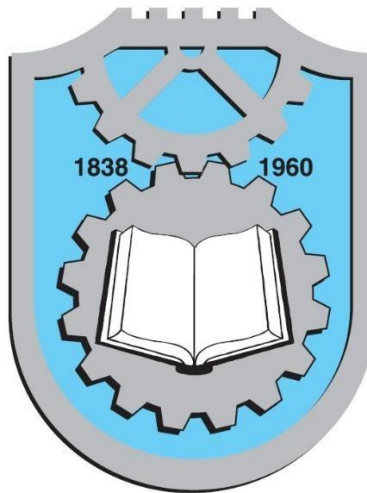


Univerzitet u Kragujevcu
Fakultet inženjerskih nauka



Seminarski rad iz Softverskog inženjeringa

Tema:
UML test

Predmetni profesor:
Nenad Filipović

Student:
Sofija Jakoljević 574/2015

Predmetni saradnik:
Tijana Šušteršić

Sadržaj

1. Postavka zadatka.....	3
2. Opis delova programa sa izvornim kodom.....	4
3. UML dijagrami	9
3.1 Dijagram slučajeve korišćenja (Use case diagrams)	9
3.2 Dijagram sekvenci (Sequence diagram).....	10
3.3 Dijagram aktivnosti (Activity diagram)	11
3.4 Dijagram stanja (State machine diagram).....	12
Literatura.....	13

1. Postavka zadatka

Napisati aplikaciju u kojoj korisnik rešava UML test. Korisnik najpre unosi lične podatke da bi mogao da pristupi testu, a zatim, pri tačnim podacima za prijavljivanje i pristupa istom. Neophodno je da korisnik odradi svih 10 pitanja, na osnovu čega dobija informaciju o osvojenim poenima kao i na koja pitanja nije tačno odgovorio, ukoliko ih ima. Svako pitanje nosi 1 poen i na svako pitanje postoji samo jedan tačan odgovor. Maksimalan broj poena je 10. Svaki prijavljeni korisnik ima na uvid tabelu svih korisnika kao i njihove rezultate. Ako korisnici nemaju naloge, neophodno je da se registruju da bi mogli da pristupe.

2. Opis delova programa sa izvornim kodom

Aplikacija je urađena u programskom jeziku PHP.

Prvo je neophodno konektovanje sa MySQL-om, preko username-a, šifre i hostname-a. Zatim se pravi baza podataka ukoliko ne postoji i povezuje se sa MySQL-om.

```
<?php
    $host = 'localhost';
    $username = 'root';
    $password = 'sofiya';

    $connection = mysqli_connect($host,$username,$password)
    or die("Nisam uspeo da se konektujem!" . mysqli_connect_error());

    $sql = "CREATE DATABASE IF NOT EXISTS Test";

    mysqli_query($connection, $sql) or die(mysqli_error($connection));

?>
```

Slika 1.

Poziva se baza podataka i konektuje se sa njom, a u okviru nje se pravi tabela USER, koja će imati polja kao na slici 2. Isto se odradi i za tabelu Pitanja (Slika 3).

```
require_once 'createDB.php';
//mysql_set_charset('utf8');
$db = mysqli_connect($host, $username, $password, 'Test');
// User tabela
$query = "CREATE TABLE IF NOT EXISTS User(
    user_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    ime VARCHAR(30) NOT NULL,
    prezime VARCHAR(30) NOT NULL,
    email VARCHAR(30) NOT NULL,
    brpoena int NULL,
    password VARCHAR(100) NOT NULL)
";

mysqli_query($db, $query);
```

Slika 2.

```
// Pitanja tabela
$query20 = "CREATE TABLE IF NOT EXISTS Pitanja(
    id_pitanja INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    pitanje VARCHAR(555) NOT NULL,
    odgovor VARCHAR(555) NOT NULL,
    odgovor1 VARCHAR(555) NOT NULL,
    odgovor2 VARCHAR(555) NOT NULL,
    odgovor3 VARCHAR(555) NOT NULL,
    odgovor4 VARCHAR(555) NOT NULL
)";
```

Slika 3.

Podešavanje da se u okviru aplikacije koristi ćirilica, slika 4

```
mysqli_query($db, $query20);
mysqli_query($db, "ALTER DATABASE Test CHARACTER SET utf8 COLLATE utf8_unicode_ci");
mysqli_query($db, "ALTER TABLE Pitanja CONVERT TO CHARACTER SET utf8 COLLATE utf8_unicode_ci");
mysqli_query($db, "SET NAMES UTF8");
```

Slika 4.

Sledi ubacivanje vrednosti u tabelu Pitanja koja je u bazi podataka zajedno sa ponudjenim odgovorima

```
// Unos pitanja

$query21 = "INSERT INTO Pitanja VALUES(1,'<b> Секвенца акција репрезентује интеракцију: </b><br> ', 'актера са субјектом и
    кључним
апстракцијама објекта.', 'актера са објектом и кључним апстракцијама субјекта.', 'актера са објектом и кључним
апстракцијама објекта.', 'актера са субјектом и кључним апстракцијама субјекта и објекта.', 'актера са субјектом и кључним
апстракцијама субјекта.')
```

Slika 5.

Kada je završen rad sa bazom, sledi deo u kojem se prikazuje korisnički interfejs. Korisniku se nude polja za Prijavljivanje ako već ima nalog. U okviru formi se popunjavaju polja za e-mail i šifru. Pritiskom tastera za prijavljivanje, korisnik prelazi na stranu login.php (slika 7). Ukoliko korisnik nema nalog, neophodno je da pritisne taster za registraciju gde prelazi na stranu signup.php (slika 8).

```
1 <form action="login.php" method="POST">
2 <h1> Добро дошли! </h1>
3 <p>Унесите своје податке:</p>
4 <hr>
5 <input type="email" placeholder="Е-адреса" name="email" style=" width:auto;" required = "required"> <br>
6 <br>
7 <input type="password" placeholder="Шифра" name="password" style=" width:auto;" required = "required"> <br>
8 <button type="submit" style="width:auto;" name="submit">Пријављивање</button>
9 </form>
10 <form action="signup.php" method="POST">
11 <button type="submit" style="width:auto;" name="submit">Регистрација</button>
12 </form>
```

Slika 6

Pri prijavljivanju se proveravaju podaci koji se unose da bi se proverila ispravnost unetih podataka. Ako su podaci validni, korisnik se uspešno prijavio i može da počne sa testiranjem

```
if (isset($_POST['submit'])){
    if (filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
        $email = $_POST['email'];
        $_SESSION['poruka']='';
    }
    else {
        $_SESSION['poruka']="Неисправна електронска адреса!";
        header("Location: pocetna.php");
    }
    $password = $_POST['password'];
    $query = "SELECT password FROM User WHERE email='$email'";
    $result = mysqli_query($db, $query);

    $query2 = "SELECT user_id FROM User WHERE email='$email'";
    $result2 = mysqli_query($db, $query2);
    $userID = mysqli_fetch_array($result2);
```

Slika 7

Ako korisnik nema nalog, sledi mu registracija. Registracijom se sakupljaju uneti podaci I ubacuju u bazu podataka da bi se kasnijim prijavljivanjem znalo, koji korisnik pristupa aplikaciji (slika 9).

```
<form method="post" action = "newacc.php">
    <input type="text" name="ime" placeholder="Име" required="required" style="width:auto;">
    <input type="text" name="prezime" placeholder="Презиме" required="required" style="width:auto;">
    <input type="text" name="email" placeholder="Е-адреса" required="required" style="width:auto;">
    <input type="password" name="password" placeholder="Шифра" required="required" style="width:auto;">
    <button type="submit" style="width:auto; left:51%;" name="submit" value="Submit"><b><i>Направите налог!</i></b></button>
```

Slika 8

```
$ime = $_POST['ime'];
$prezime = $_POST['prezime'];

$email = $_POST['email'];

$password = password_hash($_POST['password'],PASSWORD_DEFAULT);

$query = "INSERT INTO User VALUES(NULL, '$ime', '$prezime', '$email', NULL, '$password')";

mysqli_query($db, $query);

$query2 = "SELECT user_id FROM User WHERE email='$email'";
$result = mysqli_query($db, $query2);
$userID = mysqli_fetch_array($result);
$_SESSION[user_id] = $userID[0];

header('Location: html.php');
```

Slika 9

Kada se pristupi testu, prelazi se na fajl html.php. U njemu se vuku pitanja I ponudjeni odgovori iz baze kako bi korisnik imao uvid u sva pitanja I njihove moguće odgovore

```

5
6     $query = "SELECT pitanje FROM Pitanja";
7     $result = mysqli_query($db, $query);
8     $pitanja = mysqli_fetch_all($result);
9
10    $query1 = "SELECT odgovor FROM Pitanja";
11    $result1 = mysqli_query($db, $query1);
12    $odgovor = mysqli_fetch_all($result1);
13

```

Slika 10

Prolaskom kroz for petlju se prolazi kroz svih 10 pitanja I za svako pitanje kroz svih 5 odgovora. Proverom vrednosti svakog ponudjenog odgovora, se proverava koje od odgovora je tačno. Svaki odgovor koji ima vrednost 1, se gleda kao tačan odgovor, dok odgovori koji imaju vrednost 0.00001 se gledaju kao netačni. Php vrednost 0 posmatra kao odgovor koji nije popunjen, zato je neophodno da se stavi vrednost zanemarljivo veća od 0.

```

for ($i = 0; $i<10 ; $i++){
    $broj = $i+1;
    echo $broj."." . " " . $pitanja[$i][0] . "<br>";
    if ($j == 5) {
        $j = 0;
    }
    if($broj== 1){
        echo $odgovor[$j][0] . "<br>";
        echo "<input type='radio' name='pitanje".$broj."' value='0.00000001'> <br>";
        echo $odgovor1[$j][0] . "<br>";
        echo "<input type='radio' name='pitanje".$broj."' value='0.00000001'> <br>";
        echo $odgovor2[$j][0] . "<br>";
        echo "<input type='radio' name='pitanje".$broj."' value='0.00000001'> <br>";
        echo $odgovor3[$j][0] . "<br>";
        echo "<input type='radio' name='pitanje".$broj."' value='0.00000001'><br>";
        echo $odgovor4[$j][0] . "<br>";
        echo "<input type='radio' name='pitanje".$broj."' value='1'><br>";
        $j++;
    }
}

```

Slika 11

Pritiskom tastera za potvrdu odgovora tj rešavanjem testa, prolazi se kroz petlju gde se proverava da li je odgovoreno na svako pitanje kao I vrednost odgovora da bi moglo da se izračuna ukupan rezultat testa. Ukoliko na neko pitanje nedostaje odgovor, neophodno je da se opet odradi test. Odgovori se pakuju u listu, da bi se lakse štampalo koja pitanja nisu tačna

```
if (isset($_POST['submit'])){
    for ($i = 1; $i<= 10; $i++){
        if (!empty($_POST['pitanje'].$i)){
            if ($_POST['pitanje'].$i == 1){
                $brpoena++;
                $rezultati[$i-1]= 1;
            }
            else {
                $rezultati[$i-1]=0;
            }
        } else {
            echo "Нисте одговорили на питање!";
            header('Location: html.php');
        }
    }
}
```

Slika 12

Prolaskom kroz listu, proverava se svako pitanje, tj njegova vrednost. Sva pitanja koja imaju vrednost 0, znači da su netačna I štampa se upozorenje o tome

```
$user_id = $_SESSION['user_id'];
$query="UPDATE User SET brpoena = $brpoena Where user_id = $user_id";
$result = mysqli_query($db, $query);

echo "Osvojili ste " . $brpoena. " poena <br>";
for($j=1;$j<=10;$j++){
    if($rezultati[$j-1]==0){
        echo "Pitanje" . $j . ": Netacno <br>" ;
    }
}
```

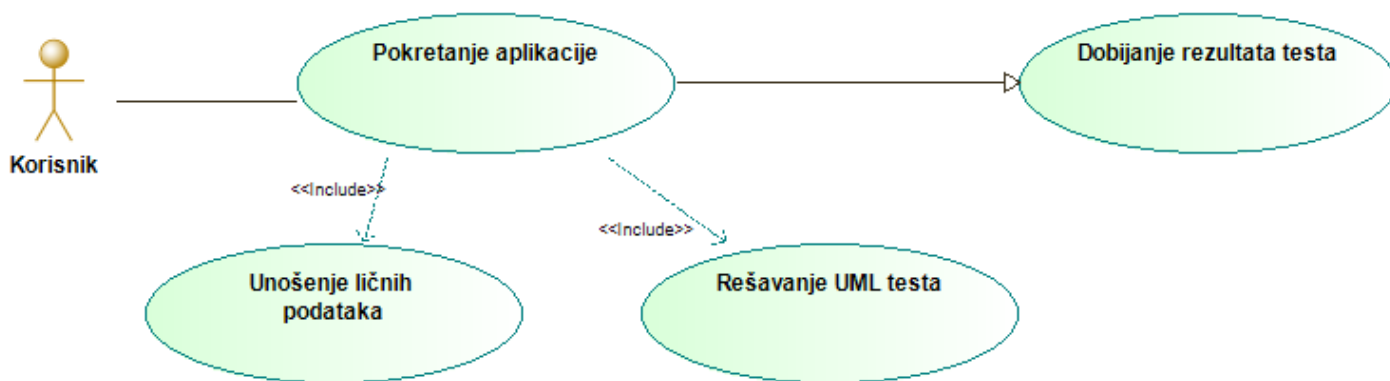
Slika 13

3. UML dijagrami

3.1 Dijagram slučajeja korišćenja (Use case diagrams)

Dijagram slučajeja korišćenja prikazuje skup slučajeja korišćenja i aktera. Vizuelizuje ponašanje sistema, podsistema ili čak klase i interfejsa. Elementi dijagrama su: slučajevi korišćenja, akteri, relacije i paketi.

Dijagram slučajeja korišćenja za datu aplikaciju predstavljen je na slici 6.

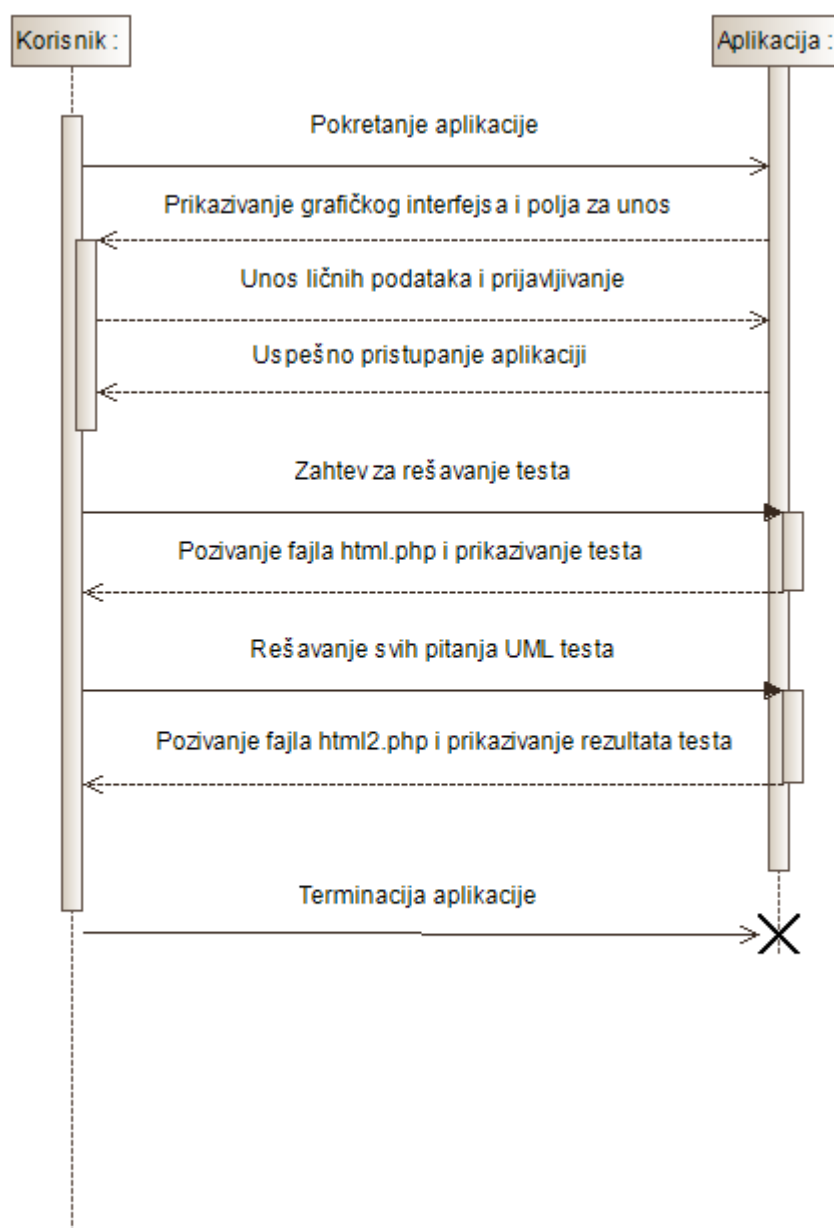


Slika 6.

3.2 Dijagram sekvenci (Sequence diagram)

Prikazuje komunikaciju između skupa objekata, koja se ostvaruje porukama koje objekti međusobno razmenjuju u cilju ostvarivanja očekivanog ponašanja. Detaljno opisuje kako se operacije izvode, koristi se za prikaz jednog ili više scenarija. Sadrži dve dimenzije: vertikalnu (označava vreme) i horizontalnu (označava objekte)

Dijagram sekvenci za datu aplikaciju predstavljen je na slici 7.

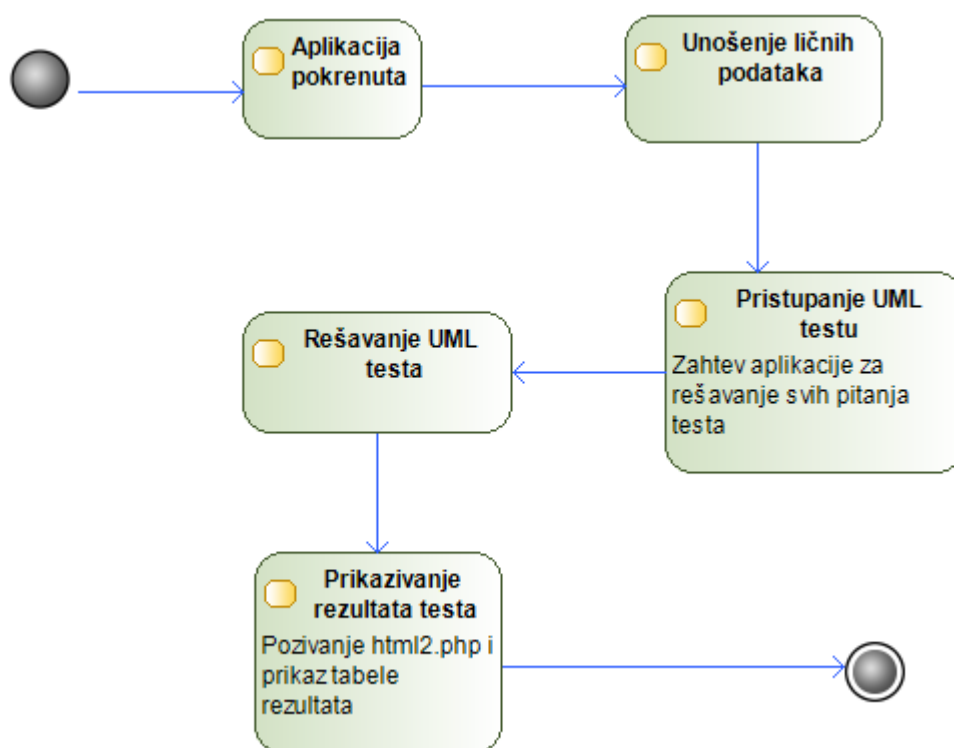


Slika 7.

3.3 Dijagram aktivnosti (Activity diagram)

Dijagrami aktivnosti su namenjeni modeliranju dinamičkih aspekata (ponašanja) Sistema. Dijagram aktivnosti prikazuje tok aktivnosti koju izvršavaju objekti, eventualno i tok objekata između koraka aktivnosti.

Dijagram aktivnosti za datu aplikaciju predstavljen je na slici 8.

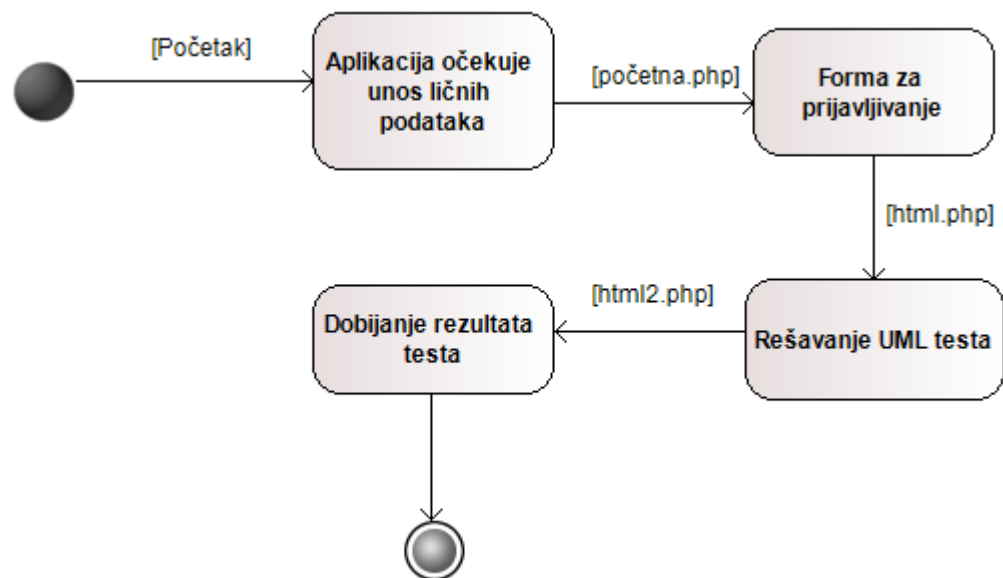


Slika 8.

3.4 Dijagram stanja (State machine diagram)

- Automat stanja/Mašina stanja (state machine)
 - ponašanje koje specificira sekvence stanja kroz koja prolazi
 - modelira ponašanje nekog entiteta ili protokol interakcije
- Entitet reaguje na događaje promenom stanja
 - promena stanja izaziva nove događaje i akcije
- Dijagram stanja je graf koji prikazuje automat stanja
 - čvorovi su stanja
 - grane su prelazi

Dijagram stanja za datu aplikaciju predstavljen je na slici 9.



Slika 9.

Literatura

- <http://moodle.mfkg.rs/course/view.php?id=524>
- <https://www.modelio.org/documentation-menu/user-manuals.html>