI  155kW     2012      3      3
6      Renault C1 1,5 dCi  66kW      2013      4      1
sqlite> _

```

Slika 2.10. Izgled tabličnog prikaza

Od važnijih naredbi koje će se koristiti prilikom izrade baze podataka, valja spomenuti i naredbu `DROP TABLE` koja briše tablicu. Uz naredbu `DROP TABLE` upisat će se naziv tablice koja se želi obrisati. Također važnije naredbe koje će se koristiti su naredbe `UPDATE` i naredba `DELETE`. Naredba `UPDATE` zamijenjuje podatke po stupcima, odnosno, ukoliko se dogodi da se negdje upisao krivi podatak, pomoću ove naredbe bit će ga moguće promijeniti. Sama sintaksa promjene podataka po stupcima izgleda:

```
UPDATE table_name SET column_name = new_value [...] WHERE
exspression
```

Pomoću naredbe `DELETE` brisat će se redovi u tablici koji neće biti potrebni ili su jednostavno krivo upisani, pa je sintaksa za to slijedeća:

```
DELETE FROM table_name WHERE exspression
```

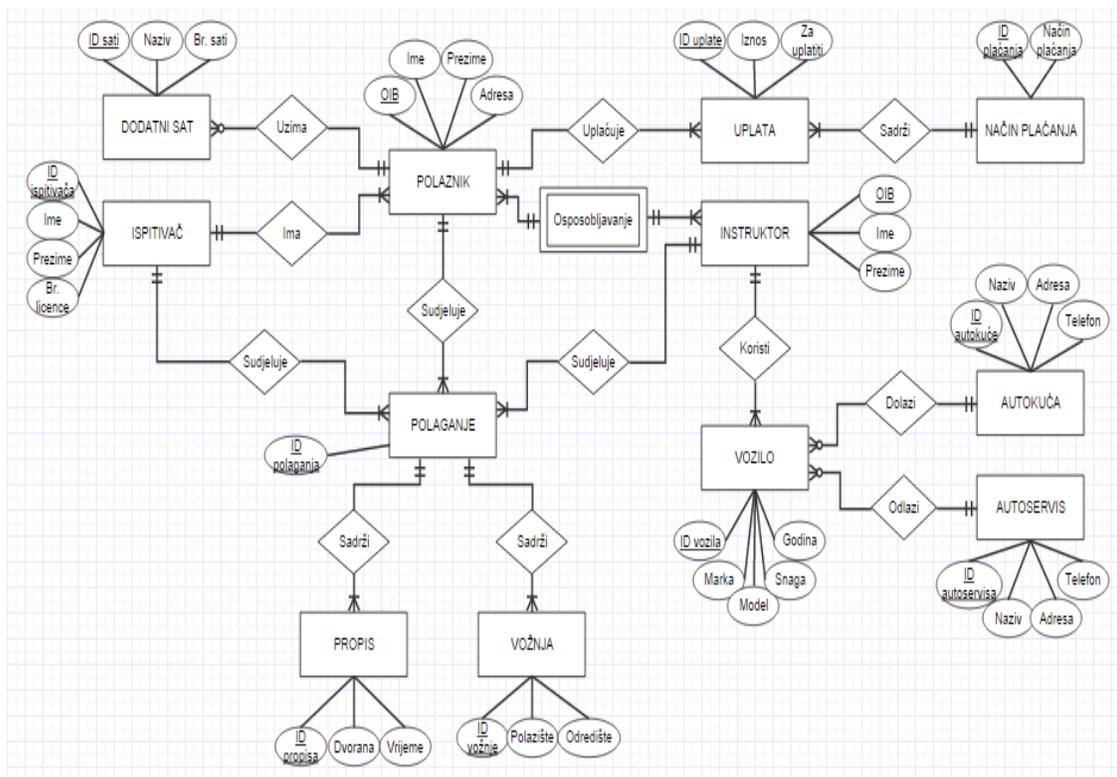
Spomenut će se još samo naredbe `ORDER BY` i `GROUP BY` koje su dio `SELECT` kaluzule. Dakle, naredba `ORDER BY` će poredati podatke po zadanom uvjetu, a naredba `GROUP BY` će ih po zadanom uvjetu grupirati.

3. MODEL BAZE PODATAKA

U svrhu izrade baze podataka „Autoškola“ izradit će se ERA model⁴ kao model baze podataka. ERA model će na apstraktan način opisati korištenje podataka iz tablica, te odnose među tablicama i veze kojima su povezane, odnosno pojednostavnit će se pristup eventualno nastalim problemima [Podaci, modeli podataka i modeliranje produkcijskih i dimenzijskih baza podataka, dostupno 22.9.2014 na http://informatika.efos.hr/wp-content/uploads/2012/04/P9_Baze%20podataka.pdf].

3.1. Era model baze podataka

ERA modelom (Slika 3.1. ERA model) baza podataka „Autoškola“ promatrat će se kroz tri kategorije: eniteti, veze među enitetima i atribute eniteta. ERA model napravit će se u obliku dijagrama u kojem će pravokutnici predstavljati enitete, rombovi će predstavljati veze, a unutar elipsa upisivat će se atributi eniteta. ERA model bit će konceptualni dijagram koji će predstavljati apstrakciju prave autoškole.



Slika 3.1. ERA model

⁴ Entity-Relationship-Attribute (enitet-veza-atribut)

Prema ERA modelu tablica „Polaznik“ će biti enitet naziva „Polaznik“. Bit će mu dodijeljeni atributi OIB, Ime, Prezime i Adresa, gdje će OIB označavati primarni ključ eniteta. Po tome atribut OIB treba biti podvučen jer je ključni atribut, te se naziva identifikator⁵, dok su ostali atributi deskriptori⁶. Enitet (tablica) „Polaznik“ je povezan sa četiri druge tablice, odnosno eniteta. Povezan je sa enitetom „Dodatni sat“ vezom jedan naprema nula ili više (1:M), što se doslovno može reći da jedan polaznik može uzeti nula ili više dodatnih sati. Enitet „Polaznik“ je vezom jedan ili više naprema jedan (M:1) povezan sa tablicom „Ispitivač“, što bi značilo da jedan ili više polaznika mogu imati jednog ispitivača koji će im se dodijeliti od strane Autoškole. Sa enitetom „Polaganje“, enitet „Polaznik“ će se povezati vezom jedan naprema jedan ili više (1:M), što bi značilo da jedan polaznik može jednom ili više puta sudjelovati na polaganju (podrazumijeva se da ukoliko polaznik na jednom polaganju padne, isti polaznik može sudjelovati i više puta, sve dok ne ostvari prolaz). Povezan je sa enitetom „Uplata“ vezom jedan naprema jedan ili više (1:M), što pokazuje da jedan polaznik može imati jednu ili više uplata (u obzir se uzima plaća li gotovinom ili u ratama). S obzirom da se kod povezivanja sa enitetom „Instruktor“ radi o vezi više naprema više (N:M), uvest će se dodatni slabi enitet⁷ naziva „Osposobljavanje“, te će tako povezanost biti da jedan ili više polaznika mogu imati jedno osposobljavanje, a jedno osposobljavanje može provoditi jedan ili više instruktora. Radi se o vezi više naprema više iz razloga što je ERA modelom pokušano prikazati da polaznik može promijeniti instruktora ukoliko želi (najčešće se može raditi o želji za promjenom vozila, osobnost instruktora ili nešto treće).

„Dodatni sat“ je enitet koji sadrži tri atributa od kojih će ID sati označavati identifikator, dok će atributi Naziv i Br. sati biti deskriptori. Povezan je samo sa enitetom „Polaznik“.

„Ispitivač“ je enitet sa četiri atributa od kojih će ID Ispitivača biti identifikator, a Ime, Prezime i Br. licence će biti deskriptori. Povezan je sa enitetom „Polaznik“ (kao što je ranije spomenuto), jedan ispitivač može imati jednog ili više polaznika (ovisi koliko mu Autoškola dodijeli), i enitetom „Polaganje“ vezom jedan naprema jedan ili više (1:M). Dakle, jedan ispitivač može sudjelovati na jednom ili više polaganja.

⁵ Koriste se kako bi se instanca eniteta jedinstveno identificirala.

⁶ Koriste se kako bi se odredila vrijednost atributa koji koji mogu biti zajednički većem broju instance (npr. ime).

⁷ Ne mogu egzistirati bez drugog eniteta, odnosno nemaju atribut koji bi mogao biti primarni ključ, već se sastoje od primarnih ključeva povezanih eniteta.

Enitet „Uplata“ sadrži tri atributa, od kojih je ID Uplate identifikator, a Iznos i Za_uplatiti identifikatori. Povezana je sa enitetima „Polaznik“ i „Način plaćanja“, pri čemu jednu ili više uplata može uplatiti jedan polaznik, a jedna uplata ili više uplata može imati samo jedan način plaćanja.

„Instruktor“ kao enitet sadrži tri atributa, od toga OIB Instruktora kao identifikator, a atributi Ime i Prezime deskriptori. Vezom više naprema više (N:M) povezat će se sa enitetom „Polaznik“, uvođenjem slabog eniteta „Osposobljavanje“. Veza prema enitetu „Polaganje“ biti će jedan naprema jedan ili više (1:M), dakle jedan instruktor može sudjelovati na jednom ili više polaganja. Vezom jedan naprema jedan ili više (1:M) povezan je sa enitetom „Vozilo“, tako da jedan instruktor može imati jedno ili više vozila.

„Vozilo“ kao enitet sadrži pet atributa, od toga ID Vozila kao identifikator, a Marka, Model, Snaga i Godina kao deskriptori. Povezanost jedan naprema nula ili više (1:M) sa enitetom „Autokuća“, gdje jedno vozilo može biti iz nijedne ili više autokuća evidentiranih u bazi podataka. Također se ista vrsta veze, jedan naprema nula ili više (1:M) odnosi i na enitet „Autoservis“, gdje nijedno ili više vozila mogu odlaziti u autoservis evidentiran u bazi podataka.

Enitet „Autokuća“ sadrži četiri atributa, od toga ID Autokuće kao identifikator, a Naziv, Adresa i Telefon kao deskriptori. Povezan je samo sa tablicom „Vozilo“, jednako kao i enitet „Autoservis“ koji također sadrži četiri atributa, ID Autoservisa kao identifikator i Naziv, Adresa i Telefon kao deskriptori.

„Polaganje“ kao enitet sadrži samo jedan atribut, ID Polaganja koji će biti deskriptor, jer je povezan sa tri druga eniteta i to „Polaznik“, „Propisi“ i Vožnja. Kako na jednom polaganju može sudjelovati jedan ili više polaznika (1:M), tako je i „Polaganje“ povezano sa enitetima „Propisi“ vezom jedan naprema jedan ili više (1:M) i „Vožnja“ vezom jedan naprema jedan ili više (1:M).

„Propisi“ su enitet koji sadrži tri atributa, ID Propisa kao identifikator, te Dvorana i Vrijeme kao deskriptori. Povezan je jednio sa enitetom „Polaganje“, jednako kao i enitet „Vožnja“. „Vožnja“ također sadrži tri atributa, ID Vožnje kao identifikator, te Polazište i Odredište kao deskriptori.

4. OPIS APLIKACIJSKE DOMENE

Za bazu podataka „Autoškola“ kreirat će se aplikacija (grafičko sučelje), koje će biti način interakcije službenika zaposlenog u jednoj autoškoli sa već postojećom i kreiranom bazom podataka. Grafičkim elementima, porukama i obavijestima olakšat će se vođenje baze podataka, te njezina pretraga i ažuriranje putem aplikacije. Aplikacija je osmišljena na način da se samo jednim klikom miša može obrisati pojedini podatak, ili pak unosom samo važnih podataka ubrzati uvođenje novih polaznika, instruktora, vozila i slično. Osmišljen je lakši pregled nad uplatama polaznika, te eventualnim njihovim dugovanjima. Dakle, aplikacijom će se rad sa bazom podataka uvelike olakšati običnim korisnicima, odnosno zaposlenima u jednoj autoškoli, bez znanja sintakse i linijskih kodova. Grafičko sučelje izradit će se u PHP programskom jeziku. „PHP je skraćenica od Hypertext Preprocessor i pripada u programske jezike koji se izvršavaju na poslužitelju“ [Baniček, 2004, str. 69.]. U ovoj će se aplikaciji baza podataka „Autoškola“ naći kao dodatak programskom jeziku PHP jer on sadrži SQLite programski kod.

4.1. Uvod u PHP

Budući da se PHP jezik ističe širokom podrškom raznim bazama podataka, koristit će se za izradu grafičkog sučelja baze podataka kreiranoj u SQLite sustavu. PHP je skriptni jezik za programiranje koji se nalazi na strani poslužitelja i za njegovo izvođenje bit će potreban web poslužitelj. PHP kodovi upisuju se između znakova `<?php i ?>`, a svaki naredbeni red mora završavati sa točkazarez (;). Za privremenu pohranu vrijednosti koriste se varijable, koje mogu biti lokalne i globalne. Lokalne varijable definirane su u funkcijama a sve varijable definirane izvan funkcija su globalne. Za potrebe funkcije, globalne varijable prosljeđuju se putem ključne riječi `global`. Varijable se označavaju znakom `$` koji dolazi ispred imena varijable. Također postoje i superglobalne varijable koje su ugrađene (`$_POST`, `$_GET`, `$_SESSION`, `$_COOKIE`), `$_SERVER`, `$_FILES`). PHP programski jezik koristi i funkcije, odnosno blok naredbi koje izvršavaju svoj dio koda, i mogu se po potrebi pozvati više puta.

Sintaksa funkcije:

```
function naziv_funkcije (argumenti) {kod koji funkcija izvršava}
```

Za potrebe izrade aplikacije koristit će se i klase koje su skup funkcija i varijabli, s namjerom da se definira objekt kao predložak za stvarne vrijednosti. Da bi se klase koristile potrebno je kreirati instancu objekta koristeći operator `new` [PHP 5, dostupno 20.09.2014. na <http://www.w3schools.com/php>].

```
Class naziv_klase {  
function naziv_funkcije () {  
}  
var $varijabla;  
}
```

Kod PHP programskog jezika spomenut će se još i petlje koje još omogućavaju da se određeni dio koda ponovi više puta, te na taj način ubrzavaju rad i štede vrijeme izvođenja određenog dijela aplikacije (`while`, `do while`, `for`, `foreach`) [Osnove PHP-a, dostupno 20.09.2014. na <http://phptutorial.odlican.net/osnovephpa.php>].

4.2. Spajanje na bazu podataka

Da bi se podaci iz baze podataka mogli čitati, ispisivati i prikazivati putem aplikacije, napravit će se spajanje na bazu podataka „Autoškola“. Spajanje na bazu podataka „Autoškola“ napravit će se putem slijedećih linija kodova:

```
class MyDB extends SQLite3  
{  
    function __construct()  
    {  
        $this->open('autoskola.db');  
    }  
}  
$db = new MyDB();  
if (!$db) {  
    echo $db->lastErrorMsg();  
} else {  
    // echo „Opened database successfully\n“;  
}
```

S obzirom da programski jezik PHP kreiranu bazu podataka tretira kao klasu, prva linija koda će dati naredbu da se klasa naziva `MyDB` povlači iz `SQLite3` datoteke, a zatim će se definirati konstruktorska funkcija naziva `construct()` u kojoj će se u varijablu `$this` dohvatiti podaci funkcijom `open()` iz postojeće baze podataka `'autoskola.db'`. U varijablu `$db` spremiće se vrijednost nove instancirane klase `MyDB` pomoću operatora za instanciranje klase `new`. Uvjetom `if` je uvjetovano da ukoliko nema konekcije, naredbom za ispis `echo` ispisuje se poruka o neuspjelom zahtjevu otvaranja baze podataka, ili se može, kao što je u komentar dodano, ukoliko je povezivanje na bazu podataka uspješno, pomoću naredbe `else` koja izvršava naredbe ako uvjet nije ispunjen, ispisat će se poruka o uspješnom otvaranju baze podataka. Ukoliko prilikom otvaranja baze podataka, baza podataka takvog naziva ne postoji, ona će se pod tim imenom kreirati.

Nakon spajanja na bazu podataka mogu se napraviti i izvršiti brojni upiti i ispisi tih upita, ali je nakon izvršavanja upita i njihovih ispisa, te ostalih određenih potrebnih radnji nad bazom podataka, vezu prema bazi podataka potrebno zatvoriti. Veza prema bazi podataka zatvara se slijedećom funkcijom:

```
$db->close();
```

Kako je u varijablu `$db` spremljena vrijednost klase `MyDB`, tako se u nju sprema i nova vrijednost funkcije `close()`, odnosno zatvaranje veze.

4.3. Ispis podataka iz baze podataka

Da bi se podaci mogli ispisati iz baze podataka, odnosno da bi na ekranu bili prikazani u obliku tablice, ili nekom drugom obliku, nakon spajanja na bazu podataka, potrebno je napraviti upit (query). Upit će obuhvatiti sve one dijelove tablice koji se žele prikazati. Tako će se za ispis tablice „Autokuća“ napraviti slijedeće linije kodova:

```
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('autoskola.db');
    }
}
$db = new MyDB();
if (!$db) {
    echo $db->lastErrorMsg();
}
```

```

} else {
    // echo „Opened database successfully\n“;
}

$sql =<<<EOF
    SELECT * from AUTO_SERVIS;
EOF;

$ret = $db->query($sql);
echo „ <table border='2' cellpadding='2' style='width:100%' >“;
echo „<tr><td align='center'><b>ID<td align='center'><b>Naziv<td
align='center'><b>Adresa<td align='center'><b>Telefon<td
align='center'><b>Vise podataka</tr>“;
while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
    echo „<tr class='highlighter'>“;

    echo „<td align='center'>“; echo $row['ID']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Naziv']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Adresa']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Telefon']; echo „</td>“;

    echo „<td align='center'>“; echo „<a href='ID.php?autoservis_id=“;
echo $row['ID']; echo „'>“; echo „Vise podataka“; echo „</a></td>“;

}
// echo „Operation done successfully\n“;
$db->close();
?>

```

Dakle, spajanjem na bazu podataka, naredbom SELECT dohvatit će se svi podaci iz tablice “Auto_servis”, u varijablu \$ret će se spremiti upit iz varijable \$sql nad bazom podataka iz varijable \$db. Nadalje, definirat će se izgled tablice kod ispisa. Tako će okvir tablice biti broj 2, razmak između ćelija također veličine 2, a širina tablice će biti 100%, što će značiti da će širina tablice biti po cijeloj širini stranice. Poravnanje ćelija u tablici će biti centrirano, naslovi stupaca ID, Naziv, Adresa i Telefon će također imati centrirano poravnanje, ali će uz to biti i podebljanog naziva. Uz to, stupac Više podataka, koji ujedno predstavlja poveznicu na popis automobila koje određeni autoservis servisira, bit će centriranog poravnanja i podebljanog naziva. Potom će uslijediti ispisivanje podataka dohvaćenih upitom u kreiranu tablicu. Koristit će se petlja while koja će dohvaćati podatke sve dok je uvjet zadovoljen u varijablu naziva \$row gdje će se u obliku asocijativnog niza⁸ spremati podaci iz varijable \$ret. Ispisivat će se \$row pod ključem “ID” (\$row ['ID']), \$row pod ključem “Naziv” (\$row ['Naziv']), \$row pod ključem

⁸ Niz čije su vrijednosti ključeva tekstualne a ne brojčane.

„Adresa“ (\$row ['Adresa']), \$row pod ključem „Telefon“ „Adresa“ (\$row ['Adresa']), te poveznica na popis automobila koje autoservis servisira, na koje će poveznica biti rečenica „Više podataka“ (ID.php). Nakon ispisa moguće je staviti poruku o uspješno izvršenoj operaciji. Na kraju će se zatvoriti veza prema bazi podataka funkcijom \$db->close(). Linije kodova prikazane u nastavku, prikazat će na koji način će se prikazivati povezivanje automobila sa odabranim autoservisom. Naime, autoservis koji koji će se primiti GET metodom (\$_GET), će se putem naredbe SELECT dohvatiti iz tablice „Vozilo“ sve podatke na način da zadovoljava uvjet gdje je autoservis jednak autoservisu primljenom GET metodom.

```

$autoservis_id=$_GET['autoservis_id'];

class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('autoskola.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    // echo „Opened database successfully\n“;
}

$sql = " select * from VOZILO where Auto_servis='$autoservis_id'";

$ret = $db->query($sql);
echo „ <table border='2' cellpadding='2' style='width:100%' >“;
echo „<tr><td align='center'><b>ID<td align='center'><b>Marka<td
align='center'><b>Model<td align='center'><b>Snaga<td
align='center'><b>Godina“;
while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
    echo „<tr class='highlighter'>“;

    echo „<td align='center'>“; echo $row['ID']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Marka']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Model']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Snaga']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Godina']; echo „</td>“;

}
// echo „Operation done successfully\n“;
$db->close();

```

Prije postavljanja bilo kojeg upita bit će potrebno spojiti se na bazu podataka, pa će se u nastavku prikazati samo dio linija koda koje se odnose na ispisivanje podataka iz tablice „Autokuća“.

```
$sql =<<<EOF
    SELECT * from AUTOKUCA;
EOF;
$ret = $db->query($sql);
echo „ <table border='2' cellpadding='2' style='width:100%' >“;
echo „<tr><td align='center'><b>ID
<td align='center'><b>Naziv
<td align='center'><b>Adresa
<td align='center'><b>Telefon<td align='center'><b>Više
podataka</tr></tr>“;
while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
    echo „<tr class='highlighter'>“;

    echo „<td align='center'>“; echo $row['ID']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Naziv']; echo „</td>“;
        echo „<td align='center'>“; echo $row['Adresa']; echo „</td>“;
        echo „<td align='center'>“; echo $row['Telefon']; echo „</td>“;
    echo „<td align='center'>“; echo „<a href='ID1.php?autokuca_id=“; echo
$row['ID']; echo „'>“; echo „Više podataka“; echo „</a></td>“;

}
// echo „Operation done successfully\n“;
$db->close();
```

Kao i kod ispisa podataka iz tablice „Autoservis“ iz baze podataka, tako će se iz tablice „Autokuća“ ispisati podaci o autokućama. Naredbom SELECT naredit će se da se dohvate svi podaci, ali će kod poveznice „Više podataka“ biti veza na popis automobila koji dolaze iz određenih autokuća.

Budući da svaki ispis tablica započinje sa spajanjem na bazu podataka, nastavlja se kroz definiranje izgleda tablice i završava zatvaranjem veze prema bazi podataka, prikazat će se ključni dijelovi ispisivanja podataka iz tablica „Dodatni sati“, „Instruktor“ i „Ispitivač“.

```

$sql = <<<EOF
    SELECT * from DODATNI_SATI;
EOF;

$ret = $db->query($sql);
echo „ <table border='2' cellpadding='2' style='width:100%' >“;
echo „<tr><td align='center'><b>ID<td align='center'><b>Naziv<td
align='center'><b>Broj sati<td align='center'><b>Dodatne
informacije</tr>“;
    while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
        echo „<tr class='highlighter'>“;

        echo „<td align='center'>“; echo $row['ID']; echo „</td>“;
        echo „<td align='center'>“; echo $row['Naziv']; echo „</td>“;
        echo „<td align='center'>“; echo $row['Br_sati']; echo „</td>“;
        echo „<td align='center'>“; echo „<a
href='ID3.php?odatnisati_id=“; echo $row['ID']; echo „'>“; echo „Vise
podataka“; echo „</a></td>“;
    }

```

Ispis podataka iz tablice „Instruktor“:

```

$sql = <<<EOF
    SELECT i.ID, i.Ime, i.Prezime, v.Marka from INSTRUKTOR i, VOZILO v
where i.Vozilo=v.ID;
EOF;

$ret = $db->query($sql);
echo „ <table border='2' cellpadding='2' style='width:100%' >“;
echo „<tr><td align='center'><b>ID<td align='center'><b>Ime<td
align='center'><b>Prezime<td align='center'><b>Vozilo<td
align='center'><b>Dodatne informacije<td align='center'><b>Brisi</tr>“;
    while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
        echo „<tr class='highlighter'>“;

        echo „<td align='center'>“; echo $row['ID']; echo „</td>“;

```

```

    echo „<td align='center'>“; echo $row['Ime']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Prezime']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Marka']; echo „</td>“;
    echo „<td align='center'>“; echo „<a
href='ID5.php?instruktor_id=“; echo $row['ID']; echo „'>“; echo „Vise
podataka“; echo „</a></td>“;
    echo „<td align='center'>“; echo „<a href='ID6.php?instruktorid=“;
echo $row['ID']; echo „'>“; echo „Brisanje“; echo „</a></td>“;

}
// echo „Operation done successfully\n“;
$db->close();

```

Ispis podataka iz tablice „Ispitivač“:

```

$sql = <<<EOF
    SELECT * from ISPITIVAC;
EOF;

$ret = $db->query($sql);
echo „ <table border='2' cellpadding='2' style='width:100%' >“;
echo „<tr><td align='center'><b>ID<td align='center'><b>Ime<td
align='center'><b>Prezime<td align='center'><b>Broj licence<td
align='center'><b>Dodatne informacije</tr>“;
while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
    echo „<tr class='highlighter'>“;

    echo „<td align='center'>“; echo $row['ID']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Ime']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Prezime']; echo „</td>“;
    echo „<td align='center'>“; echo $row['Broj_licence']; echo
„</td>“;
echo „<td align='center'>“; echo „<a href='ID4.php?ispitivac_id=“; echo
$row['ID']; echo „'>“; echo „Vise podataka“; echo „</a></td>“;

}

$db->close();

```

4.3. Unos novih podataka u bazu podataka

Unos novih podataka radit će se putem forme, s ciljem da se unos ubrza i bude što točniji i precizniji. Forma će se kreirati pomoću kodova u nastavku.

```
<form class="basic-grey" name="form1" method="post"
action="dodaj_autokucu.php">
</h1>
<label>
<span>Naziv: </span><input name="Naziv" type="text"
id="Naziv"></label>
<label>
<span>Adresa: </span><input name="Adresa" type="text"
id="Adresa"></label>
<label>
<span>Telefon: </span><input name="Telefon" type="text"
id="Telefon"></label>
<label>
<span>&nbsp;</span> <input type="submit" name="Submit"
value="Dodaj autokucu"></label>
</form>
```

Podaci unešeni preko ove forme, šalju se preko POST metode, u `dodaj_autokucu.php` datoteku. U varijable `$Naziv`, `$Adresa` i `$Telefon` će se spremi podaci iz superglobalne varijable `$_POST` pod ključem 'Naziv', 'Adresa' i 'Telefon'. Otvorit će se veza prema bazi podataka, te će se napraviti upit, odnosno naredba da se u tablicu naziva „Autokuća“ unesu vrijednosti iz varijabli koje su primljene POST metodom. U varijablu `$ret` će se ispisati nove unešene vrijednosti, odnosno autokuće. Ukoliko se u varijablu `$ret` ništa ne ispiše, ispisati se može poruka koja javlja grešku ispisivanja, a ako se u varijablu `$ret` nešto ispiše, može se ispisati poruka da je dodavanje nove autokuće uspješno napravljeno. Zatvorit se veza prema bazi podataka, i ispisati poruka da je nova autokuća uspješno dodana i ispisati njezin naziv, te će se ponuditi poveznica na dodavanje nove autokuće ili povratak na početnu stranicu.

```
<?php
$Naziv=$_POST['Naziv'];
$Adresa=$_POST['Adresa'];
$Telefon=$_POST['Telefon'];
```

```

class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('autoskola.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    // echo „Opened database successfully\n”;
}
$sql = "INSERT INTO AUTOKUCA (Naziv,Adresa,Telefon) VALUES ('$Naziv',
'$Adresa', '$Telefon' )";
$ret = $db->exec($sql);
if(!$ret){
    echo $db->lastErrorMsg();
} else {
    // echo „Records created successfully\n”;
}
$db->close();
echo „Uspješno dodana nova autokuća:”; echo $Naziv;
echo „<br><br> <a href = 'nova_autokuca.php'> Dodati novu autokuću </a>”;
echo „ ili se vratite na <a href='index.php'> početnu stranicu!</a>”;
?>

```

4.4. Brisanje podataka iz baze podataka

Ukoliko netko od polaznika uspješno završi, položi, autoškolsko osposobljavanje, njega više nije potrebno voditi u evidenciji. Isto tako, ukoliko autoškola iz nekog razloga ostane bez nekog instruktora (promjena radnog mjesta, odlazak u mirovinu, otkaz), također ga više nije potrebno voditi u bazi podataka. Ista stvar događa se ukoliko netko od ispitivača više nije zadužen za određenu autoškolu, i njegovi podaci će biti nepotrebni u bazi podataka. Iz tog razloga, napraviti će se jednostavno brisanje nepotrebnih podataka.

Za primjer će se uzeti brisanje instruktora. ID instruktora pod ključem 'instruktorid' primljen preko superglobalne varijable \$_POST spremić će se u varijablu naziva \$ID. Nakon toga napraviti će se spajanje na bazu podataka, te napraviti upit gdje će se upisati ključna riječ (naredba) DELETE, označiti da je to tablica „Instruktor“ i zadati uvjet da se obriše ID instruktora koji je jednak sadržaju koji je spremljen u varijablu \$ID. U varijablu \$ret će se spremiti rezultat upita iz varijable \$sql, te će se ispisati poruka o uspješno izbrisanom instrukturu i mogućnost povratka na stranicu sa instruktorima za još brisanja, ili na početnu stranicu. U nastavku će se prikazati brisanje instruktora izraženo u kodovima.

```
<?php
$ID=$_GET['instruktorid'];

class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('autoskola.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
}
$sql = „delete from INSTRUKTOR where ID='$ID'”;
$ret = $db->query($sql);

echo („<br>“);

echo „Uspješno izbrisan instruktor!”;

echo „<br><br><a href='instruktor.php'>Brisati još</a>“; echo „ ili se
vratite na <a href='index.php'>početak</a>“;
```

5. OPIS IMPLEMENTACIJE

Aplikacija, koja služi kao korisničko sučelje korisnicima sadržavat će i tri uloge koje će moći pristupiti. Svaka od tih uloga će prilikom prijave imati drugačiji pregled baze podataka, odnosno, vidjet će samo onaj dio koji je za nju karakterističan i služiti će samo za obavljanje njegove djelatnosti. Također će se implementirati i zapisnik koji će bilježiti tko se je i kada prijavio ili odjavio. Pregled nad prijavljenim i odjavljenim ulogama dodijelit će se administratoru, kako bi mogao voditi evidenciju, ili jednostavno aplikaciju držati pod kontrolom.

5.1. Implementacija uloga

Implementacija uloga napraviti će se na način da će to biti forma, naziva „Prijava u sustav“ koja će sadržavati polja za unos korisničkog imena i lozinke, te gumbić za prijavu. Nakon uspješne prijave, prikazat će se poruka o uspješnoj prijavi određene uloge, te će se automatski preusmjeriti na početnu stranicu aplikacije. Za potrebe implementiranja uloga u rad aplikacije, u bazi podataka „Autoškola“ napraviti će se jedna nova tablica. Tablica naziva Korisnik sadržavat će tri stupca, od kojih će ID_Korisnika biti primarni ključ te tablice. Definirat će se stupac naziva Naziv, gdje će se dodati korisnička imena za uloge koje će se implementirati. U ovom slučaju to će biti nazivi administrator, računovođa i polaznik. U trećem stupcu, naziva lozinka, dodat će se lozinke koju će svaka pojedina uloga imati. *Tablica 5.1. Korisnička imena i lozinke* prikazuje korisnička imena i lozinke uloga koje se mogu prijaviti u sustav.

Tablica 5.1. Korisnička imena i lozinke

ID_Korisnik	Korisničko ime	Lozinka
1	administrator	admin
2	racunovoda	sifra
3	polaznik	polaznik

Prijava u sustav, kao što je ranije spomenuto je forma za unos podataka. Forma za unos podataka napraviti će se u HTML (HyperText Markup Language) prezentacijskom jeziku.

Prijava u sustav

Korisnicko ime:

Lozinka:

Slika 5.1. Forma prijave

Nakon prijave, podaci sa korisničkim imenom i lozinkom poslat će se u checklogin.php datoteku, gdje će se provjeravati vjerodostojnost podataka. Slijedeće linije koda provjeravat će vjerodostojnost podataka:

```
<?php
ob_start();

$username=$_POST[„Naziv“];
$password=$_POST[„Lozinka“];

class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('autoskola.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    // echo „Opened database successfully\n“;
}

$sql="SELECT count(Naziv) as broj FROM korisnik WHERE Naziv='$username'
and Lozinka='$password'";

$ret = $db->query($sql);

$row = $ret->fetchArray(SQLITE3_ASSOC);

ob_end_flush();
?>
```

Nakon znaka za početak PHP-a, pokrenut će se funkcija `ob_start()` koja će imati ulogu međuspremnik, odnosno tek nakon svih provjera će podatke poslati serveru. U varijablu `$myusername` će se spremiti sadržaj superglobalne varijable `$_POST` pod ključem „Naziv“. Superglobalna varijabla `$_POST` predstavlja metodu kojom će podaci biti poslani. U varijablu `$mypassword` će se spremiti sadržaj superglobalne varijable `$_POST` pod ključem „Lozinka“. Varijable `$myusername` i `$mypassword` će biti globalne varijable, jer će se definirati izvan funkcije. Nakon toga napraviti će se spajanje na bazu podataka, koje je već ranije bilo objašnjeno. U varijablu `$sql` spremit će se podaci koji će se dohvatiti naredbom `SELECT`, koja dohvaća podatke. U varijablu `$ret` spremit će se upit iz varijable `$sql` koji će se izvršiti nad bazom podataka definiranoj u varijabli `$db`. U varijablu `$row` će se zatim iz varijable `$ret` ispisivati podaci u obliku asocijativnog niza. Funkcijom `ob_end_flush()` podaci će se poslati serveru.

Tko će od korisnika aplikacije, odnosno od implementiranih uloga što vidjeti, odredit će se u skripti naziva *navigacija2.php*, odnosno sljedećim linijama kodova:

```
<?php
session_start();

if(isset($_SESSION['Naziv'])==true){
$tipQ=$_SESSION['Naziv'];
if ($tipQ=='administrator') {
    echo"Ulogirani ste kao ADMIN „;
        echo"<br>";
        echo „<a href='logout.php'>Odjavite se</a><br>“;
    echo „
<div ><ul id='menu'>

    <li class='current'><a href='index.php'>Pocetna<br></a></li>
    <li class='current'><a href='zapisnik.php'>Zapisnik<br></a></li>
    <li><a href='#' class='drop'>Polaznici</a>
    <div class='dropdown_1column align='right'>
        <div class='col_1'>
    <ul class='simple'>
        <li><a href='polaznik.php'>Popis polaznika</a></li>
        <li><a href='novi_polaznik.php'>Dodaj polaznika</a></li>
    </ul>
    </div>
</div>
</li>
    <li><a href='#' class='drop'>Instruktor</a>
```

```

    <div class='dropdown_1column align='right'>
      <div class='col_1'>
    <ul class='simple'>
      <li><a href='instruktor.php'>Popis instruktora</a></li>
      <li><a href='novi_instruktor.php'>Dodaj
instruktora</a></li>
    </ul>
      </div>
    </div>
  </li>
  <li><a href='#' class='drop'>Vozni park</a>
  <div class='dropdown_1column align='right'>
    <div class='col_1'>
  <ul class='simple'>
    <li><a href='vozilo.php'>Popis vozila</a></li>
    <li><a href='novo_vozilo.php'>Dodaj vozilo</a></li>
    <li><a href='autokuca.php'>Popis autokuca</a></li>
    <li><a href='nova_autokuca.php'>Dodaj autokucu</a></li>
    <li><a href='auto_servis.php'>Popis autoservisa</a></li>
    <li><a href='novi_autoservis.php'>Dodaj
autoservis</a></li>
  </ul>
    </div>
  </div>
  </li>
  <li><a href='#' class='drop'>Ispiti</a>
  <div class='dropdown_1column align='right'>
    <div class='col_1'>
  <ul class='simple'>
    <li><a href='dodatni_sati.php'>Dodatni sati</a></li>
    <li><a href='polaganje.php'>Polaganje</a></li>
    <li><a href='propisi.php'>Propisi</a></li>
    <li><a href='voznja.php'>Voznja</a></li>
    <li><a href='ruta_voznje.php'>Dodaj rutu voznje</a></li>
    <li><a href='novi_dodsate.php'>Dodaj nove dodatne
sate</a></li>
    <li><a href='novi_propisi.php'>Dodaj termin
propisa</a></li>
    <li><a href='novo_polaganje.php'>Dodaj novo
polaganje</a></li>
  </ul>
    </div>
  </div>
  </li>
  <li><a href='#' class='drop'>Ispitivac</a>
  <div class='dropdown_1column align='right'>
    <div class='col_1'>
  <ul class='simple'>
    <li><a href='ispitivac.php'>Ispitivac</a></li>
    <li><a href='novi_ispitivac.php'>Dodaj novog
ispitivaca</a></li>
  </ul>

```

```

        </div>
    </div>
</li>
<li><a href='#' class='drop'>Placanje</a>
<div class='dropdown_1column align='right'>
    <div class='col_1'>
<ul class='simple'>
        <li><a href='placanje.php'>Placanje</a></li>
        <li><a href='uplate.php'>Uplate</a></li>
        <li><a href='dodaj_placanjepolaznika.php'>Dodaj
dugovanja polaznika</a></li>
        <li><a href='nova_vrstaplacanja.php'>Dodaj novi način
plaćanja</a></li>
    </ul>
    </div>
</div>
</li>

</ul>
</div>
"; }

elseif ($tipQ=='racunovoda') {
    echo"Ulogirani ste kao racunovoda ";
    echo"<br>";
    echo "<a href='logout.php'>Odjavite se</a><br>";
    echo "

<div ><ul id='menu'>

    <li class='current'><a href='index.php'>Pocetna<br></a></li>

    <li><a href='#' class='drop'>Placanje</a>
<div class='dropdown_1column align='right'>
        <div class='col_1'>
<ul class='simple'>
            <li><a href='placanje.php'>Placanje</a></li>
            <li><a href='uplate.php'>Uplate</a></li>
            <li><a href='dodaj_placanjepolaznika.php'>Dodaj
dugovanja polaznika</a></li>
            <li><a href='nova_vrstaplacanja.php'>Dodaj novi način
plaćanja</a></li>
        </ul>
        </div>
    </div>
</li>

</ul>
</div>
"; }

```

```

elseif ($tipQ=='polaznik') {
    echo"Ulogirani ste kao polaznik „;
        echo"<br>";
        echo „<a href='logout.php'>Odjavite se</a><br>“;
    echo „

<div ><ul id='menu'>

    <li class='current'><a href='index.php'>Pocetna<br></a></li>

    <li><a href='#' class='drop'>Ispiti</a>
    <div class='dropdown_1column align='right'>
        <div class='col_1'>
    <ul class='simple'>
        <li><a href='dodatni_sati.php'>Dodatni sati</a></li>
        <li><a href='polaganje.php'>Polaganje</a></li>
        <li><a href='propisi.php'>Propisi</a></li>
        <li><a href='voznja.php'>Voznja</a></li>

            </ul>
        </div>
    </div>
    </li>

</ul>
</div>
„; }

}

elseif (isset($_SESSION['Naziv'])==false){
    echo „<hr>“;
    echo „Niste prijavljeni, možete to učiniti klikom na link:&nbsp;“;

    echo „<a href='login.php'>Login</a><br>“;echo „<hr>“;

}
?>

```

Nakon prijave u sustav, ovisno u ulogi koja će biti prijavljena, aplikacija će prikazati samo onaj dio za koji prijavljena uloga ima pristup. Tako će administrator imati pregled nad svim podacima, računovođa će imati pregled samo nad uplatama, dok će polaznik imati pristup samo pregledu polaganja (*Slika 5.2. Pregled aplikacije – administrator, Slika 5.3. Pregled aplikacije – računovođa, Slika 5.4. Pregled aplikacije – polaznik*).



Slika 5.2. Pregled aplikacije – administrator



Slika 5.3. Pregled aplikacije – računovođa



Slika 5.3. Pregled aplikacije – polaznik

5.2. Kontrola zapisa prijavljenih i odjavljenih uloga

Kako bi se kontroliralo koliko dugo je netko bio prijavljen u sustav i tko, implementirat će se opcija zapisnik, nad kojom će pregled imati samo uloga administratora. Za potrebe zapisnika o vremenu prijave i odjave, u bazu podataka „Autoškola“ dodat će se još jedna nova tablica naziva Zapisnik, koja će imati 3 stupca, od kojih će ID_zapisnika biti primarni ključ, vrijeme i radnja. Zapisnik će raditi na način da će se prilikom prijave, u checklogin.php datoteci, uz autentičnost korisničkog imena i lozinke, pokretati sesija⁹. Iz sesije će se izvlačiti podaci o korisničkom imenu, lozinci i statusu prije, te zapisivati u tablicu naziva Zapisnik kao što Slika 5.4. Zapisnik prijavljenih i odjavljenih prikazuje.

Linije koda za provjeru prijavljenih i pokretanje sesije prilikom istog:

```
<?php
ob_start();

$myusername=$_POST[„Naziv“];
```

⁹ Vrijeme koje korisnik provede na stranicama.

```

$mypassword=$_POST[„Lozinka“];
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('autoskola.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    // echo „Opened database successfully\n“;
}

$sql=“SELECT count(Naziv) as broj FROM korisnik WHERE
Naziv='$myusername' and Lozinka='$mypassword'“;

$ret = $db->query($sql);

$row = $ret->fetchArray(SQLITE3_ASSOC);

$count=$row['broj'];

if($count==1){

    session_start();

    $sql=“SELECT Naziv, Lozinka FROM KORISNIK WHERE
Naziv='$myusername' and Lozinka='$mypassword'“;
    $ret = $db->query($sql);

    $row = $ret->fetchArray(SQLITE3_ASSOC);

    $_SESSION['Naziv'] = $myusername;
    $_SESSION['Lozinka'] = $mypassword;
    $_SESSION['logiran'] = „DA“;

    $datum=date('d-m-Y H-i-s');

    $sql=“INSERT INTO ZAPISNIK (NAZIV, VRIJEME, RADNJA) values
('$myusername', '$datum', 'Logiranje')“;

    echo $sql;
    $ret = $db->query($sql);

}
else {

```

```

echo „Krivo korisnicko ime ili lozinka“;
}
header („location:logiran.php“);
ob_end_flush();

?>

```

Upisivanje podataka u tablicu zapisnik:

```

<html>
<head>
<title>Zapis</title>
<style>

</style>

</head>
<body>
<h1 align="center">Sustav upravljanja bazom podataka Auto
škole</h1>
<br>
<?php

include („navigacija2.php“);
echo „<br>“;
echo „<h2>Popis ulogiravanja korisnika</h2>“;
echo „<br>“;
    class MyDB extends SQLite3
    {
        function __construct()
        {
            $this->open('autoskola.db');
        }
    }
    $db = new MyDB();
    if(!$db){
        echo $db->lastErrorMsg();
    } else {
        // echo „Opened database successfully\n“;
    }

    $sql =<<<EOF
        SELECT * from ZAPISNIK;
EOF;

    $ret = $db->query($sql);
echo „ <table border='2' cellpadding='2' style='width:100%' >“;
echo „<tr><td align='center'><b>ID

```

```

<td align='center'><b>Naziv
<td align='center'><b>VRIJEME
<td align='center'><b>Radnja</tr>";
    while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
        echo „<tr class='highlighter'>“;

        echo „<td align='center'>“; echo $row['ID']; echo „</td>“;
        echo „<td align='center'>“; echo $row['Naziv']; echo „</td>“;
        echo „<td align='center'>“; echo $row['VRIJEME']; echo
„</td>“;
        echo „<td align='center'>“; echo $row['RADNJA']; echo
„</td>“;

    }
    // echo „Operation done successfully\n“;
    $db->close();
?>

```

Sustav upravljanja bazom podataka Auto škole

Ulogirani ste kao ADMIN
[Odjavite se](#)

Početna Zapisnik Polaznici Instruktor Vozni park Ispiti Ispitivac Placanje

Popis ulogiravanja korisnika

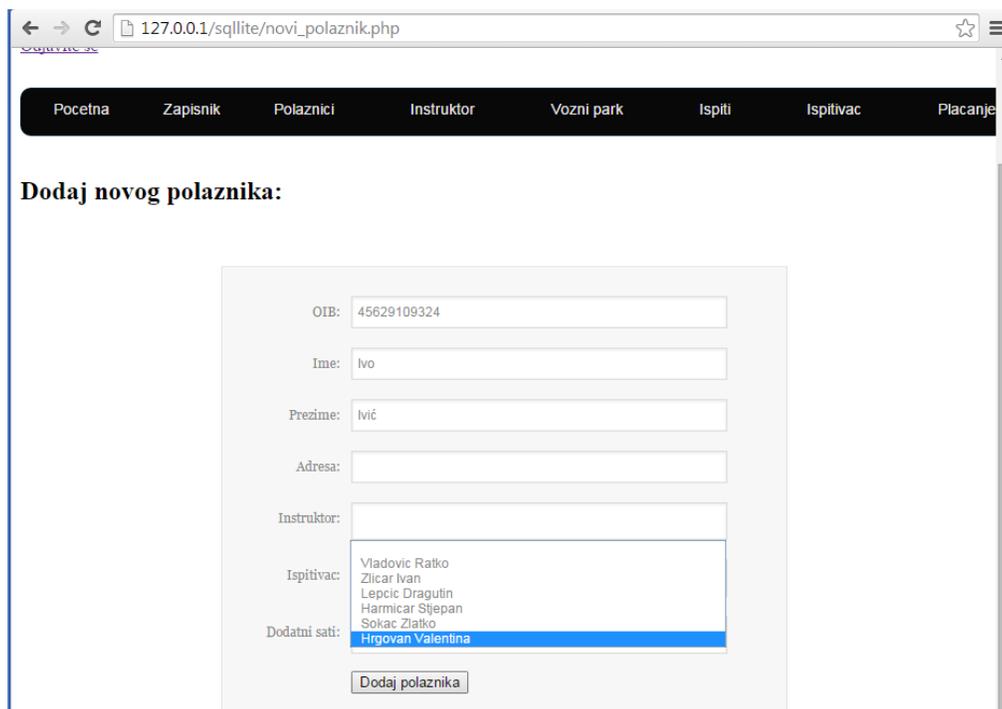
ID	Naziv	VRIJEME	Radnja
1	administrator	14-10-2014 20-30-38	Logiranje
2	administrator	14-10-2014 20-31-38	Logiranje
3	administrator	14-10-2014 20-32-38	Logiranje
4	administrator	14-10-2014 20-33-12	Odlogiran
5	racunovoda	14-10-2014 20-33-20	Logiranje
6	racunovoda	14-10-2014 20-33-22	Odlogiran
7	polaznik	15-10-2014 18-21-03	Logiranje
8	polaznik	15-10-2014 18-21-27	Odlogiran

Slika 5.4. Zapisnik prijavljenih i odjavljenih

Popis svih programskih kodova po datotekama može se pogledati na CD-u (vidjeti *Prilog A: CD sa popisom programskih kodova po datotekama*).

6. PRIMJERI KORIŠTENJA

Aplikacija je osmišljena tako da će njome biti vrlo lako rukovati. Pregled svih podataka moguć je samo jednim klikom miša, dok će se unos novih podataka u bazu podataka obavljati preko formi koje će biti oku ugodnije i preglednije. Da bi se spriječilo neovlašteno čitanje i izmjena podataka, samo administrator će imati ovlasti dodavati nove podatke. Iznimka će biti samo opcija plaćanje, koju će uz administratora moći vidjeti i računovođa. No međutim, računovođa u cijeloj aplikaciji vidi samo i isključivo opciju sa plaćanjem. U nastavku će biti prikazan način unosa novih podataka i korištenja same aplikacije.



127.0.0.1/sqlite/novi_polaznik.php

Pocetna Zapisnik Polaznici Instruktor Vozni park Ispiti Ispitivac Placanje

Dodaj novog polaznika:

OIB: 45629109324

Ime: Ivo

Prezime: Ivić

Adresa:

Instruktor:

Ispitivac: Vladovic Ratko
Zlicar Ivan
Lepcic Dragutin
Harmicar Stjepan
Sokac Zlatko
Hrgovan Valentina

Dodatni sati: Hrgovan Valentina

Dodaj polaznika

Slika 6.1. Dodavanje novog polaznika

Kao što *Slika 6.1. Dodavanje novog polaznika* prikazuje, forma je vrlo jednostavna za korištenje. Unosit će se OIB polaznika, ime, prezime, adresa. Klikom na instruktora, polaznik će reći kojeg instruktora želi, a administratoru će se otvoriti padajući izbornik sa popisom svih instruktora. Također će se kod Ispitivača otvoriti padajući izbornik sa popisom svih ispitivača, te će se polazniku od strane autoškole odrediti ispitivač kojeg će imati na polaganju. Također, ukoliko netko od polaznika iz neke vježbe želi pohađati dodatne sate, administrator će putem padajućeg izbornika

pronaći dodatni sat kojeg polaznik želi, te ga dodati. Klikom miša na gumb dodaj polaznika, novi polaznik ubacit će se u bazu podataka, i također biti ispisan u tablici u aplikaciji pod opcijom koju administrator ima, a to je popis polaznika. Sve forme u aplikaciji funkcionirat će na isti princip, a to je samo da se unesu željeni podaci, ili pak da se između ponuđenog izbora u padajućem izborniku odabere podatak.

Slika 6.2. Uređivanje i brisanje polaznika prikazat će kako vrlo jednostavno obrisati ili urediti podatke. U tablici Polaznik će uz sve stupce koji se odnose na njegove podatke, kao što su na primjer ime, prezime, adresa, odabrani instruktor i sl., pokraj podataka svih polaznika nalaziti opcija za uređivanje ili pak brisanje podataka. Promjena podataka bit će vrlo jednostavna jer će sadržavati formu za unos novih podataka, koja je slična formi za unos novih polaznika. Potrebno je samo nakon što će se unjeti novi odgovarajući podaci polaznika, kliknuti na tipku „Obnovi“ i polazniku će se promijeniti podaci. Brisanje će biti još jednostavnije tako što će pokraj opcije „Uredi“ biti opcija „Brisanje“ koja čim se klikne na nju obriše odabranog polaznika. Brisanje je predviđeno za brisanje polaznika koji su položili ili odustali od pohađanja autoškole.

Sustav upravljanja bazom podataka Auto škole

Ulogirani ste kao ADMIN
[Odjavite se](#)

Pocetna Zapisnik **Polaznici** Instruktor Vozni park Ispiti Ispitivac Placanje

Popis polaznika
 Dodaj polaznika

ID	Ime	Prezime	Adresa	Instruktor	Ispitivac	Dodatni sati	Uredi	Brisi
12345678	Josip	Jug	Vedra Nebesa 12	Dragutin Lepcic	Rene Bitorajac	Slalom	Uredi	Brisanje
12345678912	Josip	Bolfan	Gregurovec 7, Krizevci	Ratko Vladovic	Tarik Filipovic	Intenzivno kocenje	Uredi	Brisanje
12345678913	Petra	Pavlic	Guscerovec 103, Krizevci	Ivan Zlicar	Rene Bitorajac	Gradska voznja	Uredi	Brisanje
12345678914	Tanja	Novosel	Fodrovec 14, Krizevci	Valentina Hrgovan	Jasmin Stavros	Gradska voznja	Uredi	Brisanje
12345678915	Marina	Matus	Franje Rackog 3, Krizevci	Dragutin Lepcic	Rene Bitorajac	Intenzivno kocenje	Uredi	Brisanje

Slika 6.2. Uređivanje i brisanje polaznika

7. ZAKLJUČAK

Izrada baze podataka u SQLite sustavu, pomoću naredbenih linija kodova prilično je jednostavna. Za svaki pojedini upit uglavnom se koristi po jedna ključna riječ koja je ujedno i naredba. Važno je zapravo naučiti osnovnu sintaksu. Jedna riječ uglavnom čini jednu naredbu kao što je dohvati (select), obriši (delete), dodaj (insert), promijeni (update). Nakon što je baza podataka bila kreirana, i u toj bazi podataka nakon što su sve tablice bile popunjene podacima, uočio se problem kod definiranih primarnih ključeva, što je kasnije bio problem kod izrade aplikacije. Naime, svakoj tablici u bazi podataka, stupac sa nazivom primarnog ključa bio je jednak (naziv ID). Radi lakšeg snalaženja kod izrade aplikacije, uveliko bi olakšalo da je naziv stupca sa primarnim ključevima bio različit, na primjer, naziv primarnog ključa tablice Polaznik – OIB_Polaznik, a ne ID, kao i naziv primarnog ključa tablice Vozilo – ID_Vozila itd. Za izrađenu aplikaciju smatra se da je korektna i ima sve potrebne elemente da zadovolje djelatnika u autoškoli, i vrlo ju je lako koristiti. Dakle, funkcionalnost aplikacije je neupitna, no prilikom izrade neka rješenja su mogla biti i drugačije napravljena. Tako se mogla napraviti jedna datoteka, primjerice naziva `spajanje.php` u kojoj se moglo definirati spajanje na bazu podataka. Također se ta ista datoteka mogla koristiti, pozivati u svim drugim datotekama pomoću ključne riječi (funkcije) `include`. Na taj bi se način uštedjelo na vremenu da se kreiraju ostale datoteke za izradu aplikacije, jednom napravljeno spajanje na bazu podataka ne bi trebalo raditi u svakoj datoteci, time bi se i smanjio broj linija kodova, a funkcionalnost bi bila jednaka. Također, mogle su biti implementirane još neke uloge. Tako se npr. mogla implementirati uloga instruktora, koji bi vidio popis svih polaznika koje ima na osposobljavanju, mogućnost slanja zahtjeva da promijeni vozilo koje koristi i/ili da se doda opcija da instruktor onemogući da mu se dodijeli novi polaznik iz razloga što mu je kvota polaznika koju može imati popunjena. Uglavom, mogućnosti dodavanja opcija su brojne. Unatoč navedenim nedostacima ili mogućim poboljšanjima, smatra se da je baza podataka obuhvatila sve moguće podatke koji su potrebni za vođenje autoškole, a da ih je aplikacija na jednostavan način i prikazala.

Literatura

1. Baniček, D. (2004.) *PHP/MySQL razvoj dinamičnih web stranica* Varaždin: PRO - MIL
2. Kreibich, A. J. (2010.) *Using SQLite* O'Reilly Media, Inc
3. Rabuzin, K. (2011.) *Uvod u SQL* Varaždin: Fakultet organizacije i informatike
4. *Osnove PHP-a*, dostupno 20.09.2014. na <http://phptutorial.odlican.net/osnovephpa.php>
5. *PHP 5*, dostupno 20.09.2014. na <http://www.w3schools.com/php>
6. *Podaci, modeli podataka i modeliranje produkcijskih i dimenzijskih baza podataka*, dostupno 22.9.2014 na http://informatika.efos.hr/wp-content/uploads/2012/04/P9_Baze%20podataka.pdf
7. *SQLite*, dostupno 15.09.2014. na <http://www.sqlite.org>
8. *SQLite SQL Database Engine*, dostupno 15.9.2014. na <http://www.tutorialspoint.com/sqlite/>