Izrada android aplikacije za upravljanje bilježaka

Šimić, Mario

Undergraduate thesis / Završni rad

2017

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:200:069032

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-12-22**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFROMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni studij Informatike

IZRADA ANDROID APLIKACIJE ZA UPRAVLJANJE BILJEŽAKA

Završni rad

Mario Šimić

Osijek, 2017

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Osijek, 12.05.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Ime i prezime studenta:	Mario Šimić
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI4324, 29.08.2013.
OIB studenta:	37908129963
Mentor:	Doc.dr.sc. Mirko Kohler
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Krešimir Nenadić
Član Povjerenstva:	Doc.dr.sc. Ivica Lukić
Naslov završnog rada:	Izrada android aplikacije za upravljanje bilježaka
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	Zadatak ovog završnog rada je izrada android aplikacije za unos bilježaka s predavanja. Pri spremanju šalje se kopija istih u bazu podataka na serveru, kako bi korisnik imao mogućnost istima pristupiti na drugom uređaju. Potrebno je omogućiti i ubacivanje multimedijalnih datoteka.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	12.05.2017.

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, 24.05.2017.

Ime i prezime studenta:	Mario Šimić
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI4324, 29.08.2013.
Ephorus podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom**: Izrada android aplikacije za upravljanje bilježaka**

izrađen pod vodstvom mentora Doc.dr.sc. Mirko Kohler

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. Opis zadatka 1
2. TEHNOLOGIJE KORIŠTENE U IZRADI APLIKACIJE 2
2.1. Android 2
2.2. Android studio 2
2.3. HTML 2
2.4. CSS 2
2.5. JavaScript 3
2.6. PHP 3
2.7. Java 3
2.8. MySQL 4
3. POSTUPAK IZRADE APLIKACIJE 5
3.1. Android manifest 5
3.2. Android aplikacija 5
3.3. Web aplikacija 6
3.3.1. Početna stranica 7
3.3.2. Nova zabilješka 9
3.3.3. Galerija slika 10
4. BAZA PODATAKA 12
5. UPUTE ZA KORIŠTENJE 13
6. ZAKLJUČAK 16
LITERATURA 17
SAŽETAK 18
ABSTRACT 19
ŽIVOTOPIS 20

1. UVOD

U današnjem vremenu internetske tehnologije se primjenjuju u svakom koraku, bilo to za edukacijske ili zabavne svrhe pa sve do istraživačkih i poslovnih svrha. Jedna od prednosti sustava baziranih na internet tehnologiji je ta da su pristupačni na svakom uređaju koji ima pristup internetu. Uz prednosti dolaze i nedostaci, a jedan veliki nedostatak ovakvih sustava je sigurnost zbog svoje velike pristupačnosti s različitih uređaja.

U ovome radu je opisana izrada i način funkcioniranja aplikacije za upravljanje bilježaka. Aplikacija omogućava korisniku da na svojem pametnom uređaju, bio to tablet, mobitel ili računalo, piše zabilješke koje se spremaju na poslužitelju te korisnik ima kasnije mogućnost pristupa tim zabilješkama s bilo kojeg uređaja koji ima pristup internetu. Po potrebi korisnik može preuzeti te zabilješke u pdf formatu. U poglavlju dva su ukratko objašnjene tehnologije koje su korištene pri izradi aplikacija. U trećem poglavlju je dan opis izrade mobilne i web aplikacije. Izrada baze podataka opisana je u četvrtom poglavlju. U petom poglavlju nalaze se upute za rad sa aplikacijom za korisnika.

1.1. Opis zadatka

Zadatak ovog završnog rada je izrada Android aplikacije za unos bilježaka s predavanja. Pri spremanju šalje se kopija istih u bazu podataka na serveru kako bi korisnik imao mogućnost istima pristupiti na drugom uređaju. Potrebno je omogućiti i ubacivanje multimedijalnih datoteka.

2. TEHNOLOGIJE KORIŠTENE U IZRADI APLIKACIJE

2.1. Android

Android je *open source* operacijski sustav kojeg je razvio Google. Zasnovan je na Linux 2.6 jezgri koja je napisana u C/C++ programskom jeziku. Namijenjen je za uređaje sa zaslonom na dodir kao što su pametni telefoni, satovi i tableti. Android Inc. su 2003. godine osnovali Andy Rubin, Chris White, Rich Miner i Nick Sears, a 2005. godine tvrtku je kupio Google. [1]

2.2. Android studio

Android studio je integrirano razvojno okruženje za razvoj na Android platformi. Prva verzija objavljena je u svibnju 2013. godine dok je prva stabilna verzija objavljena tek u prosincu 2014. godine kao verzija 1.0. Trenutna zadnja dostupna verzija je 2.2.3. koja je puštena u rad u prosincu 2016. godine. Android studio je baziran na JetBrains' IntelliJ IDEA softveru te je namijenjen za izradu Android aplikacija. Dostupan je za sve veće operativne sustave kao što su Windows, Linux i Mac OS X. Svojstva koja trenutna verzija razvojnog okruženja podržava su: *Lint* alati za poboljšanje performansi, kompatibilnosti i ostalih problema, *Gradle build* sustav, *drag and drop* način rada, čarobnjak za kreiranje dizajna uz pomoć predložaka, virtualni uređaj za testiranje aplikacija. [2]

2.3. HTML

HTML (*eng. HyperText Markup Language*)je prezentacijski jezik za izradu web stranica odnosno pomoću njega se oblikuje sadržaj i stvaraju se hiperveze. Dosta je jednostavan za uporabu te ga se lako uči pa je zbog toga popularan. Njegova uloga je da uputi web preglednik kako da prikaže hipertekst dokument pri tome da taj dokument izgleda isto na svim web preglednicima, odnosno na svim računalima. Važno je napomenuti da HTML nije programski jezik jer s njime se ne mogu izvršavati nikakve operacije i naredbe. Osnovni elementi svake stranice su oznake (eng. *tags*) koji opisuju kako se nešto prikazati na web pregledniku. [3]

2.4. CSS

CSS (eng. *Cascading Style Sheets*) je stilski jezik koji se rabi za dizajn web dokumenta, odnosno web stranice napisane u HTML jeziku. Glavna razlog razvoja CSS-a je da se omogući razdvajanje sadržaja stranice i dizajna te stranice. To je omogućilo veću preglednost koda i mogućnost korištenja iste CSS datoteke za više različitih HTML stranica. [4]

2.5. JavaScript

JavaScript je skriptni programski jezik koji se izvršava u web pregledniku na korisničkoj strani. Zbog lakšeg korištenja rađena je na temelju Java programskog jezika, ali nije objektno orijentiran. Uz HTML i CSS, JavaScript je jedan od tri glavna programska jezika za razvoj web stranica. Podržan je od svih modernijih web preglednika i nije potreban nikakav dodatak za korištenje JavaScripta. Razvijen je od strane Netscape tvrtke. [5]

2.6. PHP

PHP (rekurzivni akronim i backronim za „PHP: *Hypertext Preprocessor"*, prije *„Personal Home Page"*) je objektno orijentirani programski jezik namijenjen za razvoj dinamičkih web stranica. PHP se izvršava na samome poslužitelju te je zbog toga *server-side* programski jezik. Za izradu kompleksnijih web aplikacija koriste se web *framework*-ci. Neki od poznatijih *framework*-a su Laravel, CodeIgniter, CakePHP, Symfony i Zend Framework. Framework omogućava jednostavniju i lakšu izradu web aplikacija zbog svojih ugrađenih funkcija i MVC strukture. [6]

Za izradu ovog rada bio je korišten Laravel *framework*. On je *open-source* framework koji je baziran na Simfony *framework*-u. Jedan je od najpopularnijih i najlakših *framework*-a, a cijeli izvorni kod je dostupan na GitHub-u.

2.7. Java

Java je objektno orijentirani programski jezik kojeg su razvili Patrick Naughton i James Gosling u tvrtki Sun Microsystems. Razvoj Jave je započeo 1991. godine kao dio projekta pod nazivom *Green*, a nakon 4 godine je bio objavljen. Tvrtka Sun posjeduje *trademark* prava na ime Java, ali okruženje je potpuno besplatno za korištenje. Velika prednost Jave je ta što se u odnosu na druge programske jezike može izvoditi na svim operativnim sustavima na kojima postoji JVM (*eng. Java Virtual Machine*) dok programe koji su pisani u C-u potrebno je prilagoditi operacijskom sustavu na kojemu se izvode. Java programski jezik je jedan od najkorištenijih, nagađa se da ima preko 10 milijuna korisnika. Iako je inspirirana C programskim jezikom Java pruža bolji stupanj sigurnosti i pouzdanosti zahvaljujući VM-u i hermetički zatvorenom okolišu u kojem se svaki program izvodi. Popularan je za razvoj programa na mobilnim telefonima te je osnovni programski jezik za programiranje Android sustava. [7]

2.8.MySQL

MySQL je *open source* sustav za upravljanje bazom podataka. Ranije u svom razvoju MySQL baza nije bila popularna zbog svojih nedostataka naspram ostale konkurencije. MySQL baze su relacijskog tipa, što je pokazano kao najbolji način za pohranjivanje i pretraživanje većih količina podataka. To se je pokazalo kao važan temelj za poslovne baze jer su brzo dohvaćale informacije. [8]

3. POSTUPAK IZRADE APLIKACIJE

Glavni dio rada je Android aplikacija koja se instalira na uređaj te se pomoću nje rade zabilješke. U početku sve se je trebalo nalaziti unutar aplikacije, ali zbog komplikacija sa pretvaranjem datoteka u pdf format i sinkronizacijom sa poslužiteljem došlo je do promjene. Glavni dio je napravljena kao web platforma na koju se putem mobilne aplikacije spaja na tu stranicu. Time je omogućena lakša komunikacija između aplikacije i baze podataka te nema potrebe za sinkronizacije uređaja sa bazom jer je sve spremljeno na poslužitelju.

3.1. Android manifest

Svaka Android aplikacija ima datoteku pod nazivom *AndroidManifest.xml* koja se nalazi u *root* direktoriju. *Manifest* sadrži bitne informacije o aplikaciji koje su potrebne Android sustavu, bez njega aplikacija se ne može pokrenuti. Jedna od tih informacija su dozvole koje zadrži aplikacija. Bez tih dozvola sama aplikacija ne bi mogla funkcionirati ispravno. Na slici 3.1 je vidljivo da su potrebne dozvole za čitanje/pisanje podataka na memoriji te pristup kameri i internetu. [9]

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.etfos.mariosimic_ai4324.organizatorbilješki">

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.INTERNET" />

    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Organizator Bilješki"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.NoActionBar">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".web_based"></activity>
    </application>

</manifest>
```

Sl 3.1 Android Manifest

3.2. Android aplikacija

Samim time što je veći dio sustava na poslužitelju aplikacija nije zahtijevala veći dio rada. Sastoji se od glavnog prozora (eng. *layout*) na kojemu se nalazi *WebView* element čije je glavno svojstvo prikazivati web sadržaj. Slika 3.2 prikazuje *XML* kod glavnog prozora gdje je vidljiv cijeli grafički

sastav aplikacije. *RelativeLayout* postavlja relativne odnose između ostalih elemenata što se nalaze u njemu. Osim *RelativeLayout*-a postoji i *LinearLayout* pomoć kojega se elementi slažu linearno, odnosno jedan na drugi.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="0dp"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    tools:context="com.etfos.mariosimic_ai4324.organizatorbiljeski.web_based">

    <WebView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/webView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true" />
</RelativeLayout>
```

Sl 3.2 *Layout* glavnog prozora

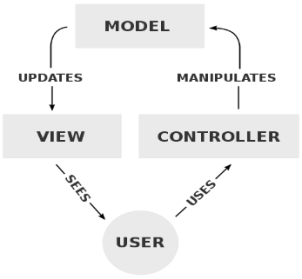
Da bi aplikacija u potpunosti funkcionirala i povezala se na poslužitelj potrebno je definirati *WebView* element kao objekt kako bi se on dalje mogao konfigurirati. Na *WebView* trebamo dodijeliti *url* na koji će se povezivati pomoću funkcije *.loadUrl()* gdje u zagradu unosimo adresu web stranice.

```
webView = (WebView)findViewById(R.id.webView);

webView.getSettings().setJavaScriptEnabled(true);
webView.getSettings().setLoadWithOverviewMode(true);
webView.getSettings().setUseWideViewPort(true);
webView.setWebViewClient(new ourViewClient());
webView.getSettings().setAllowFileAccess(true);

webView .loadUrl("http://www.punzalogaj.com/OrganizatorBiljezaka/public");
```

Sl 3.3 Definiranje *WebView* elementa i dodjela url-a

3.3. Web aplikacija

Za izradu web platforme korišten je Laravel framework koji je ukratko opisan u poglavlju 2.6.. Temelji se na MVC (model-view-controller) modelu radi lakše i jednostavnije funkcionalnosti. U MVC modelu korisniku se prikazuje pogled (eng. *View*) gdje korisnik svojim akcijama na toj tom pogledu pokreče upravitelja (eng. *controller*) koji šalje naredbu modelu koji po naredbi radi izmjene na pogledu.

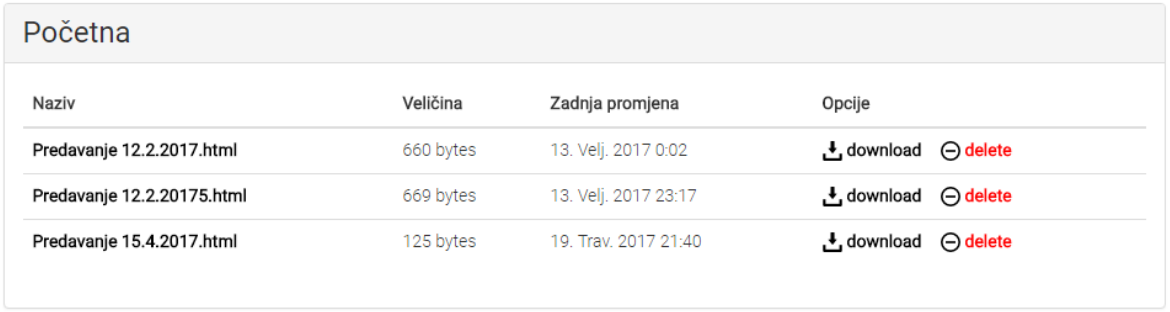


Sl 3.4 Primjer MVC modela [10]

Platforma se dijeli na tri dijela, a to su „Početna stranica“ gdje korisnik može upravljati svojim zabilješkama, „Nova datoteka“ omogućava korisniku stvaranje nove zabilješke i „Galerija“ gdje se nalaze slike koje je korisnik spremio na poslužitelj.

3.3.1. Početna stranica

Na početnoj stranici korisnik može upravljati svojim zabilješkama. Korisnik može otvoriti već postojeću zabilješku i tako nadodati zabilješke u nju ili prepraviti. Daljnje mogućnosti su preuzimanje zabilješke u pdf formatu i brisanje zabilješke iz sustava.



Sl 3.5 Prikaz početne stranice

Zabilješke korisnika se spremaju u datoteku čiji naziv je jedan korisnikovom te tako svaki korisnik ima svoju vlastitu datoteku u kojoj su spremljene zabilješke. Kada korisnik pritisne na naziv zabilješke ona se otvara te je se može uređivati i izmjenjivati. Prilikom klika u upravitelj se šalju tri vrijednosti, a to su ime korisnika, naziv zabilješke i naziv datoteke u kojoj se nalazi za zabilješka. Nakon toga s funkcijom *file_get_contents()* uzima se sadržaj te zabilješke i sprema se u varijablu *$sadrzaj*. Nakon što su dohvaćeni podaci iz zabilješke spremaju se u varijablu $*sadrzajEnc* skupa sa nazivom zabilješke, a „||“ služi za razdvajanje naziva i sadržaja.

```
public function open($user,$folder, $file)
    {
        $path="files/".$user."/".$folder."/".$file;
        $sadrzaj = file_get_contents($path);
        $sadrzajEnc = $file."||".$sadrzaj;

        return Redirect::to('/editor')->with('open', $sadrzajEnc);
    }
```

Sl 3.6 Prikaz koda za otvaranje zabilješke

Ako korisnik želi skinuti svoju zabilješku pritiskom na *download* u upravitelj se šalju tri vrijednosti, a to su ime korisnika, naziv zabilješke i naziv datoteke u kojoj se nalazi za zabilješka. Korištenjem biblioteke mPDF zabilješka u HTML formatu pretvara se u pdf format i preuzima se na uređaj. Biblioteka mPDF služi za pretvaranje HTML datoteka u pdf datoteku s odgovarajućim *CSS* stilom. Biblioteka mPDF ne može prikazati sve HTML oznake kao što bi ih prikazala web stranica, ali može prepoznati sve osnovne oznake koje se koriste u zabilješkama. [11]

```
public function download($user,$folder, $file)
    {
        $path="files/".$user."/".$folder."/".$file;
        $sadrzaj = file_get_contents($path);
        $naziv =   $naziv = substr($file, 0, -5);

          $mpdf=new mPDF('c','A4','','' , 0 , 0 , 0 , 0 , 0 , 0);
          $mpdf->SetDisplayMode('fullpage');
          $mpdf->list_indent_first_level = 1;
          $html = '<html><head><meta charset="UTF-8" /></head><body><br><br><div '
                  . 'style=" margin: 20pt 25pt 20pt 25pt;">'.$sadrzaj.'</div></body></html>';
          $mpdf->WriteHTML($html);
          $orderPdfName = "order-".$singleOrder[0]['display_name'];
          $mpdf->Output($naziv.".pdf",'D');
          header('Content-type: application/pdf');
          header("Content-Disposition: attachment; filename=".$naziv.".pdf");

         return Redirect::to('/filemanager');
    }
```

Sl 3.7 Spremanje zabilješke kao .pdf

Osim uređivanja već napisanih zabilješki i preuzimanja korisnik može i obrisati zabilješke koje mu nisu više potrebne. Kao i u prijašnje dvije mogućnosti upravitelju se šalju ime korisnika, naziv zabilješke i datoteke te s tim podacima se pronalazi ta zabilješka i briše iz sustava.

```
public function delete($user,$folder, $file)
 {
       File::delete('files/'.$user.'/doc/'.$file);
       return Redirect::to('/filemanager')->with('message', 'Datoteka obrisana');
 }
```

Sl 3.8 Prikaz dijela koda za brisanje zabilješke

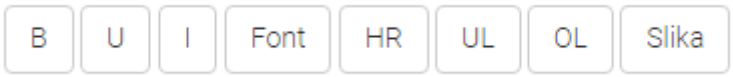
3.3.2. Nova zabilješka

Odabirom „Nova zabilješka“ na glavnom izborniku korisniku se otvara nova stranica gdje se nalazi uređivač teksta. Uređivač teksta se temelji na WYSIWYG (eng. *What You See Is What You Get*) principu jer je na taj način najlakše prikazati stvarni izgled teksta. Pri vrhu nalazi se tekstualno polje u koje korisnik upisuje ime svoje zabilješke, a s lijeve strane nalazi se gumb „Spremi“. Pritiskom na gumb POST metodom se šalju naziv zabilješke, ime korisnika i sadržaj zabilješke u *save.php* skriptu koja zapisuje to sve u HTML format i sprema u potrebnu datoteku. Nakon spremanja skripta vrača natrag na uređivač teksta i ispisuje poruku da je zabilješka spremljena.

```php
<?php
      $user = $_POST['user'];
      $nazivDat = $_POST['nazivDoc'];
      $sadrzajDat = $_POST['myTextArea'];

       $tmp = fopen('files/'.$user.'/doc/'.$nazivDat.'.html', 'w');
       fwrite($tmp,$sadrzajDat);
       fclose($tmp);

       $actual_link = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
       $link= str_replace("/save.php", "",$actual_link);
       header("Location: ".$link.'/editor/'.$nazivDat);
```

Sl 3.9 save.php skripta

Za uređivanje teksta ima nekoliko osnovnih mogućnosti kao što su podebljani, podcrtani i kosi tekst, promjena veličine slova, dodavanje horizontalne linije, uređene i neuređene liste te dodavanje slike. Te mogućnosti se nalaze iznad zabilješke u alatnoj traci. Pritiskom na neki od gumbova na alatnoj traci u zabilješku se postavlja odgovarajuća HTML oznaka pomoću koje uređivač grafički uređuje tekst.



Sl 3.10 Alatna traka

Korisnik ima mogućnost dodavanja slike u svoje zabilješke. Slika se može nalaziti već u galeriji slika, a ako se tamo ne nalazi korisnik može sa svog uređaja *uploadat* sliku te je odmah staviti u svoju zabilješku. Prilikom odabira slike korisnik mora unijeti visinu i širinu slike te se pomoću tih podataka slika smanjuje, odnosno povećava.

```
function chosePic(objButton)
{
    var visina = prompt ("Unesite visinu slike u px: ", "");
    var duzina = prompt ("Unesite dužinu slike u px: ", "");

    var imgSrc = objButton.value;
    if(imgSrc != null)
    {
      richTextField.document.execCommand('insertHTML',false,
      '<img width="'+duzina+'" height="'+visina+'" src="'+imgSrc+'">');
    }
}
```

Sl 3.11 Funkcija za postavljanje slike u zabilješku

3.3.3. Galerija slika

U galeriji slika nalaze se sve slike koje je korisnik spremio na poslužitelj tako da nema potrebe uvijek iznova *uploadat* sliku sa svojeg uređaja. Putanja slike je spremljena u bazi podataka te se putem putanje pronalazi slika za prikaz. Slike se prikazuju redoslijedom od najnovije do najstarije tako da korisniku budu lakše dostupne nove slike.

```
require 'connect.php';
$sql_selectPhoto = "SELECT path_thumb,naslov FROM ".Auth::user()->name."_slike ORDER BY id DESC";
$stmt = $conn->query($sql_selectPhoto);

foreach($stmt as $slika)
 {
   $url = url('/gallery/'.$slika['naslov']);

   echo '<div style="text-align:center;" class="col-lg-3 col-md-4 col-xs-6 thumb">';
   echo    '<a class="thumbnail" href="#">';
   echo        '<img width="150" height="150" class="img-responsive" src="'.$slika['path_thumb'].'" alt="">';
   echo    '</a>';
   echo '<form method="POST" action="'.$url.'">';
   echo '<input type="hidden" name="_token" value="'.csrf_token().'">';
   echo '<input  type="submit" class="btn btn-default" name="'.$slika['naslov'].'" value="Obriši">';
   echo '</form>';
   echo '</div>';
 }
```

Sl 3.12 Ispis slika od najnovije do najstarije

Unos je ograničen na slike s ekstenzijom *png,jpg,jpeg,bmp* i *gif*, radi sigurnosti. Ako slika ne zadovoljava te uvijete, ona neće biti *uploadana*. Da ne bi došlo do preklapanja slika sa istim imenom slika se sprema pod imenom *„vrijeme_postavljanja-datum_postavljanja.ekstenzija“*.

```
$user = Auth::user()->name;
$file = $request->file('slika');
$path= 'files/'.$user.'/pic/';
$ext=$file->getClientOriginalExtension();
$fileName=$file->getClientOriginalName();
$name= date("h:i:s")."-".date("Y.m.d").".".$ext;
$nameFile = $fileName.$ext;

if($ext == "png" || $ext == "jpg" || $ext == "jpeg" || $ext == "gif" || $ext == "bmp")
{
    $file->move($path,$fileName);
    DB::table($user.'_slike')->insert(
              ['naslov' => $fileName,
               'path' => 'files/'.$user.'/pic/'.$fileName,
               'path_thumb'=> 'files/'.$user.'/pic/'.$fileName]
              );
     return Redirect::to('/gallery')->with('message', 'Slika je spremljena');
}
else
 {
     return Redirect::to('/gallery')->with('messageBad', 'Slika nije odgovarajućeg formata(png,jpeg,jpg,gif,bmp).');
 }
```

Sl 3.13 Provjera ekstenzije slike i spremanje slike

Ako korisniku slika više nije potrebna on ima mogućnost brisanja slike iz svoje galerije pritiskom na gumb „Obriši“. Time se slika briše iz njegove mape i iz baze podataka.

```
public function delete($nazivSlike)
 {
     $naziv = $nazivSlike;

     File::delete('files/'.Auth::user()->name.'/pic/'.$naziv);
     DB::table(Auth::user()->name.'_slike')->where ('naslov', '=', $naziv)->delete();
     return Redirect::to('/gallery')->with('message', 'Slika obrisana');
 }
```

Sl 3.14 Brisanje slike

4. BAZA PODATAKA

Baza podataka je kreirana putem MySQL programskog jezika, a upravljana je putem myPhpAdmin-a radi lakše izmjene samih tablica. Baza podataka sadrži minimalno jednu tablicu, a to je tablica *„users“*. U toj tablici se nalaze podaci korisnika kao što su ime, email adresa, lozinka i kada je kreiran.

```
CREATE TABLE `users`
 (
  `id` int(255) NOT NULL,
  `name` varchar(50) NOT NULL,
  `email` varchar(50) CHARACTER SET utf16 NOT NULL,
  `password` varchar(255) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
 )
```

Sl 4.1. Kreiranje tablice „users“

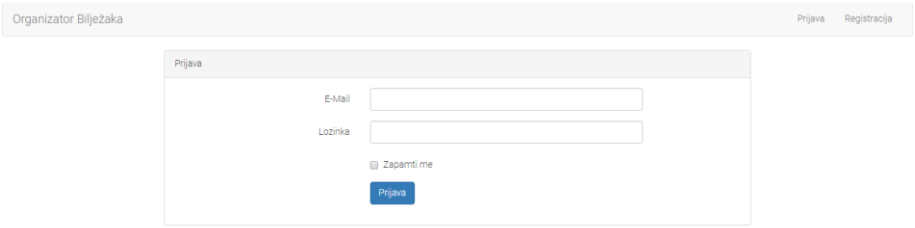
Tablica sa podacima korisnika je glavna tablica, dok ostale tablice se dinamički generiraju prilikom registracije novog korisnika. Kada se korisnik registrira po prvi puta stvara se nova tablica pod nazivom *„imeKorisnika_slike“* te se u tu tablicu spremaju putanje za slike, naslov, i redni broj slike. Uz tablicu kreiraju se i korisnikove osobne mape u kojima se nalaze zabilješke i slike.

```
public function create(array $data)
{
        mkdir('files/'.$data['name']);
        mkdir('files/'.$data['name'].'/doc');
        mkdir('files/'.$data['name'].'/pic');
        mkdir('files/'.$data['name'].'/tmp');
        require 'connect.php';

            $sql_userPhotoTable = "CREATE TABLE ".$data['name']."_slike (
                    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
                    naslov VARCHAR(255),
                    path VARCHAR(255),
                    path_thumb VARCHAR(255)
                    )";
        $stmt = $conn->query($sql_userPhotoTable);

    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => bcrypt($data['password']),
    ]);
}
```

Sl 4.2. Dinamičko kreiranje mapa i tablice

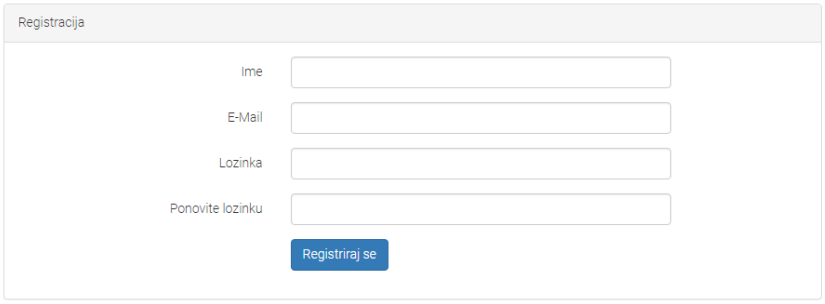
5. UPUTE ZA KORIŠTENJE

Kada korisnik otvori aplikaciju prvo što će biti prikazano je dio za prijavu korisnika. Kako bi korisnik pristupio svojim zabilješkama mora se prijaviti sa svojim podacima, a ako je novi korisnik onda se treba registrirati.



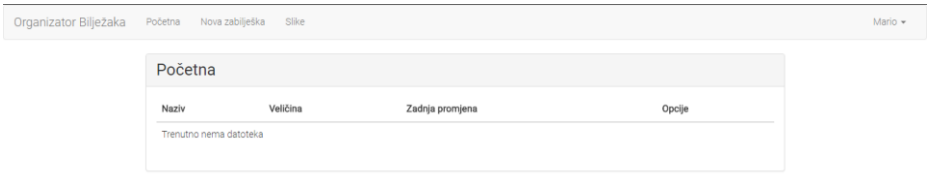
Sl 5.1 Forma za prijavu

Prilikom registracije od korisnika se traži da unese svoje ime, email adresu te lozinku koju će koristiti prilikom prijave.



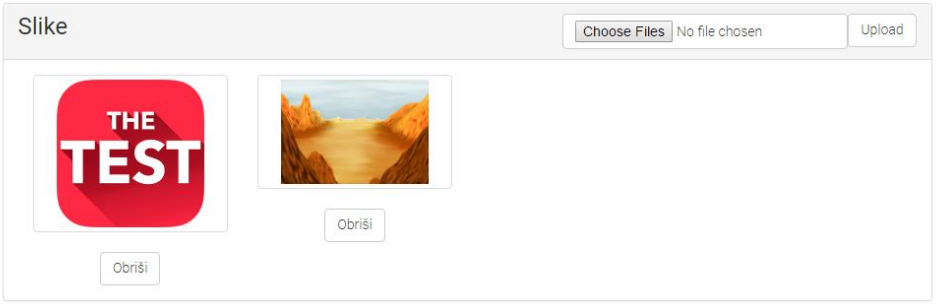
Sl 5.2 Forma za registraciju

Nakon uspješne registracije korisniku se prikazuje početni ekran gdje se nalaze njegove zabilješke, ali pošto se radi o novome korisniku neće biti prikazane nikakve zabilješke.



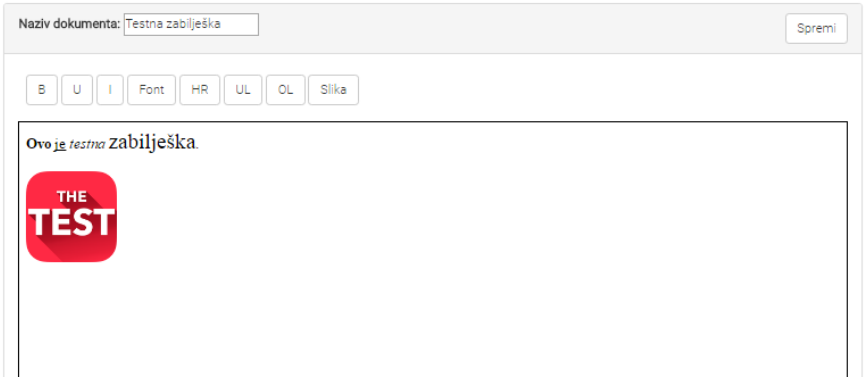
Sl 5.3 Prikaz početne stranice

U izborniku korisnik ima mogućnost za odabrati da li želi napraviti novu zabilješku ili možda želi otvoriti svoju galeriju slika. Galerija slika služi da bi korisnik samo jednom *uploado* sliku sa svoga uređaja pa da mu ona bude dostupna svakom trenutku bez ponovnog *uploada*. Korisnik ima mogućnost pritiskom na tipku „*Obriši*“ obrisati slike koje više ne želi koristiti ili koje je slučajno postavio.



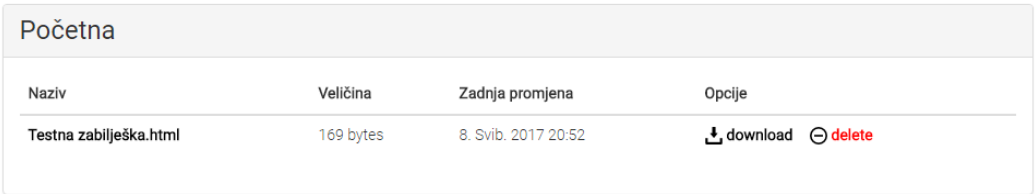
Sl 5.3. Prikaz galerije slika

Za pravljenje nove zabilješke u izborniku se treba odabrati „*Nova zabilješka*“ i onda se otvara stranica na kojoj se nalazi uređivač teksta. Korisnik ima mogućnost osnovnih opcija za uređivanje teksta kao što su podebljani, podcrtani i kosi tekst, promjena veličine slova, dodavanje horizontalne linije, uređene i neuređene liste te dodavanje slike. Nakon što je korisnik napisao svoju zabilješku treba upisati naziv zabilješke i onda je može spremiti.



Sl 5.4 Prikaz uređivača zabilješki

Nakon što je zabilješka spremljena na početnoj stranici bit će vidljiv naziv te zabilješke, kolika joj je veličina, kada je bila zadnja promjena te će biti dvije mogućnosti, jedna za skinuti zabilješku u .pdf formatu, a druga za obrisati zabilješku.



Sl 5.5 Prikaz početne stranice sa zabilješkom

Ako korisnik odluči uređivati postojeću zabilješku sve što je potrebno je pritisnuti na naziv zabilješke koju se želi uređivati i ona se otvara u uređivaču. Naziv bude već popunjen, jedino ako se želi spremiti kao nova zabilješka onda se naziv mora promijeniti.

6. ZAKLJUČAK

Android operacijski sustav je pogodan za sve vrste aplikacija najviše zbog svoje raširenosti (pametni telefoni, satovi, televizori), a i na primjeru ovog rada je vidljivo da web aplikacija sa lakoćom može pretvoriti u Android aplikaciju. Ciljevi koji su bili zadani su zadovoljeni, ali aplikacija ima još uvijek mjesta za nadogradnju. Prva veća nadogradnja bi bila mogućnost korištenja same aplikacije u izvan mrežnom načinu rada te ograničavanje prostora za slike i zabilješke da ne bi došlo do natrpanosti poslužitelja. Ostale manje nadogradnje bi se bazirale više na dizajn aplikacije i rješavanje sigurnosnih rupa.

Aplikacija bi se najviše mogla koristiti na sastancima jer ne bi bilo potrebe da se koriste papiri koji se kasnije uništavaju, što je na jednu ruku korisno za okoliš. Zbog toga što je aplikacija dostupna putem android platforme i web platforme korisnik ima pristup svojim zabilješkama na svim uređajima koji imaju pristup internetu.

LITERATURA

[1] Android sustav
https://www.android.com,
Stranica posjećena: ožujak 2017

[2] Android Studio
https://developer.android.com/studio/intro/index.html,
stranica posjećena: ožujak 2017.

[3] HTML
https://www.w3schools.com/html,
stranica posjećena: ožujak 2017.

[4] CSS
https://www.w3schools.com/css,
stranica posjećena: ožujak 2017.

[5] JavaScript
https://www.w3schools.com/js,
stranica posjećena: ožujak 2017.

[6] PHP
https://www.w3schools.com/php,
stranica posjećena: ožujak 2017.

[7] Java
https://www.java.com,
stranica posjećena: ožujak 2017.

[8] MySQL
https://www.w3schools.com/sql,
stranica posjećena: ožujak 2017.

[9] Android Developers, Android Manifest
https://developer.android.com/guide/topics/manifest/manifest-intro.html,
stranica posjećena: travanj 2017.

[10] Wikipedia, MVC model slika
https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller#/media/File:MVC-Process.svg,
stranica posjećena: travanj 2017.

[11] mPDF
https://mpdf.github.io/,
stranica posjećena: travanj 2017.

SAŽETAK

U završnome radu je opisana izrada mobilne aplikacije za pametne telefone koji koriste Android operativni sustav te web aplikacije koja se izvodi na poslužitelju. Za Android aplikaciju korišten je Java programski jezik, dok za web aplikaciju su korišteni HTML, JavaScript, PHP i CSS programski jezici. Baza podataka sadrži tablicu *users* u kojoj se nalaze podaci korisnika i svaki korisnik ima svoju tablicu za slike. *Organizator bilježaka* je aplikacija koja omogućava korisnicima zapisivanje zabilješki na svojim pametnim uređajima te im kasnije mogu pristupiti s bilo kojeg uređaja. Osim zapisivanja, korisnik može uređivati već postojeće zabilješke, preuzimati ih i brisati ih.

Ključne riječi: Android, aplikacije, baza podataka, HTML, Java, pametne telefone, PHP

ABSTRACT

Development of Android application for taking notes

This paper describes the process of creating a mobile application for smartphones that are using Android operating systems and web application that is used on the server. Java programming language was used for developing the Android application, while HTML, JavaScript, PHP and CSS were used for developing the web application. The database contains information about the users in table *users* and every user has his own table for pictures. *Note organizer* is an application that allows users to write notes on their smartphones with the ability to access them later on any device. Except writing notes users can also edit their notes, download them and delete them.

Key words: Android, application, database, HTML, Java, PHP, smarthphones

ŽIVOTOPIS

Mario Šimić rođen je 28. srpnja 1994. godine u Osijeku. Pohađao je Osnovnu školu Augusta Šenoe u Osijeku te nakon završetka 2009. godine upisuje Elektrotehničku i prometnu školu u Osijeku smjer Elektroničar. Nakon dvije godine prebacuje se u smjer Tehničar za računalstvo koje završava 2013. godine. Akademske godine 2013/2014 upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, gdje je upisao stručni studij Informatike.