Android aplikacija za praćenje cijena artikala

Čopčić, Antun

Undergraduate thesis / Završni rad

2019

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:200:909329

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstva

ANDROID APLIKACIJA ZA PRAĆENJE CIJENA ARTIKALA

Završni rad

Antun Čopčić

Osijek, 2019.

Sadržaj

1. UVOD 1

1.1. Zadatak završnog rada 1

2. OPIS KORIŠTENIH TEHNOLOGIJA 2

2.1. Operacijski sustav Android 2

2.2. Android Studio 3

2.3. Programski jezik Java 4

2.4. XML 5

3. RAZVOJ APLIKACIJE 6

3.1. Implementacija projekta 6

3.2. Baza podataka 21

3.2.1. greenDAO 21

4. ZAKLJUČAK 23

LITERATURA 24

SAŽETAK 25

ABSTRACT 25

ŽIVOTOPIS 26

1. UVOD

Ovaj završni rad je namjenjen za pomoć ljudima kod kupovine. Koliko god se trgovački lanci trudili tiskati i dijeliti letke sa svojom ponudom, ipak tehnologija napreduje, te već postoje neke aplikacije sa ponudom iz trgovačkih lanaca. Ova aplikacija je zamišljena da bi olakšala pregled proizvoda. Korisnik može u aplikaciji napraviti kategorije koje su njemu potrebne. Preko kamere mobitela očitava barkod proizvoda, unosi tražene podatke, te nakon toga sprema taj proizvod u određenu, njemu namjenjenu kategoriju. Poslije, korisnik, u svakom trenutku preko aplikacije može pogledati cijenu proizvoda koji ga zanima. Pregledavanje silnih reklama kako bi se našlo baš točno ono što se želi je za dosta ljudi zamorno. Ova aplikacija bi ubrzala taj postupak.

Kod razvoja ove aplikacije korišten je operacijski sustav Android. Potrebno je dobro znanje programskog jezika Java i integriranog razvojnog okruženja Android Studio u kojem će se ovaj rad napraviti. Za bazu podataka odabran je greenDAO, trenutno jedan od najčešće korištenih alata kod mnogih kvalitetnih programera.

Strukturu ovog završnog rada tvore uvod, opis korištenih tehnologija, razvoj aplikacije kao glavni dio, zaključak, literatura, sažeci na hrvatskom i engleskom jeziku, te životopis.

1.1. Zadatak završnog rada

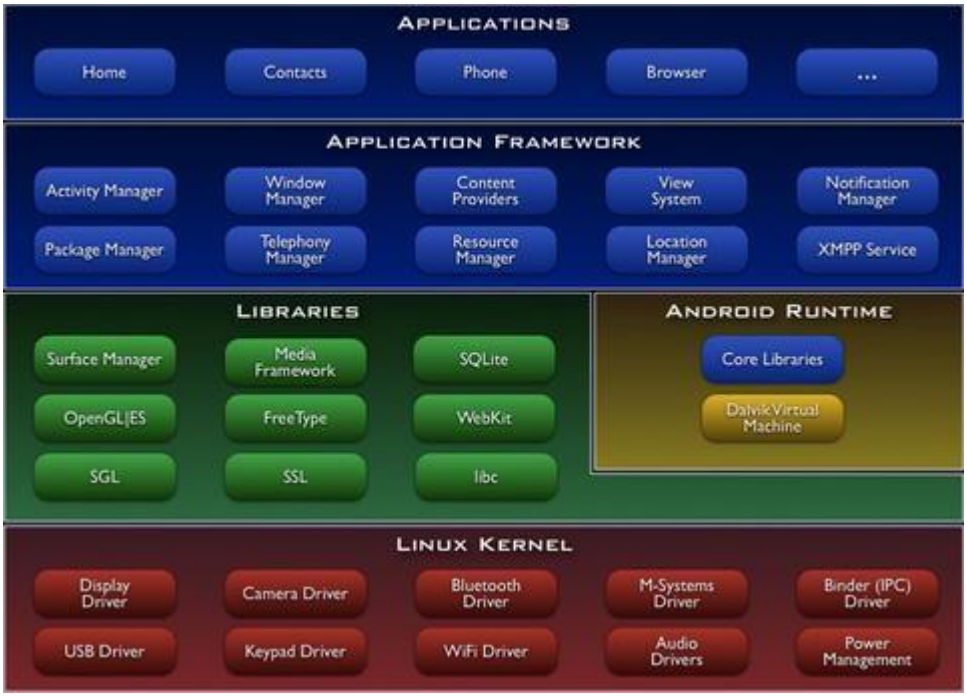
Zadatak ovog rada je izraditi Android aplikaciju čija će svrha biti da korisniku omogući lakše snalaženje kada treba obaviti neku kupovinu. Aplikacija treba imati mogućnost očitavanja barkoda s različitih proizvoda, zatim mogućnost da se dodatno unesu potrebni podaci i na kraju sve to treba spremiti u bazu podataka radi lakšeg korištenja u bližoj budućnosti.

2. OPIS KORIŠTENIH TEHNOLOGIJA

Tehnologije korištene za razvoj ove aplikacije opisane su u ovom poglavlju. To su operacijski sustav Android, Android Studio, programski jezik Java i XML.

2.1. Operacijski sustav Android

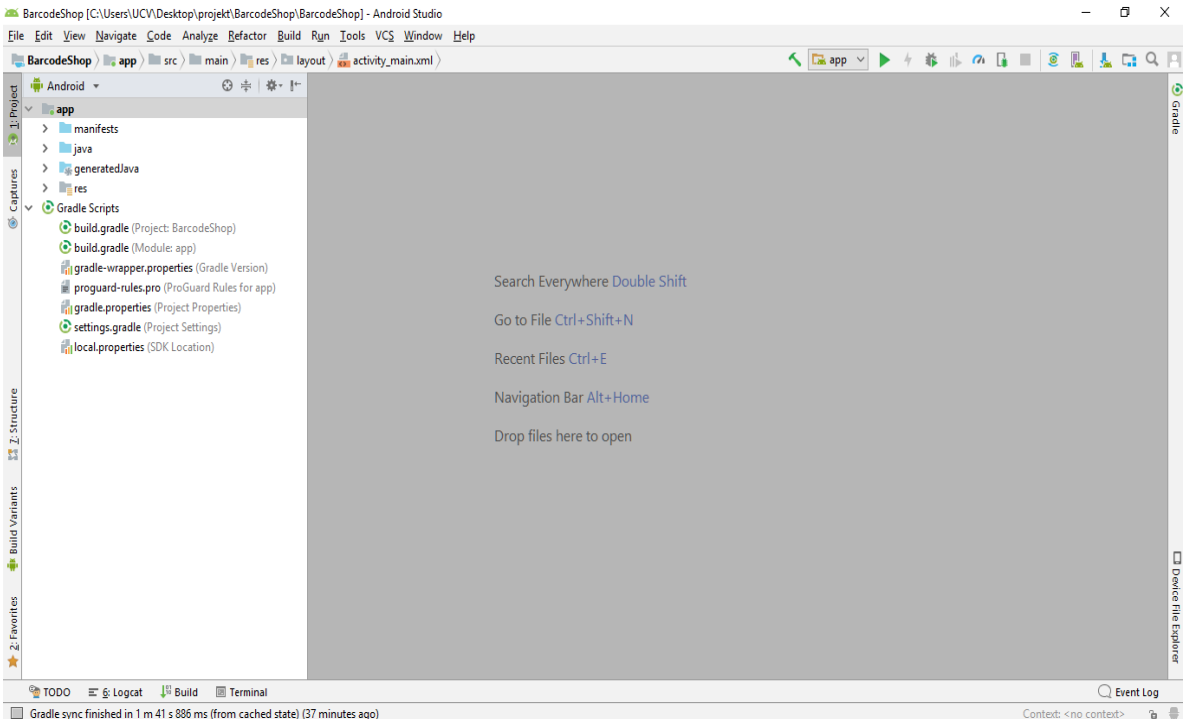
Zadnje godine 20. stoljeća smatraju se počecima pametnih mobilnih uređaja. S vremenom su rasli apetiti korisnika, a samim time i proizvođača, pa su se i razvijali uređaji. Jedna od tvrtki koja se bavila time je Android Inc. u kojoj je 2003. godine nastala ideja o sustavu Android. Sam taj razvoj sustava bio je poprilično skup. Srećom, 2005. godine Google ju je spasio od propasti jer je tu vidio svoju priliku za izlaz na tržište mobilne telefonije. Google 2007. godine osniva konzorcij *Open Handset Alliance* (OHA). Svrha tog konzorcija je stvaranje službene norme za mobilne uređaje. Ubrzo se tom konzorciju pridružio velik broj tvrtki koji imaju veze s mobilnom industrijom. Istog dana OHA izdaje Android. To je mobilna platforma otvorenog kôda koja je temeljena na Linux jezgri. To se ujedno smatra i prvim javnim predstavljanjem Androida kao operacijskog sustava. Android se danas smatra najraširenijim operacijskim sustavom za pametne uređaje. Zasnovan je na jezgri Linux 2.6.. Programski jezik u kojem je napisan je C/C++. Ipak, većina aplikacija napisana je programskim jezikom Java koristeći Android razvojne programske komponente (engl. *Android Software Development Kit*, SDK). Prema [1] arhitektura Android sustava sastoji od nekoliko razina (Slika 2.1.): prva razina je Linux jezgra koja sadrži drivere, iznad se nalaze knjižnice pisane u C/C++ programskom jeziku, potom dolazi Android Runtime - sloj koji pokreće aplikacije, nakon toga dolazi aplikacijski okvir koji sadrži mehanizme koji olakšavaju pisanje aplikacija, a gore na samom vrhu se nalaze aplikacije - sloj kojeg krajnji korisnik može vidjeti. Google Play Store je mrežna trgovina Androida u kojoj je moguće pronaći razne aplikacije, te ih korisnik može preuzeti, instalirati i nakon toga koristiti na svojem mobilnom uređaju. Tu se nalazi veliki broj besplatnih aplikacija, ali također i onih koje se mogu kupiti prije preuzimanja.



Slika 2.1. Arhitektura Android sustava, prema [3].

2.2 Android Studio

Android Studio realiziran je na temelju IntelliJ IDEA softvera. Objavljen je u svibnju 2013. godine na Googleovoj konferenciji razvojnih inženjera. Android Studio (Slika 2.2.) je službeni razvojni alat za Googleov operativni sustav Android. Prema [4] bio je u stupnju ranog pristupa, krenuvši od inačice 0.1 izdane u svibnju 2013. godine, a potom se ušlo u beta status krenuvši od inačice 0.8 koja je izdana u lipnju 2014. godine. Prva postojana konstrukcija izdana je u prosincu 2014. godine, krenuvši od inačice 1.0. Kreiran je specijalno za razvoj novih aplikacija namijenjenih za Android platformu. Može se preuzeti besplatno i podržan je za Linux, Windows i Mac OS X operativne sustave. Prema [5] Android Studio pruža mnogo značajki koje utječu na bolju efikasnost kod izrade aplikacija za Android. Te značajke su: brz i bogati emulator, podrška za C++ i NDK, Instant Run kako bi mogli unjeti izmjene u pokrenutu aplikaciju bez izrade novog APK-a, opsežni alati za testiranje itd. Najveći dio ovo rada je napravljen upravo u Android Studio alatu.



Slika 2.2. Sučelje Android Studia

2.3. Programski jezik Java

Patrick Naughton, James Gosling i drugi inženjeri u tvrtci Sun Mycrosystems razvili su Javu, objektno orijentirani programski jezik, kod kojeg se kôd nalazi unutar klasa. 1991. godine, kao dio projekta Green, počeo je razvoj Jave, a 1995. godine je objavljen. Tada je to bilo osvježenje za programere. Razlika Jave u odnosu na ostale tadašnje programe je ta da su se programi mogli izvoditi bez preinaka na svakom operativnom sustavu na kojem postoji *Java Virtual Machine* (JVM). Programi koji su pisani u C-u morali su se prilagođavati operativnom sustavu na kojem se izvode. Java se ubraja među glavne programske jezike koji se upotrebaljavaju za razvoj Android mobilnih aplikacija. Ubraja se u skupinu viših programskih jezika, koji imaju svoje prednosti i nedostatke. Jedan od najvećih nedostataka kod Jave i drugih viših programskih jezika je taj što se kôd prvo mora prebaciti u niži programski jezik, odnosno strojni jezik. Prednosti viših programskih jezika su: jednostavnije programiranje, kraće vrijeme za pisanje, kôd se lakše čita. Ako nije instalirana Java ne mogu se koristiti mnoge aplikacije i internet stranice. Srećom, Javu je moguće besplatno preuzeti. Mnogi zaljubljenici u programski jezik Java vole reći da je on brz, siguran i pouzdan programski jezik.

```java
public class AddTwoNumbers {

   public static void main(String[] args) {

      int num1 = 5, num2 = 15, sum;
      sum = num1 + num2;

      System.out.println("Sum of these numbers: "+sum);
   }
}
```

Slika 2.3. Primjer jednostavnog kôda u Javi za zbroj dva broja

2.4. XML

XML (engl. *Extensible Markup Language*) je vrlo raširen jezik za označavanje podataka. U tekstualnom formatu se prikazuju informacije poput dokumenata, raznih podataka i slično. Čitljiv je i ljudima i računalnim programima. XML dokumenti mogu se vrlo pouzdano obrađivati pomoću računalnih programa zato što obrada neće započeti dok god postoje greške u dokumentu. XML kôd piše se unutar oznaka. Sastoji se od početne oznake određenog elementa koja se obilježava < >, oznake zatvaranja </ >, te sadržaja koji se nalazi između oznaka (Slika 2.4). Prema [7] format oznaka u XML-u je sličan onome u HTML-u, ali dizajniran je za drugačije potrebe. Koristi se u različite svrhe kao što su: razmjena, pohrana i povećanje dostupnosti podataka.

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout 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"
    tools:context=".activity.CategoryActivity">

    <include layout="@layout/content_category" />

</android.support.design.widget.CoordinatorLayout>
```

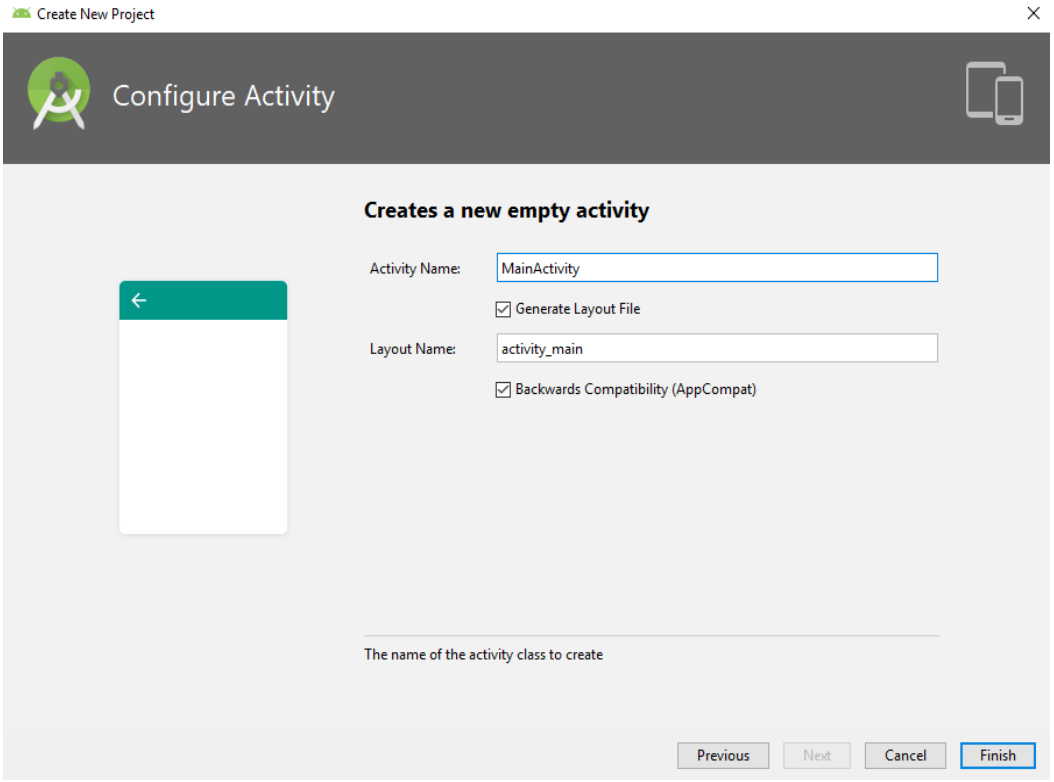
Slika 2.4. Primjer XML kôda

3. RAZVOJ APLIKACIJE

U ovom poglavlju opisan je razvoj aplikacije za pomoć pri kupovini. Ideja za ovu aplikaciju došla je nakon razgovora sa ljudima koji su izrazili svoj problem i zatražili rješenje. Slobodno se može reći da je ideja prvi korak u razvoju aplikacije. Nakon razrađivanja ideje koje uključuje utvđivanje korisničkih zahtjeva i obilazak trgovačkih lanaca, prelazi se na samu provedbu projekta. U nastavku završnog rada objašnjeni su koraci programskog rješenja za nastanak aplikacije popraćeni slikama kodova, te sama upotreba aplikacije sa slikama iz iste.

3.1. Implementacija projekta

Otvaranje Android Studia i kreiranje novog projekta je jednostavno. Taj postupak završava kreiranjem nove prazne aktivnosti (Slika 3.1.).



Slika 3.1. Kreiranje nove aktivnosti u Android Studiu

Struktura projekta sastoji se od tri glavna dijela: *manifests*, *java* i *res*. U *manifestsu* svakog projekta nalaze se bitne informacije o aplikaciji. Također se tu postavljaju različite dozvole koje aplikacija koristi. U ovom radu su to dozvole za

pristup kameri na Androidu (Slika 3.2.).

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

Slika 3.2. Dozvole za pristup kameri

U *java* se nalaze sve klase koje su pisane u Java programskom jeziku. Za lakše snalaženje u kôdu i za lakše testiranje aplikacije unutar stabla projekta napravljeno je više direktorija. Svaka aktivnost ima svoj direktorij. Također postoji direktorij za fragmente, direktorij za različite adaptere, direktorij za modele, direktorij za dijalog klasu i direktorij za *util* klasu. Dok se u *res-u* nalaze resursi kao što su boje, slike itd. Također se u *res-u* određuje izgled korisničkog sučelja. To se radi u tzv. datotekama rasporeda. One su pisane u XML-u i u njih se stavljaju pogledi (Button, ListView itd.) koji će biti prikazani na zaslonu uređaja.

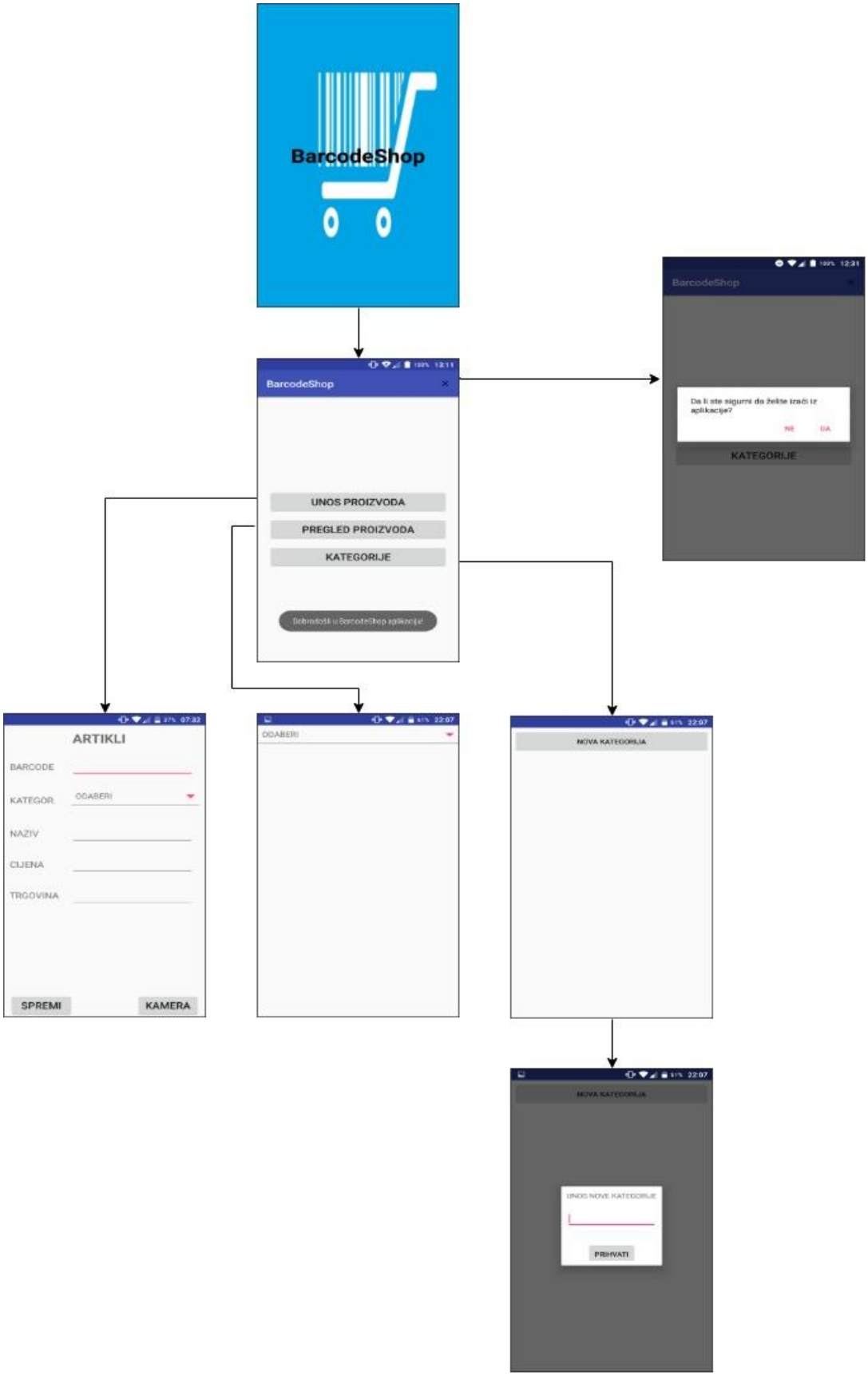
U ovom radu osnovni skup biblioteka koji se nalazi u Android Studiu nije dovoljan, pa su implementiranjem uvedene vanjske biblioteke u *gradle* datoteci (Slika 3.3.). Biblioteke su ustvari Java klase u obliku paketa koje programeri koriste po potrebi. Ovdje su dodane biblioteke za barkod, bazu podataka i padajući izbornik.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:design:27.1.1'
    implementation 'org.greenrobot:greendao:3.2.0'
    implementation 'com.facebook.stetho:stetho:1.4.1'
    implementation 'me.dm7.barcodescanner:zxing:1.8.4'
    implementation 'com.github.ganfra:material-spinner:2.0.0'
    implementation 'com.android.support:support-v4:27.1.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
}
```

Slika 3.3. Implementiranje vanjskih biblioteka

Jedan od glavnih zahtjeva kojeg su postavili budući korisnici aplikacije je da

kretanje kroz aplikaciju bude jednostavno.



Slika 3.4. Prikaz kretanja kroz aktivnosti aplikacije

Prema slici 3.4. i prikazanom tzv. Application flow-u vidi se da je zahtjev za jednostavnoću kretanja kroz aplikaciju ispunjen.

Kao ikona aplikacije korištena je ikona prema slici 3.5. Preuzeta je s web izvora *https://pngimage.net/logo-shopping-png-3/*. Slika je spremljena u mipmap mapi sa nastavkom .png. U datoteci AndroidManifest se postavlja ikona aplikacije prema slici 3.6.



Slika 3.5. Ikona aplikacije

```
android:name=".BarcodeShopApplication"
android:allowBackup="true"
android:icon="@mipmap/app_icon"
android:label="BarcodeShop"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
```

Slika 3.6. Postavljanje ikone aplikacije u AndroidManifest

Kada se pritisne ikona aplikacije pojavljuje se *splash screen* (Slika 3.7.) koji traje određen broj sekundi. To je uvod za početni zaslon koji slijedi poslije. On nema funkcionalnost u samom radu aplikacije, ali utječe na ukupni dojam korisnika o

aplikaciji. Koliko će taj *splash screen* trajati, šta će na njemu biti i ostali parametri određeno je u SplashActivity.java i u activity_splash.xml čiji su kodovi prikazani na slikama 3.8. i 3.9..



Slika 3.7. Splash screen

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"
    android:background="@drawable/first"
    android:orientation="vertical"
    android:gravity="center_vertical"
    tools:context="activity.SplashActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:textSize="40sp"
        android:text="BarcodeShop"
        android:layout_gravity="center"
        android:textStyle="bold"/>
</LinearLayout>
```

Slika 3.8. XML kôd za splash screen

```
public class SplashActivity extends AppCompatActivity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);

       requestWindowFeature(Window.FEATURE_NO_TITLE);
       getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
               WindowManager.LayoutParams.FLAG_FULLSCREEN);
       getSupportActionBar().hide();

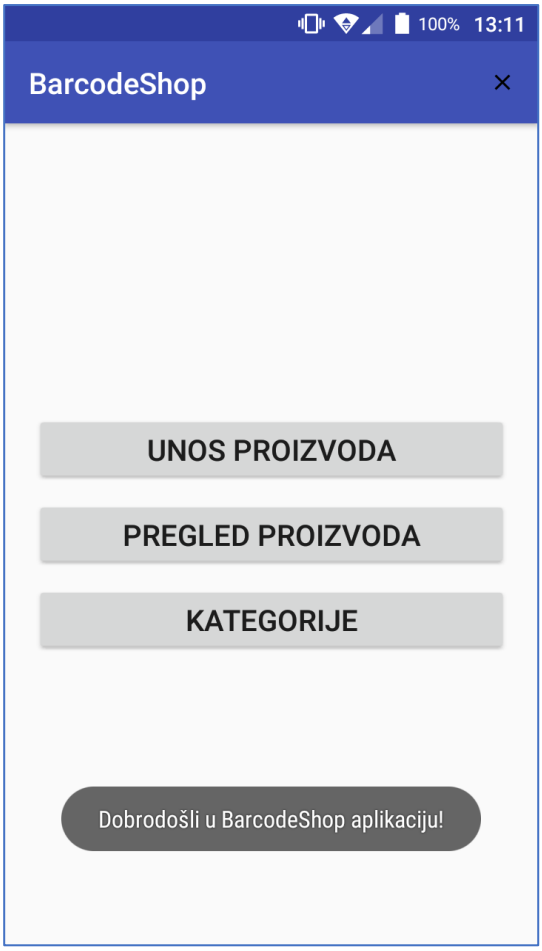
       setContentView(R.layout.activity_splash);

       int tajmer = 3000;

       new Handler().postDelayed(new Runnable() {
           @Override
           public void run() {
               Intent i = new Intent(SplashActivity.this, MainActivity.class);
               startActivity(i);
               finish();
           }
       },tajmer);
   }
}
```

Slika 3.9. Java kôd za splash screen

Nakon *splash screen-a* pojavljuje se početni zaslon (Slika 3.10.) na kojem se se vidi naziv aplikacije, gumb X, te tri gumba (UNOS PROIZVODA, PREGLED PROIZVODA I KATEGORIJE). Također, na kratko se pojavljuje skočna (eng. *Toast*) ulazna poruka s tekstom „Dobrodošli u BarcodeShop aplikaciju!“.



Slika 3.10. Početni zaslon aplikacije

Gumbi UNOS PROIZVODA, PREGLED PROIZVODA I KATEGORIJE kreirani su u *activity_main.xml* (Slika 3.11.). U klasi *MainActivity.java* (Slika 3.12.) dohvaćaju se napravljeni gumbi i definiraju. S obzirom da gumbi reagiraju na pritisak moraju se napraviti tzv. slušači događaja. Kada aplikacija otkrije pritisak nekog od ova tri gumba poziva se onClick () metoda.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    tools:context=".activity.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_marginRight="20dp"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:layout_weight="3"
        android:orientation="vertical"
      >
        <Button
            android:id="@+id/btnImportProduct"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:text="UNOS PROIZVODA"
            android:layout_marginBottom="10dp"
            />
        <Button
            android:id="@+id/btnViewProduct"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:text="PREGLED PROIZVODA"
            android:layout_marginBottom="5dp"
            />
        <Button
            android:id="@+id/btnCategory"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:layout_marginTop="5dp"
            android:textSize="20sp"
            android:text="KATEGORIJE"
            />
    </LinearLayout>
</LinearLayout>
```

Slika 3.11. XML kôd za kreiranje gumba

```
public class MainActivity extends AppCompatActivity {

    int PERMISSION_ALL = 1;
    String[] PERMISSIONS = {Manifest.permission.CAMERA};

    Button importProductButton;
    Button viewProductButton;
    Button categoryButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toast.makeText(this, "Dobrodošli u BarcodeShop aplikaciju!", Toast.LENGTH_LONG).show();

        importProductButton = (Button) findViewById(R.id.btnImportProduct);
        viewProductButton = (Button) findViewById(R.id.btnViewProduct);
        categoryButton = (Button) findViewById(R.id.btnCategory);

        setListeners();

        if(!PermissionManager.hasPermissions(this, PERMISSIONS)){
            ActivityCompat.requestPermissions(this, PERMISSIONS, PERMISSION_ALL);
        }
    }


    private void setListeners()
    {
        importProductButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startImportProductActivity();
            }
        });

        viewProductButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startViewProductActivity();
            }
        });

        categoryButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startCategoryActivity();
            }
        });
    }

    private void startCategoryActivity()
    {
        Intent intent = new Intent(MainActivity.this, CategoryActivity.class);
        startActivity(intent);
    }

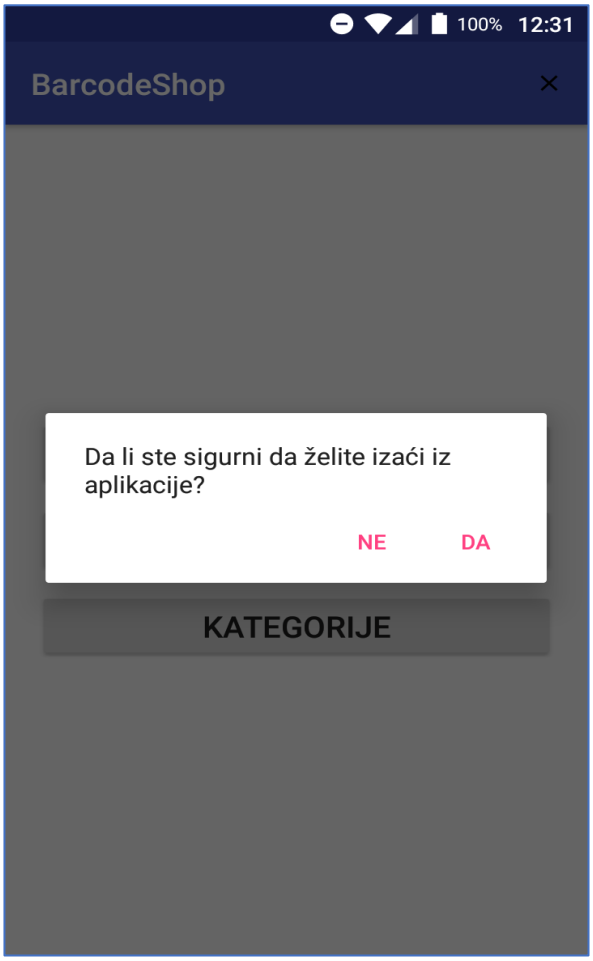
    private void startImportProductActivity()
    {
        Intent intent = new Intent(MainActivity.this, ProductActivity.class);
        startActivity(intent);
    }

    private void startViewProductActivity()
    {
        Intent intent = new Intent(MainActivity.this, ViewProductActivity.class);
        startActivity(intent);
    }
```

Slika 3.12. Kôd u MainActivity.java

Gumb X (u kodu označen kao *action_mode_close_button*) služi korisniku kao mogućnost izlaska iz aplikacije. Pritiskom na X iskoči dijalog upozorenja (eng. *Alert*

Dialog) kao na slici 3.13.. To je prozorčić koji od korisnika traži da donese odluku želi li izaći iz aplikacije ili ne. Kako bi se u MainActivity.java mogao pozvati taj gumb i kodom mu odrediti ulogu, bitno ga je prije toga stvoriti u menu_main.xml (Slika 3.14.). Kôd je prikazan na Slici 3.15..



Slika 3.13. Mogućnost izlaska iz aplikacije

```
<item
    android:id="@+id/action_mode_close_button"
    android:icon="@drawable/ic_close"
    android:orderInCategory="100"
    android:title="Close"
    app:showAsAction="always" />
```

Slika 3.14. Gumb X u menu_main.xml

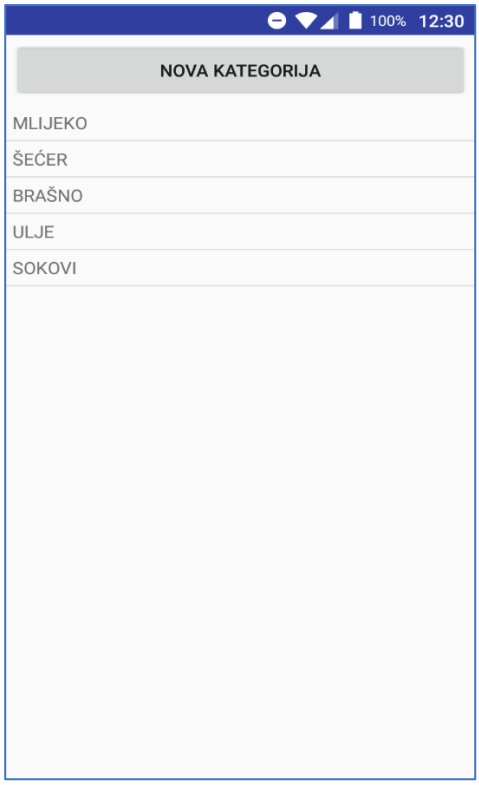
```
public boolean onCreateOptionsMenu(Menu menu) {
     //return super.onCreateOptionsMenu ( menu );
     MenuInflater inflater = getMenuInflater ();
     inflater.inflate ( R.menu.menu_main , menu );
     return true;
 }

 @Override
 public boolean onOptionsItemSelected(MenuItem item) {
     int id = item.getItemId();
     if(id == R.id.action_settings)
     {
        return true;
     }
     else if (id==R.id.action_mode_close_button) {
         AlertDialog.Builder builder = new AlertDialog.Builder(this);
         builder.setMessage("Da li ste sigurni da želite izaći iz aplikacije?")
                 .setCancelable(false)
                 .setPositiveButton("Da", new DialogInterface.OnClickListener() {
                     @Override
                     public void onClick(DialogInterface dialog, int which) {
                         MainActivity.this.finish();
                     }
                 })
                 .setNegativeButton("Ne", new DialogInterface.OnClickListener() {
                     @Override
                     public void onClick(DialogInterface dialog, int which) {
                         dialog.cancel();
                     }
                 });
         AlertDialog alert = builder.create();
         alert.show();
     }

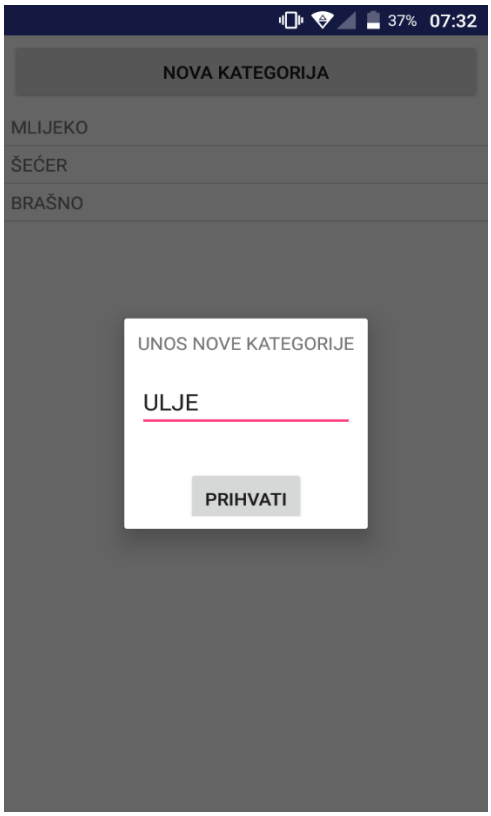
     return super.onOptionsItemSelected(item);
 }
```

Slika 3.15. Kôd dijaloga upozorenja za izlazak iz aplikacije

Ako korisnik stisne gumb KATEGORIJE otvara mu se novi prozor na kojem može napraviti novu kategoriju proizvoda i vidjeti već postojeće (Slika 3.16.). Tu se nalazi gumb NOVA KATEGORIJA preko kojeg korisnik pravi/unosi novu kategoriju proizvoda (Slika 3.17.).

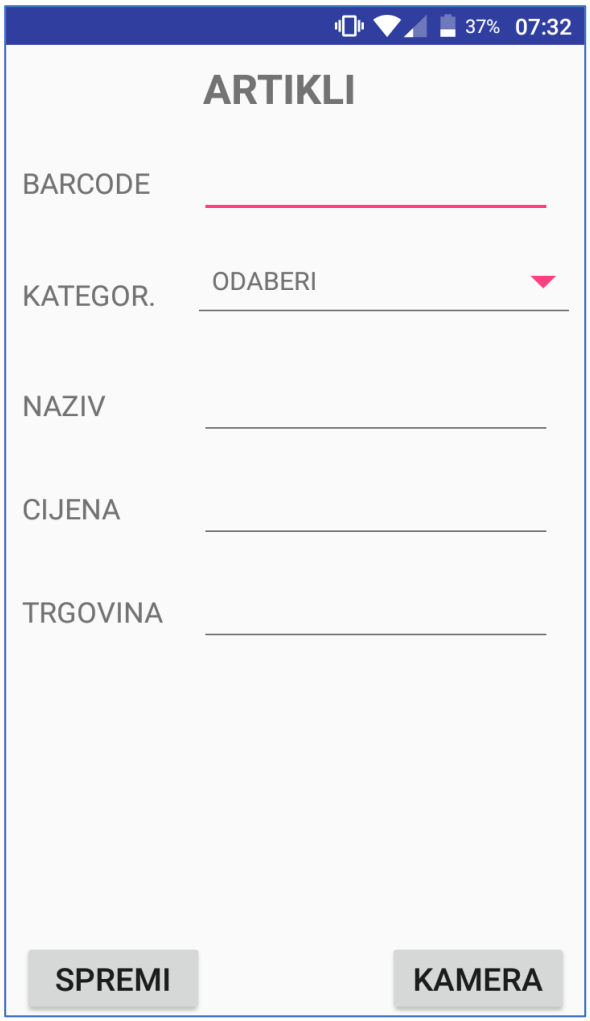


Slika 3.16. Prozor KATEGORIJE



Slika 3.17. Unos nove kategorije

Ako korisnik na početnom zaslonu stisne gumb UNOS PROIZVODA otvara mu se novi zaslon (Slika 3.18.) na kojem se osim podataka koje treba unjeti nalaze i dva gumba: SPREMI i KAMERA. Pritiskom gumba KAMERA otvara se kamera mobilnog uređaja preko koje se očita barkod proizvoda. U slučaju uspješnog očitavanja barkoda on se automatski pojavljuje na prošlom zaslonu i mogu se unjeti ostali traženi podatci. Nakon unošenja tih podataka korisnik treba stisnuti gumb SPREMI i podatci se spreme u bazu podataka. U slučaju da neki podatak nije unešen pojaviti će se skočna poruka sa tekstom „Greška kod upisa podataka!“.



Slika 3.18. Unos proizvoda

Kod izbora kategorije imamo padajući izbornik koji je napravljen u datoteci *fragment_product.xml*. Napravljen je *MaterialSpinner* (Slika 3.19.) za čije je korištenje, kao što je već ranije u radu spomenuto, implementirana vanjska biblioteka.

Kada aplikacija otkrije pritisak gumba KAMERA poziva se onClick() metoda (Slika 3.20.).

```
<fr.ganfra.materialspinner.MaterialSpinner
    android:id="@+id/spinCategory"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:ms_arrowColor="@color/colorAccent"
    android:layout_gravity="top"
    app:ms_arrowSize="16dp"
    app:ms_enableFloatingLabel="false"
    app:ms_hint="              ODABERI"
    app:ms_multiline="true" />
```

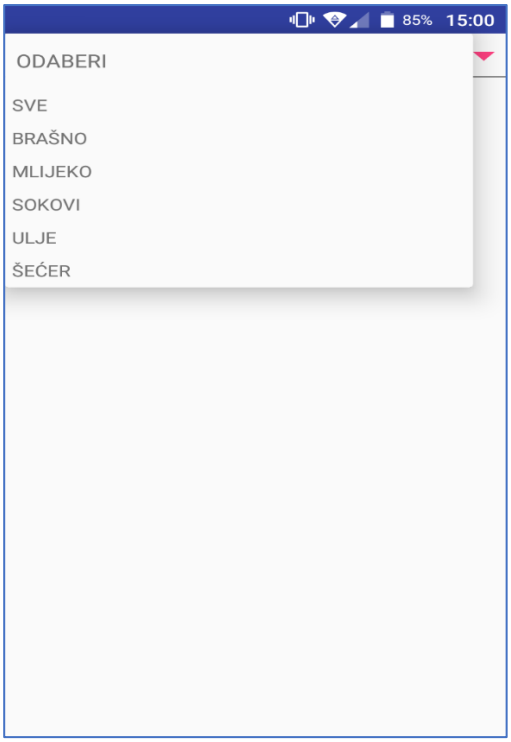
Slika 3.19. XML kôd za MaterialSpinner

```
btnCamera.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
              startCameraActivity();
          }

private void startCameraActivity()
   {
       Intent i = new Intent(ProductActivityFragment.this.getActivity(), QRScanActivity.class);
       ProductActivityFragment.this.startActivityForResult(i, QRScanActivity.QR_SCAN_REQUEST_CODE);
   }
```

Slika 3.20. onClick() metoda za gumb KAMERA

Ako korisnik na početnom zaslonu stisne gumb PREGLED PROIZVODA pojavljuje mu se prozor sa padajućim izbornikom na kojem može odabrati listu svih unešenih proizvoda ili samo listu proizvoda koji se nalaze u određenoj kategoriji (Slika 3.21. i 3.22.). Također je kod padajućeg izbornika korišten *MaterialSpinner*.



Slika 3.21. Prikaz svih kategorija i proizvoda



Slika 3.22. Prikaz svih proizvoda unutar kategorije MLIJEKO

Za pravljenje liste koristio se *ListView* u datoteci *fragment_view_product.xml* prema slici 3.23..

```
<ListView
    android:id="@+id/lv_product_list"
    android:layout_width="match_parent"
    android:longClickable="true"
    android:clickable="true"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:layout_marginBottom="10dp"
    android:drawSelectorOnTop="true"
    android:fastScrollEnabled="true"
    android:overScrollMode="never"
    android:padding="0dp"
    android:scrollbarStyle="outsideOverlay" />
```

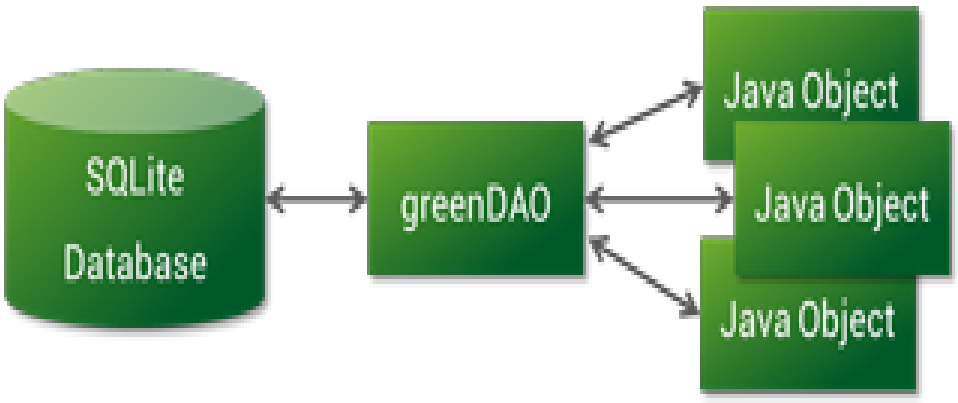
Slika 3.23. XML kôd za ListView

3.2. Baza podataka

Za pravljenje baze podataka koristio se greenDAO.

3.2.1. greenDAO

GreenDAO je alat za objektno/relacijsko mapiranje (ORM) kod Androida. Prema [9] greenDAO, jednostavno rečeno, je lagano i brzo ORM rješenje za mapiranje objekata u SQLite baze podataka (Slika 3.24.). ORM alati kao greenDAO obavljaju mnoge zadatke koji se pojavljuju i nude jednostavno sučelje za korisnikove podatke. Postoje i drugi ORM alati kao što su OrmLite, SugarORM, Active Android, Realm, no greenDAO je vjerojatno najpopularniji među programerima. Glavni ciljevi greenDAO dizajna su: jednostavni API-ji, smanjenje opterećenja memorije, moderna knjižnica, povećanje učinkovitosti, vrlo je optimiziran za Android.



Slika 3.24. Odnos između greenDAO, SQLite baze podataka i Java objekata

U ovom radu su u Android Studio napravljene klase Product i Category gdje se definiraju tablice u bazi iz te klase. Napravljene su još dvije pomoćne klase DatabaseHelper i UpgradeDatebaseHelper. GreenDAO iz svega toga sam onda kreira bazu kada se pozove u BarcodeShopApplication klasi (Slika 3.25.).

```
public class BarcodeShopApplication extends Application {

   @Override
   public void onCreate()
   {
       super.onCreate();

       try
       {
           DatabaseHelper.initialize(BarcodeShopApplication.this);

           Stetho.initialize(Stetho.newInitializerBuilder(this).enableDumpapp(Stetho.defaultDumperPluginsProvider(this))
                   .enableWebKitInspector(Stetho.defaultInspectorModulesProvider(this)).build());
       }
       catch (Exception e)
       {
           Log.d("Error", e.getMessage());
       }
   }
}
```

Slika 3.25. Kôd u BarcodeShopApplication klasi

4. ZAKLJUČAK

Upotreba mobilnih aplikacija postala je neizostavna svakodnevnica u svim dijelovima života. Izrada ovog završnog rada zahtijevala je dosta istraživanja i proučavanja pripadnih tehnologija. U fazi istraživanja posjećene su mnoge trgovine i trgovački lanci u svrhu saznavanja podatka može li se očitavati kôd proizvoda mobitelom unutar same trgovine. Također su ispitane želje i zahtjevi budućih korisnika ove aplikacije. Android Studio korišten je kao alat za razvoj aplikacije. Dosta je popularan među programerima. Aplikacija je zamišljena kao pomoć pri kupovini kako bi se skratilo vrijeme za pronalazak određenog proizvoda i njegove cijene. Ljudi dosta vremena gube obilazeći trgovine, lutajući po njima i tražeći najpovljnije cijene. Također je mnogim ljudima zamorno prelistavanje mnogobrojnih reklamnih letaka u potrazi za željenim proizvodom. Ovom aplikacijom korisnik može sam napraviti kategorije proizvoda koje ga zanimaju, unjeti cijene iz pojedinih trgovina i na taj način u svakom trenutku preko mobitela znati gdje želi kupiti neki određeni proizvod. Ono što bi ovu aplikaciju učinilo još funkcionalnijom i privlačnijom većoj masi ljudi je kada bi se preko nje mogle pratiti promjene cijena unesenih proizvoda. Znači, aplikacija sada ima svoju funkciju, ali ima i mogućnost daljneg razvijanja u budućnosti.

LITERATURA

[1] Wikipedija, Android, dostupno na: https://hr.wikipedia.org/wiki/Android_(operacijski_sustav), (ožujak 2019.)

[2] D. Radić, Mobilni uređaji - Android operativni sustav, dostupno na: https://informatika.buzdo.com/pojmovi/mobile-3.htm, (ožujak 2019.)

[3] Skupina autora, Sigurnost operacijskog sustava Android 4.0, CIS, 2012., dostupno na: https://www.cis.hr/dokumenti/sigurnostoperacijskogsustavaandroid40.html, (ožujak 2019.)

[4] Meet android studio, dostupno na: https://developer.android.com/studio/intro, (ožujak 2019.)

[5] Wikipedija, Android, dostupno na: https://hr.wikipedia.org/wiki/Android_(operacijski_sustav), (ožujak 2019.)

[6] Java, dostupno na: https://www.java.com/, (ožujak 2019.)

[7] Wikipedija, XML, https://hr.wikipedia.org/wiki/XML, (ožujak 2019)

[8] GreenDAO, dostupno na: http://greenrobot.org/greendao/, (ožujak 2019.)

[9] ORM greenDAO, dostupno na: https://imtangqi.com/2015/07/26/orm-greendao-summary/, (ožujak 2019.)

SAŽETAK

U ovom završnom radu razvijena je Android mobilna aplikacija za pomoć pri kupovini nazvana BarcodeShop. Korisnicima je omogućeno pravljenje kategorija po želji i unošenje proizvoda u te iste kategorije. Kasnije u svakome trenutku mogu pristupiti sačuvanim podatcima. Za izradu aplikacije korišten je Java programski jezik unutar Android Studia. U teorijskom dijelu objašnjene su tehnologije koje su korištene pri izradi aplikacije. Također je objašnjen i način upotrebe aplikacije. Sve navedeno je popraćeno slikama kako bi se što točnije dočaralo ono što je objašnjeno u tekstu.

Ključne riječi: Android, aplikacija, kupovina

ABSTRACT

Android product price tracking application

In this bachelor's thesis, an Android mobile application for buying assistance called BarcodeShop was developed. Customers are enabled to create the desired category and place the products in the same category. Later they can access saved data at any time. The Java program language was used to create the application within Android Studio. Theoretical part explains the technologies used in the design of the application. The way the app is used is also explained All of the above is accompanied by pictures to accurately illustrate what is explained in the text.

Keywords: Android, apps, shopping

ŽIVOTOPIS

Antun Čopčić rođen je 13. lipnja 1985. godine u Vinkovcima. Pohađao je osnovnu školu Ivana Gorana Kovačića u Vinkovcima. Nakon završenog osnovnoškolskog obrazovanja upisuje gimnaziju Matije Antuna Reljkovića u Vinkovcima, prirodoslovno-matematički smjer. 2014.godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, preddiplomski sveučilišni studij, smjer računarstvo.

_____

Antun Čopčić