Primjena informacijsko komunikacijske tehnologije kao promocijskog alata u svrhu realizacije marketinške strategije na primjeru hrvatske turističke destinacije

Blavicki, Anamarija

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:200:670737

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

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 studij elektrotehnike

PRIMJENA INFORMACIJSKO KOMUNIKACIJSKE TEHNOLOGIJE KAO PROMOCIJSKOG ALATA U SVRHU REALIZACIJE MARKETINŠKE STRATEGIJE NA PRIMJERU HRVATSKE TURISTIČKE DESTINACIJE

Završni rad

Anamarija Blavicki

Osijek, 2017.

SADRŽAJ

1. UVOD ... 1
1.1. Zadatak završnog rada ... 2
2. ANALIZA KORIŠTENIH TEHNOLOGIJA ZA IZRADU APLIKACIJE ... 3
2.1. Android operacijski sustav ... 3
2.2. Android studio ... 4
2.3. SQLite ... 4
2.4. Programski jezik Java ... 5
3. REALIZACIJA MOBILNE APLIKACIJE ... 6
3.1. Ideja i osnovni koncept ... 6
3.2. Izrada aplikacije ... 7
3.2.1. SQLite baza podataka ... 8
3.2.2. RecyclerView ... 10
3.2.3. Prikaz popisa i detalja ... 12
3.3. Testiranje aplikacije ... 15
4. ANALIZA APLIKACIJE ... 21
5. ZAKLJUČAK ... 23
LITERATURA ... 24
SAŽETAK ... 25
ABSTRACT ... 26
ŽIVOTOPIS ... 27
PRILOZI ... 28

1. UVOD

U današnje doba sve većeg razvoja informacijskih i komunikacijskih tehnologija, sve češća je upotreba mobilnih aplikacija u razne koristi. Upravo iz tog razloga, cilj ovog završnog rada je izraditi aplikaciju za Nacionalni park Paklenica. Park je 1949. proglašen zaštićenim područjem te se prostire na površini od 95 kilometara kvadratnih. Posjetitelji imaju mogućnost planinarenja raznim stazama, penjanja raznim smjerovima te educiranja o kulturnoj i prirodnoj baštini na području Paklenice. [1] Godišnje Paklenicu posjeti oko 80 tisuća posjetitelja što ju stavlja na četvrto mjesto najposjećenijih nacionalnih parkova u Hrvatskoj. Najčešći posjetitelji su strani državljani što nije slučaj samo u Nacionalnom parku Paklenica nego i u ostalim nacionalnim parkovima i parkovima prirode. [2] Upravo ta aplikacija će posjetiteljima i korisnicima omogućiti pregled biljnog i životinjskog svijeta koji se nalazi unutar parka te im ponuditi detalje o svakoj vrsti biljke ili životinje koju je moguće vidjeti u nacionalnom parku. Popis biljaka i životinja se odnosi na glavnu stazu kroz kanjon Velike Paklenice, koja se prostire od ulazne recepcije do planinarskog doma. Aplikacija također ima svrhu promoviranja nacionalnog parka te uz ostale, već primijenjene, tehnologije privući što veći broj posjetitelja i time nadopuniti hrvatski turizam i turističku ponudu. Uz promociju, aplikacija potiče edukaciju o životinjskom i biljnom svijetu koji se nalazi u parku te potiče veću ekološku osviještenost o nužnosti čuvanja prirodnih bogatstava. Kroz aplikaciju je istaknuta prisutnost zaštićenih vrsta biljaka i životinja u Nacionalnom parku Paklenica, a time i posebnost ove turističke destinacije.

Završni rad koncipiran je u pet poglavlja kroz koje je detaljno opisan postupak izvršavanja danog zadatka. Nakon uvoda u rad u prvom poglavlju, u drugom poglavlju su opisane korištene tehnologije poput Androida, Android Studia, SQLite-a te Java programskog jezika i ono predstavlja teorijsku podlogu potrebnu za razumijevanje ostatka rada. U trećem poglavlju, opisana je realizacija same aplikacije. Detaljnije su objašnjene metode i postupci izrade svake komponente od koje se aplikacija sastoji, kao i izrada popisa i baze podataka te unos raznih podataka u bazu koja sadrži detalje o biljkama i životinjama. Također je u trećem poglavlju provedeno testiranje aplikacije te je prikazan rad i funkcionalnost aplikacije na raznim uređajima. Zaključno, u četvrtom poglavlju je analizirana učinkovitost aplikacije budući da je ona zamišljena kao način promocije te kao instrument kojim će Nacionalni park Paklenica privući što veći broj posjetitelja.

1.1. Zadatak završnog rada

Napraviti analizu primjene informacijsko komunikacijske tehnologije u svrhu promocije nacionalnih parkova Hrvatske, te na primjeru iz prakse pokazati novi oblik promocije. Potrebno je izraditi Android aplikaciju koja bi koristila posjetiteljima Nacionalnog parka Paklenica kao vodič od Ulazne recepcije 1 preko glavne staze kroz kanjon Velike Paklenice. Unesena lista vrsta biljaka i životinja koje se mogu sresti kroz Nacionalni park bi bila podijeljena u kategorije prema godišnjem dobu. Koristeći aplikaciju posjetitelji bi mogli odabirom na vrstu biljke ili životinje saznati podatke o njoj prema bazi Natura 2000 vrste.

2. ANALIZA KORIŠTENIH TEHNOLOGIJA ZA IZRADU APLIKACIJE

Unutar poglavlja objašnjene su tehnologije koje su korištene pri izradi aplikacije, kako bi se omogućilo lakše razumijevanje korištenih pojmova. Detaljnije je opisan Android operacijski sustav, kao i njegova službena integrirana okolina za razvoj – Android Studio. Također je objašnjena Java kao jezik koji se koristi unutar te razvojne okoline, kao SQLite – program koji omogućava upravljanje bazama podataka.

2.1.Android operacijski sustav

Andy Rubin, Rich Miner, Nick Sears i Chris White 2003. godine izrađuju prvi operacijski sustav za mobilne uređaje – Android temeljen na inačici Linuxa te ga 2005. godine kupuje Google i nastavlja njegov razvoj. Prva inačica Android operacijskog sustava bila je 1.0 nakon čega Google razvija novije i poboljšane inačice koje prate napredak mobilnih uređaja. Android je otvoreni operacijski sustav što znači da omogućava korisnicima prepravljanje, mijenjanje i poboljšavanje izvornog koda i njegovog sadržaja te se aplikacije tretiraju ravnopravno. Aplikacije za Android OS su pisane pomoću *Android software development kita* (SDK) i u Java programskom jeziku te za njihovo pokretanje se radi Dalvik virutalni stroj budući da na takvim uređajima ne postoji Java virtualni stroj koji služi za izvršavanje Java bajtnog koda. Budući da je sustav napisan u C/C++ programskom jeziku, njegove aplikacije mogu biti napisane upravo u tom jeziku s napomenom da se tada ne upotrebljava SDK nego *Native Development kit* (NDK) kako bi se izvršio kod napisan u Androidovom razvojnom kodu. Svakim danom se razvija sve više i više aplikacija namijenjenih za Android operacijske sustave te se one mogu preuzeti s Google Play trgovine koja je instalirana na svim Android uređajima. Upravo te aplikacije čine krajnji sloj arhitekture Androida koji je vidljiv korisnicima. Sustav je modularan te prilagođen svim pametnim telefonima te uređajima s dodirnim zaslonom kao i raznim pametnim uređajima poput televizija, automobila te ručnih satova, a pri povezivanju koristi razne tehnologije poput *Bluetootha*, LTE-a i WiMAX-a. Korisnikovo sučelje se uglavnom temelji na izravnoj manipulaciji pomoću dodira kao na primjer klizanje ili tipkanje. Također sadrži i dodatne unutarnje hardvere poput GPS-a, akcelometra, žiroskopa, termometra i raznih drugih pogodnosti koje olakšavaju i omogućavaju korisniku raznovrsno korištenje pametnih uređaja. [3]

2.2.Android studio

Android Studio je službena integrirana okolina za razvoj (IDE) Android aplikacija budući da u sebi sadrži temeljne pakete za razvoj aplikacija te se mogu dodati i drugi potrebni paketi. Temelji se na IntelliJ IDEA softveru te je dostupan za Windows, iOS i Linux operacijske sustave. Jedna od bitnijih značajki Android Studia je fleksibilni *build* sustav nazvan *Gradle* koji se temelji na JVM (engl. *Java Virtual Machine*) build sustavima te na taj način izvorne programske datoteke, nakon kompajliranja i linkanja, komprimira u jednu datoteku (.apk) koju Android sustav dalje koristi pri implementaciji aplikacije na uređaje. Program također nudi brzi emulator koji emulira pametni telefon na kojem omogućava testiranje napisanog koda bez korištenja stvarnog uređaja. Omogućena je ujedinjena okolina za razvijanje aplikacija za sve vrste Android uređaja te Lint alati koji pomažu pri praćenju kvalitete koda. Korisniku je olakšan razvoj čestih značajki aplikacija pomoću raznih predložaka koda te GitHub integracije koja omogućava uvoz raznih primjera koda. [4]

Svaki projekt sadrži jedan ili više modula s datotekama koje sadrže izvorne kodove, a svaki modul sadrži sljedeće datoteke: manifest koja sadrži *AndroidManifext.xml* datoteku, java koja sadrži izvorne kodove aplikacije te *res* mapu koja sadrži sve resurse koji ne spadaju u kod aplikacije poput XML tlocrta, UI stringove te razne fotografije. [5]

2.3.SQLite

SQLite je programski sustav koji omogućava upravljanje relacijskim bazama podataka, a sadržan je u C programskoj biblioteci. Sadrži korisničko sučelje koje se lako koristiti, a podatci se mogu dijeliti između više računala. Model mu se ne temelji na klijent-poslužitelj arhitekturi, nego je baza ugrađena u krajnji program, a kod se izvršava pri pozivu. SQLite koristi SQL (engl. *Structured query language*) sintaksu te može podržavati velike baze i nizove podataka. Sustav je ugrađen u Android sustave te ne zahtijeva nikakvu dodatnu prilagodbu i administraciju. Za korištenje baze podataka je potrebno samo definirati upite za kreiranje i uređivanje nakon čega se bazom može upravljati iz Androida. SQLite podržava samo određene tipove podataka te svi podatci moraju biti zapisani u *text, integer"* i *real* tipu podatka kako bi bili uspješno spremljeni u bazu. Baza također zapisuje podatke u obliku *blob* ili *null*. [6]

2.4.Programski jezik Java

Java je objektno orijentirani programski jezik koji se temelji na programskom jeziku Oak, a razvijen je 1995.godine. Java korisniku olakšava razvoj programa budući da sadrži potporu za višeplatformske sustave te se svi programi pisani u Javi izvode pomoću Java virtualnog stroja (JVM – *Java Virtual Machine*). Jezik sadrži pravila pisanja koda, ključne riječi pomoću kojih jezični prevodilac pretvara izvorni kod u izvršni byted kod za JVM. Samom kodu je potreban skup bibloteka koje omogućavaju korištenje željenih funkcija te su one dostupne kao dio Java platforme i opisane u dokumentu *Java Application Programming Interface* (API). Dakle, programi se pišu Java jeziku (.java datoteke) te ih je potrebno pretvoriti, pomoću programa javac (java *compiler*), u bajtkod (.class datoteka) kako bi se one izvršile. Dobiveni bajtkod predstavlja izvršni kod za Java virtualni stroj. [7]

Jezik prepoznaje dvije vrste podataka: primitivne tipove te reference. Primitivni tipovi obuhvaćaju *byte, short, int, long, char, float, double* i *boolean* te je nad njima moguće djelovati raznim operatorima. Stringovi nisu uvršteni pod tip primitivnih podataka te je njih potrebno pohraniti kao primjerak razreda. Jezik također podržava i uvjetno izvođenje koda te se mogu koristiti *if-else* konstrukti. [8]

Java svojim korisnicima omogućava laki i brzi razvoj programa uz što veći stupanj sigurnosti i pouzdanosti te je upravo iz tog razloga jedan od popularnijih programskih jezika za razvoj aplikacija za sve vrste pametnih uređaja.

3. REALIZACIJA MOBILNE APLIKACIJE

U ovom poglavlju opisan je tok razvoja mobilne aplikacije. Prvenstveno je opisana ideja i osnovni koncept za izradu aplikacije. Dalje se detaljno navode sve kreirane datoteke i klase kao i njihova svrha unutar aplikacije. Također je objašnjen način izrade SQLite baze podataka kao i njeno popunjavanje podatcima dobivenim od Nacionalnog parka Paklenica. Ti podatci se prikazuju unutar popisa biljaka i životinja te je također opisan prikaz detalja o pojedinoj biljci ili životnji. Na kraju je sama aplikacija testirana, a njen izgled je prikazan na jednom stvarnom mobilnom uređaju, jednom emuliranom mobilnom uređaju i na jednom emuliranom tabletu.

3.1. Ideja i osnovni koncept

Ideja za izradu aplikacije dolazi po prijedlogu Nacionalnog parka Paklenica u svrhu promoviranja parka kao značajne turističke destinacije u Republici Hrvatskoj, a time i općenito hrvatskog turizma. Budući da trenutno skoro svaka osoba posjeduje pametni telefon, aplikacija bi trebala približiti svijet Nacionalnog parka svim posjetiteljima te na taj način pružiti im novo iskustvo u istraživanju prirode i planinarenju.

Za izradu aplikacije, korištena je Android Studio razvojna okolina pri čemu se vodilo računa o jednostavnosti aplikacije radi što jednostavnije uporabe. Pri pokretanju aplikacije otvara se početni zaslon koji sadrži logo Nacionalnog parka Paklenica. Zatim se ispisuje upozorenje kojim se posjetiteljima daje na znanje da su sve vrste unutar parka zakonom zaštićene. Također je naglašeno kako je biljne vrste strogo zabranjeno brati ili oštetiti, a životinjske vrste je zabranjeno namjerno uznemiravati. Posjetitelj pritiskom na dugme „Dalje“ koje se nalazi ispod navedenog teksta potvrđuje da je upoznat s pravilima parka te novčanim kaznama koje slijede u slučaju nepoštivanja pravila. Nakon toga dolazi zaslon s ponuđenim godišnjim dobima te odabirom jednog doba aplikacija prikazuje popis biljaka i životinja, na hrvatskom i engleskom jeziku, koje se mogu susresti šetnjom glavnom stazom kroz kanjon Velike Paklenice u odabrano godišnje doba. Klikom na pojedinu biljku ili životinju, korisniku se prikazuje uvećana fotografija odabrane vrste, njezin naziv na hrvatskom, engleskom i latinskom jeziku, detaljniji opis biljke ili životinje koristeći opise iz baze podataka Natura 2000 te lokacija na kojoj se mogu naći to jest dio staze na kojem se nalaze.

3.2.Izrada aplikacije

Unutar Android Studio projekta, nalazi se nekoliko bitnih datoteka koje sadržavaju sve potrebne kodove za realizaciju aplikacije. Ključna datoteka svake aplikacije je „AndroidManifest.xml“ datoteka koja sadrži bitne podatke kao što su ime java paketa, opisi komponenata aplikacije, korištene biblioteke, minimalna razina API-a te razna sigurnosna dopuštenja. [7] Također je bitno napomenuti da Android SDK ne uključuje određene klase te je njihove zavisnosti (engl. *support library gradle dependency*) potrebno dodati u datoteku „build.gradle“ unutar projekta. Sljedeći folder koji je značajan za projekt je java koji sadrži kodove za izvođenje aplikacije, a podijeljen je u 11 klasa – „Data“, „DatabaseHelper“, „Detalji“, „Dobrodosli“, „MainActivity“, „MyAdapter“, „Napomena“, „PopisP“, „PopisLj“, „PopisJ“ i „PopisZ“. Posljednja skupina datoteka je „res“ folder unutar koje se nalaze „drawable“ folder u kojoj su smještene sve fotografije korištene unutar projekta, „layout“ folder koji sadrži sve .xml datoteke kojima se definira izgled aplikacije te „values“ folder unutar kojeg se nalazi važna *strings.xml* datoteka koja sadrži tekstualne vrijednosti (stringove) koji se koriste u aplikaciji.

Prilikom otvaranja aplikacije, prikazuje se početni zaslon sa porukom dobrodošlice čiji se izgled – „activity_dobrodosli.xml“ i klasa – „Dobrodosli“ vide na slici 3.1.

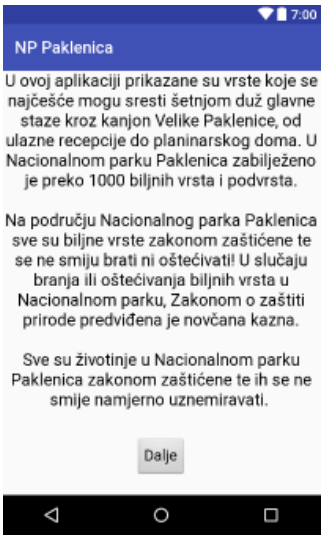
```
public class Dobrodosli extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_dobrodosli);
        Thread myThread = run() → {
                try {
                    sleep(1500);
                    Intent startMainScreen = new Intent(getApplicationContext(), Napomena.class);
                    startActivity(startMainScreen);
                    finish();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
        };
        myThread.start();
    }
}
```

Sl. 3.1. Komponente activity_dobrodosli.xml i Dobrodosli.java

Izgled klase je prilično jednostavan i sadrži jedan *ImageView* koji obuhvaća cijeli zaslon te prikazuje logo Nacionalnog parka. Unutar te klase se izvršava metoda „run()“ pomoću koje se taj zaslon zadržava određeno vrijeme što je u ovom slučaju definirano na 1500ms, a zatim preusmjerava aplikaciju na klasu „Napomena“.

Zaslon sa napomenom sadrži jedan *TextView* u kojemu su navedena opća upozorenja i napomene posjetiteljima te gumb koji, ukoliko se pritisne, pokreće metodu „gotoMain“ koja prebacuje aplikaciju na klasu „MainActivity“. Izgled XML datoteke napomene je prikazan na slici 3.2.



Sl. 3.2. Izgled activity_napomena.xml

3.2.1. SQLite baza podataka

Najbitniji dio aplikacije je zapravo baza podataka koja sadrži sve informacije o biljkama i životinjama unutar parka, kao i njihove fotografije. Baza podataka je kreirana pomoću klase „DatabaseHelper“, pod nazivom „nppbaza“ te se sastoji od tablice „popis“. Unutar te tablice je definirano 10 stupaca koji su prikazani na slici 3.3.

```
public class DatabaseHelper extends SQLiteOpenHelper {

    public static final String DATABASE_NAME = "nppbaza.db";
    public static final String TABLE_NAME = "popis";
    public static final String HRV_NAZIV = "hrvatski";
    public static final String ENG_NAZIV = "engleski";
    public static final String LAT_NAZIV = "latinski";
    public static final String OPIS = "opis";
    public static final String LOKACIJA = "lokacija";
    public static final String PROLJECE = "proljece";
    public static final String LJETO = "ljeto";
    public static final String JESEN = "jesen";
    public static final String ZIMA = "zima";
    public static final String SLIKA = "slika";
```

Sl. 3.3. Podatci o bazi

Za kreiranje baze podataka, korištena je metoda „onCreate()“ u kojoj se izvršava SQL naredba za kreiranje tablice – definirani su tipovi podataka svih entiteta, a koriste se dva tipa podataka – *text* i *integer*. Za entitete „hrvatski“, „engleski“, „latinski“, „opis“ i „lokacija“ je korišten *text* tip podatka, a za entitet „slika“ je korišten *integer* budući da se fotografije iz „drawable“ mape dohvaćaju u obliku cijelog broja. Entiteti „proljece“, „ljeto“, “jesen“ i „zima“ su definirani kao *integer-i* kojima se kasnije utvrđuje kojem godišnjem dobu biljka ili životinja pripada. Također je definiran primarni ključ tablice, a on je hrvatski naziv biljke ili životinje budući da ne postoje dvije istoimene biljne ili životinjske vrste. U slučaju potrebne nadogradnje baze podataka, koristi se metoda „onUpgrade()“ koja izbriše postojeću tablicu te kreira novu, ažuriranu tablicu podataka. Upravo te dvije metode, nužne za postojanje baze, su prikazane na slici 3.4.

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table " + TABLE_NAME +
            " (hrvatski TEXT PRIMARY KEY,  engleski TEXT, latinski TEXT, opis TEXT, lokacija TEXT, " +
            "proljece INTEGER, ljeto INTEGER, jesen INTEGER, zima INTEGER, slika INTEGER);");
}

@Override
public void onUpgrade(SQLiteDatabase db, int i, int il) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}
```

Sl. 3.4. „onCreate()“ i „onUpgrade()“ metoda

Kako bi se olakšao unos podataka u bazu, kreirana je metoda „insertData()“ kojoj se predaju sve potrebne vrijednosti koje se nalaze u jednom retku tablice te se zatim te vrijednosti pridružuju određenim entitetima što se vidi na slici 3.5., a na slici 3.6. se uočava i način na koji se koristi spomenuta metoda to jest način na koji se popunjava baza podataka sa vrijednostima dobivenim od uprave Nacionalnog parka Paklenica.

```
public boolean insertData(String hrv, String eng, String lat, String opis, String lokacija, int proljece, int ljeto, int jesen, int zima, int slika) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(HRV_NAZIV, hrv);
    contentValues.put(ENG_NAZIV, eng);
    contentValues.put(LAT_NAZIV, lat);
    contentValues.put(OPIS, opis);
    contentValues.put(LOKACIJA, lokacija);
    contentValues.put(PROLJECE, proljece);
    contentValues.put(LJETO, ljeto);
    contentValues.put(JESEN, jesen);
    contentValues.put(ZIMA, zima);
    contentValues.put(SLIKA, slika);

    long result = db.insert(TABLE_NAME, null, contentValues);

    if(result == -1)
        return false;
    else
        return true;
}
```

Sl. 3.5. Metoda „insertData()“

```
myDb.insertData("Smeđa slezenica", "Common Maidenhair", "Asplenium trichomanes", "Smeđa slezenica je vrsta paprati " +
        "koja raste u pukotinama stijena i zidova. Raste na osunčanim stijenama te na stijenama u šumama gdje često " +
        "prekriva kamene gromade zajedno s mahovinama.", "duž cijele", 1, 1, 1, 1, R.drawable.smeda_slezenica );
myDb.insertData("Crni bor", "Black Pine", "Pinus nigra", "Crni bor je je stablo koje u povoljnim uvijetima može narasti " +
        "do 30m visine, dok u jako lošim ekološkim uvjetima (primjerice izložen jakoj buri) može cijeli svoj životni vijek " +
        "provesti kao malo stabalce visoko 20ak cm. U Nacionalnom parku Paklenica moguće je vidjeti oba oblika, na početnom " +
        "dijelu staze, visoka stabla crnog bora (ta su stabla sađena), a na gornjem dijelu staze autohtone šume crnog bora sa " +
        "stablima raznih veličina i oblika. Nacionalni park Paklenica poznat je po očuvanim prirodnim šumama crnog bora.",
        "donji i gornji dio", 1, 1, 1, 1, R.drawable.crni_bor);
```

Sl. 3.6. Popunjavanje baze podataka

3.2.2. RecyclerView

Popisi biljaka i životinja koje se mogu susresti u Nacionalnom parku izrađeni su pomoću *RecyclerView-a* koji predstavlja fleksibilniju verziju *ListView-a*. Pomoću *RecyclerView-a* se prikazuje sadržaj popisa, ali se prikazuje samo određeni broj stavki popisa, a ostatak se kasnije samostalno učitava po potrebi (engl. *Lazy Load*). Na slici 3.7. prikazane su zavisnosti koje je potrebno dodati u *build.gradle(Module:app)* kako bi se omogućilo prikazivanje popisa.

```
compile 'com.android.support:recyclerview-v7:25.2.0'
compile 'com.android.support:cardview-v7:25.2.0'
```

Sl. 3.7. Dodane zavisnosti *RecyclerView*-a

Ovaj pogled ima mogućnost promjene izgleda redaka koji se nalaze u popisu te je izgled pojedinog retka kreiran u datoteci „custom_row.xml“ i sadrži *ImageView* s id-om „slika“ koji prikazuje fotografiju biljke ili životinje te dva *TextView-a* s id-ovima „hrvatski“ i „engleski“ te oni služe za prikaz hrvatskog i engleskog naziva biljke ili životinje. Vlastito kreirani redak je kreiran unutar posebnog pogleda *CardView-a* koji je vrsta *ViewGroup-e* te on obuhvaća RelativeLayout,

a upravo unutar tog pogleda je definiran raspored *ImageView-a* i *TextView-ova* pomoću specifikacija prikazanih na slici 3.8.

```
<ImageView
    android:id="@+id/slika"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="16dp" />

<TextView
    android:id="@+id/hrvatski"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_toRightOf="@+id/slika"
    android:text="hrvatski"
    android:textSize="30sp" />

<TextView
    android:id="@+id/engleski"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/hrvatski"
    android:layout_toRightOf="@+id/slika"
    android:text="engleski" />
```

Sl. 3.8. Definiranje rasporeda *ImageView*-a i *TextView*-a

Za korištenje *RecyclerView* instance, potrebno je kreirati adapter koji služi povezivanju popisa i klase unutar koje je sastavljen popis. Iz tog razloga, kreirana je klasa „MyAdapter“ koja sadrži metodu „onBindViewHolder()“ koja za parametre prima određeni *holder* unutar kojeg se nalazi sadržaj koji treba prikazati te *integer* koji specificira gdje se taj sadržaj treba pokazati. Željene fotografije bilo je potrebno dodati u datoteku „drawable“. Budući da su fotografije životinja različitih dimenzija, korištena je vanjska biblioteka Picasso pomoću koje je slikama smanjena veličina te su prilagođene veličinama raznih zaslona pri učitavanju. Također unutar klase „MyAdapter“ kreirana je „ViewHolder“ klasa koja obrađuje događaje i radi s grafičkim prikazom jednog retka.

Unutar adaptera je također definirana „onClick()“ metoda, prikazana na slici 3.9., koja omogućava korisniku da, klikom na određenu biljku ili životinju dobije ostatak podataka odnosno detalje o odabranoj stavci. Dakle, ona otvara klasu „Detalji“ unutar koje se prikazuju željene informacije. Pri izvršavanju, ta metoda u varijablu „naziv“ sprema hrvatski naziv biljke ili životinje koju je korisnik kliknuo te će se pomoću nje, budući da je to primarni ključ, kasnije dohvaćati podatci iz baze za prikaz detalja.

```
@Override
public void onClick(View v) {
    Intent intent = new Intent(context, Detalji.class);
    Bundle extras = new Bundle();
    String naziv = hrvatski.getText().toString();
    extras.putString("naziv", naziv);
    intent.putExtras(extras);
    context.startActivity(intent);
}
```

Sl. 3.9. Metoda „onClick()“

3.2.3. Prikaz popisa i detalja

Klasa „MainActivity“ sadrži četiri metode – „gotoProljece()“, „gotoLjeto()“, „gotoJesen()“ te „gotoZima()“ koje se pokreću prilikom klika na jedan od četiri gumba koja se nalaze na ekranu kao što se može vidjeti na slici 3.10.

```
public void gotoProljece (View view) {
    Intent intent = new Intent(this, PopisP.class);
    startActivity(intent);
}

public void gotoLjeto (View view) {
    Intent intent = new Intent(this, PopisLj.class);
    startActivity(intent);
}

public void gotoJesen (View view) {
    Intent intent = new Intent(this, PopisJ.class);
    startActivity(intent);
}

public void gotoZima (View view) {
    Intent intent = new Intent(this, PopisZ.class);
    startActivity(intent);
}
```

Sl. 3.10 Prikaz activity_main.xml i MainActivity.java

Upravo te metode određuju pokretanje nove aktivnosti tj. prelazak na sljedeću klasu. Drugim riječima, ukoliko korisnik klikne na gumb „Proljeće“, pokrenut će se metoda „gotoProljece()“ te će aplikacija prikazati layout klase „PopisP“ koji sadrži popis biljaka i životinja koje posjetitelji mogu naći u parku za vrijeme proljeća. Na slici 3.11. se može vidjeti prikaz XML

koda za proljeće pomoću kojeg se definira funkcija gumba; analogno tome se definiraju funkcije ostalih gumbova „Ljeto“, „Jesen“ i „Zima“.

```
<Button
    android:text="Proljeće"
    android:layout_width="match_parent"
    android:id="@+id/button11"
    android:textSize="20sp"
    android:onClick="gotoProljece"
    android:background="@drawable/proljece"
    android:layout_weight="0.46"
    android:layout_height="65dp" />
```

Sl. 3.11. XML kod za gumb „Proljeće“ unutar „activity_main“ *layouta*

Dakle, nakon što se izvrši metoda „gotoProljece“, prikazuje se layout klase „PopisP“, koji je zapravo *RecyclerView* koji preko adaptera pristupa podatcima te popunjava listom biljaka i životinja koje se mogu pronaći u proljeće. Unutar ove klase je definiran *LinearLayoutManager* pomoću kojeg se definira izgled *RecyclerView-a* te je on u ovom slučaju nalik običnom *ListView-u*. Listu popunjava metoda „getAllProljece()“ koja se nalazi u klasi „DatabaseHelper“. Upravo ta metoda, prikazana na slici 3.12, na temelju podataka unesenih u bazu, odlučuje koje će biljke i životinje biti prikazane u proljetnoj listi. Dakle, sve vrste kojima je dodijeljena vrijednost entiteta „proljece“ jednaka „1“, nalazit će se na listi za proljeće. Kako bi se olakšala razmjena podataka između baze i dijelova aplikacije, kreirana je posebna klasa „Dana“ (Sl. 3.13.) koja sadrži 10 varijabli, sukladno stupcima u bazi, te metode (*getteri i setteri*) za postavljanje i dohvaćanje vrijednosti. Prilikom izvršavanja SQL *query-a*, iz baze se odabiru oni redci koji odgovaraju zadanim uvjetima te se čitaju pojedini stupci (0, 1, 2,..., 9) svakog retka. Pomoću kursora moguće je dohvatiti bilo koju vrijednost nekog retka („cursor.getString()“, „cursor.getInt()“ koje kao parametar primaju redni broj stupca) te se zatim dohvaćena vrijednost sprema u odgovarajuću varijablu objekta klasa „Data“. Dakle pri dohvaćanju podataka iz baze, kreira se *model* – objekt klase „Data“ te se pomoću metoda „setHrvatski()“, „setEngleski()“ i „setSlika()“, vrijednosti, prethodno dohvaćene iz baze pomoću kursora, spremaju u objekt *model*. Svaki ovako kreirani model postavljenim varijablama predstavlja jedan redak tablice nastale kao rezultat gore spomenutog *query-a*. Lista, koja se vraća metodom „getAllProljece()“, popunjena je nizom takvih modela te se u konačnici prikazuje pomoću *RecyclerView-a* u layout-u klase „PopisP“. Ostala godišnja doba se definiraju analogno proljeću.

```
public List<Data> getAllProljece() {
    List <Data> modelList = new ArrayList<>();
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery("select * from " + TABLE_NAME + " where proljece==1 order by " + HRV_NAZIV + " asc", null);

    if (cursor.moveToFirst()){
        do {
            Data model = new Data();
            model.setHrvatski(cursor.getString(0));
            model.setEngleski(cursor.getString(1));
            model.setSlika(cursor.getInt(9));

            modelList.add(model);
        }while (cursor.moveToNext());
    }

    cursor.close();
    return modelList;
}
```

Sl. 3.12. Metoda „getAllProljece()“

```
public class Data {

    public String hrvatski;
    public String engleski;
    public String latinski;
    public String opis;
    public String lokacija;
    public int slika;


    public String getHrvatski() { return hrvatski; }

    public void setHrvatski(String hrvatski) { this.hrvatski = hrvatski; }
```

Sl. 3.13. Klasa "Data"

Klikom na određenu biljku ili životinju se poziva klasa „Detalji“ unutar koje se prikazuju detalji o odabranoj biljci ili životinji. XML datoteka te klase se sastoji od *ScrollView*-a unutar kojeg je definiran *RelativeView* s jednim *ImageView-om* i 5 *TextView-ova* pomoću kojih se prikazuje slika te hrvatski, engleski i latinski naziv, opis i lokacija odabrane vrste. Također je definiran i *PhotoViewAdapter* koji omogućava približavanje i udaljavanje prikazane fotografije u *Imageview-u pinch-anjem* – pomicanjem s dva prsta. U ovoj klasi se poziva metoda iz klase „DatabaseHelper“ pod nazivom „getAllData()“. Ona iz baze odabire redak u kojem je hrvatski naziv jednak varijabli „naziv“ koju prima iz metode „onClick()“ što se i uočava na slici 3.14.

```
public Data getAllData(String naziv) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery("Select * FROM " + TABLE_NAME + " WHERE " + HRV_NAZIV + " =  \"" + naziv + "\"", null);
    Data model = new Data();

    if (cursor.moveToFirst()){
        do {
            model.setEngleski(cursor.getString(1));
            model.setLatinski(cursor.getString(2));
            model.setOpis(cursor.getString(3));
            model.setLokacija(cursor.getString(4));
            model.setSlika(cursor.getInt(9));
        }while (cursor.moveToNext());
    }

    cursor.close();
    return model;
}
```

Sl. 3.14. „getAllData()“ metoda

3.3.Testiranje aplikacije

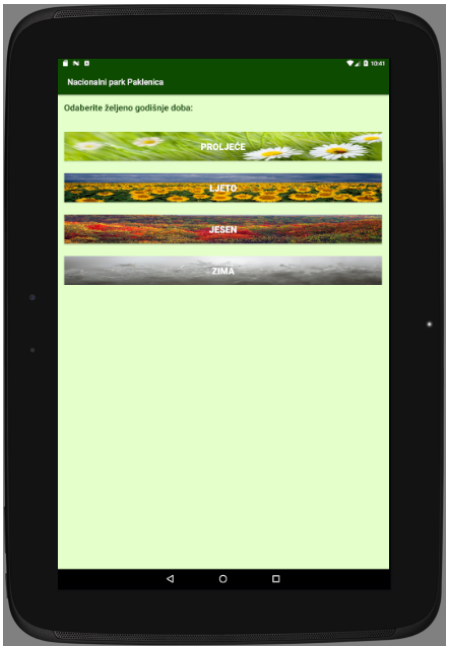
Nakon realizacije, aplikaciju je potrebno testirati kako bi se uvidjela njena funkcionalnost i ispravnost na različitim uređajima. Kroz proces izrade aplikacije korišten je stvarni Android uređaj kako bi se aplikacija pokretala. Nakon što je aplikacija finalizirana, korištena su još dva različita virtualna uređaja (*AVD – Android Virtual Device*) koji su pokrenuti pomoću Android Studio razvojnog alata. U manifest datoteci u Android Studiju, kao ikona aplikacije, postavljen je logo Nacionalnog parka Paklenica te je ona prikazana na slici 3.15.

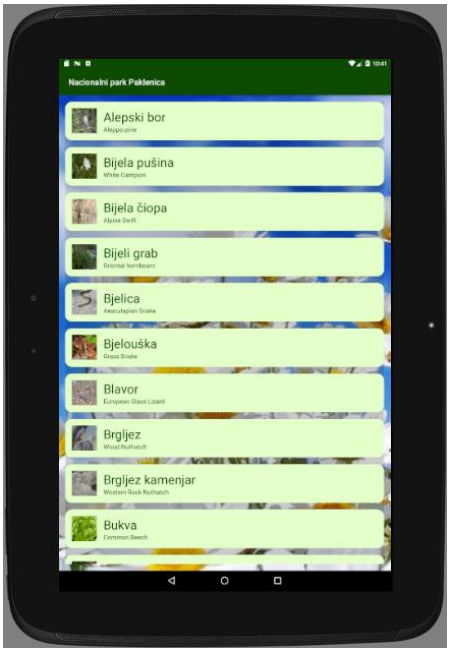
Sl. 3.15. Ikona aplikacije

Pomoću Android Studia, emulirana su dva uređaja – tablet Nexus 10 i mobilni uređaj Galaxy Nexus. Tablet Nexus 10 ima ekran veličine 10.1 inča, a rezolucija mu je 2560x1600 piksela (300ppi), inačica Android sustava je Nougat 7.0. Kao što se može uočiti na slici 3.14, aplikacija je uspješno pokrenuta i na tablet uređaju te zadržava svu svoju funkcionalnost. Budući da je aplikacija prvenstveno dizajnirana za mobilne uređaje, prikaz na tabletu je nešto razvučen, ali zadržava sve svoje karakteristike. Na slikama se može uočiti da je aplikacija funkcionalna i u horizontalnom, a i u vertikalnom položaju tableta.

Nacionalni park Paklenica
PAKLENICA
Nacionalni park
National Park

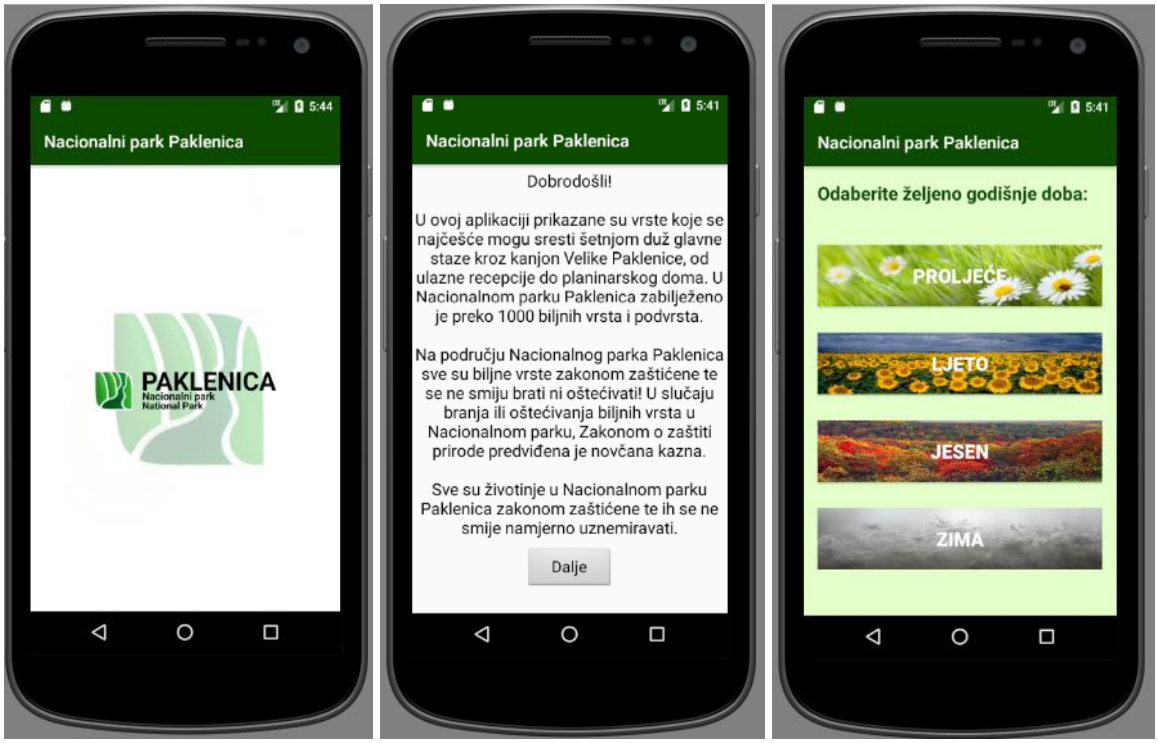
Nacionalni park Paklenica
Dobrodošli!
U ovoj aplikaciji prikazane su vrste koje se najčešće mogu sresti šetnjom duž glavne staze kroz kanjon Velike Paklenice, od ulazne recepcije do planinarskog doma. U Nacionalnom parku Paklenica zabilježeno je preko 1000 biljnih vrsta i podvrsta.
Na području Nacionalnog parka Paklenica sve su biljne vrste zakonom zaštićene te se ne smiju brati ni oštećivati! U slučaju branja ili oštećivanja biljnih vrsta u Nacionalnom parku, Zakonom o zaštiti prirode predviđena je novčana kazna.
Sve su životinje u Nacionalnom parku Paklenica zakonom zaštićene te ih se ne smije namjerno uznemiravati.
Dalje

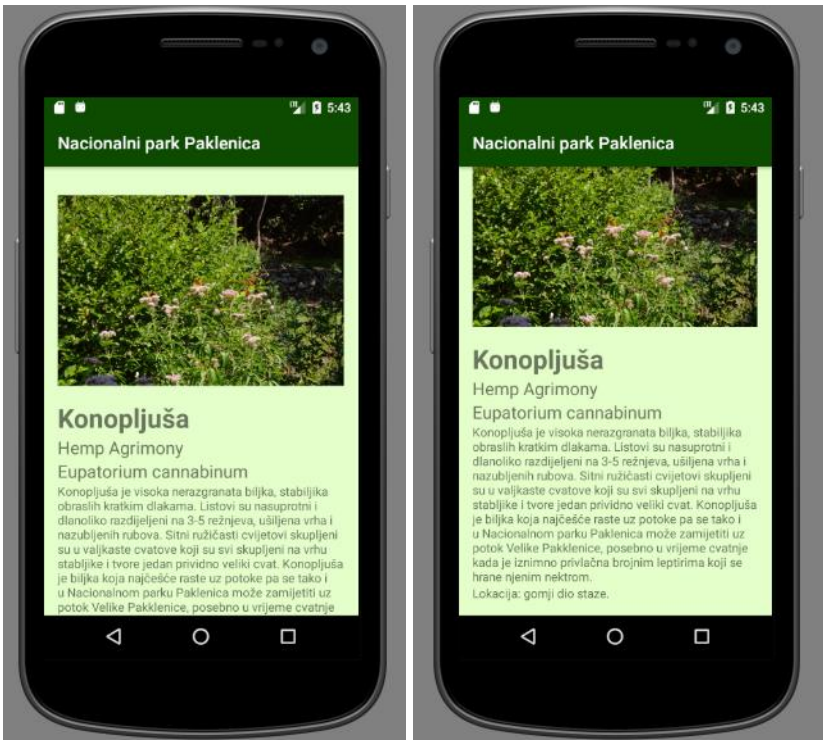

Nacionalni park Paklenica
Odaberite željeno godišnje doba:
PROLJEĆE
LJETO
JESEN
ZIMA


Nacionalni park Paklenica
Alepski bor
Bijela pušina
Bijela čiopa
Bijeli grab
Bjelica
Bjelouška
Blavor
Brgljez
Brgljez kamenjar
Bukva

Sl. 3.16. Testiranje aplikacije na tabletu

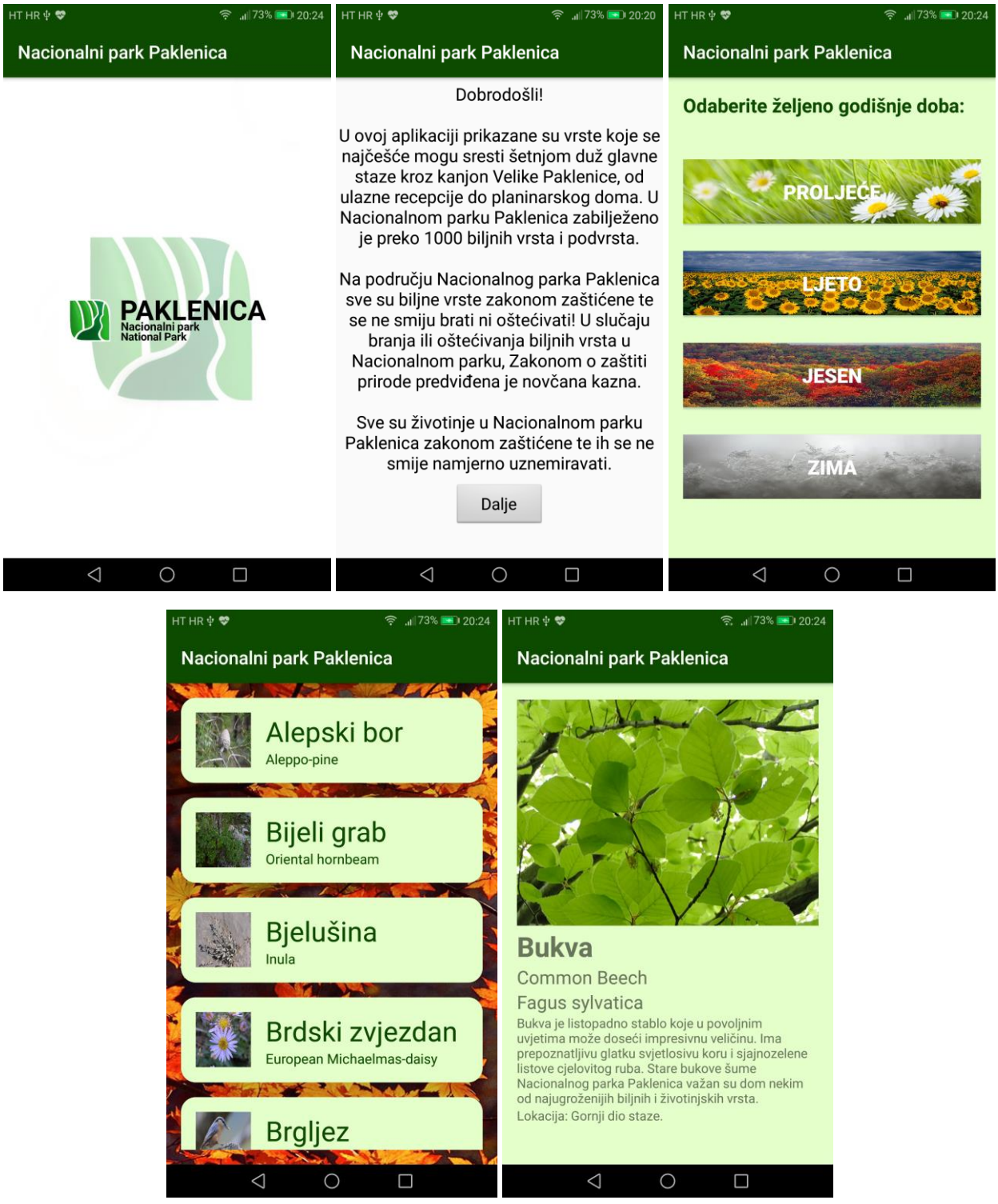
Virtualni mobilni uređaj je pokrenut na 6.0 Marshmallow inačici Androida, API 23. Veličina ekrana iznosi 4.65 inča te 1280x720 piksela to jest 216 ppi. Aplikacija je uspješno pokrenuta te se može vidjeti da, iako je ekran uređaja manje veličine, svi podatci su dostupni zahvaljujući umetnutom ScrollView-u pomoću kojega se mogu pomicati detalji biljke ili životinje gore ili dolje. Upravo to se vidi na posljednje dvije fotografije slike 3.17. Aplikacija zadržava sve svoje funkcionalnosti i na starijim inačicama Androida poput Marshmallow-a.





Sl. 3.17. Testiranje aplikacije na virtualnom mobilnom uređaju

Stvarni Android uređaj na kojem se vršilo testiranje bio je Huawei P9 Lite čija je rezolucija ekrana 1920x1080 piksela (424 ppi – *pixels per inch*), veličina zaslona je 5.2 inča, a inačica Android sustava 7.0 – Nougat, API 24. Na slici 3.18 se vidi rad aplikacije na tom uređaju te se može zaključiti kako aplikacija uspješno radi budući da su sve potrebne funkcionalnosti dostupne te ispravne.



Sl. 3.18. Testiranje aplikacije na stvarnom uređaju

Ukoliko se usporede sva tri testiranja na različitim uređajima, može se uočiti razlika u prikazanim popisima biljaka i životinja. Upravo ta razlika ovisi o odabranom godišnjem dobu to jest u popisima se prikazuju samo vrste koje se mogu pronaći kroz stazu u tom godišnjem dobu. Iz navedenih fotografija se može uočiti da je aplikacija kreirana na vrlo jednostavan način te je

prilagođena svim korisnicima bez obzira na njihovo iskustvo korištenja pametnih uređaja. Na kraju, može se zaključiti kako je aplikacija uspješno izvedena te se može implementirati u svakodnevno poslovanje i promicanje Nacionalnog parka Paklenica.

4. ANALIZA KORISNOSTI APLIKACIJE

Informacijsko komunikacijske tehnologije (ICT – engl. *Informaction and Comunications technology*) značajno napreduje iz dana u dan te razvoj i korištenje tih tehnologija omogućava sve značajniji napredak u načinima promocije. Budući da ICT ima sve veći utjecaj u čovjekovom svakodnevnom životu, upotreba takvih tehnologija omogućava lakše zadržavanje postojećih korisnika kao i lakše privlačenje novih korisnika. Detaljnom analizom postojećeg sadržaja, utvrđeno je da većina hrvatskih nacionalnih parkova i parkova prirode koriste informacijsko komunikacijske tehnologije u vidu web stranica te da, izuzev Nacionalnog parka Plitvička jezera i Nacionalnog parka Krka, nacionalni parkovi i parkovi prirode ne posjeduju aplikaciju kojima bi korisnicima olakšali organizaciju posjete i informiranje o parku. Na spomenutim web stranicama nacionalnih parkova nalaze se razne informacije o parku poput povijesti parka, cjenika, općenitim informacijama, službenim kontaktima, ukratko o flori i fauni, no nigdje ne postoji detaljan opis svih biljnih i životinjskih vrsta unutar parka.

Realizirana aplikacija će nakon implementacije donositi razna poboljšanja u poslovanju Nacionalnog parka Paklenica. Ona prvenstveno ima edukativnu svrhu posjetitelja vezano za zaštićene biljne i životinjske vrste prisutne u parku, ali i općenito podizanje svijesti o važnosti čuvanja prirode. Posjetitelji Nacionalnog parka Paklenica će imati poseban doživljaj posjeta koristeći ovu aplikaciju, jer će biti obogaćeni mnogim znanjima koja im može pružiti korištenje aplikacije kao besplatnog vodiča kroz posjećenu destinaciju. Nacionalni park aplikacijom ispunjava dio društvene odgovornosti tako što veličaju društvena pitanja o zaštiti okoliša te na taj način uvrštavaju ekološke ciljeve u svoje poslovanje. Upravo takav način poslovanja doprinosi konkurentnosti parka te povećava zadovoljstvo posjetitelja. Uz povećanje zadovoljstva, aplikacija unaprjeđuje ponudu parku, a time i dobit u konačnici jer bi mogla privući značajniji broj posjetitelja. Također utječe na smanjenje troškova poslovanja budući da se uvođenjem aplikacije smanjuje potreba za dodatnim tečajevima, kako bi omogućili to isto znanje posjetiteljima, a troškovi ulaganja i održavanja aplikacije su vrlo mali. Nadalje, park smanjuje troškove ljudskih resursa te povećava efikasnost rada obzirom da zaposlenici koju su vršili edukaciju posjetitelja sada mogu raditi na drugim radnim mjestima i tako poboljšati poslovanje parka. Pored toga, aplikacija predstavlja odličan način marketinga i diferencijacije budući da niti jedan Nacionalni park u Hrvatskoj nema takvu aplikaciju koja sadrži detaljne opise biljaka i životinja koje se nalaze u njemu te se jedino slična aplikacija može preuzeti za američke nacionalne parkove – *National Parks Wildlife Guide*. Ona sadrži sve informacije o biljkama i životinjama koje se nalaze unutar 100 američkih nacionalnih parkova te uz te detalje i fotografije sadrži i glasovne zapise glasanja

životinja kao i cjelokupnu ponudu parka. Upravo ta aplikacija nudi potencijalnu mogućnost poboljšanja realizirane aplikacije Nacionalnog parka Paklenica. Budući da je vrijeme informacijskih tehnologija koje se sve više primjenjuju u svakodnevnom životu, aplikacija predstavlja inovativnu ponudu parka koja nudi zabavan način upoznavanja prirode te privlači posjetitelje obzirom da omogućava veliku promociju Nacionalnom parku.

5. ZAKLJUČAK

Svakim danom, tehnologija se sve više i više razvija što dovodi do sve veće ovisnosti ljudi o njihovim pametnim telefonima i aplikacijama koje su instalirane na njima. Budući da je Android jedan od vodećih operacijskih sustava za pametne uređaje, glavni cilj ovog završnog rada je izvršen pomoću Android platforme ciljajući upravo korisnike tih operacijskih sustava. Postoji mnogo verzija Androida te je za izradu ove aplikacije korištena inačica 7.1 (API *level* 25), a mogu ju koristiti i korisnici API *level*-om, ne nižim od, 15 kako bi se što većem broju ljudi sa raznoraznim uređajima i verzijama operacijskog sustava omogućilo nesmetano korištenje aplikacije.

U završnom radu, prikazan je razvoj aplikacije koja sadržava popis i detalje biljnog i životinjskog svijeta unutar Nacionalnog parka Paklenica. Aplikacija je prvenstveno namijenjena posjetiteljima Paklenice u svrhu promocije i porasta broja posjetitelja. Također omogućava edukaciju korisnika o biljnim i životinjskim vrstama te ekologiji unutar parka. Pri dolasku u Nacionalni park, posjetitelji mogu preuzeti aplikaciju te ju koristiti za vrijeme šetnje kroz kanjon Velika Paklenica kako bi spoznali koje vrste biljaka i životinja mogu očekivati na stazi u određeno godišnje doba te dodatne informacije o pojedinim vrstama. Zadani ciljevi završnog rada su u potpunosti postignuti budući da je aplikacija realizirana i dostupna za preuzimanje na službenim stranicama Nacionalnog parka Paklenica (http://www.np-paklenica.hr/) te ju je moguće koristiti pri šetnji u Nacionalnom parku Paklenica. Kako bi se omogućio lakše preuzimanje aplikacije, kroz park će biti postavljeni plakati sa sadržanim QR kodom koji će sadržavati poveznicu na preuzimanje aplikacije.

Aplikacija nudi brojna unaprjeđenja koje je moguće postići ukaže li se za to potreba. Moguće je dodavati, mijenjati te brisati sadržaj biljaka i životinja, kao i dodavati ili uklanjati biljke i životinje iz željenog godišnjeg doba. Također je moguće unaprijediti aplikaciju dodavanjem stranih jezika tako da njen sadržaj bude dostupan posjetiteljima koji ne poznaju hrvatski jezik. Još jedna mogućnost unaprjeđenja aplikacije je prikaz, na karti parka, točnog staništa određene vrste biljke unutar Nacionalnog parka, kao i dodavanje glasovnog zapisa glasanja životinja nalik prethodno spomenutoj američkoj aplikaciji. Isto tako bi se moglo omogućiti korisnicima da, u slučaju neznanja ili neprepoznavanja biljke, postupnim odabirom karakteristika biljke, saznaju o kojoj se točno vrsti radi.

LITERATURA

[1] Nacionalni park Paklenica, http://www.np-paklenica.hr/index.php/park/o-parku, (20.06.2017.)

[2] Ministarstvo turizma Republike Hrvatske, http://www.mint.hr/default.aspx?id=5396, (21.06.2017.)

[3] Aliferi, C.: **Android Programming Cookbook**, Exelixis Media P.C., 2016.

[4] Smyth, N.: **Android Studio Development Essentials – Second Edition**, 2015.

[5] Android Developers, https://developer.android.com/studio/intro/index.html, (20.06.2017.)

[6] Sistemac Srce, http://www.srce.unizg.hr/arhiva_weba/sistemac2015/index.php%3Fid=35&no_cache=1&tx_ttnews%255Btt_news%255D=778.html, (21.06.2017.)

[7] Schildt, H.: **Java: The Complete Reference, Seventh Edition**, The McGraw-Hill Companies, 2007.

[8] Čupić, M.: Programiranje u Javi, Zagreb, 2015.

[9] Android Developers, https://developer.android.com/guide/topics/manifest/manifest-intro.html, (21.06.2017.)

SAŽETAK

Cilj završnog rada je bio realizirati Android aplikaciju koja korisnicima omogućava pregled životinjskog i biljnog svijeta u Nacionalnom parku Paklenica. Korisnici, unutar aplikacije, imaju mogućnosti biranja željenog godišnjeg te pregleda raznih vrsta životinja i biljaka koje se mogu pojaviti za vrijeme tog doba u parku. Detalji o pojedinim životinjama i biljkama napisane su prema bazi podataka Natura 2000. Pri realizaciji, korišten je Android Studio kao razvojna platforma Android aplikacije te Java programski jezik za logički dio aplikacije, a XML programski jezik za izgled aplikacije to jest njezin *layout*. Primjenom aplikacije omogućavaju se razna poboljšavanja u vidu poslovanja Nacionalnog parka Paklenica te uz edukaciju posjetitelja, ona također povećava dobit parka jer privlači značajan broj posjetitelja.

Ključne riječi: Android, Android Studio, Java, NP Paklenica,

ABSTRACT

The use of communication and information technologies as a promotial tool for the implementation of a marketing strategy on an example of a Croatian tourist destination

The main goal of this project was to develop an Android application which provides the user an overview of the animal and plant species located in the National park Paklenica. In the application, the user has the ability to choose what season he or she wants and then see what kind of animals and plants are sighted in the park during that time of year. The details about the animals and the plants found in the park were taken from the Natura 2000 database. Android Studio was used as the developing platform for the application, Java as the programming language and XML for defining the visual structure for the user interface. Many improvements in the business of National park of Paklenica are provided by using the application. Not only does it enable the education of visitors, it also increases the profit of the National park as a result of a growth of visitors each year.

Keywords: Android, Android Studio, Java, NP Paklenica

ŽIVOTOPIS

Anamarija Blavicki rođena je 30. srpnja 1996. godine u Slavonskom Brodu, Hrvatska. Godine 2002. započinje osnovnoškolsko obrazovanje u OŠ Earl M. Lawson, Leavenworth, Kansas, SAD. Nakon završenog prvog razreda, prebacuje se u OŠ Orašje te tamo nastavlja s obrazovanjem do sedmog razreda. 2008. godine se s obitelji ponovo seli u Sjedinjene Američke države gdje dovršava svoje osnovnoškolsko obrazovanje u Walter Colton Middle School, Monterey, California. Zatim upisuje jezičnu gimnaziju Matija Mesić u Slavonskom Brodu gdje se nastavlja školovati do 2014. kada je upisala Elektrotehnički fakultet u Osijeku.

Anamarija Blavicki

PRILOZI

CD

- Rad u .doc formatu i pdf formatu
- Android studio projekt
- .apk datoteka aplikacije