Mobilna aplikacija za anketiranje u stvarnom vremenu

Pavičić, Marko

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:846337

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-24**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni preddiplomski studij Elektrotehnike

MOBILNA APLIKACIJA ZA ANKETIRANJE U STVARNOM VREMENU

Završni rad

Marko Pavičić

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 27.08.2021.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Marko Pavičić
Studij, smjer:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4569, 24.07.2018.
OIB studenta:	23689261882
Mentor:	Izv. prof. dr. sc. Mirko Köhler
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Mobilna aplikacija za anketiranje u stvarnom vremenu
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	27.08.2021.
Datum potvrde ocjene Odbora:	

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 01.09.2021.

Ime i prezime studenta:	Marko Pavičić
Studij:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4569, 24.07.2018.
Turnitin podudaranje [%]:	12

Ovom izjavom izjavljujem da je rad pod nazivom**: Mobilna aplikacija za anketiranje u stvarnom vremenu**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Mirko Köhler

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. PRIKAZ POSTOJEĆIH RJEŠENJA 2
2.1. CARNET Loomen – Ankete 2
2.2. Kahoot! 3
3. KORIŠTENE TEHNOLOGIJE 4
3.1. HTML 4
3.2. Bootstrap 4
3.3. JavaScript 4
3.4. Chart.js 4
3.5. Visual Studio Code 4
3.6. Firebase 5
3.7. Java 5
3.8. XML 5
3.9. Android Studio 6
4. PROGRAMSKO RJEŠENJE 7
4.1. Web administratorsko sučelje 7
4.1.1. Stranica prijave 7
4.1.2. Glavna stranica 9
4.1.3. Stranica povijesti anketiranja 13
4.2. Android aplikacija 15
4.2.1. Zaslon prijave 15
4.2.2. Zaslon registracije 17
4.2.3. Početni zaslon 18
4.2.4. Zaslon upravljanja grupama 20
4.2.5. Zaslon unosa koda 21
4.2.6. Zaslon pripreme za anketu 22
4.2.7. Zaslon anketiranja 22
5. PRIMJENA APLIKACIJE I ADMINISTRATORSKOG SUČELJA 24
5.1. Primjena administratorskog sučelja 24
5.2. Primjena Android aplikacije 28
6. ZAKLJUČAK 33
LITERATURA 34
SAŽETAK 36
ABSTRACT 37
ŽIVOTOPIS 38
PRILOZI 39

1. UVOD

Anketiranjem prikupljamo podatke u korisničkom istraživanju. Provođenje anketa u stvarnom vremenu predstavlja brzo i jednostavno rješenje problema prikupljanja velike količine podataka i pregleda rezultata. S obzirom na to da su ovakve ankete anonimne ispitanici su voljniji rješavati ih i spremni su odgovoriti na osjetljivija pitanja.

U završnom radu napravljen je pregled postojećih rješenja na tržištu i opisana je izrada administratorskog sučelja za izradu i pregled rezultata anketa te mobilne aplikacije za ispunjavanje anketa.

Za izradu administratorskog sučelja korišten je Visual Studio Code, HTML, JavaScript te Bootstrap dok je za mobilnu aplikaciju korišten Android Studio i programski jezik Java. Podaci se spremaju u bazu podataka pomoću Firebase platforme.

Administratorsko sučelje sastoji se od dijela za izradu anketa i dijela za pregled rezultata. Prijava u administratorsko sučelje vrši se pomoću e-pošte i lozinke. Anketi se pristupa preko mobilne aplikacije pomoću jedinstvenog koda generiranog putem administratorskog sučelja. Anketi je moguće pristupiti samo do određenog vremena kojeg definira administrator.

U sljedećem poglavlju opisana su postojeća rješenja. U trećem poglavlju opisane su tehnologije korištene prilikom izrade aplikacije. Četvrto poglavlje opisuje postupak izrade aplikacije te programski kod. Peto poglavlje prikazuje primjenu izrađene aplikacije. Posljednje poglavlje je zaključak ovog završnog rada.

1.1. Zadatak završnog rada

U završnom radu potrebno je napraviti mobilnu aplikaciju za anketiranje u stvarnom vremenu. Korisnik se može prijaviti u anketu putem lozinke sobe ili korisničkih podataka. Glasati može samo anonimno. Anketno pitanje ima vremensko ograničenje i nije ga moguće naknadno izmijeniti. Istekom vremena u administratorskom sučelju pojavljuju se grafički rezultati ankete. U istom tom sučelju kreiraju se pitanja i postavljaju vremenska ograničenja. Rezultate ankete moguće je spremiti.

2. PRIKAZ POSTOJEĆIH RJEŠENJA

Anketiranje putem Interneta provodi se još devedesetih godina prošlog stoljeća, te su se kroz vrijeme razvile mnoge aplikacije koje omogućuju anonimno anketiranje korisnika i grafički prikaz rezultata osobi koja provodi anketu. U nastavku ovog poglavlja bit će opisana postojeća rješenja te će biti napravljena usporedba postojećih rješenja i rješenja ovog završnog rada

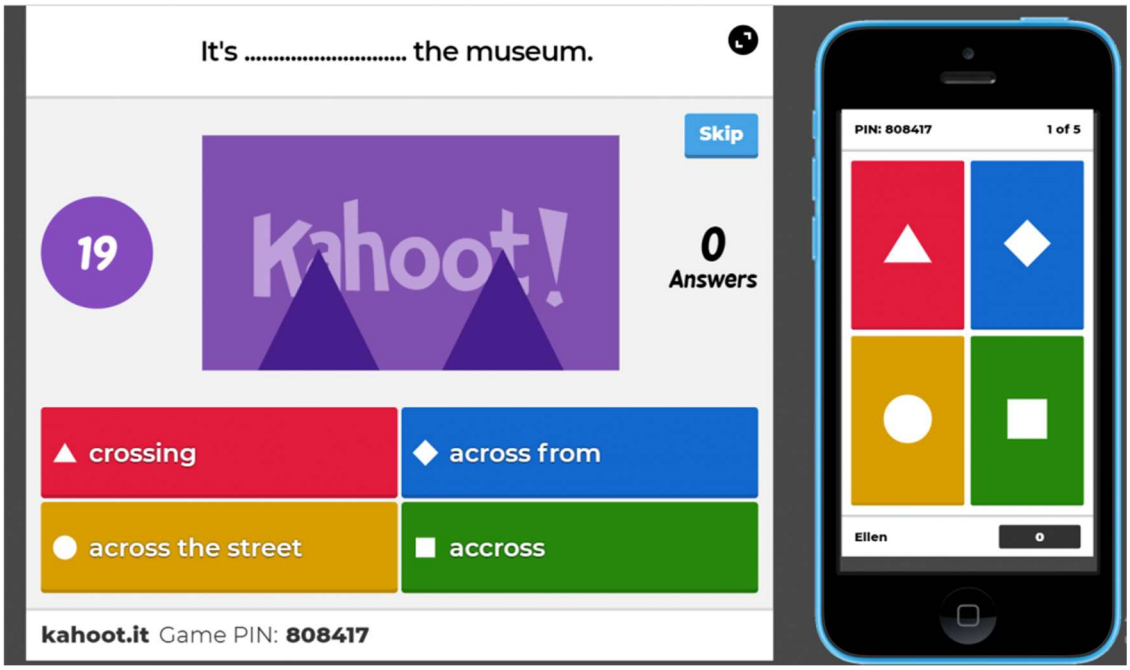
2.1. CARNET Loomen – Ankete

Ankete na CARNET Loomen-u nastavniku omogućuju da dobije povratnu informaciju od polaznika vezanu za e-kolegij ili o nekoj drugoj određenoj temi [1]. Nastavnik mora unijeti naziv i opis ankete, također se može odrediti period u kojemu će se anketa moći rješavati, te postoji mogućnost odabira hoće li anketa biti anonimna ili ne za razliku od aplikacije iz ovog završnog rada gdje anketa može biti samo anonimna. Nakon što nastavnik odredi postavke ankete, izrađuje pitanja koja će biti u anketi, koja mogu biti različitih vrsta (kraći tekstualni odgovor, višestruki odabir, brojčani odgovor…) za razliku od rješenja ovog završnog rada gdje su sva pitanja višestrukog odabira. Nakon provedbe ankete nastavnik može pogledati rezultate preko interaktivnih grafikona i tablica što je sličnost s aplikacijom koja se izrađuje ovim završnim radom. Slika 2.1. prikazuje izgled jedne ankete na CARNET Loomen sustavu.

Slika 2.1. *Izgled ankete na CARNET Loomen sustavu*

2.2. Kahoot!

Preko 70 milijuna ljudi diljem svijeta koriste Kahoot!, platformu za učenje temeljenu na igri [2]. Korisnici ne moraju imati račun da bi pristupili Kahoot! kvizu tako da mogu ostati anonimni ali i dalje se njihovi rezultati prikazuju preko nadimka koji si korisnik sam dodjeli, dok u aplikaciji iz ovog završnog rada nema zapisivanja nikakvih podataka o korisniku, pa ni nadimka. Kahoot! je platforma za učenje koja je interaktivna i zanimljiva pa privlači pozornost i zadržava korisnike te pomaže učenicima u učenju što joj je i primarna namjena. Namjena aplikacije iz ovog završnog rada je isključivo anonimno anketiranje putem mobilnog uređaja. Izgled Kahoot! kviza prikazan je na slici 2.2.



Slika 2.2. *Izgled Kahoot! kviza [3]*

3. KORIŠTENE TEHNOLOGIJE

U ovom poglavlju navedene su i opisane tehnologije i razvojni alati korišteni za izradu web administratorskog sučelja i mobilne aplikacije za anketiranje.

3.1. HTML

Osnovni jezik koji se upotrebljava za izradu web stranica je HTML [4]. HTML web pregledniku daje podatke o sadržaju i strukturi web stranice, zatim web preglednik pomoću tih podataka prikazuje stranicu kakvu mi vidimo. HTML nije programski jezik jer njime ne možemo izvršiti nikakvu zadaću već mu je jedina primjena opis hipertekstualnih dokumenata.

3.2. Bootstrap

Bootstrap je najpopularniji framework za izradu responzivnih web stranica na svijetu, prvenstveno s mobilnim uređajima na umu [5]. Bootstrap je besplatan i CSS front-end framework otvorenog izvornog koda.

3.3. JavaScript

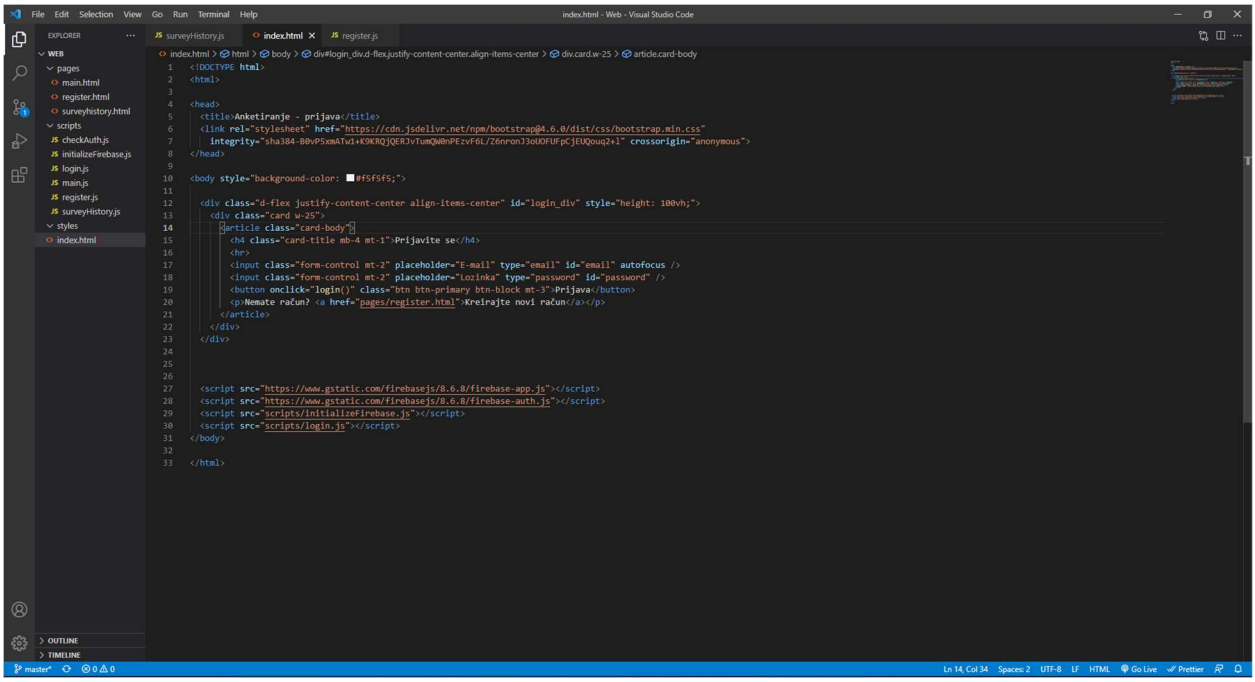
JavaScript je jednostavan programski jezik kojemu je glavna namjena razvoj interaktivnih HTML-stranica. Jezgra JavaScripta dolazi ugrađena u većinu modernih preglednika (Google Chrome, Mozilla Firefox, Safari i drugi) [6]. JavaScript ima sintaksu sličnu Javi te je prvi put inačica JavaScript 1.0 implementirana u preglednik Netscape 2.

3.4. Chart.js

Za grafički prikaz rezultata anketa korištena je besplatna Chart.js JavaScript biblioteka otvorenog izvornog koda [7]. Chart.js podržava razne vrste grafikona (tortni, linijski, prstenasti, …). Chart.js repozitorij na GitHub-u je jedan od najpopularnijih, nalazi se na 79. mjestu na svijetu s preko pedeset tisuća „Star-ova“.

3.5. Visual Studio Code

Visual Studio Code je lagan, ali moćan uređivač izvornog koda koji radi na osobnom računalu dostupan na Windows, macOS i Linux operativnim sustavima [8]. Dolazi s podrškom za JavaScript, TypeScript i Node.js te ima mogućnost podrške raznih drugih jezika poput C++, C#, Jave, Pythona, PHP i Go-a putem mnoštva proširenja koja se nude. Slika 3.1. prikazuje izgled Visual Studio Code-a.



Slika 3.1. *Visual Studio Code*

3.6. Firebase

Firebase je platforma razvijana od strane Google-a koja služi razvoju mobilnih i web aplikacija. Prvotno je osnovana 2011. godine od strane neovisne tvrtke Envolve da bi 2014. godine Google kupio platformu i napravio ju svojom vodećom ponudom za razvoj aplikacija [9]. Od široke ponude Firebase platforme aplikacija iz ovog završnog rada koristi usluge Firebase provjere identiteta za registraciju, prijavu i organizaciju korisnika te Firebase baza podataka stvarnog vremena za pohranu informacija, pitanja i rezultata anketa kao i informacija o grupama i njihovim članovima.

3.7. Java

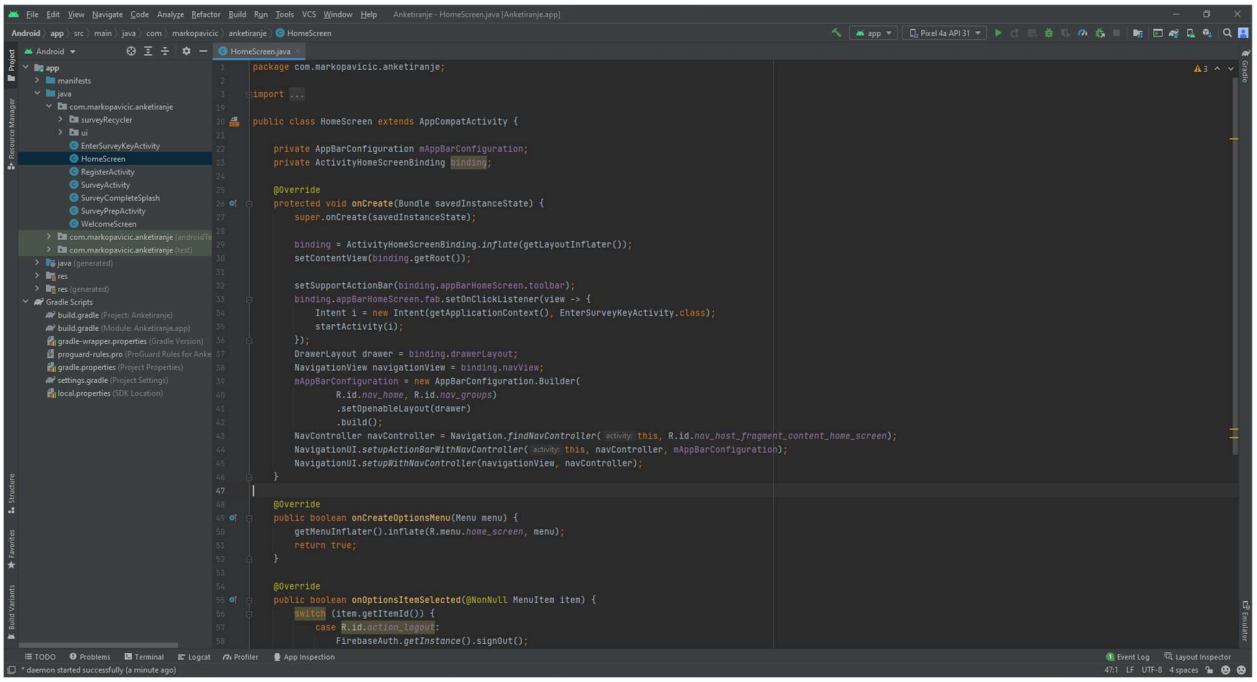
Java je objektno orijentirani programski jezik razvijen od strane James Gosling-a, Patrick Naughton-a i drugih inženjera iz tvrtke Sun Microsystems. Razvoj je počeo 1991. godine, kao dio projekta Green, a objavljen je 1995. godine [10]. Java je do 2019. godine bila temeljni programski jezik za razvoj aplikacija Android operativnog sustava kada je Google na I/O događaju objavio da Android postaje „Kotlin-first“.

3.8. XML

XML kratica je za EXtensible Markup Language, odnosno jezik za obilježavanja podataka. Svi rasporedi zaslona u mobilnoj aplikaciji opisani su XML opisnim jezikom.

3.9. Android Studio

Android Studio je službeno integrirano razvojno okružje (IDE) za razvoj Android aplikacija, temeljeno na IntelliJ IDEA [11]. Preuzimanje Android Studio-a besplatno je i omogućeno Apache 2.0 licencom, za razvoj zahtjeva instalaciju Java razvojnog okruženja (engl. Java Development Kit - JDK). Može se pokrenuti na Windows, macOS i Linux operativnim sustavima. Android Studio također nudi i Android Emulator za jednostavno pokretanje aplikacija na virtualnom uređaju koji se pokreće iz samog IDE-a. Slika 3.2. prikazuje izgled Android Studio-a.



Slika 3.2. *Android Studio*

4. PROGRAMSKO RJEŠENJE

U ovom poglavlju opisano je programsko rješenje završnog rada. Potrebno je izraditi web administratorsko sučelje za osobu koja kreira ankete iščitava rezultate, te mobilnu aplikaciju pomoću koje se pristupa anketi.

4.1. Web administratorsko sučelje

Administratorsko sučelje služi kreiranju anketa i prikazu rezultata anketa. Za izradu web administratorskog sučelja korišten je HTML, Bootstrap, JavaScript te Chart.js biblioteka za grafički prikaz rezultata.

4.1.1. Stranica prijave

Stranica prijave sastoji se od Bootstrap kartice (engl. card) unutar kojega se nalaze dva unosa za e-mail i lozinku te gumba za prijavu. Klikom na gumb poziva se funkcija *login()* kojom se vrši provjera sadržaja polja te prijava putem Firebase provjere identiteta. Na dnu kartice je veza na stranicu registracije ako korisnik još nema račun. HTML kod stranice za prijavu prikazan je na slici 4.1.

```
<!DOCTYPE html>
<html>

<head>
  <title>Anketiranje - prijava</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
    integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+l" crossorigin="anonymous">
</head>

<body style="background-color: #f5f5f5;">

  <div class="d-flex justify-content-center align-items-center" id="login_div" style="height: 100vh;">
    <div class="card w-25">
      <article class="card-body">
        <h4 class="card-title mb-4 mt-1">Prijavite se</h4>
        <hr>
        <input class="form-control mt-2" placeholder="E-mail" type="email" id="email" autofocus />
        <input class="form-control mt-2" placeholder="Lozinka" type="password" id="password" />
        <button onclick="login()" class="btn btn-primary btn-block mt-3">Prijava</button>
        <p>Nemate račun? <a href="pages/register.html">Kreirajte novi račun</a></p>
      </article>
    </div>
  </div>



  <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-auth.js"></script>
  <script src="scripts/initializeFirebase.js"></script>
  <script src="scripts/login.js"></script>
</body>

</html>
```

Slika 4.1. *HTML kod stranice za prijavu*

Na dnu su ugrađene Firebase skripte za provjeru identiteta te skripte *initializeFirebase.js* i *login.js*. Skripta *initializeFirebase.js* služi za inicijalizaciju Firebase servisa s konfiguracijom za projekt aplikacije te je prikazana na slici 4.2.

```
const firebaseConfig = {
  apiKey: "AIzaSyDvIg1ZO-JsQ-2R4bsZsNnFJAZfVrNlZLg",
  authDomain: "anketiranje-180f6.firebaseapp.com",
  databaseURL: "https://anketiranje-180f6-default-rtdb.europe-west1.firebasedatabase.app",
  projectId: "anketiranje-180f6",
  storageBucket: "anketiranje-180f6.appspot.com",
  messagingSenderId: "1065651984396",
  appId: "1:1065651984396:web:cf13394f1118d7d233f629"
};
firebase.initializeApp(firebaseConfig);
```

Slika 4.2. *Skripta initializeFirebase.js*

Skripta *login.js* (Slika 4.3.) na početku izvođenja provjerava je li korisnik već prijavljen te ga, ako je prijavljen, preusmjerava na glavnu stranicu. Funkcija *login()* poziva se na klik gumba što je definirano u HTML kodu pomoću *onclick* događaja, pokretanjem provjerava ima li podataka u unosima te ako ima sprema te podatke u varijable *userEmail* i *userPass* nakon čega se poziva funkcija *firebase.auth().signInWithEmailAndPassword(userEmail, userPass)* kojom se vrši prijava putem Firebase provjere identiteta, ako dođe do pogreške prilikom prijave ta se pogreška prikazuje na zaslonu korisnika.

```
firebase.auth().onAuthStateChanged(function (user) {
  if (user) {
    var user = firebase.auth().currentUser;
    var userID = firebase.auth().currentUser.uid;
    sessionStorage.setItem('loggedInUser', userID);
    if (user == null) {
      document.getElementById("login_div").style.display = "block";
    }
    if (user != null) {
      window.location.href = 'pages/main.html';
    }
  }
});

function login() {
  var userEmail = document.getElementById("email").value;
  var userPass = document.getElementById("password").value;
  if (userEmail != "" && userPass != "") {
    firebase.auth().signInWithEmailAndPassword(userEmail, userPass).catch(function (error) {
      var errorMessage = error.message;
      window.alert("Error : " + errorMessage);
    });
  } else
    window.alert("Neka od polja su prazna.");
}
```

Slika 4.3. *Skripta login.js*

4.1.2. Glavna stranica

Početna stranica sastoji se od navigacijskog okna te Bootstrap *jumbotron* elementa u kojemu se nalaze gumbi za kreiranje ankete i prikaz povijesti anketiranja. Klikom na gumb *Kreirajte anketu* otvara se dijalog za kreiranje anketa

Klikom na + pored unosa odgovora poziva se funkcija *addAnswer()* koja je prikazana na slici 4.4. Funkcija mijenja plus u minus na pritisnutom gumbu, povećava brojač pitanja te dodaje novi unos pozivom funkcije *createAnswerInput()*.

```
function addAnswer() {
    document.getElementById("answerGroup" + (answerCounter) + "Btn").innerHTML = '<svg xmlns="http://www.w3.org/2000/svg"
    document.getElementById("answerGroup" + (answerCounter) + "Btn").classList.remove('btn-success');
    document.getElementById("answerGroup" + (answerCounter) + "Btn").classList.add('btn-danger');
    document.getElementById("answerGroup" + (answerCounter) + "Btn").setAttribute('onclick', "removeAnswer(this.id);");

    answerCounter++;

    createQuestion();
}
```

Slika 4.4. *Funkcija addAnswer()*

Klikom na gumb + *Dodaj pitanje* poziva se funkcija *newQuestion()* prikazana slikom 4.5. koja poziva funkciju *loadQuestion()* koja sprema podatke upisane u unose u polje pitanja i odgovora. Funkcija zatim provjerava jesu li podaci uspješno spremljeni te ako jesu čisti unose te povećava brojač pitanja.

```
function newQuestion() {
    loadQuestion();

    if (questions.length != 0) {
        answerCounter = 0;
        createQuestionForm();
        questionCounter++;
    }
}
```

Slika 4.5. *Funkcija newQuestion()*

Klikom na gumb *Dalje* poziva se funkcija *loadQuestion()* koja prvo provjerava a onda sprema sve upisane podatke u polje pitanja i odgovora te se otvara dijalog unosa trajanja vremena ankete. Funkcija for petljom prolazi kroz cijeli dokument i dohvaća elemente te ako elementi imaju zapisane znakove sprema ih u polje *questions.* Na slici 4.6. priakzana je funkcija *loadQuestion()*

```
function loadQuestion() {
    var question = document.getElementById("question").value;

    for (var i = 0; i <= answerCounter; i++) {
        var answer = document.getElementById("answer" + i);
        if (answer) {
            answer = answer.value;
            if (answer.length != 0)
                answers.push([answer, 0]);
        }
    }
    if (question.length != 0 && answers.length != 0) {
        questions.push([question, answers]);
        answers = [];
    } else
        alert("Neka polja su prazna.");
    chosenGroups = [];
    $("#chooseGroups input[type='checkbox']:checked").each((_, {
        value
    }) => {
        chosenGroups.push(value);
    });
}
```

Slika 4.6. *Funkcija loadQuestion()*

Klikom na gumb *Završi* na drugom dijalogu kreiranja anketa poziva se funkcija *uploadToFirebase()* koja sve unesene podatke prenosi na Firebase bazu podataka.

Podaci koji se spremaju su: kod ankete, pitanja i odgovori, ID tvorca ankete, trajanje ankete u minutama te grupe kojima će anketa biti dostupna. Unutar *uploadToFirebase()* (Slika 4.7.) funkcije poziva se funkcija *makecode()* koja kreira kod ankete te provjerava ako anketa s generiranim kodom već postoji, ako postoji, generira se novi kod sve dok ne bude unikatan. Kod se generira iz uzorka od 58 znakova, mješavine velikih i malih slova te brojeva. Dužine je šest znakova tako da postoji preko $3{,}8 * 10^{10}$ mogućih kombinacija. Nakon što se anketa prenese na bazu podataka korisniku se prikazuje kod za pristup anketi.

```
function uploadToFirebase() {
    var database = firebase.database();
    var user = firebase.auth().currentUser;
    var code = makecode(6, "Survey");

    document.getElementById("code").innerHTML = code;

    var newSurveyKey = firebase.database().ref().child('ankete').push().key;

    if (questions.length == 0) {
        alert("Ostavili ste neka polja prazna, molim provjerite.")
        code = "Pokušajte ponovo."
        document.getElementById("code").innerHTML = code;
    } else {
        database.ref('ankete/' + newSurveyKey).set({
            kod: code,
            pitanja: questions,
            tvorac: user.uid,
            trajanje: surveyLength,
            odabraneGrupe: chosenGroups
        });
        reload();
    }
}
```

Slika 4.7. *Funkcija uploadToFirebase()*

Ako korisnik odabere poveznicu *Upravljanje grupama* iz navigacijskog okna otvara mu se dijalog u kojemu vidi sve grupe koje je kreirao, njihove pripadajuće pristupne kodove te ima mogućnost brisanja starih i kreiranja novih grupa. Grupe se učitavaju pomoću funkcije *loadGroups()*. Funkcija kreira tablicu te ju ispunjava grupama koje je korisnik izradio. Prvi stupac tablice predstavlja naziv grupe dok drugi stupac predstavlja kod za pristup grupi. Naziv i kod grupe dohvaćaju se sa Firebase baze podataka. Provjerava se postoji li ijedna grupa te ukoliko postoji prikazuje se tablica. Ukoliko korisnik nema niti jednu grupu prikazuje se odgovarajuća poruka umjesto tablice. Na slici 4.8. prikazana je funkcija *loadGroups()*.

```
function loadGroups() {
    var checkIfAnyGroupExists = false;
    var user = firebase.auth().currentUser;
    firebase.database().ref().child("grupe").child(user.uid).once("value", function (snapshot) {
        document.getElementById("table").innerHTML = "";
        var table = document.createElement('table');
        table.classList.add('table');
        var tr = document.createElement('tr');
        var thName = document.createElement('th');
        thName.innerHTML = "Naziv grupe";
        var thCode = document.createElement('th');
        thCode.innerHTML = "Kod grupe";
        tr.appendChild(thName);
        tr.appendChild(thCode);
        table.appendChild(tr);
        snapshot.forEach(function (childSnapshot) {
            var tr = document.createElement('tr');
            var tdName = document.createElement('td');
            tdName.innerHTML = childSnapshot.child("ime").val();
            var tdCode = document.createElement('td');
            tdCode.innerHTML = childSnapshot.child("kod").val();
            var deleteGroupBtn = document.createElement('button');
            deleteGroupBtn.classList.add('btn');
            deleteGroupBtn.classList.add('btn-danger');
            deleteGroupBtn.classList.add('mt-1');
            deleteGroupBtn.innerHTML = '<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
            deleteGroupBtn.id = childSnapshot.key;
            deleteGroupBtn.setAttribute('onclick', "deleteGroup(this.id); loadGroups();");
            tr.appendChild(tdName);
            tr.appendChild(tdCode);
            tr.appendChild(deleteGroupBtn);
            table.appendChild(tr);
            if (tdName.length != 0 && tdCode.length != 0)
                checkIfAnyGroupExists = true;
        });
        if (checkIfAnyGroupExists)
            document.getElementById("table").appendChild(table);
        else {
            var h4 = document.createElement('h4');
            h4.classList.add('text-center');
            h4.innerHTML = "Nemate niti jednu grupu";
            document.getElementById("table").appendChild(h4);
        }
    });
}
```

Slika 4.8. *Funkcija loadGroups()*

Popis grupa sprema se na bazu podataka, te se i prilikom brisanja brišu s baze podataka pozivom funkcija *createGroup()* i *deleteGroup()*.

Funkcija *createGroup()* (Slika 4.9.) učitava ime grupe te kreira kod za pristup grupi koji se kreira na isti način kao i kodovi za ankete. Zatim se na Firebase bazu podataka prenosi ime grupe i kod grupe te se kod prikazuje u dijalogu.

```
function createGroup() {
    var groupName = document.getElementById("groupName").value;

    var groupCode = makecode(6, "Group");
    var database = firebase.database();
    var user = firebase.auth().currentUser;
    var newGroupKey = firebase.database().ref().child('grupe').push().key;
    if (groupName.length != 0) {
        database.ref('grupe/' + user.uid + "/" + newGroupKey).set({
            ime: groupName,
            kod: groupCode
        });
        document.getElementById("groups").innerHTML = "";
        var h4 = document.createElement('h4');
        h4.innerHTML = "Kod za pristup grupi:"
        h4.classList.add("diplay-4");
        h4.classList.add("row");
        h4.classList.add("justify-content-center");
        h4.classList.add("align-self-center");
        var h2 = document.createElement("h2");
        h2.classList.add("diplay-4");
        h2.classList.add("row");
        h2.classList.add("justify-content-center");
        h2.classList.add("align-self-center");
        h2.innerHTML = groupCode;
        document.getElementById("groups").appendChild(h4);
        document.getElementById("groups").appendChild(h2);
        document.getElementById("newGroupBtn").remove();

    } else
        alert("Niste unjeli ime grupe")
}
```

Slika 4.9. *Funkcija createGroup()*

Svaki gumb za brisanje dobiva unikatan ID koji predstavlja prilikom učitavanja podataka s Firebase baze podataka. Funkcija *deleteGroup()* (Slika 4.10.) dobiva taj ID kao argument te pomoću njega briše grupu s Firebase baze podataka.

```
function deleteGroup(id) {
    var user = firebase.auth().currentUser;
    firebase.database().ref().child("grupe").child(user.uid).child(id).remove();
    alert('Grupa uspješno obrisana.');

}
```

Slika 4.10. *Funkcija deleteGroup()*

4.1.3. Stranica povijesti anketiranja

Klikom na gumb *Povijest anketiranja* na početnoj stranici korisnik je preusmjeren na stranicu *../surveyhistory.html* koja se sastoji od popisa anketi koje je prijavljeni korisnik kreirao pomoću administratorskog sučelja. Stranica se sastoji od gumba za vraćanje na početnu stranicu te od liste anketi. Prilikom pokretanja stranica poziva se funkcija *loadSurveys()*.

Na dnu HTML koda (Slika 4.11.) ugrađene su prije spomenute funkcije te Chart.js biblioteka te skripte potrebne za njen rad.

```
<body onload="loadSurveys();" style="background-color: #f5f5f5;">

  <a class="btn btn-secondary mt-4 ml-3" href="main.html" role="button">Početna stranica</a>

  <div class="list-group m-5 w-50 mx-auto" id="surveyList"></div>

  <div class="modal fade" id="statsModal" tabindex="-1" aria-labelledby="statsLabel" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="statsLabel"> </h5>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        <div class="modal-body">
          <div id="charts"></div>
        </div>
        <div class="modal-footer">
          <div id="deleteSurveyBtn" class="mr-auto"></div>
          <button type="button" class="btn btn-secondary" data-dismiss="modal">Završi</button>
        </div>
      </div>
    </div>
  </div>

  <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-auth.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-database.js"></script>
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
    integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous">
  </script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-Piv4xVNRyMGpqkS2by6br4gNJ7DXjqk09RmUpJ8jgGtD7zP9yug3goQfGII0yAns" crossorigin="anonymous">
  </script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.4.0/chart.min.js"
    integrity="sha512-JxJpoAvmomz0MbIgw9mx+zZJLEvC6hIgQ6NcpFhVmbK1Uh5WynnRTTSGv3BTZMNBpPbocmdSJfldgV5lVnPtIw=="
    crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="../scripts/initializeFirebase.js"></script>
  <script src="../scripts/checkAuth.js"></script>
  <script src="../scripts/surveyHistory.js"></script>
</body>
```

Slika 4.11. *HTML kod stranice povijesti anketa*

Funkcija *loadSurveys()* (Slika 4.12.) učitava sve korisnikove ankete s Firebase baze podataka. Ako korisnik nije izradio niti jednu anketu, upozorava ga se na to te ga se vraća na početnu stranicu.

```
function loadSurveys() {
    var checkIfSurveysExist = false;
    document.getElementById("surveyList").innerHTML = "";
    firebase.database().ref().child("ankete").once("value", function (snapshot) {
        snapshot.forEach(function (childSnapshot) {
            var user = firebase.auth().currentUser;
            if (user.uid == childSnapshot.child("tvorac").val()) {
                var listItem = document.createElement('button');
                var title = '';
                listItem.classList.add('list-group-item');
                listItem.classList.add('list-group-item-action');
                listItem.dataset.toggle = "modal";
                listItem.dataset.target = "#statsModal";
                listItem.style = "cursor:pointer;"
                listItem.id = childSnapshot.key;
                listItem.addEventListener('click', loadStats);
                childSnapshot.child("pitanja").forEach(function (questionSnapshot) {
                    title += questionSnapshot.child("0").val();
                    title += ", ";
                });
                listItem.innerHTML = truncateString(title, 30);
                document.getElementById("surveyList").appendChild(listItem);
                checkIfSurveysExist = true;
            }
        });
        if (!checkIfSurveysExist) {
            window.location.href = '/pages/main.html'
            window.alert("Nemate niti jednu kreiranu anketu.");
        }
    });

}
```

Slika 4.12. *Funkcija loadSurveys()*

Funkcija *loadSurveys()* poziva funkciju *truncateString()* koja skraćuje dani niz znakova kako bi se naziv ankete ispravno prikazao na ekranu korisnika.

Klikom na ime ankete prikazat će se grafički rezultati te ankete. To se događa pozivom funkcije *loadStats()*. Funkcija *loadStats()* učitava podatke za traženu anketu s baze podataka te kreira tortni grafikon kako bi te podatke prikazala. Na dnu dijaloga s grafikonima nalazi se gumb za brisanje ankete s kojim je moguće ukloniti pregledanu anketu s baze podataka. Ako anketa nema rezultata korisniku će se prikazati odgovarajuća poruka.

4.2. Android aplikacija

Android aplikacija služi rješavanju anketa. Korisnik ima mogućnost prijave i pristupa grupama putem koda grupe. Također, korisnik ako nije registriran može izraditi račun ili pristupiti anketi bez prijave putem koda ankete. Aplikacija se sastoji od pet različitih aktivnosti.

4.2.1. Zaslon prijave

Na zaslonu prijave korisnik ima mogućnost prijave, odlaska na zaslon registracije ili nastavak bez registracije te se na njemu nalaze dva polja unosa za e-mail i lozinku te gumb za prijavu.

U *onCreate()* metodi (Slika 4.13.) vrši se inicijalizacija svih elemenata zaslona te se vrši provjera je li korisnik već prijavljen. Ako je korisnik prijavljen prebacuje ga se na glavni zaslon aplikacije.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_welcome_screen);
    progressBar = findViewById(R.id.progressBar);
    btnLogin = findViewById(R.id.btnLogin);
    btnLogin.setOnClickListener(this);
    tvRegister = findViewById(R.id.tvRegister);
    tvContinueWithoutRegistration = findViewById(R.id.tvContinueWithoutRegistration);
    tvContinueWithoutRegistration.setOnClickListener(this);
    tvRegister.setOnClickListener(this);
    etEmail = findViewById(R.id.etEmail);
    etPassword = findViewById(R.id.etPassword);
    mAuth = FirebaseAuth.getInstance();
    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    if (user != null && user.isEmailVerified()) {
        Intent i = new Intent( packageContext: WelcomeScreen.this, HomeScreen.class);
        i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(i);
        finish();
    }
}
```

Slika 4.13. *Metoda onCreate() aktivnosti WelcomeScreen*

U *onClick()* metodi (Slika 4.14.) definirane su radnje na klik gumba, te dva TextView-a. Ako korisnik klikne gumb *Prijava* prvo se vrši provjera je li korisnik ostavio neka od polja praznima, zatim se provjerava je li format e-maila odgovarajući na posljetku se provjerava je li unesena lozinka duga barem 6 znakova što je minimalan broj znakova za Firebase provjeru identiteta. Ako unosi u polja zadovoljavaju sve uvjete poziva se funkcija *userLogin()* (Slika 4.15.) koja dohvaća unesene znakove te ih sprema u varijable *email* i *password* tipa *String*. Nakon toga poziva se Firebase funkcija provjere identiteta *signInWithEmailAndPassword()* te joj se za argumente predaju prije spomenute varijable. Vrši se provjera je li korisnik potvrdio e-mail, ako je korisnika se prebacuje na glavni zaslon te se aktivnost prijave završava, a ako nije korisniku se na ekran ispisuje poruka putem *Toast-a* da potvrdi svoj račun te mu se šalje e-mail za potvrdu računa. Ako dođe do pogreške prijave korisnika se o tome obavijesti. Na klik TextView elemenata vrši se prebacivanje na odgovarajuće aktivnosti registracije ili nastavka bez registracije.

```java
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnLogin:
            if (etEmail.getText().toString().equals("") || etPassword.getText().toString().equals("")) {
                etEmail.setError("Neka polja su prazna");
                etPassword.setError("Neka polja su prazna");
                etEmail.requestFocus();
                etPassword.requestFocus();
            } else if (!Patterns.EMAIL_ADDRESS.matcher(etEmail.getText().toString()).matches()) {
                etEmail.setError("Provjerite e-mail adresu");
                etEmail.requestFocus();
            } else if (etPassword.getText().toString().length() < 6) {
                etPassword.setError("Molimo koristite lozinku dugu barem 6 znakova");
                etPassword.requestFocus();
            } else
                userLogin();
            break;

        case R.id.tvRegister:
            startActivity(new Intent( packageContext: WelcomeScreen.this, RegisterActivity.class));
            break;
        case R.id.tvContinueWithoutRegistration:
            Intent i = new Intent(getApplicationContext(), EnterSurveyKeyActivity.class);
            startActivity(i);
    }
}
```

Slika 4.14. *Metoda onClick aktivnosti WelcomeScreen*

```java
private void userLogin() {
    String email = etEmail.getText().toString().trim();
    String password = etPassword.getText().toString().trim();
    progressBar.setVisibility(View.VISIBLE);
    mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(task -> {
        if (task.isSuccessful()) {
            FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
            if (user.isEmailVerified()) {
                progressBar.setVisibility(View.GONE);
                startActivity(new Intent( packageContext: WelcomeScreen.this, HomeScreen.class));
                finish();
            } else {
                user.sendEmailVerification();
                Toast.makeText( context: WelcomeScreen.this, ("Provjerite e-mail da potvrdite svoj račun"), Toast.LENGTH_LONG).show();
            }
        } else {
            Toast.makeText( context: WelcomeScreen.this, ("Greška u logiranju"), Toast.LENGTH_LONG).show();
            progressBar.setVisibility(View.GONE);
        }
    });
}
```

Slika 4.15. *Funkcija userLogin()*

4.2.2. Zaslon registracije

Zaslon registracije sastoji se od tri polja unosa za e-mail, lozinku te ponovljenu lozinku, gumba za registraciju te TextView elementa za povratak na zaslon prijave. U *onCreateO* metodi vrši se inicijalizacija svih elemenata zaslona.

U *onClick* metodi definirane su radnje na klik gumba, te TextView-a. Ako korisnik klikne gumb *Registracija* prvo se vrši provjera je li korisnik ostavio neka od polja praznima, zatim se provjerava je li format e-maila odgovarajući, nakon toga se provjerava jesu li unesene lozinke iste te na posljetku se provjerava je li unesena lozinka duga barem 6 znakova. Ako unosi u polja zadovoljavaju sve uvjete poziva se funkcija *registerUser()* koja dohvaća unesene znakove te ih sprema u varijable *email* i *password* tipa *String*. Nakon toga poziva se Firebase funkcija registracije *createUserWithEmailAndPassword()* te joj se za argumente predaju prije spomenute varijable. Nakon uspješne registracije korisnika se prebacuje na ekran prijave. Na klik TextView elemenata vrši se prebacivanje na aktivnost prijave. Funkcija *registerUser()* prikazana je na slici 4.16.

```
private void registerUser() {
    String email = etEmail.getText().toString().trim();
    String password = etPassword.getText().toString().trim();

    progressBar.setVisibility(View.VISIBLE);
    mAuth.createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener(task -> {
                if (task.isSuccessful()) {

                    Toast.makeText( context: RegisterActivity.this, ("Korisnik uspješno registriran"), Toast.LENGTH_LONG).show();
                    progressBar.setVisibility(View.GONE);
                    etEmail.getText().clear();
                    etPassword.getText().clear();
                    etPassword2.getText().clear();
                    startActivity(new Intent( packageContext: RegisterActivity.this, WelcomeScreen.class));
                    finish();
                } else {
                    Toast.makeText( context: RegisterActivity.this, ("Greška u registriranju, pokušajte ponovno"), Toast.LENGTH_LONG).show();
                    progressBar.setVisibility(View.GONE);
                    etPassword.getText().clear();
                    etPassword2.getText().clear();
                }
            });
}
```

Slika 4.16. *Funkcija registerUser()*

4.2.3. Početni zaslon

Na početnom zaslonu aplikacije nalazi se RecyclerView koji sadrži popis svih anketa dostupnih korisniku koje se pokreću kada korisnik na njih klikne. Ankete se dohvaćaju iz grupa u kojima je korisnik prijavljen. Korisnik ima mogućnost odjave iz padajućeg izbornika klikom na gumb *Odjava,* kod koji se poziva na klik gumba prikazan je na slici 4.18. S lijeve strane nalazi se navigacijsko okno pomoću kojega se korisnik može prebaciti na fragment upravljanja grupama. Izgled stranice opisan je XML-om koji je prikazan na slici 4.17.

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$Scrolli..."
    tools:showIn="@layout/app_bar_home_screen">

    <fragment
        android:id="@+id/nav_host_fragment_content_home_screen"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Slika 4.17. *XML opis sadržaja početnog zaslona*

```java
case R.id.action_logout:
    FirebaseAuth.getInstance().signOut();
    Intent i = new Intent( packageContext: HomeScreen.this, WelcomeScreen.class);
    i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(i);
    finish();
```

Slika 4.18. *Odjava korisnika na odabir opcije Odjava*

U metodi *onCreateView()* fragmenta *HomeFragment* inicijaliziran je raspored zaslona te se učitavaju podatci o riješenim anketama iz lokalne baze podataka pomoću funkcije *loadData()* (Slika 4.19.). Tako je osigurano da korisnik ne rješava istu anketu više puta bez da zapisujemo podatke o korisniku na Firebase bazu podataka.

```java
private void loadData() {
    SharedPreferences sharedPreferences = getContext().getSharedPreferences( name: "shared preferences", MODE_PRIVATE);
    Gson gson = new Gson();
    String json = sharedPreferences.getString( key: "surveys",  defValue: null);
    Type type = new TypeToken<ArrayList<String>>() {
    }.getType();
    solvedSurveysArrayList = gson.fromJson(json, type);
    if (solvedSurveysArrayList == null) {
        solvedSurveysArrayList = new ArrayList<>();
        solvedSurveysArrayList.add("");
    }
}
```

Slika 4.19. *Funkcija loadData()*

U metodi *onViewCreated()* pokreću se funkcije *setupRecycler()* i *setupRecyclerData()*. Funkcija *setupRecycler()* inicijalizira RecyclerView i njegov pripadajući adapter. Funkcija *setupRecyclerData()* (Slika 4.20.) učitava podatke s Firebase baze podataka. Prvo se dohvaćaju sve grupe u koje je korisnik prijavljen. Nakon toga se vrši provjera je li anketa objavljena za korisnike ikoje od grupa u kojima se korisnik nalazi. Ako je anketa objavljena u grupi u kojoj je korisnik prelazi se na provjeru je li korisnik već riješio tu anketu te ako nije ime i grupa ankete se dodaju u pripadajuće liste te se dodaju u RecyclerView putem funkcije *addData()* koja je definirana u adapteru. Ako korisnik nema niti jednu dostupnu anketu u danom trenutku prikazuje mu se odgovarajuća poruka.

```
private void setupRecyclerData() {
    userGroups = new ArrayList<>();
    List<String> surveyNames = new ArrayList<>();
    List<String> surveyGroups = new ArrayList<>();
    FirebaseDatabase.getInstance().getReference( path: "korisnici").child(FirebaseAuth.getInstance().getCurrentUser().getUid())
            .addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    for (DataSnapshot groupSnapshot : snapshot.getChildren()) {
                        userGroups.add(groupSnapshot.child("ime").getValue().toString());
                    }
                    FirebaseDatabase.getInstance().getReference( path: "ankete")
                            .addListenerForSingleValueEvent(new ValueEventListener() {
                                @Override
                                public void onDataChange(@NonNull DataSnapshot snapshot) {
                                    for (DataSnapshot surveysSnapshot : snapshot.getChildren()) {

                                        for (DataSnapshot surveyGroupsSnapshot : surveysSnapshot.child("odabraneGrupe").getChildren()) {
                                            for (int i = 0; i < userGroups.size(); i++) {

                                                if (surveyGroupsSnapshot.getValue().toString().equals(userGroups.get(i))) {
                                                    Boolean check;
                                                    check = true;
                                                    for (int j = 0; j < solvedSurveysArrayList.size(); j++) {
                                                        if (surveysSnapshot.getKey().equals(solvedSurveysArrayList.get(j))) {
                                                            check = false;
                                                        }
                                                    }
                                                    if (check) {
                                                        surveyNames.add("Anketa: " + surveysSnapshot.child("kod").getValue().toString());
                                                        surveyGroups.add("Grupa: " + userGroups.get(i));
                                                    }
                                                }
                                            }
                                        }
                                    }
                                    surveyRecyclerAdapter.addData(surveyNames, surveyGroups);
                                    if (surveyNames.size() == 0) {
                                        Snackbar.make(getActivity().findViewById(android.R.id.content),  text: "Nemate niti jednu anketu u svojim grupama", Snackbar.LENGTH_LONG).show();
                                    }
                                }
```

Slika 4.20. *Funkcija setupRecyclerData()*

4.2.4. Zaslon upravljanja grupama

Prelaskom na *Upravljanje grupama* otvara se zaslon upravljanja grupama gdje se korisnik može pridružiti nekoj od grupa putem koda grupe ili izlaziti iz grupa. Na gornjem dijelu zaslona nalazi se polje unosa za kod grupe kojoj se korisnik želi pridružiti te pripadajući gumb za pridruživanje ispod polja unosa. Ispod dijela za pridruživanje nalazi se RecyclerView sa svim grupama korisnika na koje kada klikne pojavljuje se dijalog koji mu nudi mogućnost izlaska iz njih što je definirano u *GroupClick()* metodi (Slika 4.21.).

```
public void GroupClick(int position) {
    AlertDialog.Builder alertBuilder = new AlertDialog.Builder(getContext());
    alertBuilder.setTitle("Uklanjanje grupe");
    alertBuilder.setMessage("Želite li ukloniti ovu grupu?");
    alertBuilder.setPositiveButton( text: "Da", (dialog, which) -> FirebaseDatabase.getInstance().getReference( path: "korisnici").child(FirebaseAuth.getInstance()
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            for (DataSnapshot groupSnapshot : snapshot.getChildren()) {
                if (userGroupCodes.get(position).equals(groupSnapshot.child("kod").getValue().toString())) {
                    groupSnapshot.getRef().removeValue();
                    Snackbar.make(getActivity().findViewById(android.R.id.content),  text: "Grupa uspješno uklonjena.", Snackbar.LENGTH_LONG).show();
                    setupRecyclerData();
                }
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    }));
    alertBuilder.setNegativeButton( text: "Ne", (dialog, which) -> {

    });
    alertBuilder.show();
}
```

Slika 4.21. *Metoda GroupClick()*

Prilikom prijave korisnika u grupu vrši se provjera postoji li takva grupe te je li korisnik već član te grupe te ako nije u bazu podataka se sprema ime i kod grupe te se RecyclerView ponovo postavlja s novim podacima što je prikazano slikom 4.22.

```
for (DataSnapshot creatorsSnapshot : snapshot.child("grupe").getChildren()) {
    for (DataSnapshot groupSnapshot : creatorsSnapshot.getChildren()) {
        if (groupSnapshot.child("kod").getValue().toString().equals(groupCode)) {
            check = true;
            groupName = groupSnapshot.child("ime").getValue().toString();
        }
    }
}
if (check) {
    for (int i = 0; i < userGroupCodes.size(); i++) {
        if (groupCode.equals(userGroupCodes.get(i)))
            checkIfUserAlreadyBelongsToGroup = true;
    }
} else {
    Snackbar.make(getView(),  text: "Nepostojeća grupa.", Snackbar.LENGTH_LONG).show();
}
if (checkIfUserAlreadyBelongsToGroup) {
    Snackbar.make(getView(),  text: "Već ste član te grupe.", Snackbar.LENGTH_LONG).show();
} else if (check) {
    snapshot.getRef().child("korisnici").child(FirebaseAuth.getInstance().getCurrentUser().getUid()).child(pushId).child("ime").setValue(groupName).addOnCompleteListener(task -> {
    });
    snapshot.getRef().child("korisnici").child(FirebaseAuth.getInstance().getCurrentUser().getUid()).child(pushId).child("kod").setValue(groupCode).addOnCompleteListener(task -> {
    });
    Snackbar.make(getActivity().findViewById(android.R.id.content),  text: "Uspješno ste se pridružili grupi", Snackbar.LENGTH_LONG).show();
}
etGroupCode.getText().clear();
setupRecyclerData();
```

Slika 4.22. *Dio metode setOnClickListener() gumba za pridruživanje grupi*

4.2.5. Zaslon unosa koda

Zaslon unosa koda pokreće se ako korisnik sa zaslona prijave odabere opciju *Nastavi bez registracije* ili ako korisnik pritisne gumb u donjem desnom uglu na početnom zaslonu. Zaslon unosa koda omogućuje pristup anketi putem koda ankete. Sastoji se od polja za unos koda ankete i gumba za pristup anketi. Na pritisak gumba iz polja unosa iščitava se kod grupe te ga se predaje sljedećoj aktivnosti putem *Intent-a* (Slika 4.23.).

```
btnStartSurvey.setOnClickListener(v -> {
    Intent i = new Intent(getApplicationContext(), SurveyPrepActivity.class);
    String id = etSurveyCode.getText().toString().trim();
    if (!id.equals("")) {
        i.putExtra( name: "surveyKey", id);
        startActivity(i);
    } else {
        etSurveyCode.setError("Neka polja su prazna");
        etSurveyCode.requestFocus();
    }
});
```

Slika 4.23. *Metoda setOnClickListener() gumba za pristup anketi*

4.2.6. Zaslon pripreme za anketu

Na ovome zaslonu nalazi se informacija koliko vremena anketa traje te gumb za početak ankete. U aktivnosti zaslona pripreme za anketu vrši se provjera postoji li anketa s kodom ankete predanom putem *Intent-a* te provjera je li korisnik već riješio anketu s tim kodom. Ako je korisnik već riješio tu anketu ili anketa ne postoji prikazuje mu se odgovarajuća poruka. Kod učitavanja podataka o anketi prikazan je na slici 4.24.

```
for (DataSnapshot surveysSnapshot : snapshot.getChildren()) {
    if (surveysSnapshot.child("kod").getValue().equals(code)) {
        tvSurveyName.setText("Anketa: " + surveysSnapshot.child("kod").getValue().toString());
        tvSurveySolvingTime.setText("Vrijeme rješavanja ove ankete je: \n" + surveysSnapshot.child("trajanje").getValue().toString() + " min");
        solvingTimeMins = Integer.parseInt(surveysSnapshot.child("trajanje").getValue().toString());
        id = surveysSnapshot.getKey();
        check = true;

        for (int i = 0; i < solvedSurveysArrayList.size(); i++) {
            if (id.equals(solvedSurveysArrayList.get(i))) {
                checkIfAlreadySolved = true;
            }
        }
    }
}
if (!check) {
    tvSurveyName.setVisibility(View.GONE);
    tvSurveySolvingTime.setVisibility(View.GONE);
    btnStartSurvey.setVisibility(View.GONE);
    Snackbar.make(getWindow().getDecorView().getRootView(), text: "Nepostojeća anketa.", Snackbar.LENGTH_LONG).show();
}
if (checkIfAlreadySolved) {
    tvSurveyName.setVisibility(View.GONE);
    tvSurveySolvingTime.setVisibility(View.GONE);
    btnStartSurvey.setVisibility(View.GONE);
    Snackbar.make(getWindow().getDecorView().getRootView(), text: "Ova anketa je već riješena.", Snackbar.LENGTH_LONG).show();
}
```

Slika 4.24. *Dio funkcije setupData()*

4.2.7. Zaslon anketiranja

Zaslon anketiranja sastoji se od TextView-a na vrhu zaslona koji predstavlja preostalo vrijeme rješavanja ankete u sekundama, od TextView-a ispod prvog koji predstavlja trenutno pitanje u anketi te od RecyclerView-a u kojemu se nalaze ponuđeni odgovori za trenutno pitanje u anketi. Pritiskom na jedan od ponuđenih odgovora, odgovor se sprema u Firebase bazu podataka te se

prikazuje sljedeće pitanje u anketi. Na kraju ankete prikazuje se ekran zahvale korisniku na rješavanju. Anketa staje onoga trenutka kada korisnik riješi zadnje pitanje u anketi ili kada istekne vrijeme rješavanja. Na klik odgovora dohvaća se prijašnjih broj odabira tog odgovora te se ažurira vrijednošću većom za jedan. Brojač se uvećava te se provjerava je li došao do kraja polja, ako je dolazi do kraja ankete a ako nije poziva se funkcija *setupData()* (Slika 4.26.) koja pomoću brojača učitava sljedeće pitanje iz polja dohvaćenih iz Firebase baze podataka. Slikom 4.25. prikazana je metoda klika na odgovor ankete.

```
@Override
public void AnswerClick(int position) {

    FirebaseDatabase.getInstance().getReference( path: "ankete").child(id).child("pitanja").child(Integer.toString(counter))
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            previousValue = snapshot.getValue(Integer.class);
            snapshot.getRef().setValue(previousValue += 1).addOnCompleteListener(task -> {
                counter++;
                if (counter < questionsList.size()) {
                    setupData();
                } else {
                    Intent i = new Intent(getBaseContext(), SurveyCompleteSplash.class);
                    i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    i.putExtra( name: "loggedIn", loggedIn);
                    i.putExtra( name: "surveyID", id);
                    startActivity(i);
                    finish();
                }

            });
        }
```

Slika 4.25. *Dio metode AnswerClick()*

```
private void setupData() {
    tvQuestion.setText(questionsList.get(counter));
    answersRecyclerAdapter.addData(answersList.get(counter));
}
```

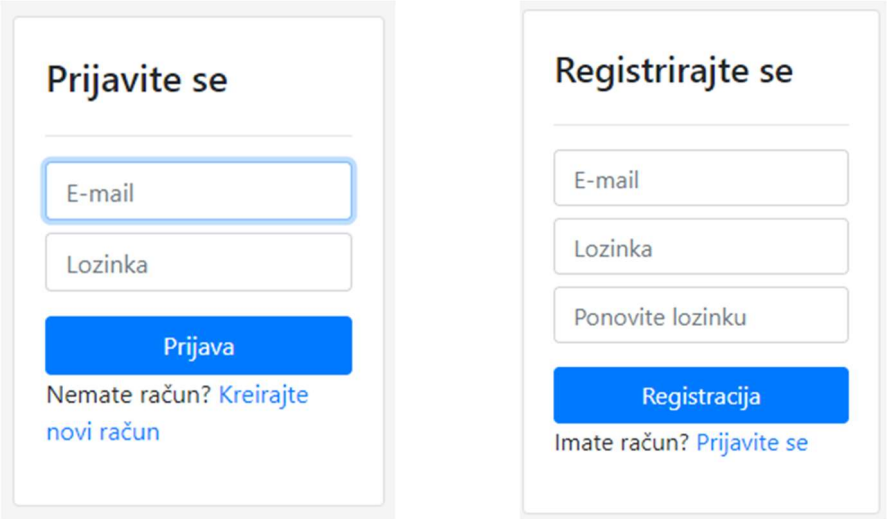
Slika 4.26. *Funkcija setupData()*

5. PRIMJENA APLIKACIJE I ADMINISTRATORSKOG SUČELJA

U ovome poglavlju opisan je izgled i primjena programskog rješenja zadatka ovog završnog rada.

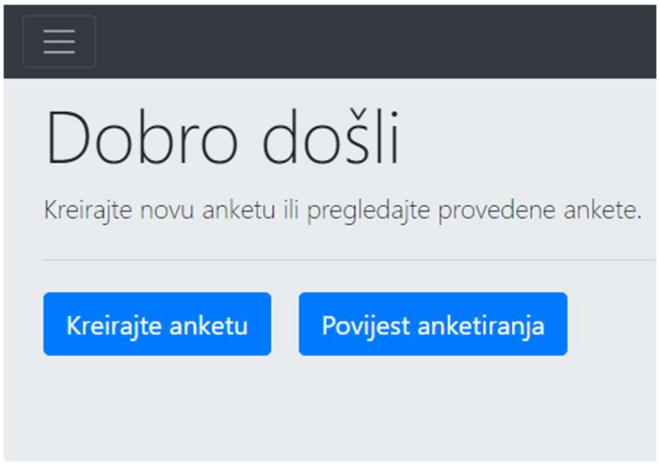
5.1. Primjena administratorskog sučelja

Korisnika dolaskom na stranicu dočekuje kartica za prijavu. Ako korisnik nema račun može odabrati opciju *Kreirajte novi račun* kojom mu se prikazuje kartica za registraciju. Izgled kartica za prijavu i registraciju prikazan je slikom 5.1.



Slika 5.1. *Izgled kartica za prijavu i registraciju*

Nakon uspješne prijave korisnika se preusmjerava na glavnu stranicu administratorskog sučelja iz kojega može upravljati grupama, odjaviti se, kreirati anketu, ili preusmjeriti se na stranicu povijesti provedenih anketa (Slika 5.2.).

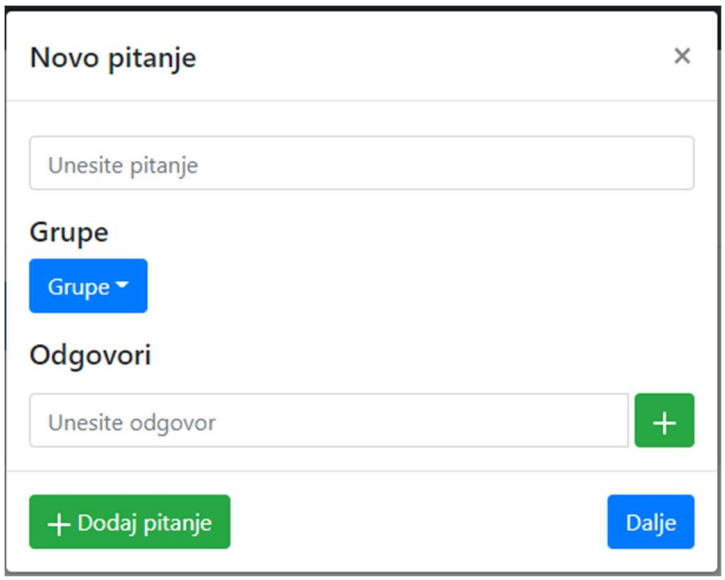


Slika 5.2. *Izgled glavne stranice*

U gornjem lijevom uglu nalazi se navigacijsko okno koje se otvara na klik gumba, iz njega korisnik ima mogućnosti otvaranja dijaloga za upravljanje grupama i odjave (Slika 5.3.).

Slika 5.3. *Sadržaj padajućeg izbornika navigacije*

Odabirom opcije *Kreirajte anketu* otvara se dijalog s poljima unosa pitanja i odgovora (Slika 5.4.). Korisnik klikom na zeleni + gumb dodaje novi odgovor a klikom na zeleni + *Dodaj pitanje* dodaje novo pitanje u anketu. Korisnik iz izbornika ima mogućnost biranja grupa kojima će anketa koju izrađuje biti dostupna.



Slika 5.4. *Prvi dijalog kreiranja ankete*

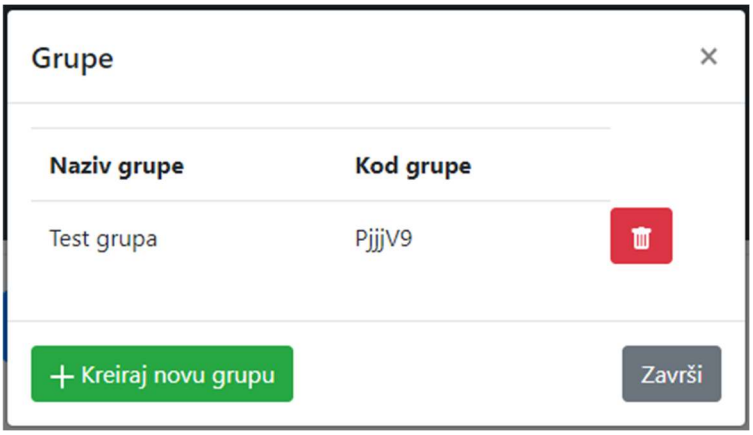
Klikom na gumb *Dalje* otvara se sljedeći dijalog u kojemu korisnik unosi vrijeme rješavanja ankete u minutama (Slika 5.5.).

Slika 5.5. *Drugi dijalog kreiranja ankete*

Nakon što je sve unio klikom na *Završi* dovršava proces izrade ankete te mu se na ekranu pokaže kod kojim korisnici mogu pristupiti anketi (Slika 5.6.).

Slika 5.6. *Prikaz koda za pristup anketi*

Odabirom opcije *Upravljanje grupama* iz padajućeg izbornika navigacije otvara se dijalog u kojemu korisnik vidi sve svoje grupe i pristupne kodove te može izrađivati nove grupe ili brisati stare (Slika 5.7.).

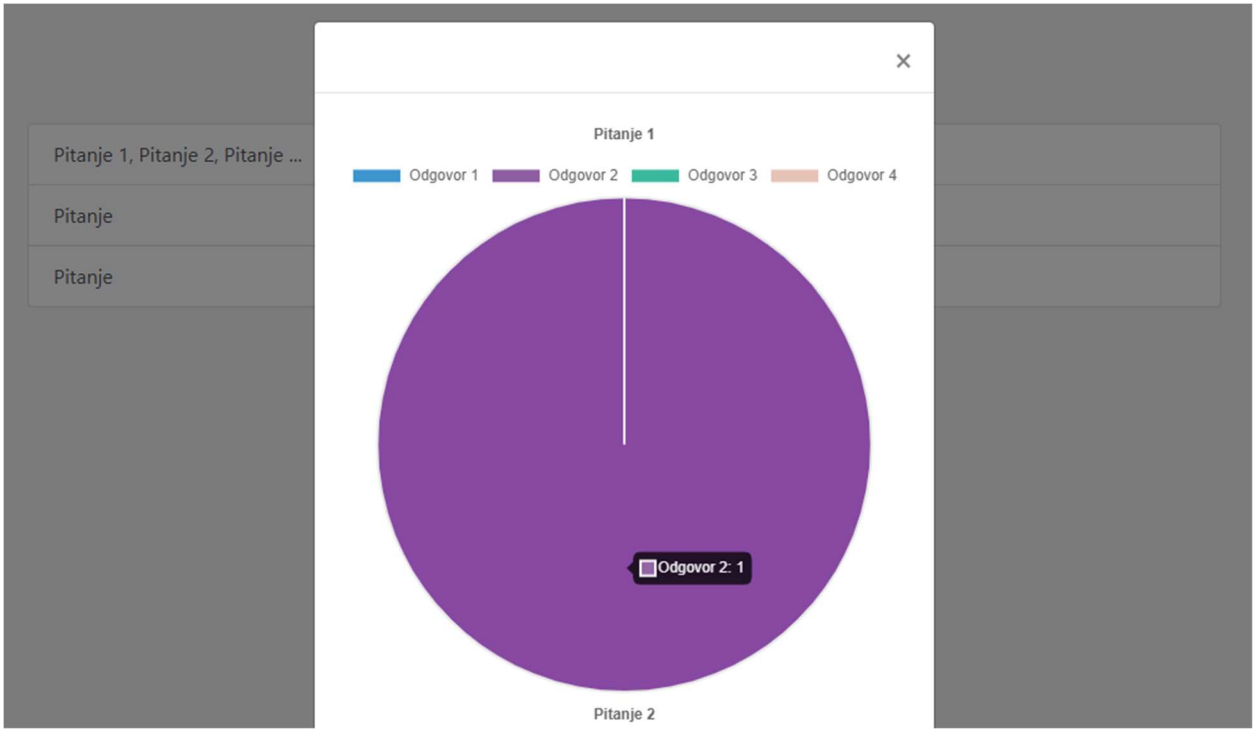


Slika 5.7. *Izgled dijaloga za upravljanje grupama*

Pritiskom na gumb *Povijest anketiranja* s početnog zaslona korisnika se preusmjerava na novu stranicu na kojoj su izlistane sve ankete koje je korisnik izradio (Slika 5.8.). Ako ih nema, korisnika se o tome obavještava te ga se preusmjerava nazad na početni zaslon. U gornjem desnom uglu nalazi se gumb za povratak na početni zaslon. Klikom na jednu od ponuđenih anketa otvara se grafički prikaz rezultata ankete.

Slika 5.8. *Izgled stranice povijesti anketa*

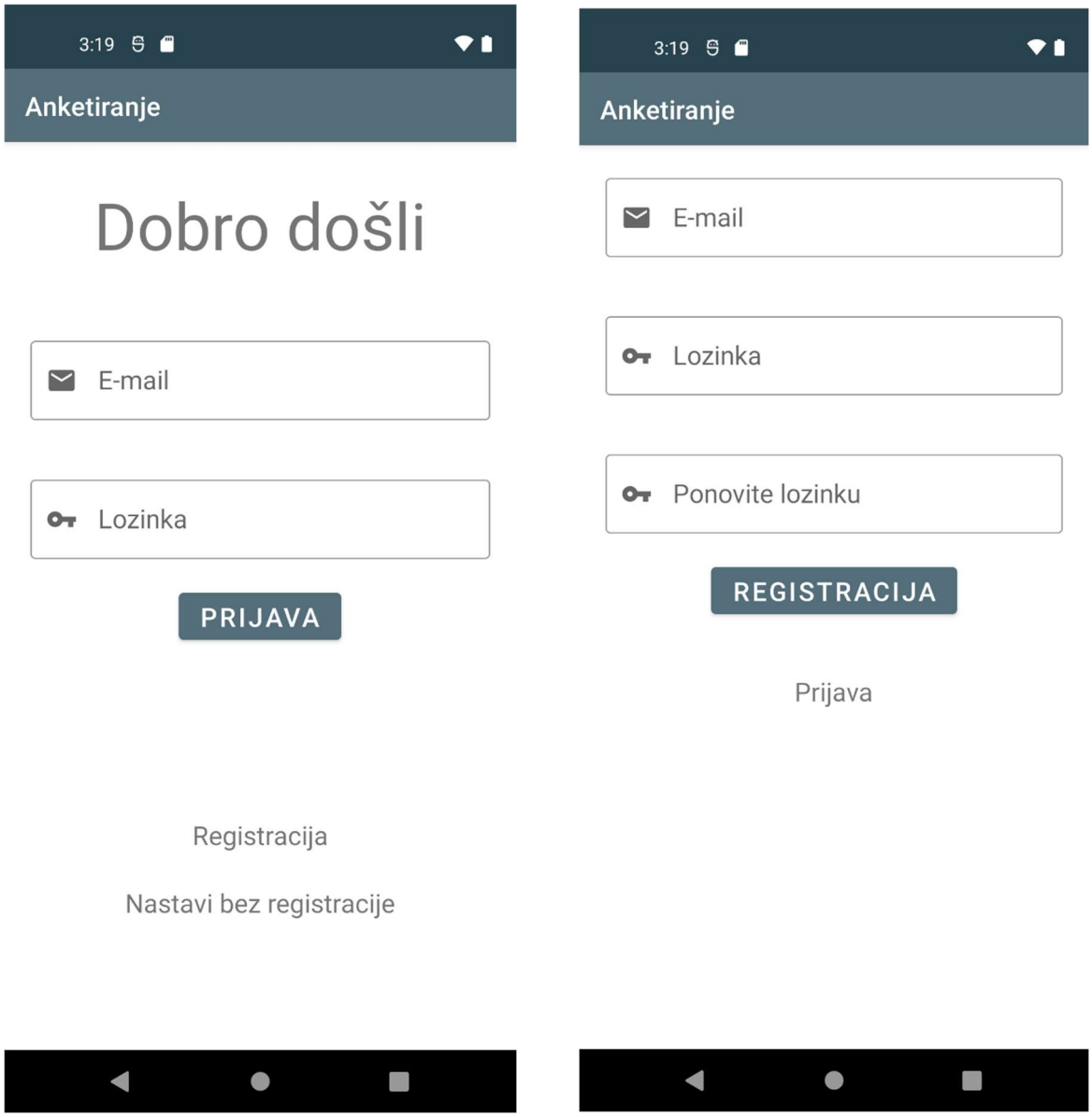
Grafički prikaz rezultata ankete sastoji se od postavljenog pitanja na vrhu, ponuđenih odgovora ispod te od tortnog grafikona koji prikazuje omjer odabranih odgovora (Slika 5.9.). Na dnu ispod grafičkih rezultata nalazi se gumb za brisanje odabrane ankete.



Slika 5.9. *Izgled grafičkog prikaza rezultata ankete*

5.2. Primjena Android aplikacije

Prilikom prvog paljenja aplikacije korisnika dočeka zaslon prijave s kojega se može prijaviti, prijeći na zaslon registracije ili prijeći na zaslon pristupa anketi bez prijave. Ako korisnik odabere opciju *Registracija* prelazi na zaslon registracije. Izgled zaslona prijave i registracije prikazan je na slici 5.10.

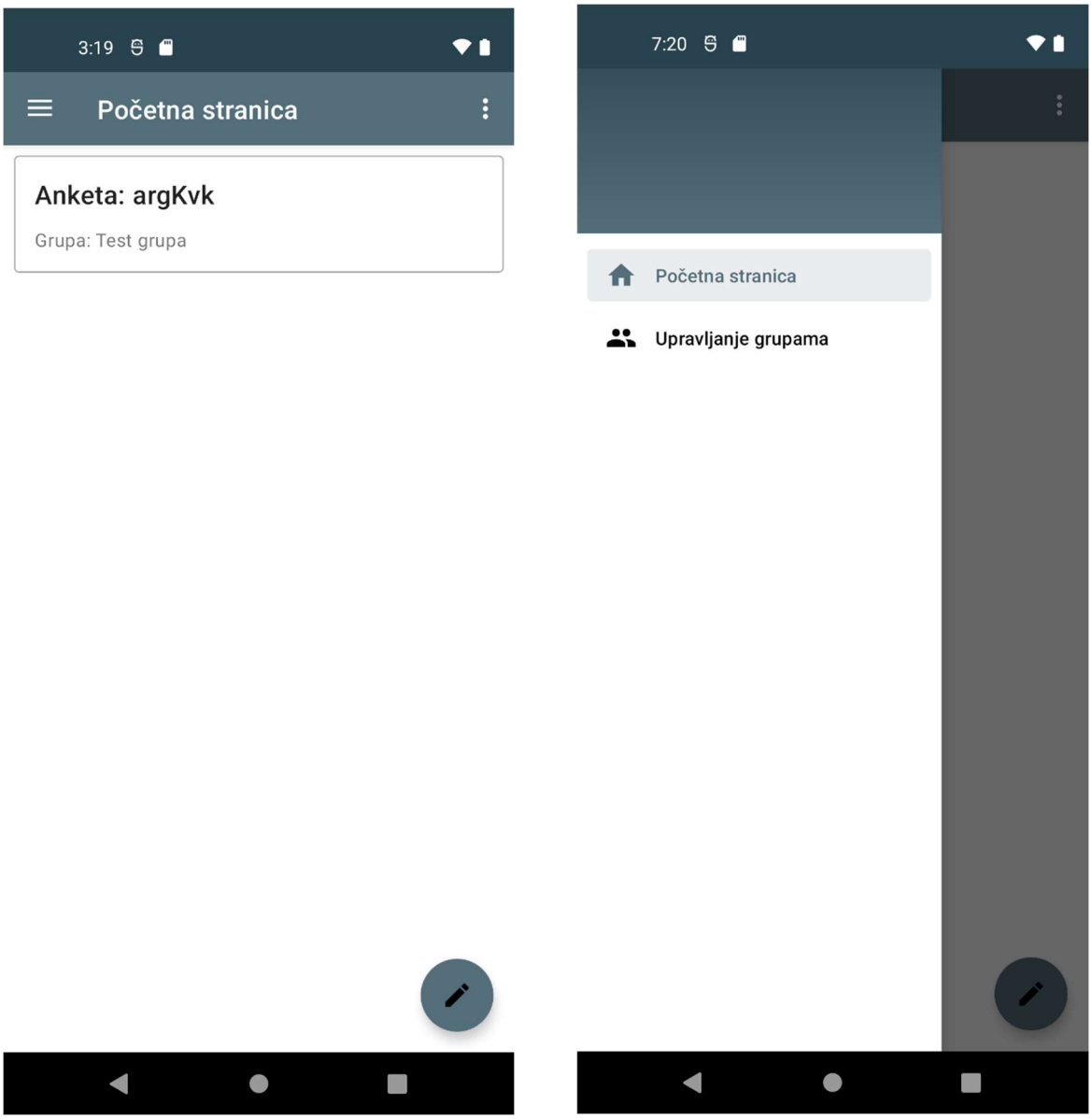


Slika 5.10. *Izgled zaslona prijave i zaslona registracije*

Klikom na *Nastavi bez registracije* otvara se zaslon unosa koda u kojemu korisnik unosi kod ankete generiran u administratorskom sučelju (Slika 5.11.). Unosom koda i klikom na *Pristupite anketi* korisnik bez prijave pristupa anketi.

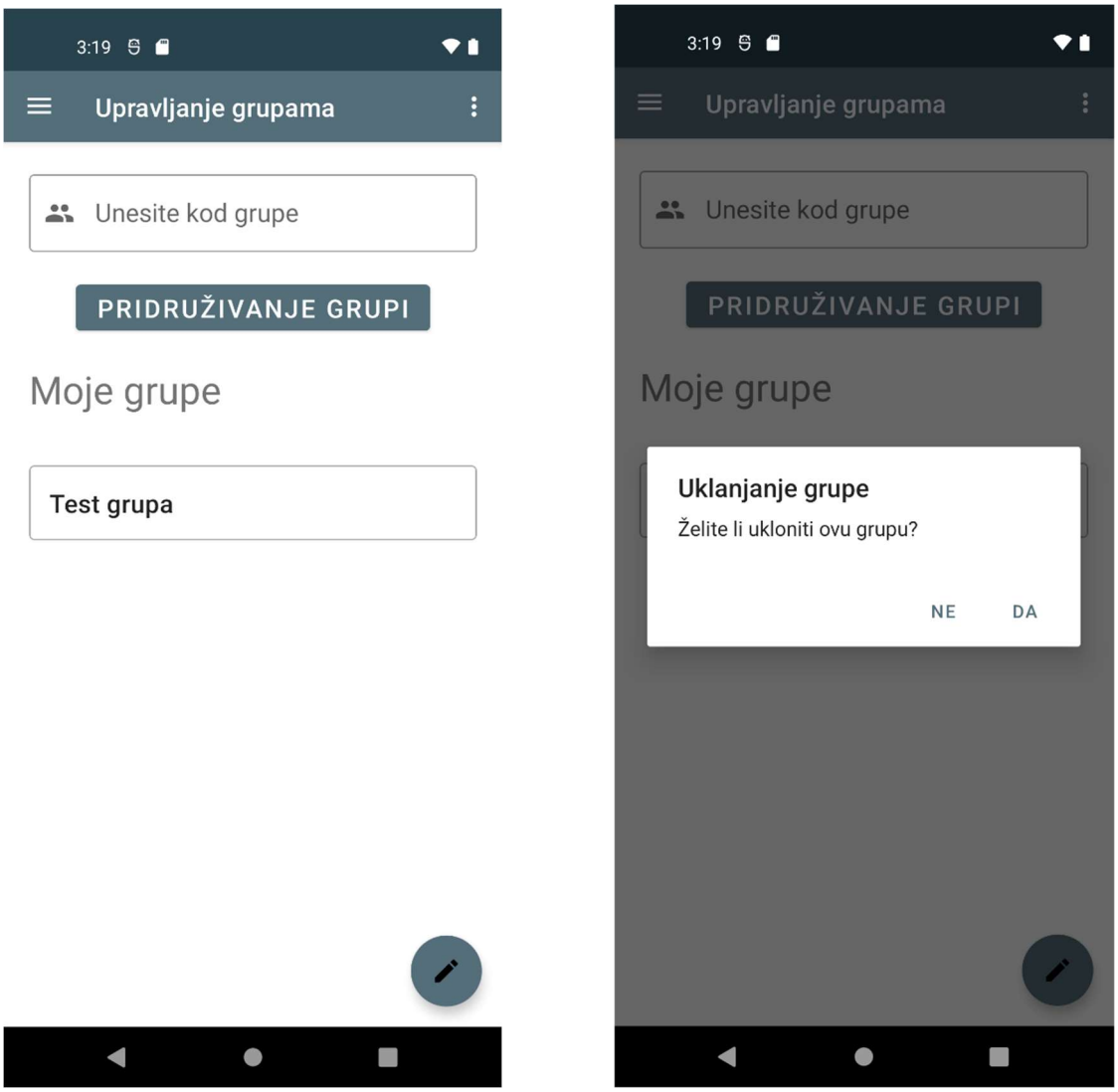
Slika 5.11. *Izgled zaslona unosa koda*

Ako se korisnik prijavi prebacuje ga se na početni zaslon aplikacije gdje vidi sve ankete njemu dostupne putem njegovih grupa. Anketu pokreće klikom na jednu od ponuđenih anketa. Ako nema niti jednu dostupnu anketu, anketi može pristupiti putem gumba u donjem desnom uglu koji otvara zaslon unosa koda. Korisnik može upravljati grupama odabirom opcije *Upravljanje grupama* iz navigacijskog okna koje se nalazi s lijeve strane i može se otvoriti povlačenjem s lijeva na desno ili klikom na gumb u gornjem lijevom uglu. Izgled početnog zaslona i navigacijskog okna prikazan je na slici 5.12.



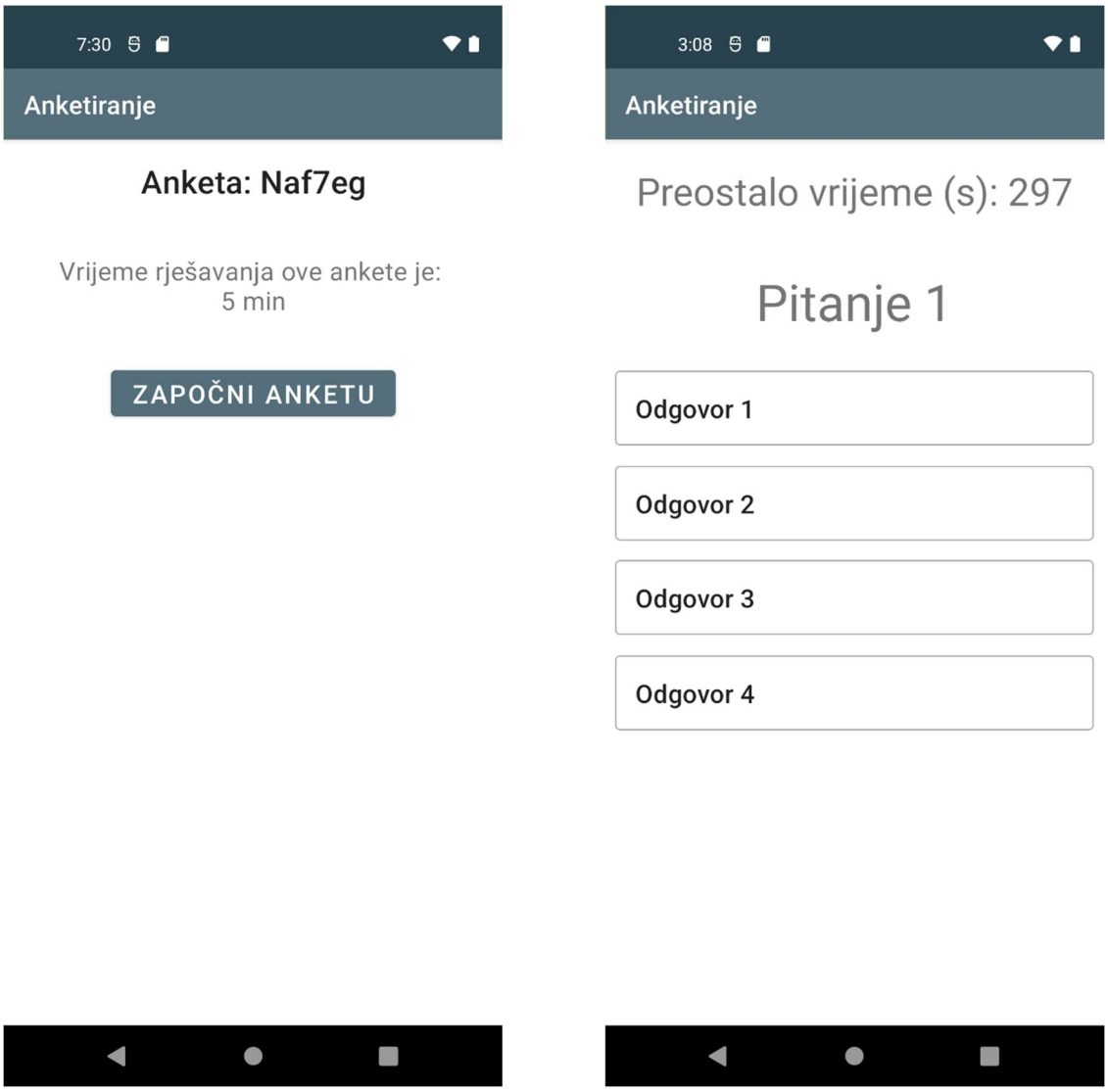
Slika 5.12. *Izgled početnog zaslona i navigacijskog okna*

Na zaslonu upravljanja grupama korisnik se može pridružiti novoj grupi ili izaći iz grupe u koju je prijavljen. Klikom na grupu otvara se dijalog koji provjerava želi li korisnik stvarno obrisati grupu te ako odgovori potvrdno, uklanja grupu iz popisa svojih grupa. Izgled zaslona za upravljanje grupama i dijaloga za uklanjanje grupe prikazan je na slici 5.13.



Slika 5.13. *Izgled zaslona za upravljanje grupama i dijaloga za uklanjanje grupe*

Nakon što korisnik pristupi anketi putem početnog zaslona ili putem zaslona unosa koda prikazuje mu se vrijeme trajanja ankete te gumb za početak ankete kojim pokreće anketu. Zaslon anketiranja se sastoji od brojača na vrhu koji prikazuje preostalo vrijeme rješavanja i od pitanja i ponuđenih odgovora. Korisnik klikom na odgovor prelazi na sljedeće pitanje, odnosno ako je došao do kraja ankete završava anketu. Anketa također staje istekom vremena rješavanja. Izgled zaslona pripreme za anketu i zaslona anketiranja prikazan je na slici 5.14.



Slika 5.14. *Izgled zaslona pripreme za anketu i zaslona anketiranja*

6. ZAKLJUČAK

Zadatak ovog završnog rada bio je izraditi web administratorsko sučelje te mobilnu aplikaciju za anketiranje u stvarnom vremenu. Tijekom izrade korištene su razne moderne tehnologije te su aplikacija i administratorsko sučelje izrađene sa što lakšim korištenjem na umu. Administratoru je jednostavno kreirati i pregledati rezultate anketa, također ima i mogućnost izrade grupa kako bi što lakše došao do krajnjih korisnika koji će dati povratnu informaciju o temi ankete. Administrator u stvarnom vremenu dobiva rezultate koji su prikazani u grafičkom obliku u administratorskom sučelju.

Mobilna aplikacija izrađena je s privatnošću na umu, s obzirom na to da se rezultati anketa ne vežu za pojedinog korisnika, krajnji korisnici mogu bezbrižno rješavati anketu. Time administrator može očekivati veći broj i kvalitetnije rezultate jer će krajnji korisnici biti otvoreniji i spremniji odgovoriti na pitanja.

Rješenje ovog završnog rada može se proširiti dodatnim mogućnostima i poboljšanjima, npr. različiti tipovi odgovora (unos korisnika putem tipkovnice) te vizualna poboljšanja web administratorskog sučelja i mobilne aplikacije. Također moguća je i izrada iOS aplikacije koja bi imala iste funkcionalnosti kao i Android verzija.

LITERATURA

[1] M. Levar, Izrada e-kolegija, CARNet, [Mrežno], dostupno na: https://loomen.carnet.hr/pluginfile.php/4517773/mod_resource/content/1/Priru%C4%8Dnik%20za%20izradu%20e-kolegija.pdf/. [Datum zadnje posjete stranici: 10 Kolovoz 2021.].

[2] Kahoot!, Kahoot-for-schools-solution-brief-August-2019.pdf, [Mrežno], dostupno na: https://kahoot.com/files/2018/09/Kahoot-for-schools-solution-brief-August-2019.pdf. [Datum zadnje posjete stranici: 10 Kolovoz 2021.].

[3] A. Amany, Kahoot! as an Engaging Game-based Learning Tool, [Mrežno], dostupno na: https://educationalgamedesign.com/kahoot-as-an-engaging-game-based-learning-tool.html. [Datum zadnje posjete stranici: 10 Kolovoz 2021.].

[4] CARNet, Uvod u HTML, [Mrežno], dostupno na: https://tesla.carnet.hr/mod/book/view.php?id=5430&chapterid=885. [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

[5] Bootstrap, Introduction, [Mrežno], dostupno na: https://getbootstrap.com/docs/5.1/getting-started/introduction/. [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

[6] D. Stančer, Osnove JavaScripta, [Mrežno], dostupno na: https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501_polaznik.pdf. [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

[7] Chart.js, [Mrežno], dostupno na: https://www.chartjs.org/. [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

[8] Documentation for Visual Studio Code, [Mrežno], dostupno na: https://code.visualstudio.com/docs. [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

[9] Wikipedija, Firebase, Wikipedija, Slobodna enciklopedija, [Mrežno], dostupno na: https://en.wikipedia.org/wiki/Firebase. [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

[10] Wikipedija, Java (programski jezik), Wikipedija, Slobodna enciklopedija, [Mrežno], dostupno na: https://hr.wikipedia.org/wiki/Java_(programski_jezik). [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

[11] Android, Meet Android Studio [Mrežno], dostupno na: https://developer.android.com/studio/intro. [Datum zadnje posjete stranici: 11 Kolovoz 2021.].

SAŽETAK

U završnom radu prikazana je izrada aplikacije za mobilno anketiranje u stvarnom vremenu. Aplikacija se sastoji od web administratorskog sučelja, te od mobilne aplikacije za anketiranje. Izrađeno je administratorsko sučelje putem kojega korisnik može izraditi anketu, pregledati rezultate provedenih anketa i izraditi te upravljati grupama. Prilikom izrade ankete definiraju se pitanja, odgovori te vremensko ograničenje ankete. Također korisnik prilikom izrade ankete može odabrati grupe za koje će ta anketa biti dostupna. Mobilna aplikacija služi prikazivanju dostupnih anketa te rješavanju istih. Korisnik ima mogućnost pristupa anketi putem koda, ili putem grupa u kojima je anketa objavljena. Izvorni kod web administratorskog sučelja pisan je u HTML i JavaScript jezicima a uređen je u programu Visual Studio Code. Izvorni kod mobilne aplikacije pisan je u Java programskom jeziku a uređen je u programu Android Studio. Svi podaci o anketama i pripadajućim rezultatima spremaju se na Firebase bazu podataka u stvarnom vremenu.

Ključne riječi: Android aplikacija, Anketiranje, Firebase, Java, Web aplikacija

ABSTRACT

Mobile application for real-time survey

The final paper presents the development of a real-time mobile survey application. The application consists of a web administrator interface and a mobile survey application. An administrator interface has been created through which the user can create a survey, view the results of conducted surveys and create and manage groups. When creating a survey, questions, answers and the time limit of the survey are defined. Also, when creating a survey, the user can select the groups for which the survey will be available. The mobile application is used for displaying and solving available surveys. The user has the option of accessing the survey via survey code, or through the groups in which the survey was published. The source code of the web administrator interface is written in HTML and JavaScript languages and was edited in Visual Studio Code. The source code of the mobile application is written in the Java programming language and was edited in the Android Studio program. All survey data and associated results are stored in a Firebase Realtime Database.

Key words: Android application, Firebase, Java, Survey, Web application

ŽIVOTOPIS

Marko Pavičić rođen je u Osijeku 26. lipnja 1999. godine. U Osijeku završava osnovnu školu Jagode Truhelke. Nakon završene osnovne škole upisuje Elektrotehničku i prometnu školu Osijek gdje 2018. stječe zvanje Elektrotehničar. Iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku studij Elektrotehnike, smjer komunikacije i informatika.

PRILOZI

Na optičkom disku nalaze se sljedeći prilozi:

- Završni rad u .docx formatu
- Završni rad u .pdf formatu
- Izvorni kod web administratorskog sučelja
- Izvorni kod mobilne aplikacije