

Tehnički fakultet „Mihajlo Pupin“, Zrenjanin

RAZVOJ SOFTVERA OTVORENOG KODA

Školska 2019/2020. godina

SEMINARSKI RAD

1. Uvod

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

2. Specifikacija zahteva korisnika

Spisak zahteva korisnika.

3. Faze razvoja softvera

Spisak faza, aktivnosti, izvođača.

Ideja, koncept rešenja.

Arhitektura softvera: troslojna ili višeslojna arhitektura softvera

Specifikacija ili modeli svakog sloja, konceptualni dizajn

4. Prikaz softvera

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

5. 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).

6. Korišteni alati i softveri

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

7. Literatura

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

1. UVOD

Seminarski rad iz predmeta Razvoj softvera otvorenog koda obuhvata izradu web aplikacije za igru na sreću LOTO 7/39 koju organizuje Državna lutrija Srbije počev od 1985. godine. Na sajtu državne lutrije (<https://www.lutrija.rs/LottoGame>) se dva puta sedmično objavljuju kombinacije koje su izvučene, a na sajtu se nalaze i dodatne informacije, druge igre i arhiva ranije itvučenih brojeva, grafički prikazi i sl.

Softver obuhvata: unos novih kombinacija, pretraga i pregled kombinacija, izmenu i brisanje kombinacija, elementarnu statistiku, kao i generisanje slučajne kombinacije.

2. SPECIFIKACIJA ZAHTEVA KORISNIKA

Zahtev 1: Unos izvučenih kombinacija (sa arhive sajta lutrije ili tv prenosom), prioritet realizacije 1.

Upis 7 izvučenih celih brojeva, kola i godine.

Provera da li je kombinacija ispravno uneta? (uslov: broj1<broj2<broj3<broj4<broj5<broj6<broj7 uslov br. 2 da su svi uneti brojevi u rasponu od 1 do 39, uslov br. 3: kombinacija se ne sme memorisati više puta)

Zahtev 2: Pregled i pretraga unetih i memorisanih izvučenih kombinacija, prioritet realizacije 2.

Kriterijumi pretrage: (godina i/ili kolo od) i/ili (godina i/ili kolo do)

Pregled i prikaz podataka: tabelarno.

Zahtev 3: Izmena unetih i memorisanih izvučenih kombinacija, prioritet realizacije 4.

Nakon pretrage i prikaza unetih i memorisanih izvučenih kombinacija.

Izmena svih 7 brojeva, kola i godine.

Validacija da li je kombinacija izvučena?

Zahtev 4: Brisanje unetih i memorisanih izvučenih kombinacija, prioritet realizacije 3.

Nakon pretrage i prikaza unetih i memorisanih izvučenih kombinacija.

Potvrda brisanja.

Zahtev 5: Automatsko izračunavanje i prikaz elementarne statistike, prioritet realizacije 7.

Izračunavanje prilikom unosa izvučenih kombinacija.

Prikaz P,N,V1,V2,V3 prilikom pretrage i prikaza kombinacija.

Statistika: Tabelarni prikaz koliko puta je svaki broj izvučen (ukupno za celu bazu)

Zahtev 6: Generisanje slučajne kombinacije, prioritet realizacije 5.

Uslov 1: moraju biti različiti brojevi u rasponu od 1 do 39!

Uslov2: sortirati brojeve po rastućem redosledu!

Zahtev 7: Provera da li je kombinacija (slučajno generisana ili uneta od strane korisnika) već među izvučenim i memorisanim kombinacijama, prioritet realizacije 6.

Povezati sa realizacijom zahteva 6.

Omogućiti unos 7 brojeva koje igrač želi da uplati i proveriti da li je kombinacija već bila izvučena.

Zahtev 8: Kreiranje izveštaja o izvučenim brojevima i statistici, prioritet realizacije 8.

“Printer-friendly” stranica u formi dokumenta koji se može odštampati iz web čitača.

Vezati realizaciju sa zahtevom broj 2, nakon pretrage.

3. Faze razvoja softvera

a. Projektne ideje

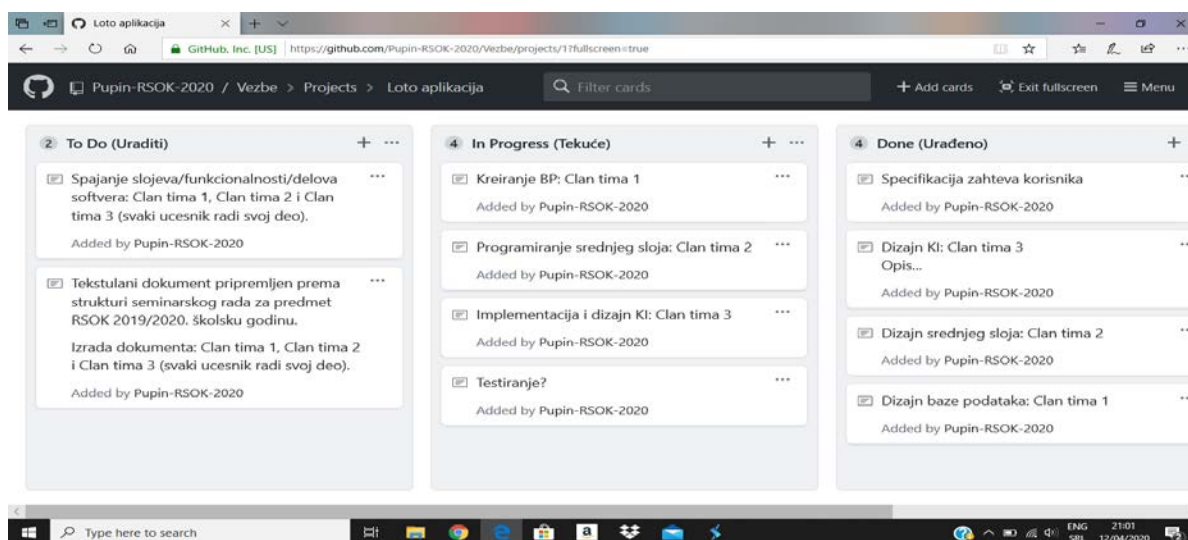
Softver: web aplikacija (PHP/HTML/CSS/JS), relaciona baza podataka (MySQL sistem za rukovanje bazama podataka), troslojna arhitektura softvera (Baza podataka – Srednji aplikacioni sloj/poslovna logika/engl. “Backend” – Korisnički interfejs KI (“User interface”, UI, engl. “Frontend”))

Alati za razvoj: Sybase/SAP Power Designer CASE alata za projektovanje softvera, PHPMyAdmin web aplikaciju za administraciju baze podataka, Apache 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. Dokumentovanje u MS Word tekst procesoru. Testiranje.

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

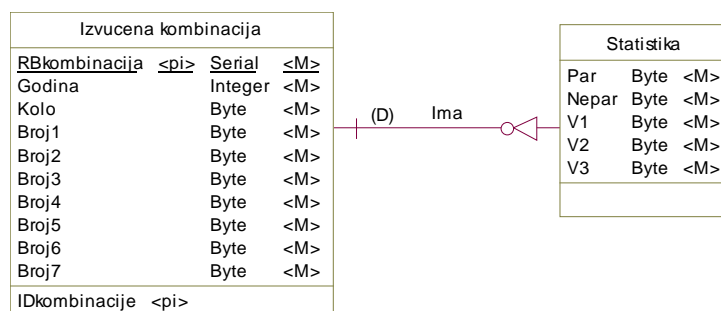
b. Dizajn softvera i arhitekture

Faze razvoja softvera:



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

EER dijagram:

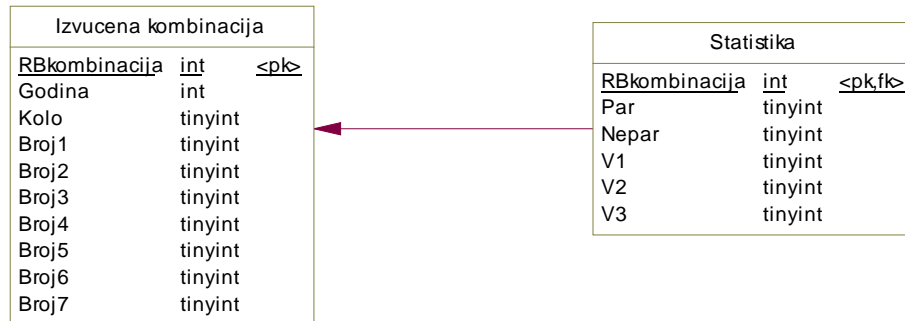


Rečnik podataka:

Naziv	Kod	Tip podatka	Enitet
Broj1	Broj1	Byte	Izvucena kombinacija
Broj2	Broj2	Byte	Izvucena kombinacija
Broj3	Broj3	Byte	Izvucena kombinacija
Broj4	Broj4	Byte	Izvucena kombinacija
Broj5	Broj5	Byte	Izvucena kombinacija
Broj6	Broj6	Byte	Izvucena kombinacija
Broj7	Broj7	Byte	Izvucena kombinacija
Godina	Godina	Integer	Izvucena kombinacija
Kolo	Kolo	Byte	Izvucena kombinacija
Nepar	Nepar	Byte	Statistika
Par	Par	Byte	Statistika
RBkombinacija	RBkombinacija	Serial	Izvucena kombinacija
V1	V1	Byte	Statistika
V2	V2	Byte	Statistika
V3	V3	Byte	Statistika

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

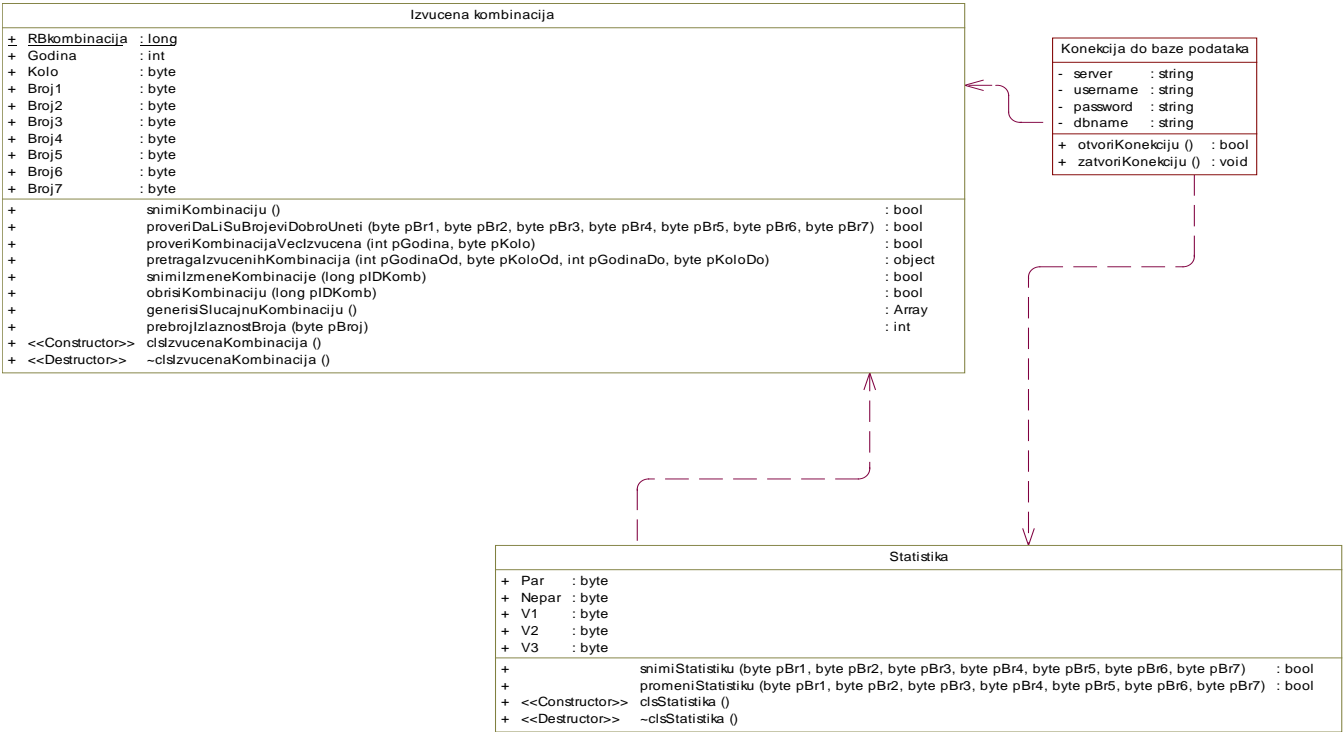
Dijagram relacionog modela:



Rečnik podataka:

Name	Code	Table	Data Type	PK	FK	Mandatory	CheckConstName
Broj1	Broj1	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Broj1
Broj2	Broj2	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Broj2
Broj3	Broj3	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Broj3
Broj4	Broj4	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Broj4
Broj5	Broj5	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Broj5
Broj6	Broj6	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Broj6
Broj7	Broj7	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Broj7
Godina	Godina	Izvucena kombinacija	int	FALSE	FALSE	TRUE	CK_Godina
Kolo	Kolo	Izvucena kombinacija	tinyint	FALSE	FALSE	TRUE	CK_Kolo
Nepar	Nepar	Statistika	tinyint	FALSE	FALSE	TRUE	CK_Nepar
Par	Par	Statistika	tinyint	FALSE	FALSE	TRUE	CK_Par
RBkombinacija	RBkombinacija	Statistika	int	TRUE	TRUE	TRUE	CK_RBkombinacija
RBkombinacija	RBkombinacija	Izvucena kombinacija	int	TRUE	FALSE	TRUE	CK_RBkombinacija
V1	V1	Statistika	tinyint	FALSE	FALSE	TRUE	CK_V1
V2	V2	Statistika	tinyint	FALSE	FALSE	TRUE	CK_V2
V3	V3	Statistika	tinyint	FALSE	FALSE	TRUE	CK_V3

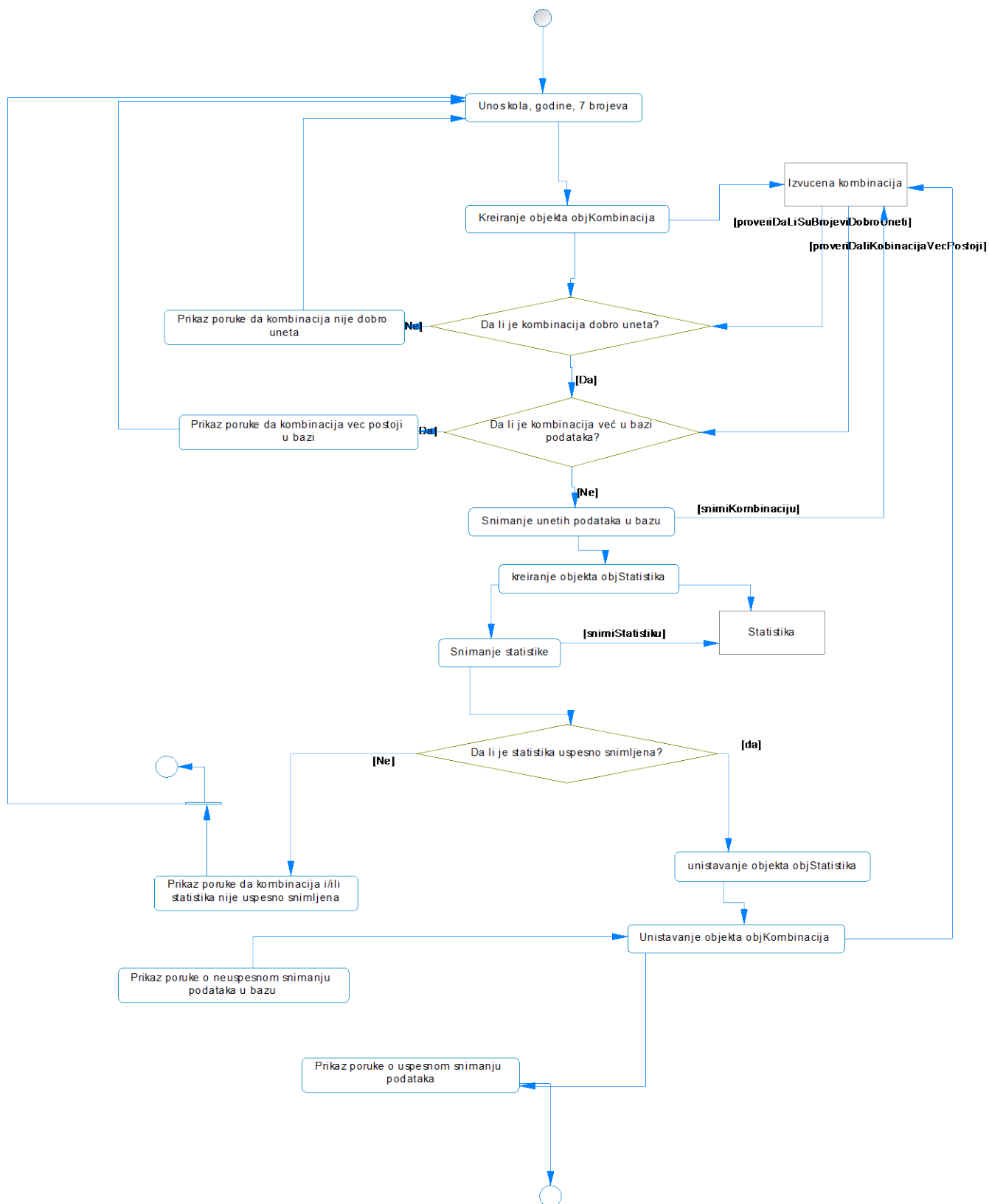
Srednji aplikacioni sloj: Objektni model srednjeg aplikacionog sloja (dijagram klasa objektnog modela):



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



Sloj KI: Logika i aktivnosti pojedinačnih slučajeva korišćenja (primer UNOS IZVUCENE KOMBINACIJE):



Sloj KI: Dizajn korisničkog interfejsa (KI, UI)

LOTO link (povratak na pocetnu)

pozadinska boja crna
slova bele boje...

Unos nove izvucene kombinacije Pretraga kombinacija Statistika Generisanje kombinacije Provera izlaznosti Informacije

Naslov stranice

- 1. Forma za unos kombinacije(dve kolone)
- 2.Kriterijum pretrage(Forma)
- 2.3.Tabelarni prikaz kombinacija i statistike
- 4.Ispis generisane kombinacije
- 5.Forma za unos kombinacije
- 6.O autorima i projektu

Pozadinska slika sa loto motivom
Fiksna slika bez skrolovanja

Tasteri za izvršavanje funkcija softvera(snimi, obriši, proveriti)

Slika sa loto motivom
Skrolovanje stranice (vertikalno i horizontalno)

4. Prikaz softvera

Pokretanje aplikacije:

Nakon instaliranja XAMP softverskog paketa, aplikacija se kopira u folder C:\Xamp\htdocs\Loto2020

U PHPMyAdmin softveru se kreira baza podataka sa nazivom: loto2020rsok

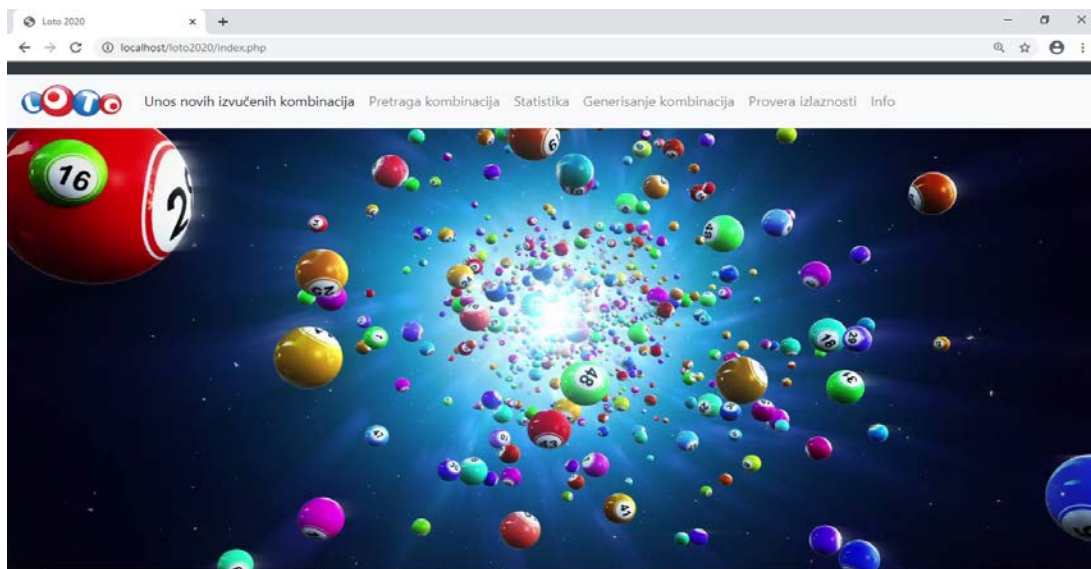
Zatim se u izvrši SQL DDL skript za kreiranje objekata baze podataka.

Pokretanje aplikacije je iz web browser-a, u URL polje potrebno je upisati:

<http://localhost/loto2020>

ili <http://localhost/loto2020/index.php>

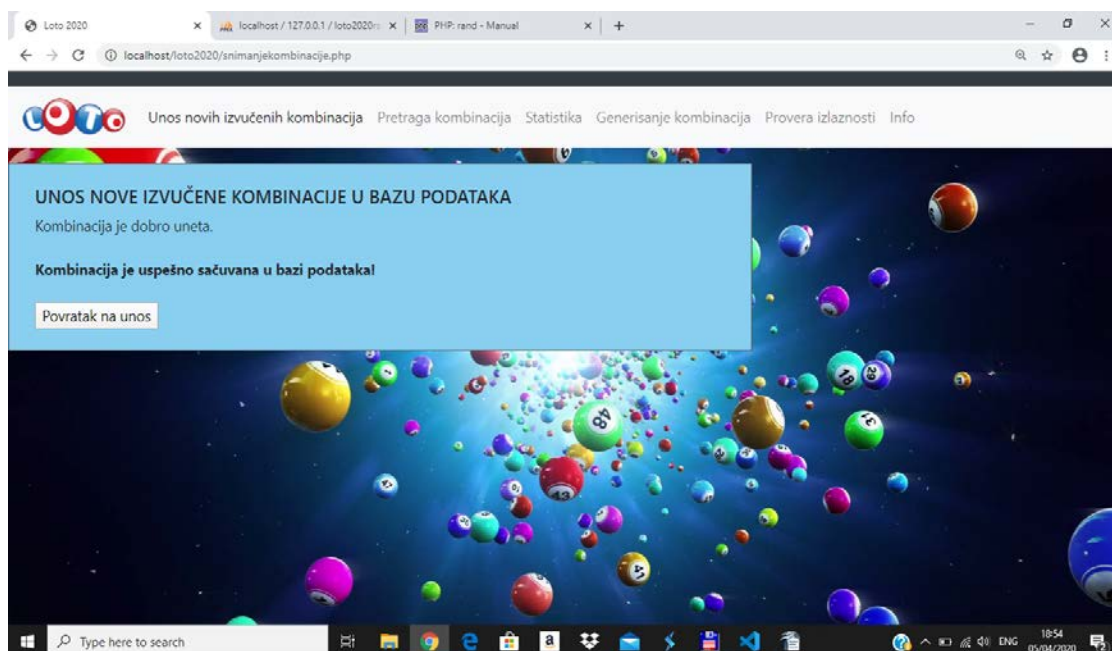
Početna stranica sa osnovni menijem aplikacije:



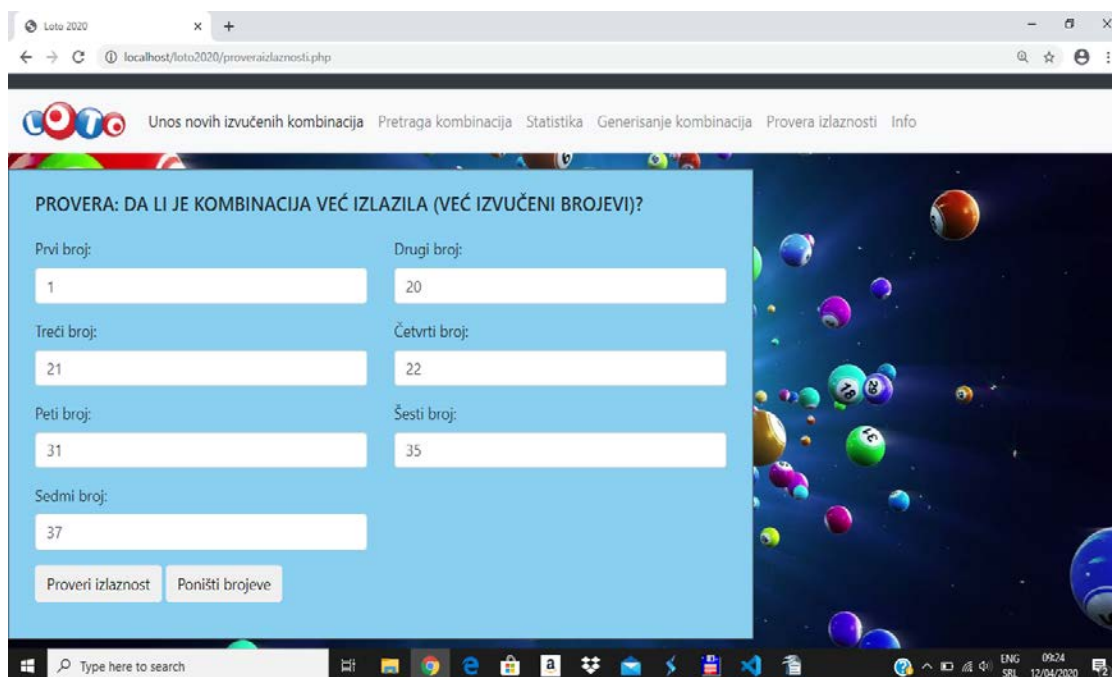
Stranica za unos novih kombinacija i brojeva u bazu preko forme:

The screenshot shows the 'Unos novih izvučenih kombinacija' page. The form is titled 'UNOS NOVE IZVUČENE KOMBINACIJE U BAZU PODATAKA'. It contains two columns of input fields. The left column has fields for 'Godina:' (Upišite godinu), 'Prvi broj:' (Unesite 1. broj), 'Treći broj:' (Unesite 3. broj), 'Peti broj:' (Unesite 5. broj), and 'Sedmi broj:' (Unesite 7. broj). The right column has fields for 'Kolo:' (Upišite broj kola), 'Drugi broj:' (Unesite 2. broj), 'Četvrti broj:' (Unesite 4. broj), and 'Šesti broj:' (Unesite 6. broj). At the bottom left of the form is a button labeled 'Sačuvaj kombinaciju'. The background of the page is the same dark blue space with colorful lottery balls as seen in the previous screenshot.

Stranica za snimanje unete kombinacije u bazu podataka i prikaz odgovarajućih poruka:



Stranica za proveru izlaznosti kombinacije:



Stranica za generisanje slučajne kombinacije:

Unos novih izvučenih kombinacija Pretraga kombinacija Statistika Generisanje kombinacija Provera izlaznosti Info

GENERISANJE SLUČAJNE KOMBINACIJE

Napomena: Ukoliko se slučajno dogodi da program generiše dva ista broja, možete im promeniti vrednosti ili generisati novu kombinaciju!

Prvi broj: 1 Drugi broj: 6

Treći broj: 12 Četvrti broj: 22

Peti broj: 29 Šesti broj: 31

Sedmi broj: 37

Proveri izlaznost

Stranica za pretragu i pregled kombinacija:

PRETRAGA KOMBINACIJA

Godina od 2019 do 2020

Redni broj kola od 1 do 10

Pronađi kombinacije Nov kriterijum pretrage

Pronađeno kombinacija: 5

Godina	Kolo	1.	2.	3.	4.	5.	6.	7.	Akcija 1	Akcija 2
2019	1	1	5	10	15	20	25	30	Izmena	Brisanje
2020	1	1	5	15	18	22	29	38	Izmena	Brisanje
2020	2	5	20	25	28	29	37	39	Izmena	Brisanje

Stranica za izmenu kombinacija:

Loto 2020

Unos novih izvučenih kombinacija Pretraga kombinacija Statistika Generisanje kombinacija Provera izlaznosti Info

IZMENA KOMBINACIJE U BAZI PODATAKA

Godina: 2019 Kolo: 1

Prvi broj: 1 Drugi broj: 5

Treći broj: 10 Četvrti broj: 15

Peti broj: 20 Šesti broj: 25

Sedmi broj: 30

Promeni kombinaciju

Brisanje kombinacije iz baze podataka:

Loto 2020

Unos novih izvučenih kombinacija Pretraga kombinacija Statistika Generisanje kombinacija Provera izlaznosti Info

BRISANJE KOMBINACIJE IZ BAZE PODATAKA

Kombinacija je uspešno obrisana iz baze podataka!

Povratak na pretragu

Statistika:

Broj	Ukupan broj izlazaka (koliko puta je broj izvučen)
1	0
2	1
3	0
4	1
5	1
6	2
7	0
8	1
9	0
10	1
11	0

Izveštaj:

The screenshot shows a web browser window with the address bar displaying 'localhost/lotto2020/zvestaj.php'. The page content includes the title 'LOTO 2020', a subtitle 'Izvučene kombinacije - 21.05.2020.', and a table of winning numbers for the years 1991, 2020, and 2020. A print dialog is open on the right side of the browser window, showing options for destination (Save as PDF), pages (All), layout (Portrait), and more settings. The print dialog also indicates '1 page' and has 'Save' and 'Cancel' buttons at the bottom.

LOTO 2020

Izvučene kombinacije - 21.05.2020.

Godina Kolo 1. broj 2. broj 3. broj 4. broj 5. broj 6. broj 7. broj Parnih Neparnih V1 V2 V3

1991	1	5	6	15	16	17	28	29	3	4	2	3	2
2020	1	15	25	26	33	35	37	38	2	5	2	3	2
2020	2	2	4	6	8	10	12	14	7	0	2	3	2

Ukupno prikazano kombinacija: 3

Print 1 page

Destination Save as PDF

Pages All

Layout Portrait

More settings

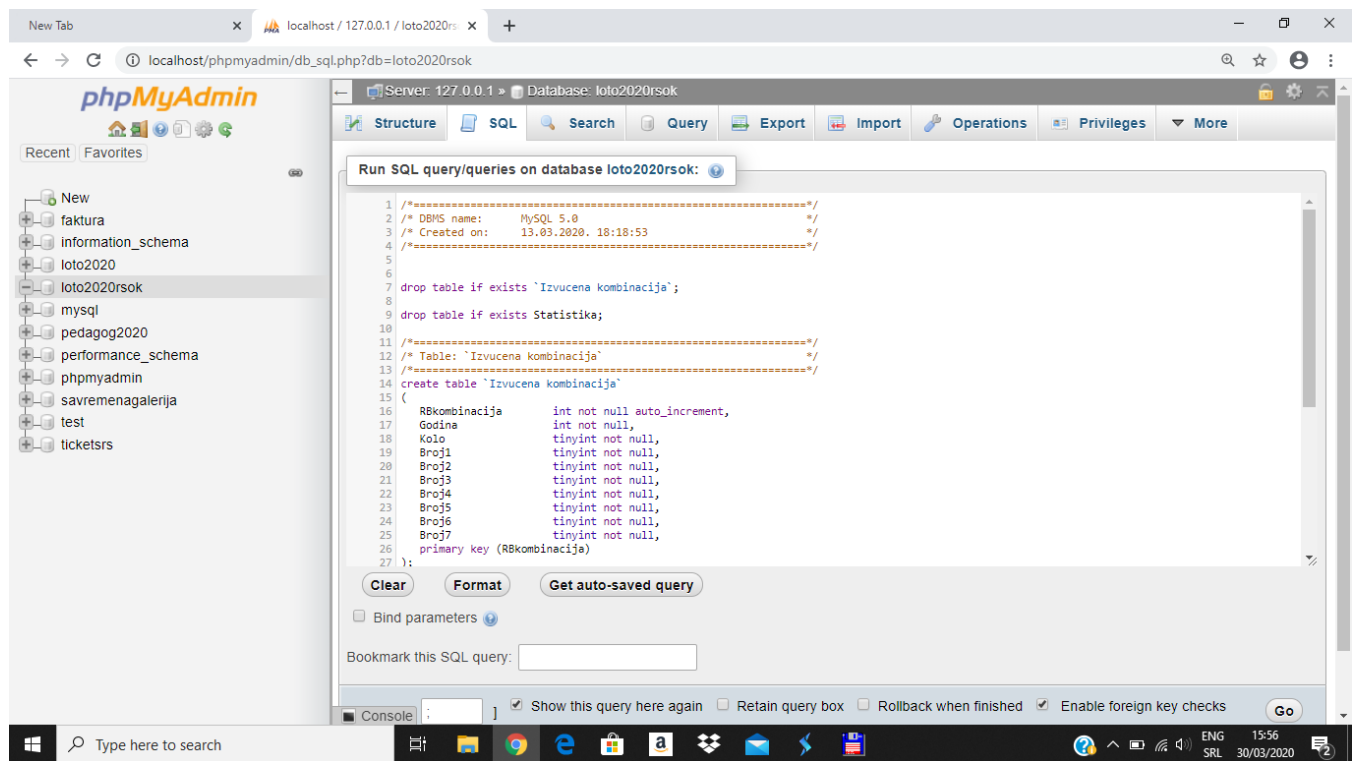
Save Cancel

5. Prikaz realizacija i implementacije

SQL DDL skript za kreiranje baze podataka:

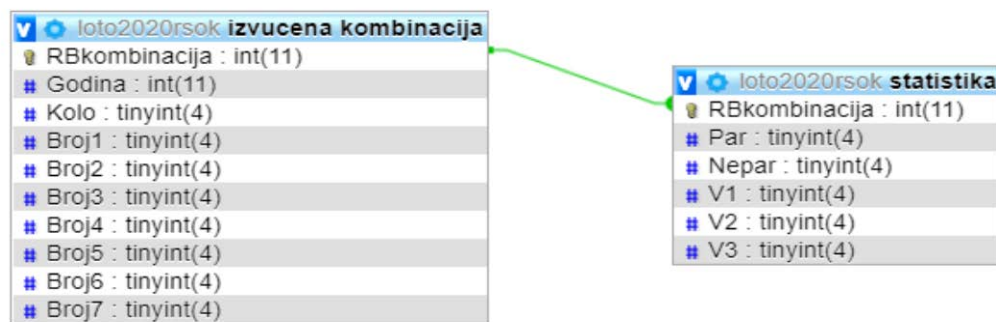
```
/*=====*/
/* DBMS name:    MySQL 5.0                                */
/* Created on:   13.03.2020. 18:18:53                      */
/*=====*/
drop table if exists `Izvucena kombinacija`;
drop table if exists Statistika;
/*=====*/
/* Table: `Izvucena kombinacija`                            */
/*=====*/
create table `Izvucena kombinacija`
(
  RBkombinacija    int not null auto_increment,
  Godina           int not null,
  Kolo             tinyint not null,
  Broj1            tinyint not null,
  Broj2            tinyint not null,
  Broj3            tinyint not null,
  Broj4            tinyint not null,
  Broj5            tinyint not null,
  Broj6            tinyint not null,
  Broj7            tinyint not null,
  primary key (RBkombinacija)
);
/*=====*/
/* Table: Statistika                                        */
/*=====*/
create table Statistika
(
  RBkombinacija    int not null,
  Par              tinyint not null,
  Nepar            tinyint not null,
  V1               tinyint not null,
  V2               tinyint not null,
  V3               tinyint not null,
  primary key (RBkombinacija)
);
alter table Statistika add constraint FK_Ima foreign key (RBkombinacija)
  references `Izvucena kombinacija` (RBkombinacija) on delete cascade on update cascade;
```

Kreiranje baze podataka, izvršavanjem SQL DDL skripta, u okviru PHPMyAdmin web aplikacije:



Naziv baze podataka: loto2020rsok.

Prikaz šeme baza podataka:



Prikaz izvornog koda aplikacije

Početna stranica **index.php** se sastoji od html oznaka u koje je umetnuta php naredba za uključivanje datoteke sa sadržajem menija. Ova stranica poseduje <html> <head> i <body> odeljke. Set karaktera je UTF-8, a korsiiti stil iz css datoteke. Sadržaj je formiran Bootstrap radnim okvirom, korišćenjem klase za navigacioni meni.

```
<html>
<head>
  <title>Loto 2020</title>
  <!-- Bootstrap --> <link href="css/bootstrap.min.css" rel="stylesheet">
  <!-- Skaliranje --> <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Set karaktera --> <meta charset="UTF-8">
  <!-- CSS --> <link rel="stylesheet" type="text/css" href="stil.css">
</head>
<body>

<nav class="navbar navbar-dark bg-dark">
  <!-- Prazan, prva linija -->
</nav>

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="index.php"></a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <!-- Meni -->
      <?php
        include 'meni.htm';
      ?>
    </div>
  </div>
</nav>

</body>
</html>
```

Sadržaj menija "meni.html" je nekoliko hiperlinkova koji otvaraju pojedinačne stranice istog izgleda i organizacije podataka:

```
<a class="nav-item nav-link active" href="unoskombinacije.php">Unos novih izvučenih kombinacija <span class="sr-
only">(current)</span></a>
<a class="nav-item nav-link" href="pretragakombinacija.php">Pretraga kombinacija</a>
<a class="nav-item nav-link" href="statistika.php">Statistika</a>
<a class="nav-item nav-link" href="generisanjekombinacije.php">Generisanje kombinacija</a>
<a class="nav-item nav-link" href="proveraizlaznosti.php">Provera izlaznosti</a>
<a class="nav-item nav-link" href="info.php">Info</a>
```

Dodatni stil stranice stil.css:

```
body {
  /* Slika */
  background-image: url("Lotoback.png");

  /* Minimalna širina pozadinske slike */
  min-height: 500px;

  /* Skrolovanje(scroll, fixed) */
  background-attachment: fixed;
```

```

/* Centriranje pozadinske slike */
background-position: center;

/* Neponavljanje pozadinske slike */
background-repeat: no-repeat;

/* Skaliranje na najveću moguću veličinu */
background-size: cover;
}

```

Forma za unos brojeva “unos.php” je Bootstrap dvokolonska forma sa poljima za unos brojeva, godine i kola. Forma prosleđuje svoje podatke drugoj web stranici POST metodom prenosa i na kojoj se vršisanje unetih brojeva u bazu:

```

<div class="col-md-6 col-lg-8" style="background-color: #89cff0; box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
  <p> </p>
  <div class="container">
    <!-- Naslov --></br>
    <h5>UNOS NOVE IZVUČENE KOMBINACIJE U BAZU PODATAKA</h5>
    <!-- Forma sa dve kolone -->
    <form class="form-horizontal" role="form" method='POST' action='snimanjekombinacije.php'>
      <div class="form-group row">
        <div class="col-sm-6">
          <label for="Godina">Godina:</label>
          <input type="text" class="form-control" id="Godina" name="Godina" required placeholder="Upišite godinu">
        </div>
        <div class="col-sm-6">
          <label for="Kolo">Kolo:</label>
          <input type="text" class="form-control" id="Kolo" name="Kolo" required placeholder="Upišite broj kola">
        </div>
      </div>
      <div class="form-group row">
        <div class="col-sm-6">
          <label for="Broj1">Prvi broj:</label>
          <input type="text" class="form-control" id="Broj1" name="Broj1" required placeholder="Unesite 1. broj">
        </div>
        <div class="col-sm-6">
          <label for="Broj2">Drugi broj:</label>
          <input type="text" class="form-control" id="Broj2" name="Broj2" required placeholder="Unesite 2. broj">
        </div>
      </div>
      <div class="form-group row">
        <div class="col-sm-6">
          <label for="Broj3">Treći broj:</label>
          <input type="text" class="form-control" id="Broj3" name="Broj3" required placeholder="Unesite 3. broj">
        </div>
        <div class="col-sm-6">
          <label for="Broj4">Četvrti broj:</label>
          <input type="text" class="form-control" id="Broj4" name="Broj4" required placeholder="Unesite 4. broj">
        </div>
      </div>
      <div class="form-group row">
        <div class="col-sm-6">
          <label for="Broj5">Peti broj:</label>
          <input type="text" class="form-control" id="Broj5" name="Broj5" required placeholder="Unesite 5. broj">
        </div>
        <div class="col-sm-6">
          <label for="Broj6">Šesti broj:</label>
          <input type="text" class="form-control" id="Broj6" name="Broj6" required placeholder="Unesite 6. broj">
        </div>
      </div>
    </div>
  </div>

```

```

<div class="form-group row">
  <div class="col-sm-6">
    <label for="Broj7">Sedmi broj:</label>
    <input type="text" class="form-control" id="Broj7" name="Broj7" required placeholder="Unesite 7. broj">
  </div>
</div>
<div class="form-group row">
  <div class="col-sm-offset-2 col-sm-10">
    <button type="submit" class="btn btn-default">Sačuvaj kombinaciju</button>
  </div>
</div>
</form>
</br>
</div>
</div>

```

Snimanje unetih brojeva u bazu podataka – Prvo se u promenljive pruzimaju vrednosti iz POST objekta, kreira se objekat tipa klase kombinacija, poziva metoda za proveru da li su brojevi uneti u skladu sa zahtevima korisnika, proverava da li kombinacija već postoji drugom metodom, te ako je sve u redu, vrši se snimanje podataka u relacionu bazu odgovarajućom metodom (preko INSERT SQL upita, tj. komande), a pre poziva metode se dodeljuju vrednosti atributima objekta klase na osnovu vrednosti promenljivih. Takođe se na ovoj stranici prikazuju i odgovarajuće poruke u zavisnosti od toga koji deo PHP koda je izvršen.

<h5>UNOS NOVE IZVUČENE KOMBINACIJE U BAZU PODATAKA</h5>

```

<!-- Provera da li kombinacija dobro uneta? -->
<!-- Kreiranje objekta kombinacija -->
<?php
$dobrouneta=false;
$godina=$_POST['Godina'];
$kolo=$_POST['Kolo'];
$broj1=$_POST['Broj1'];
$broj2=$_POST['Broj2'];
$broj3=$_POST['Broj3'];
$broj4=$_POST['Broj4'];
$broj5=$_POST['Broj5'];
$broj6=$_POST['Broj6'];
$broj7=$_POST['Broj7'];

/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();
/*provera da li su brojevi dobro uneti*/
$dobrouneta = $objKombinacija->proveriDaLiSuBrojeviIspravnoUneti($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7,$godina);
if ($dobrouneta)
{
  echo "Kombinacija je dobro uneta.";
  /*provera da li vec postoji to kolo i godina*/
  $postojivec = "";
  $postojivec = $objKombinacija->proveriDaLiKombinacijaPostoji($godina,$kolo);
  if ($postojivec)
  {
    echo '<b><br/><br/>';
    echo "Greška: Kombinacija sa unetom godinom i brojem kola već postoji u bazi podataka. ";
    echo '</b><br/><br/>';
    echo "Proverite unete podatke na prethodnoj stranici ili odustanite od unosa! ";
    echo '<br/>';
  }
  if (!$postojivec)
  {
    /*dodela vrednosti atributima objekta*/

```

```

$objKombinacija->godina = $godina;
$objKombinacija->kolo = $kolo;
$objKombinacija->broj1 = $broj1;
$objKombinacija->broj2 = $broj2;
$objKombinacija->broj3 = $broj3;
$objKombinacija->broj4 = $broj4;
$objKombinacija->broj5 = $broj5;
$objKombinacija->broj6 = $broj6;
$objKombinacija->broj7 = $broj7;
$result="";
$resultstat="";
/*snimanje objekta u bazu metodom snimiKombinaciju*/
/*snimanje statistickih parametara + citanje snimljenog rbkombinacije*/
$result = $objKombinacija->snimiKombinaciju();
$rbres = "";
$rbres = $objKombinacija->odrediRB();
$redrb = 0;
$redrb = mysqli_fetch_array($rbres);
$rb = $redrb['novirb'];
require 'class/clsstatistika.php';
$objStatistika = new clsstatistika();
$resultstat = $objStatistika->snimiStatistiku($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7,$rb);
if ($result && $resultstat)
//if ($result)
{
    echo '<br/><br/><b>';
    echo "Kombinacija sa statističkim parametrima je uspešno sačuvana u bazi podataka!";
    echo '</b><br/>';
}
else
{
    echo "Kombinacija nije snimljena u bazu podataka!";
}
}
}
else /*$dobrouneta*/
{
    echo '<b>';
    echo "Greška: Kombinacija nije dobro uneta. ";
    echo '</b><br/><br/>';
    echo "Svaki naredni broj mora biti veći od prethodnog! ";
    echo '<br/>';
    echo "Svi brojevi moraju biti u opsegu od 1 do 39! ";
    echo '<br/><br/>';
    echo "Proverite unete podatke na prethodnoj strani...";
    echo '<br/>';
} /*$dobrouneta*/
?>

```

Stranica za proveru izlaznosti kombinacija – Preko web forme se unosi 7 brojeva na isti način kao i prilikom unosa nove kombinacije, ali bez rubrika kola i godine. Podaci se POST metodom prosleđuju stranici za proveru izlaznosti gde se prihvataju u promenljive, kreira se objekat tipa klase kombinacija, poziva metoda za proveru da li su brojevi ispravno uneti, a zatim ako jesu poziva se metoda za proveru izlaznosti koja za parameter ima vrednosti promenljivih. Nakon izvršavanja metode, u zavisnosti od rezultata, sledi prikaz odgovarajuće poruke.

```
<?php
```

```

$dobrouneta=false;
$godina=2000;
/*godina se ne proverava da li je dobro uneta, pošto nije bitna za proveru izlaznosti brojeva*/
/*ali je vrednost potrebna da bi se mogla pozvati postojeća metoda za proveru da li je 7*/

```

```

/*izabranih brojeva uneto na pravilan način: 1<2<3<4<5<6<7 1..39*/
$broj1=$_POST['Broj1'];
$broj2=$_POST['Broj2'];
$broj3=$_POST['Broj3'];
$broj4=$_POST['Broj4'];
$broj5=$_POST['Broj5'];
$broj6=$_POST['Broj6'];
$broj7=$_POST['Broj7'];

/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();
/*provera da li su brojevi dobro uneti*/
$dobrouneta = $objKombinacija->proveriDaLiSuBrojeviIspravnoUneti($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7,$godina);
if ($dobrouneta)
{
    echo "Kombinacija je dobro uneta.";
    /*provera da li vec postoji to kolo i godina*/
    $postojivec = false;
    $postojivec = $objKombinacija->proveriIzlaznostKombinacije($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7);
    if ($postojivec)
    {
        /*prikaz poruke da kombinacija (7 brojeva) već postoji u bazi podataka*/
        echo '<b><br/><br/>';
        echo "Rezultat pretrage baze podataka: Kombinacija (";
        echo $broj1.", ".$broj2.", ".$broj3.", ".$broj4.", ".$broj5.", ".$broj6.", ".$broj7;
        echo ") je pronađena u bazi podataka. JESTE IZVUČENA!";
        echo '</b><br/>';
    }
    else /*od postoji vec*/
    {
        /*prikaz poruke da kombinacija (7 brojeva) ne postoji u bazi podataka*/
        echo '<b><br/><br/>';
        echo "Rezultat pretrage baze podataka: Kombinacija (";
        echo $broj1.", ".$broj2.", ".$broj3.", ".$broj4.", ".$broj5.", ".$broj6.", ".$broj7;
        echo ") nije pronađena u bazi podataka. NIJE IZVUČENA!";
        echo '</b><br/>';
    } /*od postoji vec*/
}
else /*dobrouneta*/
{
    echo '<b>';
    echo "Greška: Kombinacija nije dobro uneta. ";
    echo '</b><br/><br/>';
    echo "Svaki naredni broj mora biti veći od prethodnog! ";
    echo '<br/>';
    echo "Svi brojevi moraju biti u opsegu od 1 do 39! ";
    echo '<br/><br/>';
    echo "Proverite unete podatke na prethodnoj strani...";
    echo '<br/>';
} /*dobrouneta*/
?>

```

Forma za generisanje slučajne kombinacije – na ovoj stranici se prvo kreira objekat tipa klase kombinacija i odmah poziva metoda za generisanje slučajne kombinacije metodom slučajnog izbora od 1 do 39, a zatim se prihvata kombinacija u strukturi niza od 7 brojeva, čiji se elementi prikazuju u dvokolonskoj Bootstrap formi u kontroli Input type tipa tekst sa uključenim atributom value čiji je sadržaj određen elementom niza, a vrednost se prikazuje PHP funkcijom echo.

<h5>GENERISANJE SLUČAJNE KOMBINACIJE</h5>

<!-- Generisanje slučajne kombinacije -->

<p>Napomena: Ukoliko se slučajno dogodi da program generiše dva ista broja, možete im promeniti vrednosti ili generisati novu kombinaciju!</p>

```
<?php
/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();
/*generisanje slučajne kombinacije kao niza od 7 brojeva i
sortiranje elemenata u rastućem poretku*/
$slucajnakomb = $objKombinacija->generisiSlucajnuKombinaciju();
?>
```

<!-- Forma sa dve kolone -->

```
<form class="form-horizontal" role="form" method='POST' action='proveraizlaznostislucajne.php'>
<div class="form-group row">
</div>
<div class="form-group row">
<div class="col-sm-6">
<label for="Broj1">Prvi broj:</label>
<input type="text" class="form-
control" id="Broj1" name="Broj1" value="<?php echo $slucajnakomb[0];?>" tabindex=1 required>
</div>
<div class="col-sm-6">
<label for="Broj2">Drugi broj:</label>
<input type="text" class="form-
control" id="Broj2" name="Broj2" value="<?php echo $slucajnakomb[1];?>" tabindex=2 required>
</div>
</div>
<div class="form-group row">
<div class="col-sm-6">
<label for="Broj3">Treći broj:</label>
<input type="text" class="form-
control" id="Broj3" name="Broj3" value="<?php echo $slucajnakomb[2];?>" tabindex=3 required>
</div>
<div class="col-sm-6">
<label for="Broj4">Četvrti broj:</label>
<input type="text" class="form-
control" id="Broj4" name="Broj4" value="<?php echo $slucajnakomb[3];?>" tabindex=4 required>
</div>
</div>
<div class="form-group row">
<div class="col-sm-6">
<label for="Broj5">Peti broj:</label>
<input type="text" class="form-
control" id="Broj5" name="Broj5" value="<?php echo $slucajnakomb[4];?>" tabindex=5 required>
</div>
<div class="col-sm-6">
<label for="Broj6">Šesti broj:</label>
<input type="text" class="form-
control" id="Broj6" name="Broj6" value="<?php echo $slucajnakomb[5];?>" tabindex=6 required>
</div>
</div>
<div class="form-group row">
<div class="col-sm-6">
<label for="Broj7">Sedmi broj:</label>
<input type="text" class="form-
control" id="Broj7" name="Broj7" value="<?php echo $slucajnakomb[6];?>" tabindex=7 required>
</div>
</div>
<div class="form-group row">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" class="btn btn-default" tabindex=8 autofocus>Proveri izlaznost</button>
</div>
```

```
</div>
</form>
```

Forma za pregled i pretragu kombinacija - Za pretragu kombinacija se koristi html forma sa poljima tipa input type tekst za kriterijum pretrage (Kolo od i do, Godina od i do izlaska kombinacije memorisane u bazi podataka). Korišten je POST metod sa akcijom tipa self post, što znači da stranica ne predaje podatke iz forme drugoj, već istoj stranici. Promenljive u tom slučaju prihvataju vrednosti iz POST objekta, kreira se objekat tipa klase kombinacija i poziva metoda za pretragu izvučenih kombinacija koja kao ulazne parametre sadrži vrednosti promenljivih. Metoda vraća result set, koji ukoliko nije prazan, se prikazuje o formi html tabele odgovarajuće bootstrap klase.

```
<h5>PRETRAGA KOMBINACIJA</h5>
```

```
<!-- Forma sa dve kolone -->
<form class="form-horizontal" role="form" method='POST' action=''>
  <div class="form-group row">
  </div>
  <div class="form-group row">
    <div class="col-sm-6">
      <label for="Godinaod">Godina od</label>
      <input type="number" class="form-control" id="Godinaod" name="Godinaod" autofocus tabindex=1>
    </div>
    <div class="col-sm-6">
      <label for="Godinado">do</label>
      <input type="number" class="form-control" id="Godinado" name="Godinado" tabindex=2>
    </div>
  </div>
  <div class="form-group row">
    <div class="col-sm-6">
      <label for="Kolood">Redni broj kola od</label>
      <input type="number" class="form-control" id="Kolood" name="Kolood" tabindex=3>
    </div>
    <div class="col-sm-6">
      <label for="Kolodo">do</label>
      <input type="number" class="form-control" id="Kolodo" name="Kolodo" tabindex=4>
    </div>
  </div>
  <div class="form-group row">
    <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-default" name="pronadji" tabindex=5>Pronađi kombinacije</button>
      <button type="reset" class="btn btn-default" name="ponisti" tabindex=6>Nov kriterijum pretrage</button>
    </div>
  </div>
</form>
</br>

<!-- Pronalaženje kombinacija za kriterijum pretrage -->
<?php
if(isset($_POST['pronadji']))
{
  //ako je pritisnuta pretraga, tj. taster Pronađi
  $Godinaod=$_POST['Godinaod'];
  $Godinado=$_POST['Godinado'];
  $Kolood=$_POST['Kolood'];
  $Kolodo=$_POST['Kolodo'];

  /*kreiranje objekta tipa klase clskombinacija*/
  require 'class/clskombinacija.php';
  $objKombinacija = new clskombinacija();
  /*pretraga kombinacija*/
  $pronadjena="";
  $brojredova=0;
  $pronadjena=$objKombinacija->pretragaIzvucenihKombinacija($Godinaod,$Kolood,$Godinado,$Kolodo);
  $brojredova=mysqli_num_rows($pronadjena);
  if ($brojredova>0)
  {
```

```

$red=0;
echo "Pronađeno kombinacija: ".$brojredova;
echo "<table class='table table-striped'>
  <thead>
    <tr>
      <th scope='col'>Godina</th>
      <th scope='col'>Kolo</th>
      <th scope='col'>1.</th>
      <th scope='col'>2.</th>
      <th scope='col'>3.</th>
      <th scope='col'>4.</th>
      <th scope='col'>5.</th>
      <th scope='col'>6.</th>
      <th scope='col'>7.</th>
      <th scope='col'>Akcija 1</th>
      <th scope='col'>Akcija 2</th>
    </tr>
  </thead><tbody>";
while($red = mysqli_fetch_array($pronadjena))
{
  echo "
    <tr>
      <th scope='row'>".$red['Godina'].</th>
      <td>".$red['Kolo'].</td>
      <td>".$red['Broj1'].</td>
      <td>".$red['Broj2'].</td>
      <td>".$red['Broj3'].</td>
      <td>".$red['Broj4'].</td>
      <td>".$red['Broj5'].</td>
      <td>".$red['Broj6'].</td>
      <td>".$red['Broj7'].</td>";
      $rb=$red['RBkombinacija'];
      echo "<td> <form action='izmenakombinacije.php' method='POST'>";
      echo "<input type='hidden' name='rb' value='$rb'>";
      echo "<input type='submit' name='izmena' value='Izmena'>";
      echo "</form> </td>";
      echo "<td> <form action='brisanjekombinacije.php' method='POST'>";
      echo "<input type='hidden' name='rb' value='$rb'>";
      echo "<input type='submit' name='izmena' value='Brisanje'>";
      echo "</form> </td> </tr>";
    }
  echo "</tbody>
</table>";
}
else
{
  echo "U bazi podataka nije pronađena nijedna kombinacija!";
  echo "<br><br>";
}
} //kraj php-a ako je pritisnuta pretraga
?>

```

Stranica sa formom za izmenu kombinacija – Nakon izvršene pretrage i prikaza tabele sa kombinacijama, formiraju se forme sa tasterima za izmenu i brisanje kombinacije sa skrivenim poljem koje sadrži rb kombinacije kao primarnim ključem sa jedinstvenom vrednošću, što će omogućiti izmenu i brisanje podataka u bazi. Kod izmene se poziva stranica sa formom za unos kombinacije u koju su upisane vrednosti podataka pročitanih iz baze podataka. Kreira se objekat tipa klase kombinacija, poziva metoda za učitavanje kombinacije reko rb kombinacije, odredi se broj redova u result setu, preko fetch_array funkcije vrši pristup kolonama preko imena polja i čija se vrednost dodeli nekoj promenljivoj čije se vrednosti zatim prikazuju u dvokolonskoj Bootstrap formi, u kontroli Input type tipa tekst, sa uključenim atributom value čiji je sadržaj određen elementom niza, a vrednost se prikazuje PHP funkcijom echo. Na kraju se, opet POST metodom, promenjeni podaci šalju stranici za snimanje promena.

<h5>IZMENA KOMBINACIJE U BAZI PODATAKA</h5>

<!-- Učitavanje brojeva u kombinaciji prema rbkombinacije preuzetog sa stranice za pretragu -->


```

<?php
$ucitana=false;
$brojredova=0;
$Kolo=0;
$Godina=0;
$Broj1=0;
$Broj2=0;
$Broj3=0;
$Broj4=0;
$Broj5=0;
$Broj6=0;
$Broj7=0;
$rb=$_POST['rb'];

/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();
/*učitavanje kombinacije po rbkombinacije*/
$ucitana=$objKombinacija->ucitajKombinaciju($rb);
$brojredova=mysqli_num_rows($ucitana);
if ($brojredova>0)
{
    $red=0;
    while($red = mysqli_fetch_array($ucitana))
    {
        $Kolo=$red['Kolo'];
        $Godina=$red['Godina'];
        $Broj1=$red['Broj1'];
        $Broj2=$red['Broj2'];
        $Broj3=$red['Broj3'];
        $Broj4=$red['Broj4'];
        $Broj5=$red['Broj5'];
        $Broj6=$red['Broj6'];
        $Broj7=$red['Broj7'];
    }
}
?>

```

```

<!-- Forma sa dve kolone uz dodeljenje vrednosti poljima iz učitane kombinacije -->
<form class="form-horizontal" role="form" method='POST' action='snimanjeizmenekombinacije.php'>
    <div class="form-group row">
        <div class="col-sm-6">
            <label for="Godina">Godina:</label>
            <input type="text" class="form-control" id="Godina" name="Godina" required value="<?php echo $Godina; ?>">
        </div>
        <div class="col-sm-6">
            <label for="Kolo">Kolo:</label>
            <input type="text" class="form-control" id="Kolo" name="Kolo" required value="<?php echo $Kolo; ?>">
        </div>
    </div>
    <div class="form-group row">
        <div class="col-sm-6">
            <label for="Broj1">Prvi broj:</label>
            <input type="text" class="form-control" id="Broj1" name="Broj1" required value="<?php echo $Broj1; ?>">
        </div>
        <div class="col-sm-6">
            <label for="Broj2">Drugi broj:</label>
            <input type="text" class="form-control" id="Broj2" name="Broj2" required value="<?php echo $Broj2; ?>">
        </div>
    </div>
    <div class="form-group row">
        <div class="col-sm-6">

```

```

        <label for="Broj3">Treći broj:</label>
        <input type="text" class="form-control" id="Broj3" name="Broj3" required value="<?php echo $Broj3; ?>">
    </div>
    <div class="col-sm-6">
        <label for="Broj4">Četvrti broj:</label>
        <input type="text" class="form-control" id="Broj4" name="Broj4" required value="<?php echo $Broj4; ?>">
    </div>
</div>
<div class="form-group row">
    <div class="col-sm-6">
        <label for="Broj5">Peti broj:</label>
        <input type="text" class="form-control" id="Broj5" name="Broj5" required value="<?php echo $Broj5; ?>">
    </div>
    <div class="col-sm-6">
        <label for="Broj6">Šesti broj:</label>
        <input type="text" class="form-control" id="Broj6" name="Broj6" required value="<?php echo $Broj6; ?>">
    </div>
</div>
<div class="form-group row">
    <div class="col-sm-6">
        <label for="Broj7">Sedmi broj:</label>
        <input type="text" class="form-control" id="Broj7" name="Broj7" required value="<?php echo $Broj7; ?>">
    </div>
</div>
<div class="form-group row">
<div class="col-sm-offset-2 col-sm-10">
    <input type="hidden" name='rb' value='<?php echo $rb; ?>'>
    <button type="submit" class="btn btn-default">Promeni kombinaciju</button>
</div>
</div>
</form>

```

Snimanje izmena – Funkcioniše na gotovo identičan način kao i stranica za snimanje nove izvučene kombinacije. Prvo se u promenljive pruzimaju vrednosti iz POST objekta, kreira se objekat tipa klase kombinacija, poziva metoda za proveru da li su brojevi uneti u skladu sa zahtevima korisnika, te ako je sve u redu, vrši se menjanje podataka u relacionoj bazi (SQL upitom UPDATE tipa) odgovarajućom metodom za izmenu, a pre poziva metode se dodeljuju vrednosti atributima objekta klase na osnovu vrednosti promenljivih. Takođe se na ovoj stranici prikazuju i odgovarajuće poruke u zavisnosti od toga koji deo PHP koda je izvršen.

<h5>SNIMANJE IZMENJENE KOMBINACIJE U BAZU PODATAKA</h5>

```

<!-- Provera da li kombinacija dobro uneta? -->
<!-- Kreiranje objekta kombinacija -->
<?php
$dobrouneta=false;
$rb=$_POST['rb'];
$godina=$_POST['Godina'];
$kolo=$_POST['Kolo'];
$broj1=$_POST['Broj1'];
$broj2=$_POST['Broj2'];
$broj3=$_POST['Broj3'];
$broj4=$_POST['Broj4'];
$broj5=$_POST['Broj5'];
$broj6=$_POST['Broj6'];
$broj7=$_POST['Broj7'];

/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();
/*provera da li su brojevi dobro uneti*/
$dobrouneta = $objKombinacija->proveriDaLiSuBrojeviIspravnoUneti($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7,$godina);

```

```

if ($dobrouneta)
{
    echo "Kombinacija je dobro uneta.";
    /*provera da li vec postoji to kolo i godina*/
    $postojivec = false;
    //$postojivec = $objKombinacija->proveriDaLiKombinacijaPostoji($godina,$kolo);
    if ($postojivec)
    {
        echo '<b><br/><br/>';
        echo "Greška: Kombinacija sa unetom godinom i brojem kola već postoji u bazi podataka. ";
        echo '</b><br/><br/>';
        echo "Proverite unete podatke na prethodnoj stranici ili odustanite od unosa! ";
        echo '<br/>';
    }
    else /*postoji?*/
    {
        /*dodela vrednosti atributima objekta*/
        $objKombinacija->godina = $godina;
        $objKombinacija->kolo = $kolo;
        $objKombinacija->broj1 = $broj1;
        $objKombinacija->broj2 = $broj2;
        $objKombinacija->broj3 = $broj3;
        $objKombinacija->broj4 = $broj4;
        $objKombinacija->broj5 = $broj5;
        $objKombinacija->broj6 = $broj6;
        $objKombinacija->broj7 = $broj7;

        /*izmena objekta u bazi metodom promeniKombinaciju*/
        $result = $objKombinacija->promeniKombinaciju($rb);
        if ($result)
        {
            echo '<br/><br/><b>';
            echo "Kombinacija je uspešno promenjena u bazi podataka!";
            echo '</b><br/>';
        }
        else
        {
            echo '<br/>';
            echo "Kombinacija nije promenjena u bazi podataka!";
            echo '<br/>';
        }
    }
}
else /*$dobrouneta*/
{
    echo '<b>';
    echo "Greška: Kombinacija nije dobro uneta. ";
    echo '</b><br/><br/>';
    echo "Svaki naredni broj mora biti veći od prethodnog! ";
    echo '<br/>';
    echo "Svi brojevi moraju biti u opsegu od 1 do 39! ";
    echo '<br/><br/>';
    echo "Proverite unete podatke na prethodnoj strani... ";
    echo '<br/>';
} /*$dobrouneta*/
?>

```

Stranica za brisanje kombinacije iz baze podataka – Pokreće se samo sa stranice za pretragu kombinacija. Prvo se u promenljivu smešta vrednost rb kombinacije potreban za identifikaciju kombinacije koja će biti obrisana SQL upitom tipa DELETE. Prvo se u promenljivu pružima vrednost iz POST objekta, kreira se objekat tipa klase kombinacija, poziva metoda za brisanje kombinacije, te ukoliko je brisanje uspešno ili neuspešno izvršeno prikazuju se odgovarajuće poruke.

```
<h5>BRISANJE KOMBINACIJE IZ BAZE PODATAKA</h5>
```

```
<!-- Kreiranje objekta kombinacija -->
<?php
$uspesnoobrisana=false;
$rb=$_POST['rb'];

/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();
/*pozvi metode za brisanje*/
$uspesnoobrisana = $objKombinacija->obrisiKombinaciju($rb);
if ($uspesnoobrisana)
{
    echo '<br/><b>';
    echo "Kombinacija je uspešno obrisana iz baze podataka!";
    echo '</b><br/>';
}
else
{
    echo "Kombinacija nije obrisana iz baze podataka!";
}
?>
```

Izveštaj – je realizovan “printer-frindly” stranicom sa belom pozadinom koja se referencira na poseban css za izveštaj. Prvo se prihvataju vrednosti promenljivih dobijenih POST metodom sa stranice za pretragu, kreira se objekat tipa klase kombinacija i poziva metoda za pretragu izvučenih kombinacija koja kao ulazne parametre sadrži vrednosti promenljivih. Metoda vraća result set, koji ukoliko nije prazan, prikazuje se u formi html tabele odgovarajuće bootstrap klase.

```
<html>
<head>
<meta charset='utf-8'>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="reportstyles.css">
<script src="script.js"></script>
<title>Loto 2020</title>
</head>
<body>
<div style="font-family:arial; color:black; padding-left: 20px;">
<h3>LOTO 2020</h3>
<h4>Izvučene kombinacije - <?php echo date('d.m.Y')." ";?></h4><hr>
<?php
$Godinaod=$_POST['Godinaod'];
$Godinado=$_POST['Godinado'];
$Kolood=$_POST['Kolood'];
$Kolodo=$_POST['Kolodo'];
$result="";
$brredova=0;

/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();
/*pretraga kombinacija*/
$pronadjena="";
$brojredova=0;
```

```

$pronadjena=$objKombinacija->pretragalzvucenihKombinacija($Godinaod,$Kolood,$Godinado,$Kolodo);
$brojredova=mysqli_num_rows($pronadjena);
if ($brojredova>0)
{
$red=0;

echo "<table class='table table-striped'>
    <thead>
        <tr>
            <th scope='col'>Godina</th>
            <th scope='col'>Kolo</th>
            <th scope='col'>1. broj</th>
            <th scope='col'>2. broj</th>
            <th scope='col'>3. broj</th>
            <th scope='col'>4. broj</th>
            <th scope='col'>5. broj</th>
            <th scope='col'>6. broj</th>
            <th scope='col'>7. broj</th>
            <th scope='col'>Parnih</th>
            <th scope='col'>Neparnih</th>
            <th scope='col'>V1</th>
            <th scope='col'>V2</th>
            <th scope='col'>V3</th>

        </tr>
    </thead><tbody>";
while($red = mysqli_fetch_array($pronadjena))
{
    echo "
        <tr>
            <th scope='row'>". $red['Godina']. "</th>
            <td>". $red['Kolo']. "</td>
            <td>". $red['Broj1']. "</td>
            <td>". $red['Broj2']. "</td>
            <td>". $red['Broj3']. "</td>
            <td>". $red['Broj4']. "</td>
            <td>". $red['Broj5']. "</td>
            <td>". $red['Broj6']. "</td>
            <td>". $red['Broj7']. "</td>
            <td>". $red['Par']. "</td>
            <td>". $red['Nepar']. "</td>
            <td>". $red['V1']. "</td>
            <td>". $red['V2']. "</td>
            <td>". $red['V3']. "</td>
        </tr>";
    }
    echo "</tbody>
    </table><br>";
}
else
{
    echo "U bazi podataka nije pronađena nijedna kombinacija!";
    echo "<br><br>";
}
echo "<hr>";
echo "Ukupno prikazano kombinacija: ". $brojredova;
//kraj php-a za izvestaj
$objKombinacija=null;
?>
</div>
</body>
</html>

```

Statistika - Statistika se ogleda u prebrojavanju izlaznosti svakog od 39 brojeva igre. Kreira se objekat tipa klase kombinacija i poziva 39 puta metoda za prebrojavanje pojedinačnog broja čija se vrednost menja for PHP ciklusom od 1 do 39. Metoda vraća result set, čiji se sadržaj (vrednost prebrojavanja izlaznosti SQL upitom preko count funkcije) prikazuje kao red html tabele odgovarajuće bootstrap klase.

```
<h5>STATISTIKA</h5>

<!-- Prebrojavanje izlaznosti svih 39 brojeva -->
<?php

/*kreiranje objekta tipa klase clskombinacija*/
require 'class/clskombinacija.php';
$objKombinacija = new clskombinacija();

/*kreiranje tabele sa dve kolone, prva prikazuje broj 1-39,
a druga rezultat izvršavanja metode za prebrojavanje izalznosti*/
echo "<table class='table table-striped'>
    <thead>
        <tr>
            <th scope='col'>Broj</th>
            <th scope='col'>Ukupan broj izlazaka (koliko puta je broj izvučen)</th>
        </tr>
    </thead><tbody>";

for ($i = 1; $i <= 39; $i++)
{
    $rezultat="";
    $rezultat=$objKombinacija->prebrojIzlaznostBroja($i);
    $red=0;
    $red = mysqli_fetch_array($rezultat);
    echo "<tr>
        <td>". $i. "</td>
        <td>". $red['IzasaoPuti']. "</td>
    </tr>";
}
echo "</tbody>
</table>";
?>
```

Klasa za konekciju biblioteke klasa sa bazom podataka:

```
<?php

class clskonekcijadb
{
    private $server = "localhost";
    private $username = "root";
    private $serverpassword = "";
    private $database = "loto2020rsok";
    public $konekcija;

    public function otvoriKonekciju()
    {
        $this->konekcija = "";
        $this->konekcija = mysqli_connect($this->server, $this->username, $this->serverpassword, $this->database);
        if (!$this->konekcija)
        {
            echo("Nije uspostavljena veza sa serverom baze podataka!");
            echo "<br/>";
        }
        return $this->konekcija;
    }
}
```

```

    }

    public function zatvoriKonekciju($pkonekcija)
    {
        mysqli_close($pkonekcija);
    }

} //kraj klase
?>

```

Klasa kombinacija - prikaz metoda: za unos kombinacije, provere da li je ista dobro uneta i da li već postoji u bazi podataka kombinacija sa istim brojem kola i godinom, generisanje slučajne kombinacije, određivanje RB kombinacije, brisanje, izmenu i prikaz kombinacije, kao i prebrojavanje izlaznosti broja pozivom metode klase statistika.

```
<?php
```

```

class clskombinacija
{
    public $idkombinacije;
    public $godina;
    public $kolo;
    public $broj1;
    public $broj2;
    public $broj3;
    public $broj4;
    public $broj5;
    public $broj6;
    public $broj7;
    private $konekcija;

    function __construct()
    {
        include_once "clskonekcijadb.php";
        $objkonbp = new clskonekcijadb();
        $this->konekcija = $objkonbp->otvoriKonekciju();
    } //kraj konstruktora

    public function proveriDaLiSuBrojeviIsppravnoUneti($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7,$godina)
    {
        $suspeh=false;
        if (($godina>1984) && ($broj1>0 && $broj1<40) && ($broj2>0 && $broj2<40) && ($broj3>0 && $broj3<40) && ($broj4>0 && $broj4
<40) && ($broj5>0 && $broj5<40) && ($broj6>0 && $broj6<40) && ($broj7>0 && $broj7<40))
        {
            if ($broj1<$broj2 && $broj2<$broj3 && $broj3<$broj4 && $broj4<$broj5 && $broj5<$broj6 && $broj6<$broj7)
            {
                $suspeh=true;
            }
        }
        return $suspeh;
    } //metoda proverilIsppravnoUnetuKombinaciju

    public function proveriDaLiKombinacijaPostoji($godina,$kolo)
    {
        $suspeh=false;
        $brredova=0;
        $upit = "SELECT * FROM `izvucena kombinacija` WHERE godina=$godina and kolo=$kolo;";
        //echo $upit;
        $result = mysqli_query($this->konekcija, $upit);
        $brredova = mysqli_num_rows($result);
        if ($brredova>0)
        {
            $suspeh=true;
        }
    }
}

```

```

    }
    return $suspeh;
} //metoda daLiKombinacijaPostoji

public function snimiKombinaciju()
{
    $result = "";
    $sqlupit = "INSERT INTO `izvucena kombinacija` VALUES (',$this->godina,$this->kolo,$this->broj1,$this->broj2,$this->broj3,$this->broj4,$this->broj5,$this->broj6,$this->broj7);";
    $result = mysqli_query($this->konekcija, $sqlupit);
    return $result;
} //metoda snimiKombinaciju

public function odrediRB()
{
    $result = "";
    $sqlupit = "SELECT max(RBkombinacija) as novirb from `Izvucena kombinacija`;";
    $result = mysqli_query($this->konekcija, $sqlupit);
    return $result;
} //metoda snimiKombinaciju

public function proverilaznostKombinacije($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7)
{
    $suspeh=false;
    $brredova=0;
    $upit = "SELECT * FROM `izvucena kombinacija` WHERE broj1=$broj1 and broj2=$broj2 and broj3=$broj3 and broj4=$broj4 and broj5=$broj5 and broj6=$broj6 and broj7=$broj7;";
    $result = mysqli_query($this->konekcija, $upit);
    $brredova = mysqli_num_rows($result);
    if ($brredova>0)
    {
        $suspeh=true;
    }
    return $suspeh;
} //metoda proverilaznostKombinacije

public function generisiSlucajnuKombinaciju()
{
    $niz[0]=rand(1,39);
    $niz[1]=rand(1,39);
    if ($niz[1]==$niz[0])
        {$niz[1]=rand(1,39);}
    $niz[2]=rand(1,39);
    if ($niz[2]==$niz[0] || $niz[2]==$niz[1])
        {$niz[2]=rand(1,39);}
    $niz[3]=rand(1,39);
    if ($niz[3]==$niz[2] || $niz[3]==$niz[2] || $niz[3]==$niz[0])
        {$niz[3]=rand(1,39);}
    $niz[4]=rand(1,39);
    if ($niz[4]==$niz[3] || $niz[4]==$niz[2] || $niz[4]==$niz[2] || $niz[4]==$niz[0])
        {$niz[4]=rand(1,39);}
    $niz[5]=rand(1,39);
    if ($niz[5]==$niz[4] || $niz[5]==$niz[3] || $niz[5]==$niz[2] || $niz[5]==$niz[2] || $niz[5]==$niz[0])
        {$niz[5]=rand(1,39);}
    $niz[6]=rand(1,39);
    if ($niz[6]==$niz[5] || $niz[6]==$niz[4] || $niz[6]==$niz[3] || $niz[6]==$niz[2] || $niz[6]==$niz[2] || $niz[6]==$niz[0])
        {$niz[6]=rand(1,39);}
    sort($niz);
    return $niz;
} //metoda generisiSlucajnuKombinaciju

public function pretragalzvucenihKombinacija($godinaod,$kolood,$godinado,$kolodo)

```



```

{
    $result = "";
    $sqlupit = "SELECT * FROM `izvucena kombinacija`, Statistika WHERE `izvucena kombinacija`.RBkombinacija=Statistika.RBkombinacija";
    ";
    if (($godinaod=="") && ($godinado=="") && ($kolood=="") && ($kolodo==""))
        {$sqlupit = "SELECT * FROM `izvucena kombinacija`, Statistika WHERE `izvucena kombinacija`.RBkombinacija=Statistika.RBkombinacija ORDER BY Godina, Kolo;";}
    if (($godinaod!="") && ($godinado!="") && ($kolood=="") && ($kolodo==""))
        {$sqlupit = "SELECT * FROM `izvucena kombinacija`, Statistika WHERE `izvucena kombinacija`.RBkombinacija=Statistika.RBkombinacija AND Godina>=$godinaod AND Godina<=$godinado ORDER BY Godina, Kolo;";}
    if (($godinaod!="") && ($godinado=="") && ($kolood=="") && ($kolodo==""))
        {$sqlupit = "SELECT * FROM `izvucena kombinacija`, Statistika WHERE `izvucena kombinacija`.RBkombinacija=Statistika.RBkombinacija AND Godina>=$godinaod ORDER BY Godina, Kolo;";}
    if (($godinaod=="") && ($godinado!="") && ($kolood=="") && ($kolodo==""))
        {$sqlupit = "SELECT * FROM `izvucena kombinacija`, Statistika WHERE `izvucena kombinacija`.RBkombinacija=Statistika.RBkombinacija AND Godina<=$godinado ORDER BY Godina, Kolo;";}
    if (($godinaod!="") && ($godinado!="") && ($kolood!="") && ($kolodo!=""))
        {$sqlupit = "SELECT * FROM `izvucena kombinacija`, Statistika WHERE `izvucena kombinacija`.RBkombinacija=Statistika.RBkombinacija AND Godina>=$godinaod AND Godina<=$godinado AND Kolo>=$kolood AND Kolo<=$kolodo ORDER BY Godina, Kolo;";}
    if (($godinaod=="") && ($godinado=="") && ($kolood!="") && ($kolodo!=""))
        {$sqlupit = "SELECT * FROM `izvucena kombinacija`, Statistika WHERE `izvucena kombinacija`.RBkombinacija=Statistika.RBkombinacija AND Kolo>=$kolood AND Kolo<=$kolodo ORDER BY Godina, Kolo;";}
    $result = mysqli_query($this->konekcija, $sqlupit);
    //echo $sqlupit;
    return $result;
} //metoda pretragalizvucenihKombinacija

public function obrisiKombinaciju($rbkomb)
{
    $result = false;
    $sqlupit = "DELETE FROM `izvucena kombinacija` WHERE RBkombinacija=$rbkomb;";
    $result = mysqli_query($this->konekcija, $sqlupit);
    return $result;
} //metoda obrisiKombinaciju

public function promeniKombinaciju($rb)
{
    $result = "";
    $sqlupit = "UPDATE `izvucena kombinacija` SET Godina='$this->godina', Kolo='$this->kolo', Broj1='$this->broj1', Broj2='$this->broj2', Broj3='$this->broj3', Broj4='$this->broj4', Broj5='$this->broj5', Broj6='$this->broj6', Broj7='$this->broj7' WHERE RBkombinacija='$rb';";
    $result = mysqli_query($this->konekcija, $sqlupit);
    return $result;
} //metoda promeniKombinaciju

public function ucitajKombinaciju($rbkomb)
{
    $result = "";
    $sqlupit = "SELECT * FROM `izvucena kombinacija` WHERE RBkombinacija='$rbkomb';";
    $result = mysqli_query($this->konekcija, $sqlupit);
    return $result;
} //metoda ucitajKombinaciju

public function prebrojIzlaznostBroja($broj)
{
    $result = "";
    $sqlupit = "SELECT Count(*) as IzasaoPuti FROM `izvucena kombinacija` WHERE Broj1=$broj Or Broj2=$broj Or Broj3=$broj Or Broj4=$broj Or Broj5=$broj Or Broj6=$broj Or Broj7=$broj;";
    $result = mysqli_query($this->konekcija, $sqlupit);
    return $result;
} //metoda prebrojIzlaznostBroja

```

```

function __destruct()
{
    /*$this->konekcija = null;*/
    unset($this->konekcija);
} //kraj destruktora

} //kraj klase
?>

```

Klasa clsstatistika – metoda za snimanje statistike.

```
<?php
```

```

class clsstatistika
{
    public $rbkombinacije;
    public $p;
    public $n;
    public $v1;
    public $v2;
    public $v3;
    private $konekcijas;

    function __construct()
    {
        include_once "clskonekcijadb.php";
        $objkonbps = new clskonekcijadb();
        $this->konekcijas = $objkonbps->otvoriKonekciju();
    } //kraj konstruktora

    public function snimiStatistiku($broj1,$broj2,$broj3,$broj4,$broj5,$broj6,$broj7,$rbkomb)
    {
        $v1=0;
        $v2=0;
        $v3=0;
        $p=0;
        $n=0;
        $y = 2;
        //rezultat ostatka pri deljenju brojem 2 ako je 1 znači da je broj neparan:
        $rdb1 = fmod($broj1,$y);
        $rdb2 = fmod($broj2,$y);
        $rdb3 = fmod($broj3,$y);
        $rdb4 = fmod($broj4,$y);
        $rdb5 = fmod($broj5,$y);
        $rdb6 = fmod($broj6,$y);
        $rdb7 = fmod($broj7,$y);
        //koliko ima neparnih:
        $n=$rdb1+$rdb2+$rdb3+$rdb4+$rdb5+$rdb6+$rdb7;
        //parnih=7-neparnih:
        $p=7-$n;
        //prebrojavanje vertikal v1
        if ($broj1==1 || $broj1==4 || $broj1==7 || $broj1==10 || $broj1==13 || $broj1==16 || $broj1==19 || $broj1==22 || $broj1==25 ||
$broj1==28 || $broj1==31 || $broj1==34 || $broj1==37)
        {
            $v1++;
        }
        if ($broj2==1 || $broj2==4 || $broj2==7 || $broj2==10 || $broj2==13 || $broj2==16 || $broj2==19 || $broj2==22 || $broj2==25 ||
$broj2==28 || $broj2==31 || $broj2==34 || $broj2==37)
        {
            $v1++;
        }
    }
}

```

```

    if ($broj3==1 || $broj3==4 || $broj3==7 || $broj3==10 || $broj3==13 || $broj3==16 || $broj3==19 || $broj3==22 || $broj3==25 ||
$broj3==28 || $broj3==31 || $broj3==34 || $broj3==37)
    {
        $v1++;
    }
    if ($broj4==1 || $broj4==4 || $broj4==7 || $broj4==10 || $broj4==13 || $broj4==16 || $broj4==19 || $broj4==22 || $broj4==25 ||
$broj4==28 || $broj4==31 || $broj4==34 || $broj4==37)
    {
        $v1++;
    }
    if ($broj5==1 || $broj5==4 || $broj5==7 || $broj5==10 || $broj5==13 || $broj5==16 || $broj5==19 || $broj5==22 || $broj5==25 ||
$broj5==28 || $broj5==31 || $broj5==34 || $broj5==37)
    {
        $v1++;
    }
    if ($broj6==1 || $broj6==4 || $broj6==7 || $broj6==10 || $broj6==13 || $broj6==16 || $broj6==19 || $broj6==22 || $broj6==25 ||
$broj6==28 || $broj6==31 || $broj6==34 || $broj6==37)
    {
        $v1++;
    }
    if ($broj7==1 || $broj7==4 || $broj7==7 || $broj7==10 || $broj7==13 || $broj7==16 || $broj7==19 || $broj7==22 || $broj7==25 ||
$broj7==28 || $broj7==31 || $broj7==34 || $broj7==37)
    {
        $v1++;
    }
    //koliko ima u v3, ostatak pri deljenju sa 3 je 0:
    $y = 3;
    $v1br1 = fmod($broj1,$y);
    $v1br2 = fmod($broj2,$y);
    $v1br3 = fmod($broj3,$y);
    $v1br4 = fmod($broj4,$y);
    $v1br5 = fmod($broj5,$y);
    $v1br6 = fmod($broj6,$y);
    $v1br7 = fmod($broj7,$y);
    if ($v1br1==0)
    {
        $v3++;
    }
    if ($v1br2==0)
    {
        $v3++;
    }
    if ($v1br3==0)
    {
        $v3++;
    }
    if ($v1br4==0)
    {
        $v3++;
    }
    if ($v1br5==0)
    {
        $v3++;
    }
    if ($v1br6==0)
    {
        $v3++;
    }
    if ($v1br7==0)
    {
        $v3++;
    }

```

```

//v2=7-v1-v3:
$v2=7-$v1-$v3;
//snimanje statistickih parametara kombinacije u bazu podataka
$results = "";
$sqlupits = "INSERT INTO `Statistika` VALUES ('$rbkomb','$p','$n','$v1','$v2','$v3')";
//echo $sqlupits;
$results = mysqli_query($this->konekcijas, $sqlupits);
return $results;
} //metoda snimiStatistiku


function __destruct()
{
    /*$this->konekcija = null;*/
    unset($this->konekcijas);
} //kraj destruktora

} //kraj klase
?>

```

Opis testiranja softvera

Testiranje softvera je izvršeno na dva načina: funkcionalno, prilikom povezivanja korisničkog interfejsa i srednjeg aplikacionog sloja i pojedinih metoda klasa srednjeg sloja metodom bele kutije tako što su kreirani slučajevi testiranja.

Testiranje izvršavanja metode klase clskombinacija:

proveriDaLiSuBrojeviIspravnoUneti(\$broj1,\$broj2,\$broj3,\$broj4,\$broj5,\$broj6,\$broj7,\$godina)

Slučajevi testiranja:

Broj slučaja	Opis slučaja	Test podaci	Očekivani rezultat	Trenutni rezultat	Status (uspeh/neuspeh)
#1.1	Provera da li je kombinacija ispravno uneta? Potrebno je zadovoljiti uslove: a) broj1 < broj2 < broj3 < broj4 < broj5 < broj6 < broj7 b) svi uneti brojevi treba da su u rasponu od 1 do 39 Sedmi broj je veći od 39.	39, 17, 20, 22, 29, 30, 44	Kombinacija nije dobro uneta.	Kombinacija nije dobro uneta.	uspeh
#1.2	Provera da li je kombinacija ispravno uneta? Potrebno je zadovoljiti uslove: a) broj1 < broj2 < broj3 < broj4 < broj5 < broj6 < broj7 b) svi uneti brojevi treba da su u rasponu od 1 do 39 Prvi broj je najveći, dok je sedmi broj najmanji.	39, 17, 20, 22, 29, 30, 14	Kombinacija nije dobro uneta.	Kombinacija nije dobro uneta.	uspeh
#1.3	Provera da li je kombinacija ispravno uneta? Potrebno je zadovoljiti uslove: a) broj1 < broj2 < broj3 < broj4 < broj5 < broj6 < broj7 b) svi uneti brojevi treba da su u rasponu od 1 do 39 Ispravno unetih sedam brojeva.	14, 17, 20, 22, 29, 30, 39	Kombinacija je dobro uneta.	Kombinacija je dobro uneta.	uspeh

Testiranje izvršavanja metode klase clskombinacija:
 proverilzlaznostKombinacije (\$broj1,\$broj2,\$broj3,\$broj4,\$broj5,\$broj6,\$broj7)

Slučajevi testiranja:

Broj slučaja	Opis slučaja	Test podaci	Očekivani rezultat	Trenutni rezultat	Status (uspeh/neuspeh)
#2.1	Provera da li je kombinacija već među izvučenim i memorisanim kombinacijama? Kombinacija je već memorisana u bazi.	14, 17, 20, 22, 29, 30, 39	Kombinacija je već izlazila	Kombinacija nije pronađena u bazi podataka.	neuspeh
#2.2	Provera da li je kombinacija već među izvučenim i memorisanim kombinacijama? Kombinacija nije memorisana u bazi.	14, 17, 20, 22, 29, 30, 38	Kombinacija nije izlazila	Kombinacija nije pronađena u bazi podataka.	uspeh

Na osnovu slučaja #2.1 analizom PHP koda metode proverilzlaznostKombinacije pronađena greška u SQL komandi za proveru da li unetih sedam brojeva već postoji u bazi podataka:

\$upit = "SELECT * FROM `izvucena kombinacija` WHERE broj1=\$broj1 and broj2=\$broj2 and broj3=\$broj3 and broj4=\$broj4 and broj5=\$broj5 and broj6=\$broj6 and broj7=\$broj6;";

Upit nakon korekcije:

\$upit = "SELECT * FROM `izvucena kombinacija` WHERE broj1=\$broj1 and broj2=\$broj2 and broj3=\$broj3 and broj4=\$broj4 and broj5=\$broj5 and broj6=\$broj6 and broj7=\$broj7;";

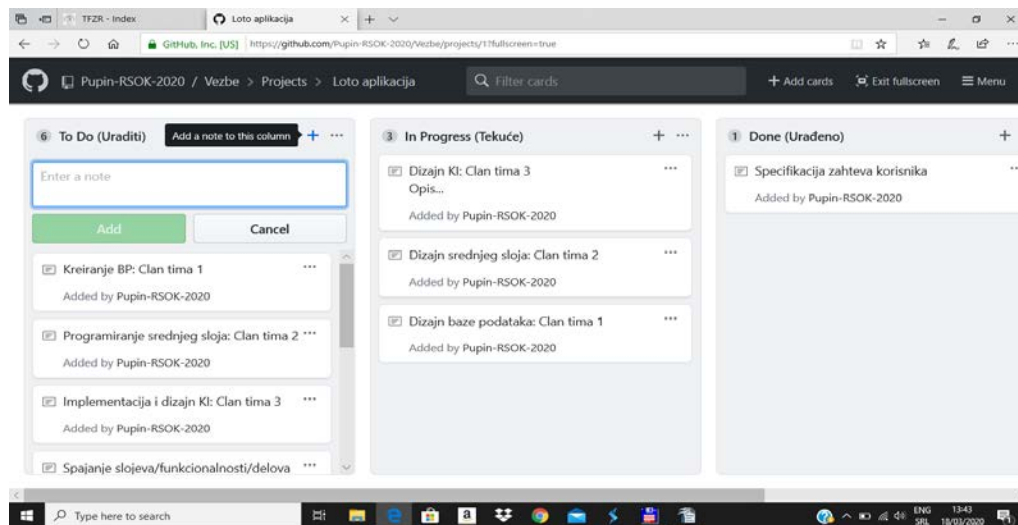
Broj slučaja	Opis slučaja	Test podaci	Očekivani rezultat	Trenutni rezultat	Status (uspeh/neuspeh)
#2.1	Provera da li je kombinacija već među izvučenim i memorisanim kombinacijama? Kombinacija je već memorisana u bazi.	14, 17, 20, 22, 29, 30, 39	Kombinacija je već izlazila	Kombinacija jeste pronađena u bazi podataka.	uspeh

Rad sa Git/Github sistemom

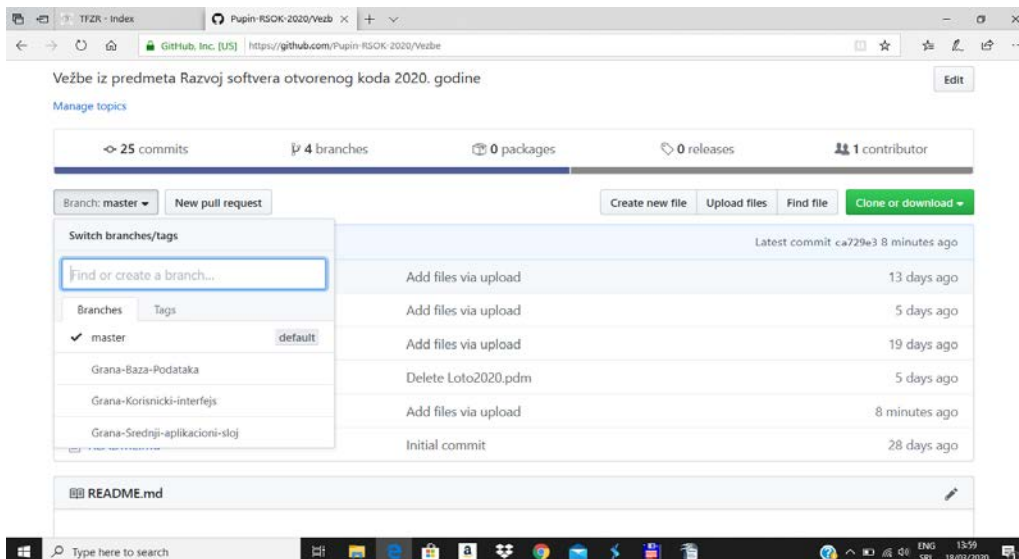
Kreirani repozitorijum – Github.com/Vezbe

Učesnici projekta – 1. Doc. Dr Kazi Zoltan (creator repozitorijuma)

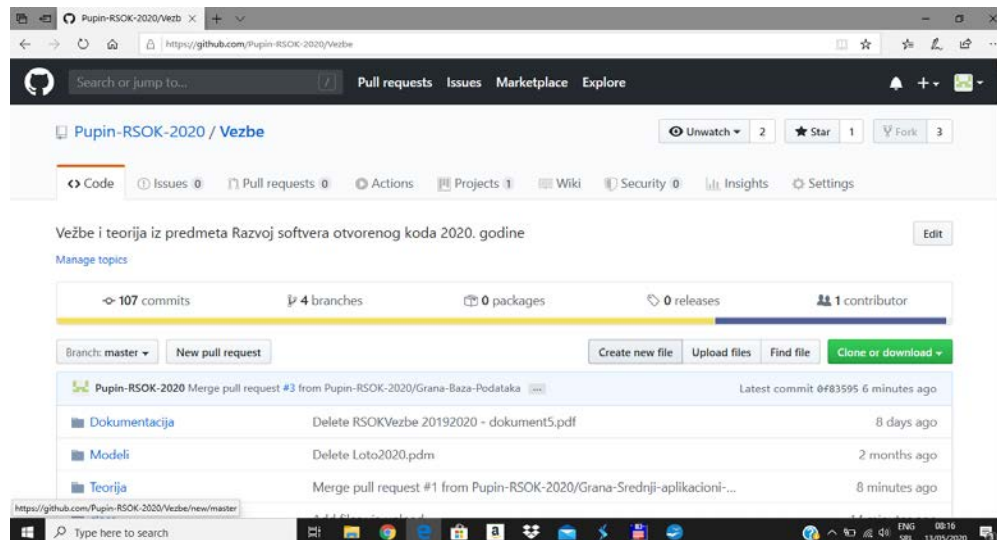
Plan projekta – Faze razvoja: 1. Specifikacija zahteva 2. Idejno rešenje 3. Dizajn aplikacije: Baze podataka, srednjeg aplikacionog sloja I baze podataka 4. Implementacija slojeva 5. Testiranje i integracija 6. Dokumentovanje, realizacija i licenca.



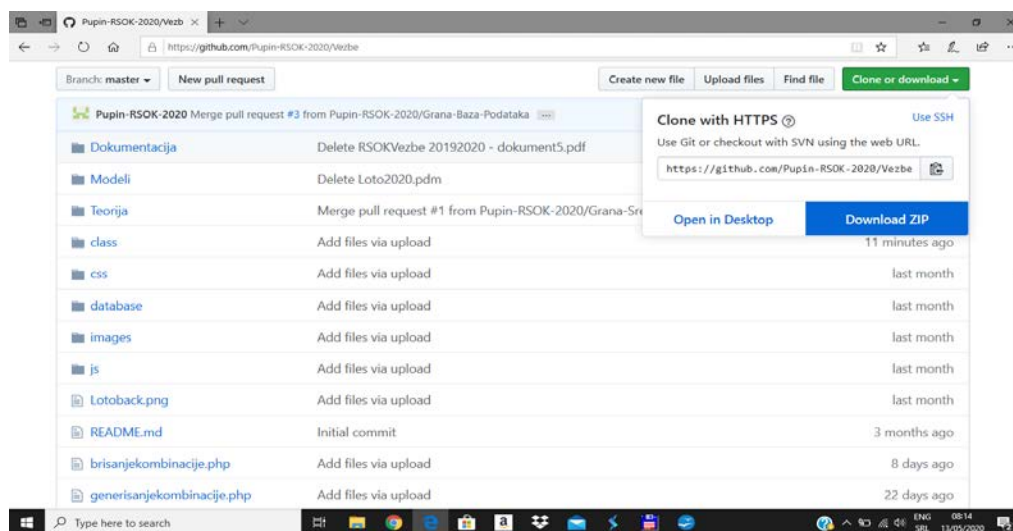
Kreriiane grane – 1. Korisnički interfejs, 2. Srednji aplikacioni sloj, 3. Baza podataka



Postavljanje datoteka - urađeno upload tehnikom za sve datoteke, na svim granama, osim licencnog fajla koji je kreiran opcijom "Create new file".



Preuzimanje datoteka - izvršeno opcijom "Download ZIP" čime se kompletna aplikacija preuzima kao zapakovana arhiva.



Spajanje grana – Izvršeno nakon završetka postavljanja ("Push") svih datoteka na granama opcijom "Pull".

Licenca – Postavljena GPL licenca na osnovu šablona koji se već nalaze na Github.com, kreirana datoteka LICENCE.TXT.

6. Korišteni alati i softveri

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

7. Literatura

- Državna lutrija Srbije, 2019., <https://www.lutrija.rs/LottoGame>
- Github repozitorijum, 2020., <https://github.com/>
- W. Jason Gilmore Beginning PHP and MySQL: From Novice to Professional, Fourth Edition Copyright © 2010 by W. Jason Gilmore Apress, USA.
- Fabio Cimo, Bootstrap Programming Cookbook, Web Code Geeks.