# Tehnički fakultet „Mihajlo Pupin“, Zrenjanin RAZVOJ SOFTVERA OTVORENOG KODA

Školska 2019/2020. godina

SEMINARSKI RAD

# Uvod

Izabrana tema, opis problema, sistema, firme, institucije, organizacije ili dela sistema za koji se razvija softver.

# Specifikacija zahteva korisnika

Spisak zahteva korisnika.

1. **Faze razvoja softvera** Spisak faza, aktivnosti, izvođača. Ideja, koncept rešenja.

Arhitektura softvera: troslojna ili višeslojna arhitektura softvera Specifikacija ili modeli svakog sloja, konceptualni dizajn

# Prikaz softvera

Opis stranica, pokretanje softvera, formi, izveštaja, delova sa ekranskim prikazima.

# Prikaz realizacija i implementacije

Prikaz karakterističnih elemenata realizacije i objašnjenja realizacije i programskog koda, oznaka, biblioteka klasa i ostalih elemenata softvera.

Rad sa Git/Github sistemom.

Opis testiranja softvera ili njegovih delova (klasa).

# Korišteni alati i softveri

Navesti spisak korištenih alata, softvera, programskih jezika, razvojnih okruženja i njihovih verzija.

# Literatura

Spisak knjiga (štampanih i e-izdanja), praktikuma, zbirki, biblioteka, radnih prostora, izvora sa Internet-a (sajtovi, linkovi, materijali, url adrese)

# UVOD

Seminarski rad iz predmeta Razvoj softvera otvorenog koda obuhvata izradu web aplikacije na temu Zoo vrt. Imamo unos ljudi koji kupuju karte u Zoo vrt.

Softver obuhvata: unos novih musterija, pretraga i pregled musterija, izmenu i brisanje.

# SPECIFIKACIJA ZAHTEVA KORISNIKA

Zahtev 1: Uci u program i upisati informacije sa tabela.

Zahtev 2: Kliknuti na dugme snimi.

Zahtev 3. Pregled svih karata iz baza

Zahtev 4. Izmena svih karata

Zahtev 5. Filtriranje pretrage

Zahtev 6. Stampanje karata

# Faze razvoja softvera

## Projektne ideje

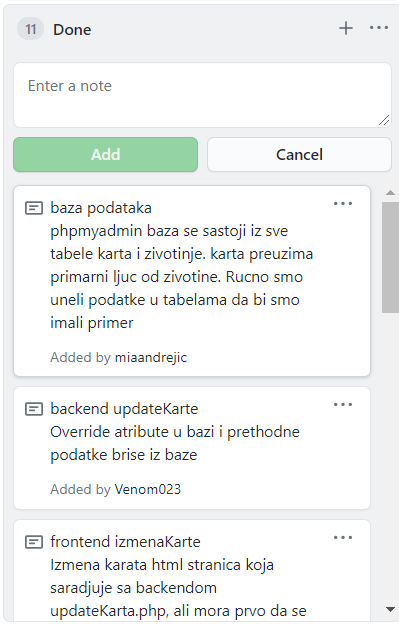
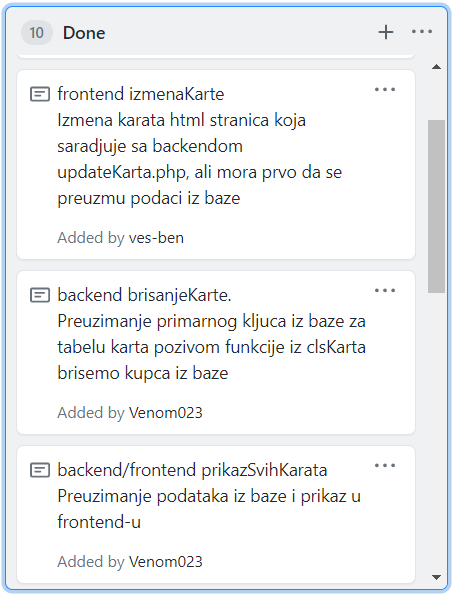
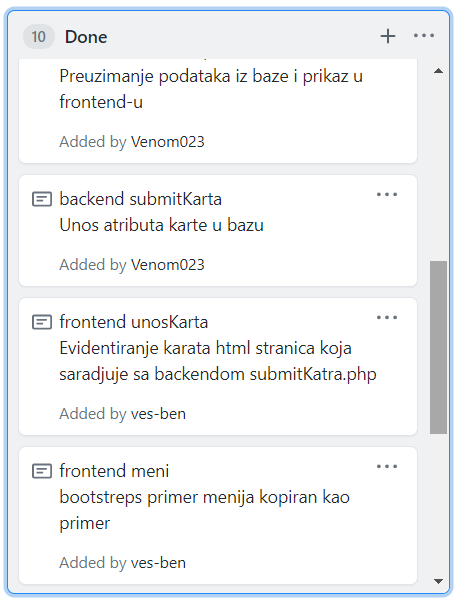
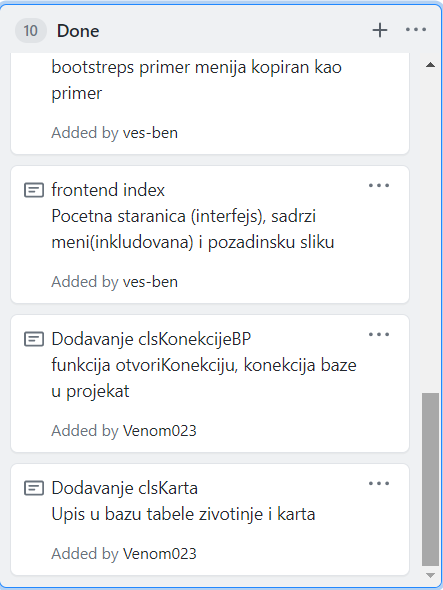
Softver: web aplikacija (PHP/HTML/CSS/JS), relaciona baza podataka (MySQL sistem za rukovanje bazama podataka), troslojna arhitektura softvera (Baza podataka – Srednji aplikacioni sloj/poslovna logika/engl. “Backend” – Korisnički interfejs KI (“User interface”, UI, engl. “Frontend”)

Alati za razvoj: Sybase/SAP Power Designer CASE alata za projektovanje softvera, PHPMyAdmin web ap- likaciju za administraciju baze podataka, Appache web server, MySQL server baze podataka, PHP pro- gramski jezik za srednji aplikacioni sloj (Editor: Visual Studio Code), PHP/HTML/CSS/JS za KI uz korišćenje Bootstrap radnog okvira. Dokumentovanje u MS Word tekst procesoru. Testiranje.

Izvršavanje softvera: hostovanje baze podataka i softvera na lokalnom serveru (MS Windows OS, XAMP)

## Dizajn softvera i arhitekture

Faze razvoja softvera:

****

Sloj baze podataka: Konceptualni model baze podataka (BP, DB) EER dijagram:



Karta

RBkom binaci ja <pi> Serial < M >

Id Karte Naziv Karte Email kupca

Cena

Vrsta karte

Id Zivot.

Integer < M >

Zivotinje

Char CharChar

Char

Char

Char

Char

< M >

< M >

< M >

< M >

< M >

< M >

< M >

< M >

(D) Im a

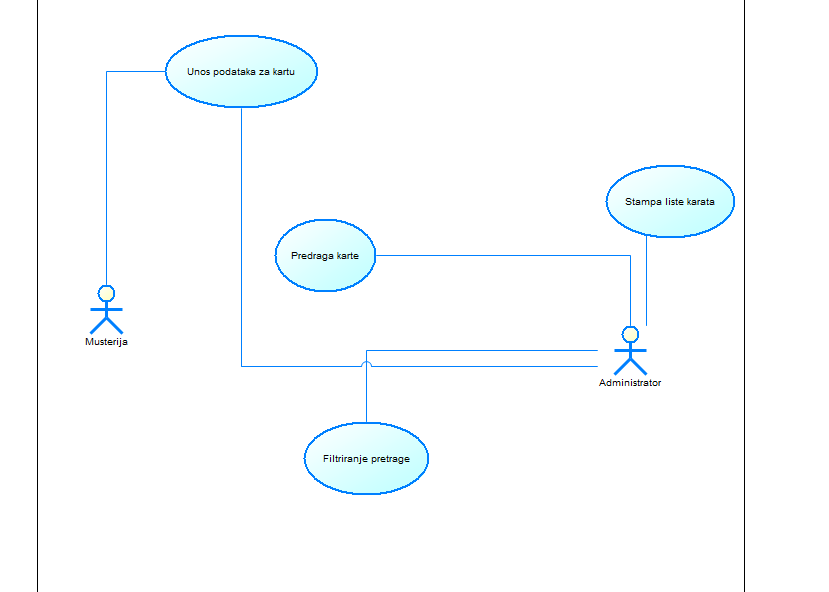
NazivZiv

Char < M >

IdZiv. Char< M >

Sloj baze podataka: Fizički model baze podataka (BP,DB)

Sloj KI: Softverske funkcije i korisnici (dijagram slučajeva korišćenja):

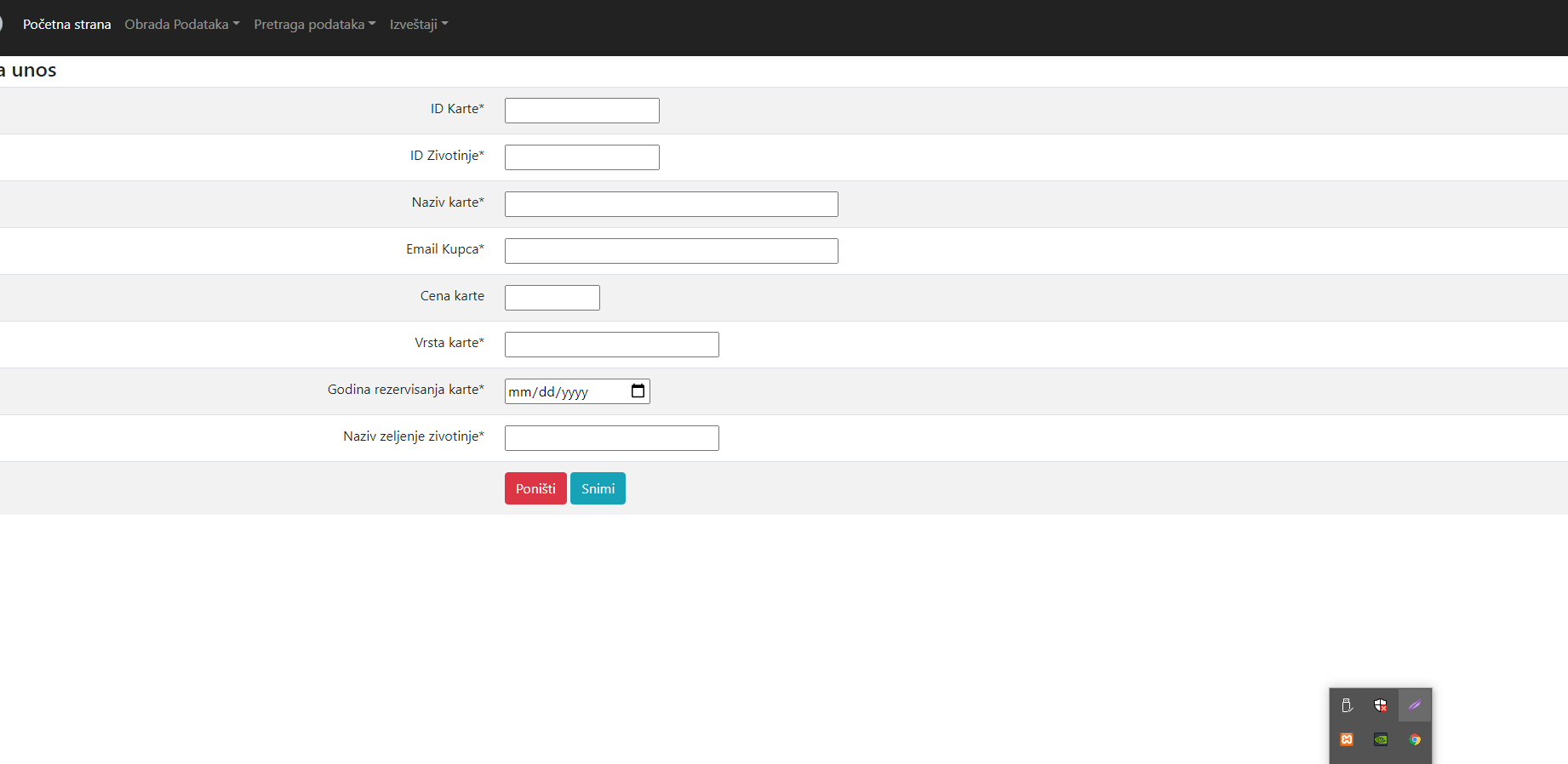


# Prikaz softvera

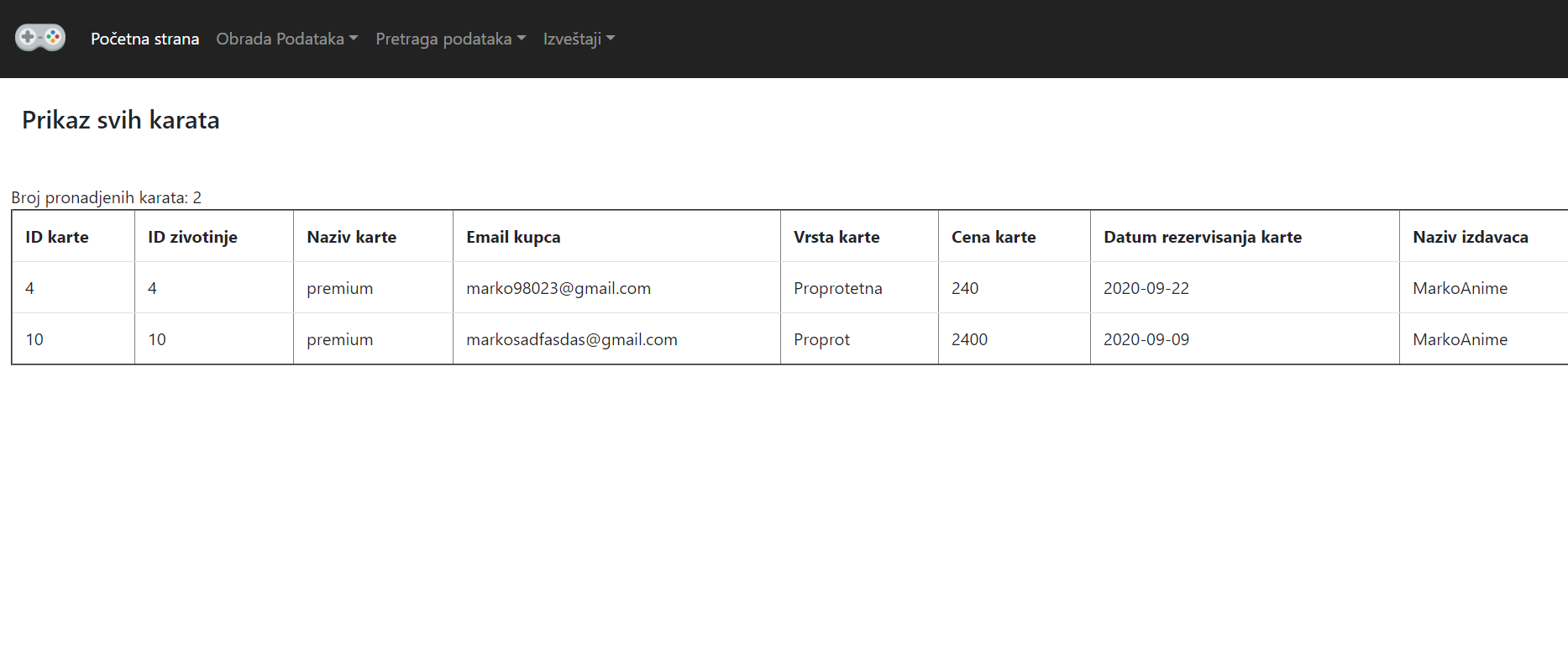
Sloj KI: Dizajn korisničkog interfejsa (KI, UI)



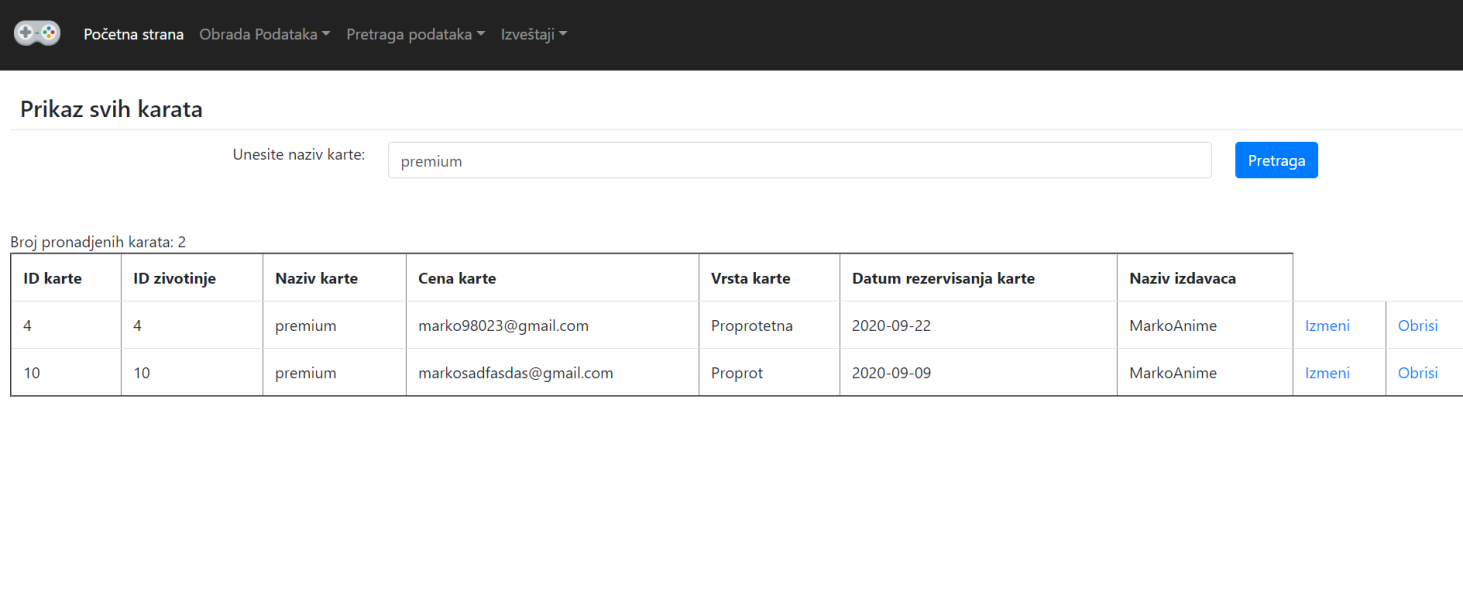
*Slika 5. Korisnicki interfejs*



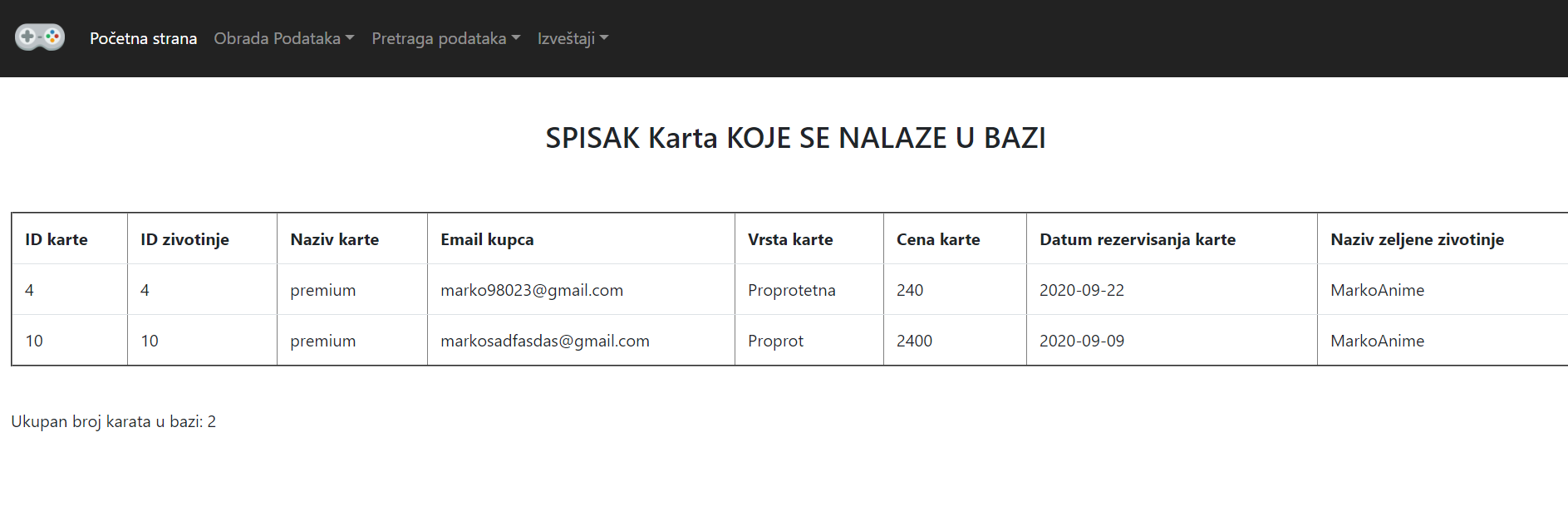
*Slika 6. Admin interfejs*



*Slika 7. Prikaz svih karata*



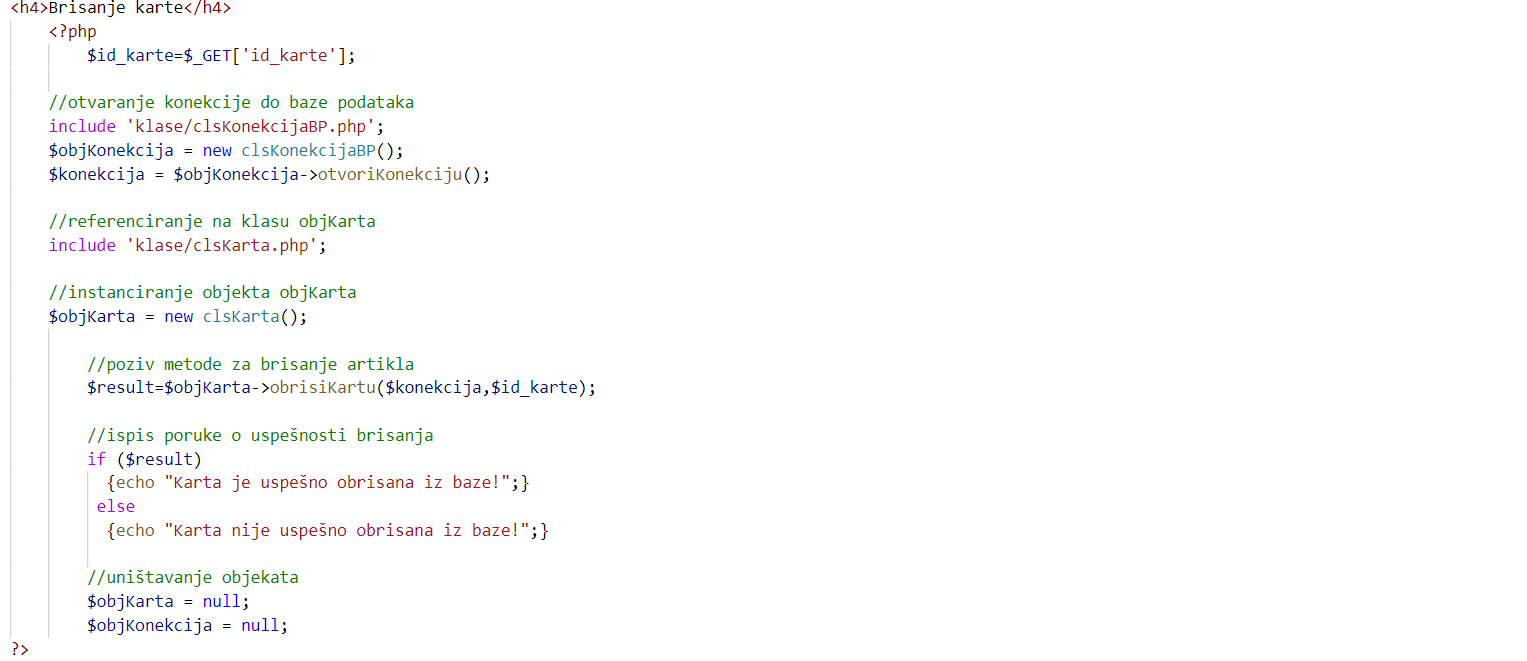
*Slika 8. Pretraga karata*



*Slika 9. Spisak karata iz baze*

1. **Prikaz realizacija i implementacije**

|  |
| --- |
| -- phpMyAdmin SQL Dump |
|  | -- version 4.9.2 |
|  | -- https://www.phpmyadmin.net/ |
|  | -- |
|  | -- Host: 127.0.0.1 |
|  | -- Generation Time: Jun 01, 2020 at 11:26 PM |
|  | -- Server version: 10.4.11-MariaDB |
|  | -- PHP Version: 7.2.26 |
|  |  |
|  | SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO"; |
|  | SET AUTOCOMMIT = 0; |
|  | START TRANSACTION; |
|  | SET time\_zone = "+00:00"; |
|  |  |
|  |  |
|  | /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/; |
|  | /\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/; |
|  | /\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/; |
|  | /\*!40101 SET NAMES utf8mb4 \*/; |
|  |  |
|  | -- |
|  | -- Database: `zoovrt` |
|  | -- |
|  |  |
|  | -- -------------------------------------------------------- |
|  |  |
|  | -- |
|  | -- Table structure for table `karta` |
|  | -- |
|  |  |
|  | CREATE TABLE `karta` ( |
|  | `id\_karte` int(11) NOT NULL, |
|  | `nazivkarte` varchar(256) NOT NULL, |
|  | `emailKupca` varchar(60) NOT NULL, |
|  | `cena` int(11) NOT NULL, |
|  | `vrstaKarte` varchar(30) NOT NULL, |
|  | `id\_zivotinje` int(11) NOT NULL |
|  | ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4; |
|  |  |
|  | -- |
|  | -- Dumping data for table `karta` |
|  | -- |
|  |  |
|  | INSERT INTO `karta` (`id\_karte`, `nazivkarte`, `emailKupca`, `cena`, `vrstaKarte`, `id\_zivotinje`) VALUES |
|  | (2, 'premium', 'djoleIvanovic23@gmail.com', 1200, 'Proprotetna', 2), |
|  | (4, 'standard', 'nikolasJovankov@yahoo.com', 750, 'Obicna', 4); |
|  |  |
|  | -- -------------------------------------------------------- |
|  |  |
|  | -- |
|  | -- Table structure for table `zivotinje` |
|  | -- |
|  |  |
|  | CREATE TABLE `zivotinje` ( |
|  | `id\_zivotinje` int(11) NOT NULL, |
|  | `godina\_zivotinje` date NOT NULL, |
|  | `nazivzivotinje` varchar(50) NOT NULL |
|  | ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4; |
|  |  |
|  | -- |
|  | -- Dumping data for table `zivotinje` |
|  | -- |
|  |  |
|  | INSERT INTO `zivotinje` (`id\_zivotinje`, `godina\_zivotinje`, `nazivzivotinje`) VALUES |
|  | (2, '2000-05-10', 'Jole'), |
|  | (4, '2014-10-02', 'Radisa'); |
|  |  |
|  | -- |
|  | -- Indexes for dumped tables |
|  | -- |
|  |  |
|  | -- |
|  | -- Indexes for table `karta` |
|  | -- |
|  | ALTER TABLE `karta` |
|  | ADD PRIMARY KEY (`id\_karte`), |
|  | ADD KEY `id\_zivotinje` (`id\_zivotinje`); |
|  |  |
|  | -- |
|  | -- Indexes for table `zivotinje` |
|  | -- |
|  | ALTER TABLE `zivotinje` |
|  | ADD PRIMARY KEY (`id\_zivotinje`); |
|  |  |
|  | -- |
|  | -- AUTO\_INCREMENT for dumped tables |
|  | -- |
|  |  |
|  | -- |
|  | -- AUTO\_INCREMENT for table `karta` |
|  | -- |
|  | ALTER TABLE `karta` |
|  | MODIFY `id\_karte` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9; |
|  |  |
|  | -- |
|  | -- AUTO\_INCREMENT for table `zivotinje` |
|  | -- |
|  | ALTER TABLE `zivotinje` |
|  | MODIFY `id\_zivotinje` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9; |
|  | COMMIT; |
|  |  |
|  | /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/; |
|  | /\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/; |
|  | /\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/; |



*Slika 10. Prikaz koda brisanje karte*



*Slika 11. Prikaz koda unos karte*

.

*Slika 12. SubmitKarta*

## Rad sa Git/Github sistemom

GitHub je dizajniran imajući u vidu jedan određeni tok rada za kolaboraciju, koji je centralizovan oko zahteva za povlačenjem. Ovaj način rada funkcioniše bilo da sarađujete sa malim timom prijatleja koji imaju samo jedan deljivi repozitorijum, ili sa globalno distrbuiranom kompanijom ili mrežom stranaca koji doprinose projektu preko brojnih forkova.

Evo kako radi u opštem slučaju:

1. Napravite tematsku granu iz master.
2. Napravite neke komitove da poboljšate projekat.
3. Gurnite ovu granu na svoj GitHub projekat.
4. Otvorite zahtev za povlačenjem na GitHub-u.
5. Diskutujte, i opciono nastavite da komitujete.
6. Vlasnik projekta obradi zahtev za povlačenjem i zatvori ga.

**Opis testiranja softvera**

Testiranje smo izvrsili redovnim unosom, izmenom I brisanje podataka I recesivno ispravljali greske.

Imali smo problema sa button-om na stranici Evidencija pisi/brisi dugme za izmenu je brisalo unete podatke.

Kvalitet softvera se može definisati na različite načine kao na primer:

1. Usaglašenost sa zahtevima i potrebama korisnika – jedan od najbitnijih uslova da bi se softver ocenio kao kvalitetan je da pomaže krajnjim korisnicima u radu.
2. Dobri atributi proizvoda kao što su brzina rada, malo zauzeće memorije i prostora na disku, brzina pokretanja.
3. Lakoća održavanja i promena u softveru, kao i prenošenja na druge platforme. Kvalitet dokumentacije, zahteva, dizajna, uputstava za upotrebu i bilo kojih drugih pratećih dokumenata kojima se opisuje softver.
4. Usaglašenost sa standardima što podrazumeva usaglašenost sa organizacionim standardnima pisanja programskog koda, poštovanje opštih standarda (ISO), usaglašenost sa zakonima i slično.
5. Rad u ekstremnim uslovima sa ogromnim količinama podataka, slabim vezama, ograničenim resursima koji su na raspolaganju i slično.
6. **Korišteni alati i softveri**

* MySQL sistem za rukovanje bazama podataka
* PHPMyAdmin web aplikaciju za administraciju baze podataka,
* Appache web server,
* MySQL server baze podataka,
* PHP programski jezik za srednji aplikacioni sloj (Editor: Visual Studio Code),
* PHP/HTML/CSS/JS za KI uz korišćenje Bootstrap radnog okvira.

# Literatura

[1] <https://www.phpmyadmin.net/about/>

[2] <https://blog.getbootstrap.com/>

[3] <https://www.awwwards.com/blog/>

[4] <https://www.wpbeginner.com/beginners-guide/what-is-a-blog-and-how-is-it-different-from-a-website-explained/>

[5] <https://zooplanet.org.rs/wp-content/uploads/2017/10/top-image.jpg>