Razvoj aplikacije za upravljanje rezervacijama u iznajmljivačkim objektima

Milić, Roberto

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:813702

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike

Roberto Milić

Izrada aplikacije za pregled iznajmljivačkih objekata i rezervacija

Završni rad

Pula, 2019.

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike

Roberto Milić

Izrada aplikacije za pregled iznajmljivačkih objekata i rezervacija

Završni rad

JMBAG: 00690658469, redoviti student / redovita studentica

Studijski smjer: Informatika

Predmet: Softversko inženjerstvo

Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijski sustavi i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor / Mentorica: doc. dr. sc. Tihomir Orehovački

Pula, Rujan 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika ekonomije/poslovne ekonomije, smjera _____ ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA
o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile
u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom _____
_____
____ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

Sadržaj

1. **Uvod……..……..1**
2. **Ideja Aplikacije……...…..2**
3. **Korišteno okruženje izrade aplikacije…….3**
4. **Prikaz aplikacije……..….5**
5. **Zaključak……...21**
6. **Literatura……..….22**
7. **Sažetak…….….23**

1. Uvod

U zadnjih nekoliko desetljeća, napredak tehnologije nikada nije bio brži veći sada nego u cijeloj ljudskoj povijesti. Pogotovo u zadnjih četrdeset godina. Ovaj brzi i nagli razvoj tehnologije donio je mnoga unapređenja u dnevnim života ljudi, te je donio mnogih olakšica. Od kućanskih aparata, to industrijskog napretka. Jedan od ključnih faktora ovog brzog razvoja tehnologije je mogućnosti dijeljenja informacija. Činjenica da se može ogromne količine podataka podijeliti veoma olako, podosta je pomoglo razvoju tehnologije, te shvaćanja današnjeg svijeta. To se pogotovo može vidjeti u razvoju mobilnih tehnologija. Nekoć mobilni uređaju nisu bili nešto pretjerano mobilni, poprilično su bili ne praktični, te generalna javnosti ih je sporo prihvaćala. No danas? Trenutno danas se ne može zamisliti svijet bez mobilnih uređaja, isto poznati kao pametni mobiteli. Danas pametni mobiteli imaju poveliki broj namjena. Nekoć su mogli samo zvati, te slati i primati poruke. U sadašnje vrijeme se njima plaćaju račune, koriste se kao novčanik, digitron, te slične namjene. Služe da za zabavu, može se čitati knjige na njima, gledati serije i filmove, igrati videoigre. Njihove namjene su skoro pa bezbrojne. A mogu postati bezbrojne ako se ovakav brz razvoj nastavlja. Odlični su za skoro svaki aspekt. Mogu pomoći sa spremanjem važnih trenutaka, mogu pomoći s organiziranjem bilo čega.

To dovodi do iduće stavke. Jasno je poznatu u svijetu da je Hrvatska jedna velika turistička destinacija. Hrvatska nudi prekrasnu prirodu, jedinstvenu kulturu, te bogatu povijesnu baštinu. Kako je turizam podosta jaka grana u Hrvatskoj, mnogi hrvatski državljani su se odlučiti baviti turizmom. Fokus će se baciti na uslužnu djelatnosti, pogotovo na privatno iznajmljivanje. Hrvatska ima mnogo iznajmljivača, koji iznajmljuju svoje iznajmljivačke objekte turistima tijekom sezone. Kako ima velik broj iznajmljivača, tako ima i velik broj načina kako svatko od njih organizira svoj biznis. Neki koriste stranice agencija koji im iznajmljuju iznajmljivačke objekte, ali i dalje velika većina vode evidenciju na papiru. Iako efektivno, nije praktično. Bilježnica ima neku svoju veličinu, te zauzima svoj neki prostor. Doduše, prostor za mobitel uvijek ima. Tako da mobilna aplikacija idealno rješenje za ovaj problem. Iznenađujuće je kako nema aplikacije na hrvatskom jeziku, koji bi pomogao iznajmljivačima da vode evidenciju svojih rezervacija. Mobilna aplikacija koja ima mogućnosti dodavanja iznajmljivačkih objekata, te rezervacija na istih, prikazivanje stanja rezervacija je poprijeko potrebno.

Kada se baci pogled na javnu statistiku, na primjer statistike web stranice gs.statcounter, može se vidjeti kako je mobilni operacijski sustav Android uveliko najpopularniji. Iz tog

razloga je ova aplikacija napravljena isključivo za Android, te je pisana u programskom jeziku Kotlin. Kotlin je preporučeni programski jezik za pisanje Android aplikacija, koji je preporučio Google Inc., stvaratelji Android operacijskog sustava. Ova aplikacija ima prijašnje navede mogućnosti te mnogo više koje će biti poslije prikazani. Zašto se čini ova aplikacija korisna? Mali broj ljudi napuštaju danas svoje domove bez mobitela. U slučaju da iznajmljivač dobiva upit za rezervaciju, odmah na licu mjesta može provjeriti da li ima slobodan termin, te upisati u aplikaciji. Što se tiče aplikacije, nema negativne strane, osim što ne može raditi ako se mobitelu isprazni baterija. Podaci su spremljeni lokalno, tako da nema potrebe za internetskom vezom.

2. Ideja

Hrvatska je jedna od vodećih zemalja u Europi što se tiče turizma. Ima sjajne plaže, prirodu i bogatu kulturu za ponuditi turistima. Nažalost je turizam i glavni unosni sektor države. Naizgled svemu, turizam nudi druge mogućnosti privrednicima. Tako i u IT sektoru. Kreiranjem raznih aplikacija i programa za vođenje raznih aktivnosti u turizmu. Kako sam zadnje desetljeće proveo duboko u turizmu, te imao bliske susrete sa raznim turistima i iznajmljivačima, dobio sam velik uvid u situaciju. Imajući sliku situacije, primjetio sam kako nema praktične aplikacije za pregled rezervacija u iznajmljivačkim objektima. Poneke booking web stranice nude tu opciju, ali ima podosta iznajmljivača koji ne koriste takve stranice za iznajmljivanje svojih objekata. Zbog toga je sinula ideja da pokušam izraditi jednu jednostavnu aplikaciju na hrvatskom, za pregled i vođenje rezervacija. Zašto na hrvatskom? Iako je engleski danas svjetski jezik, te se služi kao međunarodni univerzalni jezik. Postoji dobar postotak iznajmljivača koji nisu dovoljno vješti u korištenje engleskog jezika. Između ostalog, uvijek je jednostavnije koristiti aplikaciju na vlastitom jeziku. Cilj aplikacije je da se jednostavno i lokalno može voditi pregled rezervacija vlastitih iznajmljivačkih objekta. Jednostavnost je isto izuzetno bitan faktor ove aplikacije, pošto je dobar dio iznajmljivača srednje ili starije dobi, te generalno imaju veće poteškoće korištenja aplikacija nego mlađe generacije.

3. Korišteno okruženje za izradu aplikacije

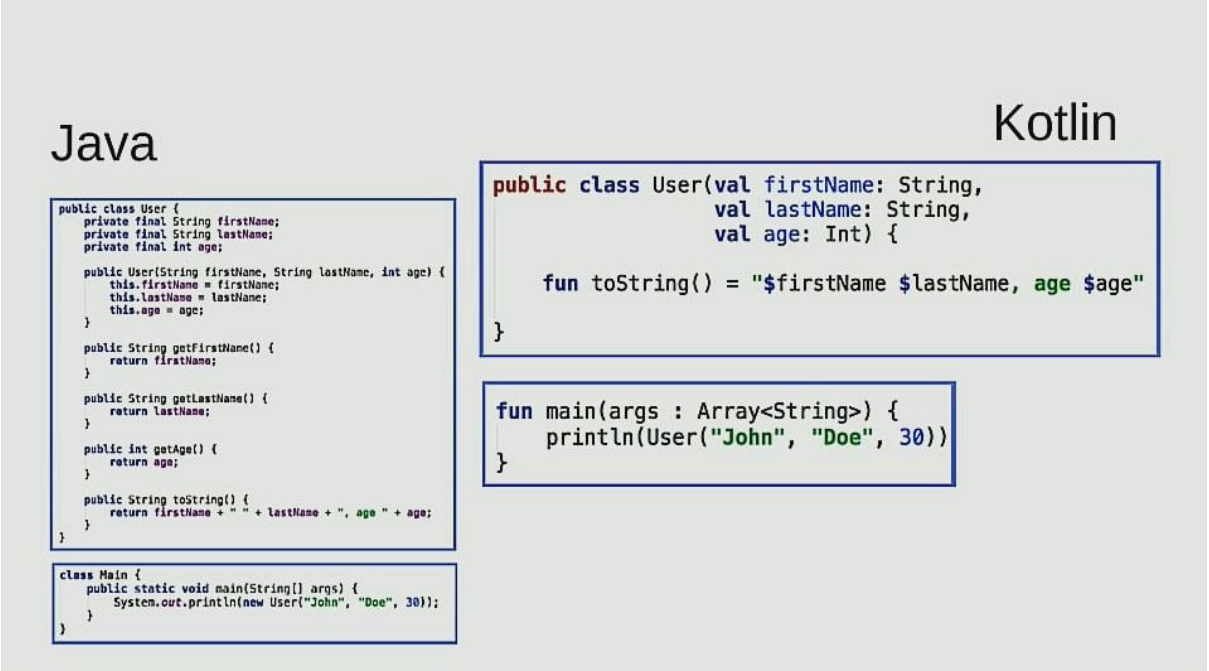
Za izradu aplikacije korišten je Android Studio kao IDE (engl. *Intergrated Development Environment*). Android Studio je službeni IDE za izradu android mobilnih aplikacija, koji je razvio Google Inc. Osnovan je na temelju IntelliJ IDEA.
Prateći stranicu Android Developer-a, prednosti Android Studio-a su:

- Fleksibilan Gradle-baziran sustav
- Bogat svojstvima i brz emulator
- Jedinstvena okolina gdje se može razvijati za sve Android uređaje
- Instant Run mogućnost
- Jednostavna intergracija s Firebase, Kotlin, Google Cloud i Github
- Opsežan broj alata i okvira (*engl. Frameworks*)
- C++ i NDK support
- Lint alati za izvođenje, koristivosti, kompatibilnost inačica i ostalih problema

Standardni programski jezik za izradu Android aplikacija je uglavnom bila Java. Iako se može izraditi Android aplikacija u više programskim jezicima, Java je bila glavna, sve do nedavno. Kotlin je druga službena varijanta za izradu Android aplikacija. Otkad je Android Studio 3.0 izašao 2017. godine, Kotlin je postao druga alternativa uz Javu kao glavni programski jezik za izradu Android aplikacije.
Kotlin je po definiciji opći (*engl. General Purpose*), statično tipkan programski jezik za razvoj multiplatformskih aplikacija. Interoperatibilnost, konciznosti i sigurnosti su neke značajke Kotlina. Razvijen je sa strane tvrtke JetBrains-a, sa sjedištem u Sankt Petersburgu, Rusija. Kotlin sintaksa nije kontabilna s Javom, ali kako radi u Javinoj virtualnoj stroju (JVM), kompajlira se u bytecode-u kao i Java. To znači da jedan Android Studio projekt može imati izmješane klase Kotlina i Jave, bez pojavljivanje ikakvih grešaka u kompajliranju, te Kotlin može koristiti Java biblioteke (engl. *Libraries*). Sintaksa Kotlina, uspoređujući ju s Javom, je veoma jednostavnija, što je bio i cilj JetBrains-a. Isto tako, Kotlin se može kompajlirati u LLVM-u te i u JavaScript kodu. Od 7. Svibnja 2019. Google je proglasio Kotlina kao službeni predloženi programirani jezik za izradu Android aplikacija.
Gledajući u Kotlin dokumentaciju, prednosti Kotlina su:

- Kompatibilnost
- Performansa
- Interoperabilnost
- Otisak Stopala (*engl. Footprint*)

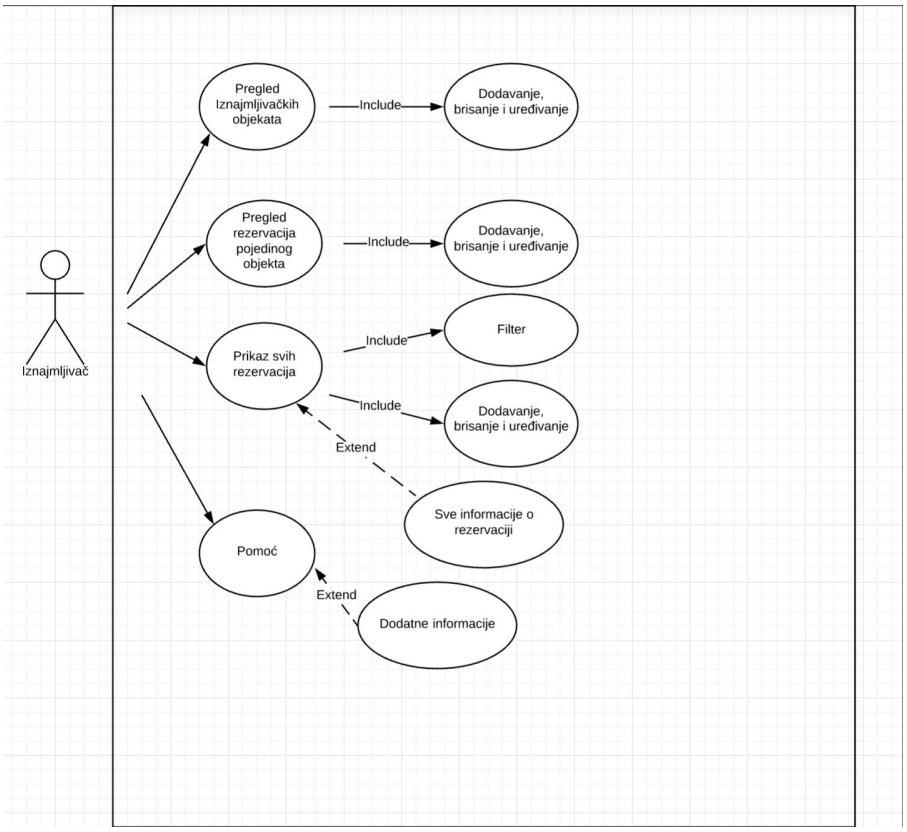
- Vrijeme kompajliranja
- Krivulja učenja (*engl. Learning Curve*)



3.1 Usporedba sintakse Java-e i Kotlina

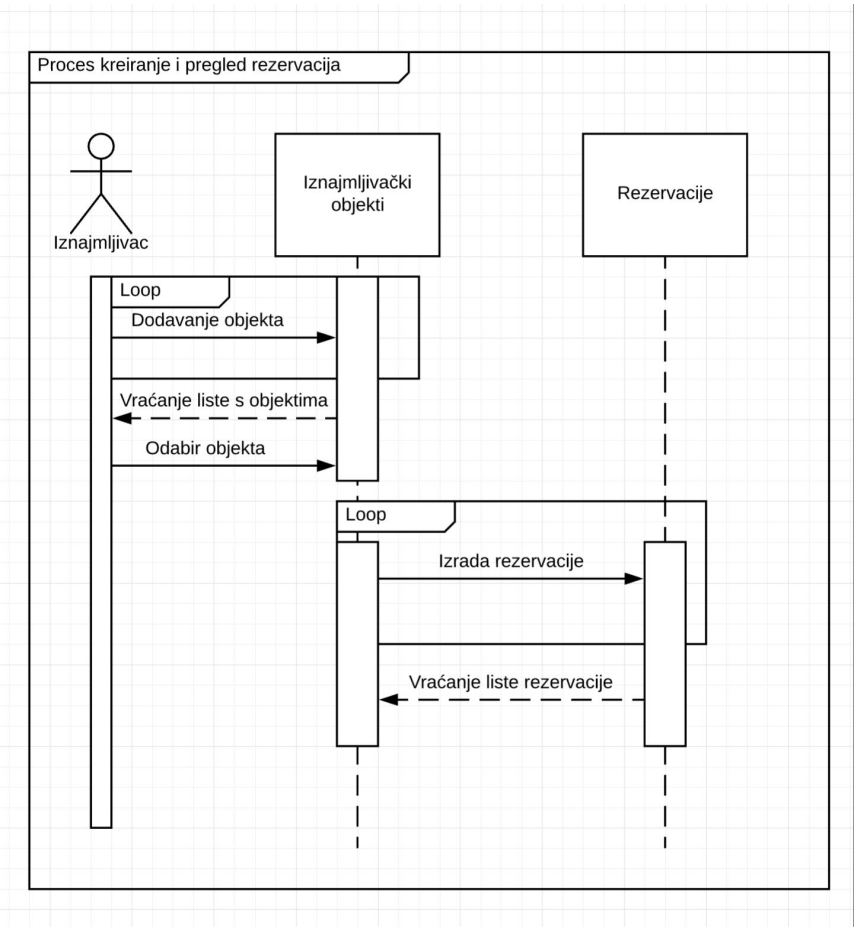
4. Prikaz aplikacije

U use case dijagramu se može vidjeti mogućnosti koja aplikacija nudi za korisnika, u ovom slučaju iznajmljivaču.



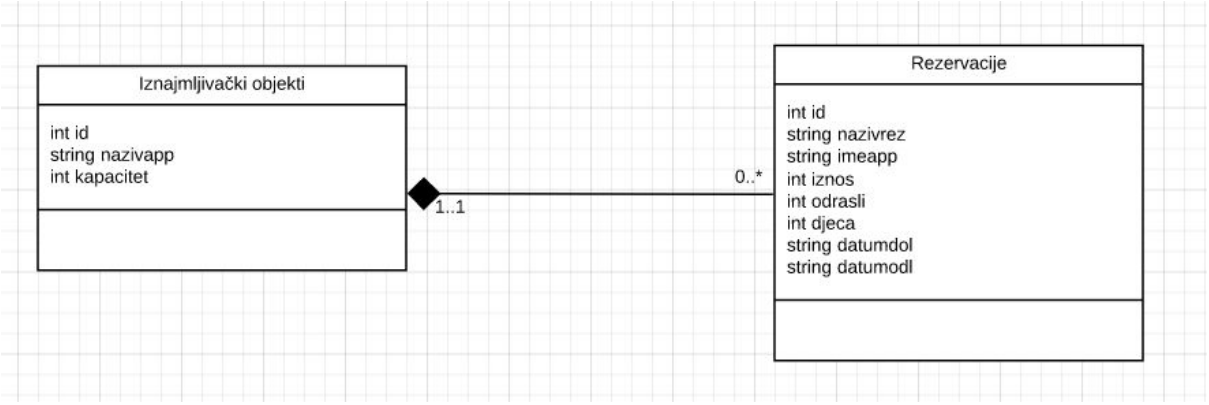
4.1 Use Case dijagram

U sekvencijalnom dijagramu je prikazan proces dodavanje nove rezervacije. Ako uzmemo u obzir da nema novi Iznajmljivačkih objekata, iznajmljivač može dodati nove objekte, više njih za redom dok ne stisne povratak. Aplikacija će prikazati listu s dodanim objektima, te iznajmljivač odabire objekt te će ući u aktivitet s rezervacijama. Tu će dobiti mogućnost da doda nove rezervacije, više njih u slučaju potrebe, dokle god ne pritisne tipku povratak. Aplikacija će prikazati dodane rezervacije u listi.



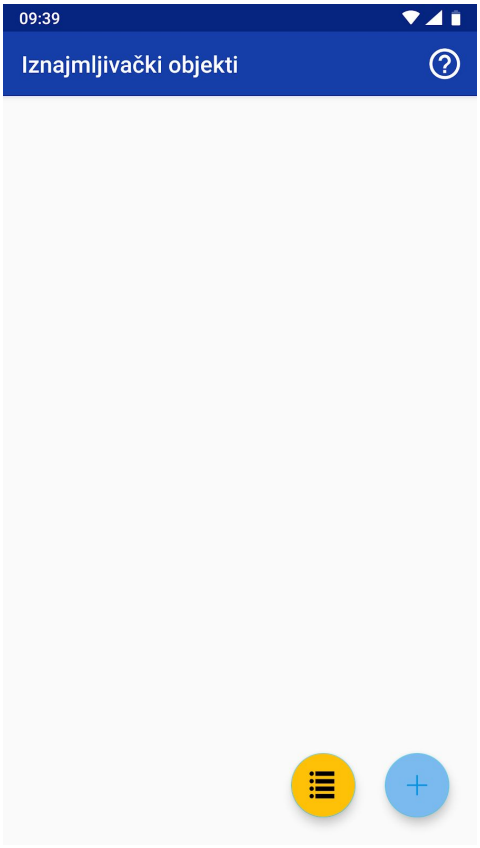
4.2 Sekvencijalni dijagram

Klasni dijagram prikazuje bazu koja se sastoji od dvije tablice. Jedna tablica služi za spremanje podataka vezane za iznajmljivačkih objekata, dok druga za rezervacije. Svaki iznajmljivački objekt ne mora imati rezervaciju ili može sadržavati jednu ili više rezervacija. Dok rezervacija može biti vezana samo za jedan iznajmljivački objekt.



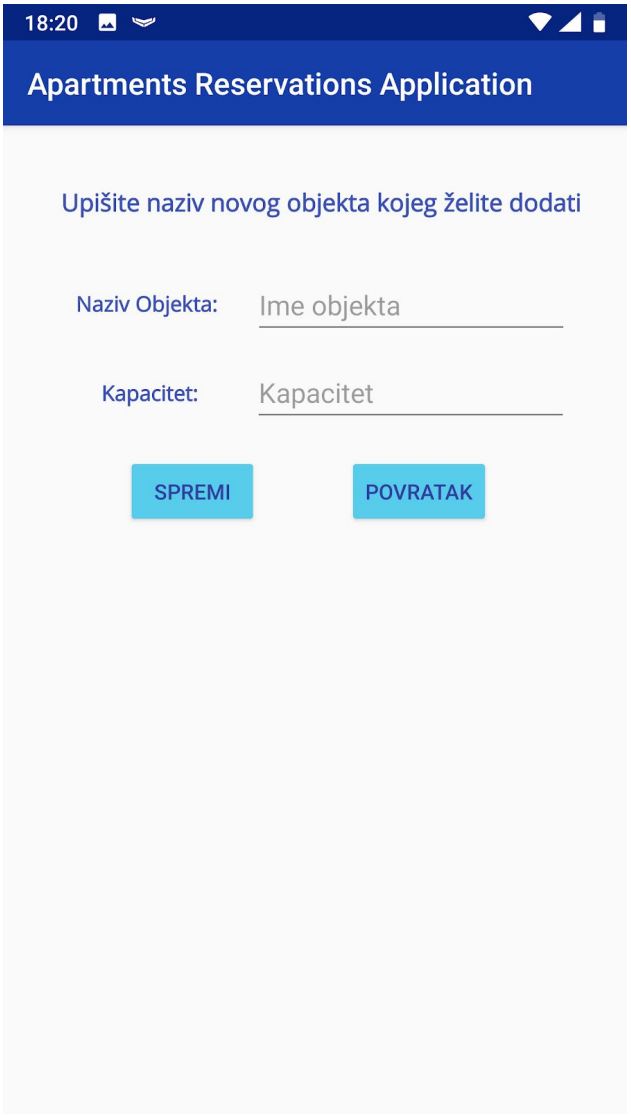
4.3 Klasni dijagram

Radi ciljane grupe korisnika, aplikacija ima jednostavan dizajn, koji je lako shvatljiv.



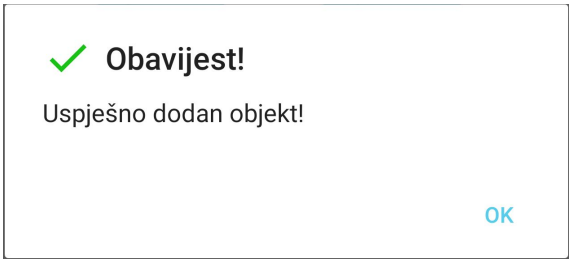
4.4 Početni aktivitet aplikacije

Početni aktivitet aplikacije kada se prvi put pokreče izgleda prazno, razlog tomu jer je lista prazna. U akcijskoj traci se može vidjeti naslov aktiviteta i bijeli upitnik. Kada se stisne bijeli upitnik, otvara se prozorčić koji prikazuje ukratko o aplikaciji, te upute. Na dnu se može vidjeti dvije okrugle tipke. Žuta je za prikaz svih rezervacija spremljeno u aplikaciji, dok plava služi za dodavanje novog iznajmljivačkog objekta.



4.5 Dodavanje iznajmljivačkog objekta

Pritiskom na tipku za dodavanje objekta, otvorit će se novi aktivitet koji će omogućiti željenu funkciju. Za dodavanje objekta, dovoljno je samo unijeti ime objekta po kojem će se prepoznati. U polje se može unositi svo znakovlje. Za dodavanje objekta dovoljno upisati ime, te stisnuti tipku spremi. Iskočiti će mali prozor s obavijesti da je spremanje uspješno provedeno.



4.6 Obavijest o uspješnosti dodavanja objekta

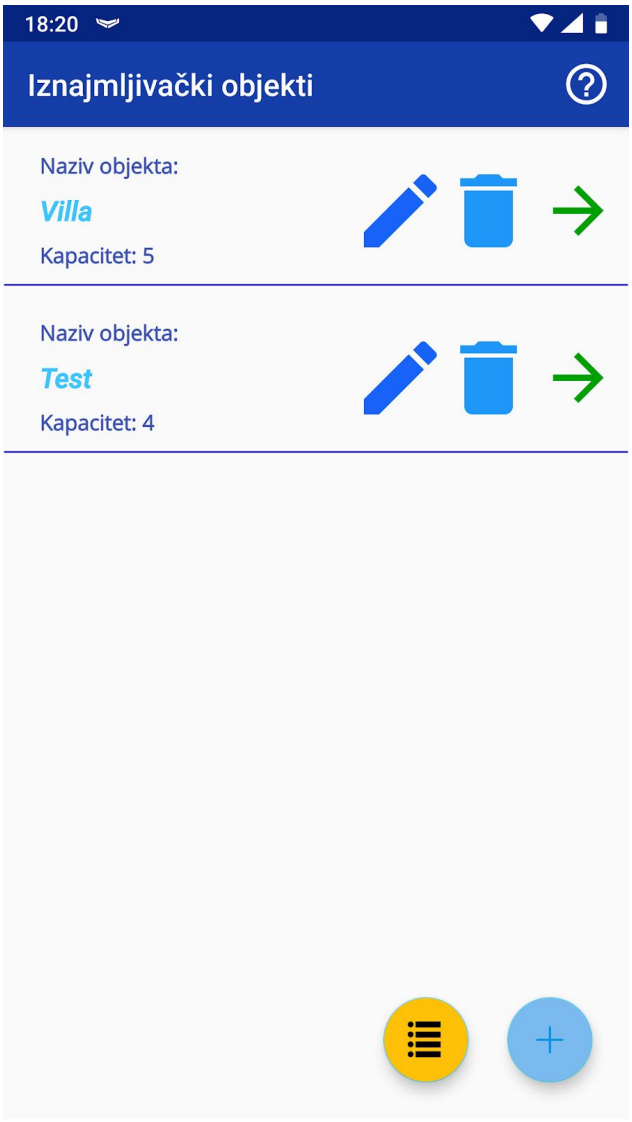
```
btn_spremi.setOnClickListener(){ it: View!
    if (edittxt_Naziv.text.isEmpty()){
        Toast.makeText( context: this,  text: "Unesite ime objekta", Toast.LENGTH_SHORT).show()
        edittxt_Naziv.requestFocus()
    }
    else if (edittxt_Kapacitet.text.isEmpty()){
        Toast.makeText( context: this,  text: "Unesite kapacitet objekta", Toast.LENGTH_SHORT).show()
        edittxt_Kapacitet.requestFocus()
    }
    else{
        val objekat = Iznajmljivacki()
        objekat.nazivApp = edittxt_Naziv.text.toString()
        objekat.kapacitet = edittxt_Kapacitet.text.toString().toInt()

        MainActivity.dbHandler.addObjekti( context: this, objekat)

        val builder = AlertDialog.Builder( context: this)
        builder.setTitle("Obavijest!")
        builder.setMessage("Uspješno dodan objekt!")
        builder.setIcon(R.drawable.ic_check_black_24dp)
        builder.setPositiveButton( text: "Ok"){dialog, which ->}
        val dialog: AlertDialog = builder.create()
        dialog.show()

        clearEdits()
        edittxt_Naziv.requestFocus()
    }
}
```

4.7 kod za spremanje objekta

Nakon spremanja, aktivitet će ostati isti, tako da se može dodati odmah i sljedeći objekt. Za povratak na prethodni aktivitet, dovoljno je samo stisnuti tipku povratak.

```
@SuppressLint( ...value: "WrongConstant")
fun prikazObjekti(){
    val objektLista : ArrayList<Iznajmljivacki>  = dbHandler.getObjekti( context: this)
    viewManager = LinearLayoutManager( context: this)
    viewAdapter = ObjAdapter( context: this, objektLista)
    recyclerView = findViewById<RecyclerView>(R.id.rv).apply{ this: RecyclerView!
        setHasFixedSize(true)
        layoutManager = viewManager
        adapter = viewAdapter
    }

}

override fun onResume(){
    prikazObjekti()
    super.onResume()
}
```

4.8 Prikaz Iznajmljivačkih objekata preko RecyclerView-a

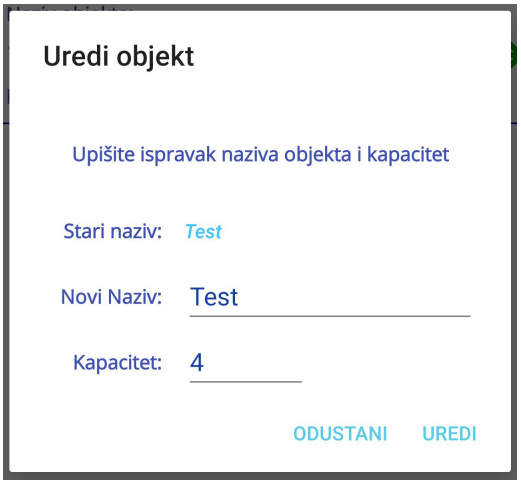


4.9 Prikaz objekata kada u listi ima elemente

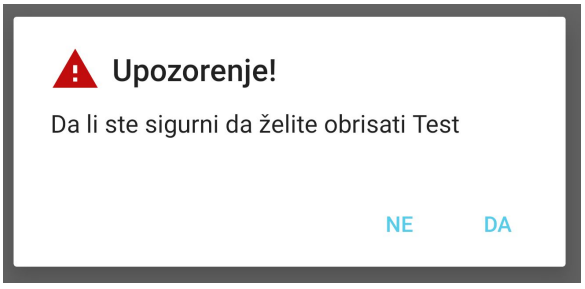
Kada u listi ima već spremljene objekte, otvaraju se dodatni mogućnosti. Na svaki objekt se pojavljuju tri nove tipke. Olovka je za uređivanje objekta, u slučaju da je došlo do promjene naziva ili je krivo upisan naziv. Kanta za smeće je za brisanje objekta. Zelena strelica je za odabir objekta, te odlazak u pregled rezervacija od odabranog objekta.

```
fun editObjekt(id: String, objektIme: String, staroIme: String, kapacitet: Int): Boolean{
    var result = false
    val db : SQLiteDatabase! = this.writableDatabase
    val contentValues = ContentValues()
    val contentValues2 = ContentValues()
    contentValues.put(COLUMN_APP_NAME, objektIme)
    contentValues.put(COLUMN_APP_CAP, kapacitet)
    contentValues2.put(COLUMN_REZ_APPNAME, objektIme)
    try{
        db.update(TABLE_APP_NAME, contentValues, whereClause: "$COLUMN_APP_ID = ?", arrayOf(id))
        db.update(TABLE_REZ_NAME, contentValues2, whereClause: "$COLUMN_REZ_APPNAME = ?", arrayOf(staroIme))
        result = true
    }catch (e: Exception){
        Log.e(ContentValues.TAG, msg: "Greška u editiranju")
        result = false
    }
    return result
}
```

4.10 Prikaz koda kod uređivanje objekta u DBHandler-u

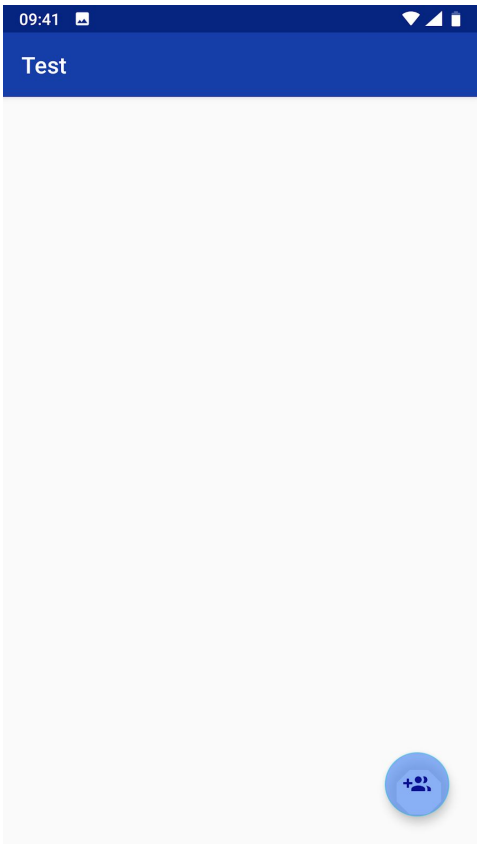


4.11 Uređivanje iznajmljivačkog objekta



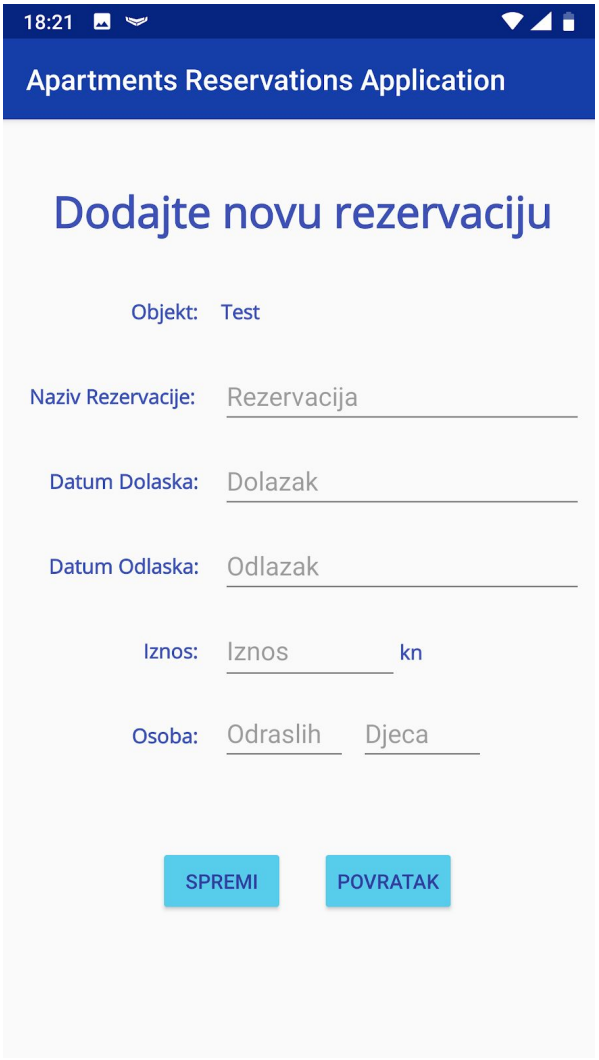
4.12 Brisanje iznajmljivačkog objekta

Pritiskom na plavu olovku, otvorit će se manji prozor (engl. *AlertDialog*) koji će dati mogućnosti preimenovanje objekta. Pritiskom na plavi koš za smeće, otvorit će se prozor s upozorenjem da korisnik je u procesu brisanja odabranog objekta.



4.13 Prikaz rezervacije pojedinog objekta

Pritiskom na zelenu strelicu, otvorit će se novi aktivitet koji prikazuje sve buduće rezervacije odabranog objekta. U akcijskoj traci u gornjem lijevom kutu će se prikazati naziv objekta. Okrugla plava tipka u donjem desnom kutu je za dodavanje nove rezervacije.

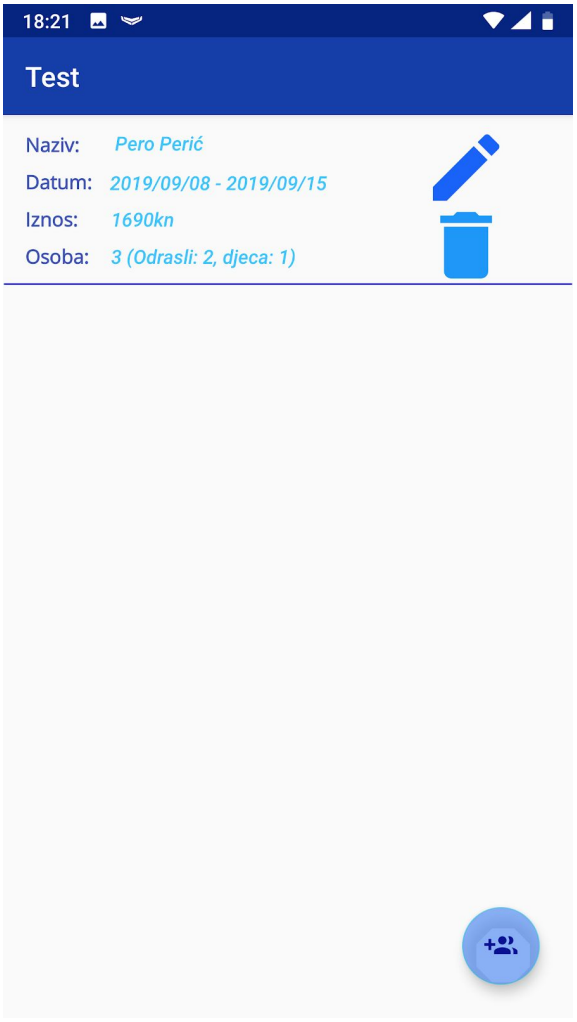


4.14 Dodavanje nove rezervacije

Pritiskom tipke za dodavanje rezervacije, otvorit će se novi aktivitet za dodavanje nove rezervacije. Ima 5 obvezna polja za ispunjavanje i jedno ne obavezno (polje djeca), te ako je i jedno polje neispunjeno, aplikacija neće dozvoliti spremanje rezervacije. Osim kod polje Djeca, ako ostane prazna, aplikacija će sama staviti nulu kao vrijednost. Isto tako, datum odlaska mora biti poslije datuma dolaska. U suprotnom će se pojaviti tekst u crvenoj boji, koji će naznačiti istu grešku.

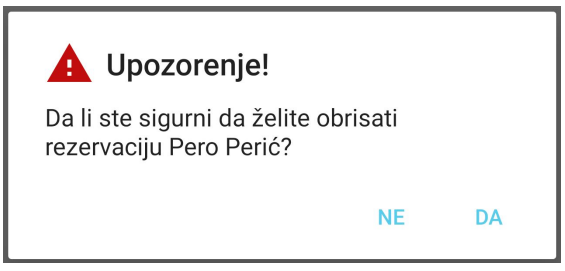
```
fun getRezSpecific(context: Context, appIme: String): ArrayList<Rezervacije>{
    val qry = "SELECT * FROM $TABLE_REZ_NAME WHERE $COLUMN_REZ_APPNAME = ? ORDER BY $COLUMN_DATE_AR ASC"
    val db : SQLiteDatabase! = this.writableDatabase
    val cursor : Cursor! = db.rawQuery(qry, arrayOf(appIme))
    val rezerv = ArrayList<Rezervacije>()

    var brojac = 0

    if(cursor.count == 0)
        Toast.makeText(context, text: "Nema rezervacija u objektu $appIme", Toast.LENGTH_SHORT).show()
    else{
        cursor.moveToFirst()
        while (!cursor.isAfterLast){
            val rezervac = Rezervacije()
            var datumPih : String = cursor.getString(cursor.getColumnIndex(COLUMN_DATE_LE)).toString()
            if(MainActivity.Usporedba.usporedba(datumPih)==1){
            rezervac.idRez = cursor.getInt(cursor.getColumnIndex(COLUMN_REZ_ID))
            rezervac.imeRez = cursor.getString(cursor.getColumnIndex(COLUMN_REZ_NAME))
            rezervac.rezAppNaziv = cursor.getString(cursor.getColumnIndex(COLUMN_REZ_APPNAME))
            rezervac.datumDOL = cursor.getString(cursor.getColumnIndex(COLUMN_DATE_AR))
            rezervac.datumODL = cursor.getString(cursor.getColumnIndex(COLUMN_DATE_LE))
            rezervac.placanje = cursor.getString(cursor.getColumnIndex(COLUMN_REZ_PLAC))
            rezervac.odrasli = cursor.getInt(cursor.getColumnIndex(COLUMN_REZ_ODR))
            rezervac.djeca = cursor.getInt(cursor.getColumnIndex(COLUMN_REZ_DJ))
            rezerv.add(rezervac)
            brojac++
            }else{
                Log.e( tag: "TAG",  msg: "Nepripadam")
            }
            cursor.moveToNext()
        }
        Toast.makeText(context, text: "Pronadeno $brojac rezervacija", Toast.LENGTH_SHORT).show()
    }
    cursor.close()
    db.close()
    return rezerv
```

4.15 Prikaz čitanje podataka iz baze za upis u RecyclerView za rezervacije



4.16 Prikaz liste pojedinog objekta, s rezervacijom

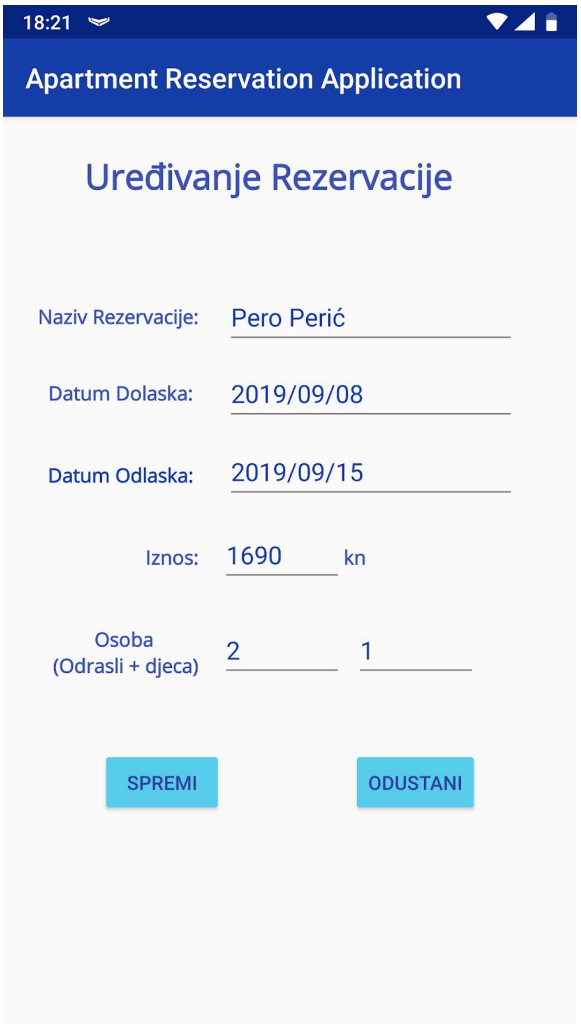
Nakon spremanja rezervacija, sve buduće rezervacije će biti prikazane u listi, vezani za isti objekt. Uz to pojavljuju se nove mogućnosti. Rezervacija se može urediti pritiskom na plavu olovku, te obrisati po potrebi pritiskom na plavi kanta za smeće. U listi se uredno prikazuje naziv rezervacije, te period trajanja rezervacije. Pritiskom na olovku, otvoriti će se novi aktivitet s trima poljima, koji će već biti ispunjeni trenutnim podatcima. Krivo unešeni podatci se mogu izmijeniti, te spremiti.



4.17 Upozorenje koje se pojavljuje kada je proces brisanja pokrenut

```
val rezervator: Rezervacije = rezerv[position]
holder.nazivRez.text = rezervator.imeRez
holder.datumRez.text = "${rezervator.datumDOL} - ${rezervator.datumODL}"
holder.iznosRez.text = "${rezervator.placanje}kn"
holder.osobaRez.text = "${rezervator.odrasli + rezervator.djeca} (Odrasli: ${rezervator.odrasli}, djeca: ${rezervator.djeca})"

holder.rezDelete.setOnClickListener() { it: View!
    val rezervacijaIme : String  = rezervator.imeRez

    var alertDialog : AlertDialog!  = AlertDialog.Builder(context)
        .setTitle("Upozorenje!")
        .setMessage("Da li ste sigurni da želite obrisati rezervaciju $rezervacijaIme?")
        .setPositiveButton( text: "Da", DialogInterface.OnClickListener { dialog, which ->
            if (MainActivity.dbHandler.deleteRez(rezervator.idRez)) {
                rezerv.removeAt(position)
                notifyItemRemoved(position)
                notifyItemRangeChanged(position, rezerv.size)
                Toast.makeText(context,  text: "Obrisana rezervacija $rezervacijaIme", Toast.LENGTH_SHORT).show()
            } else
                Toast.makeText(context,  text: "Greška brisanja $rezervacijaIme", Toast.LENGTH_SHORT).show()
        })
        .setNegativeButton( text: "Ne", DialogInterface.OnClickListener { dialog, which -> })
        .setIcon(R.drawable.ic_warning_black_24dp)
        .show()
}
```

4.18 Kreiranje AlertDialog-a za funkciju brisanje rezervacije



4.19 Uređivanje rezervacije

Povratkom na glavni meni, to jest, lista iznajmljivačkih objekta, stisnuti se može žuta tipka za listu svih rezervacija. Pritiskom, otvorit će se novi aktivitet koji će izlistati sve rezervacije koje su spremljene u bazi. Lista će prikazati naziv rezervacije, datum, kojem objektu je vezan, te status rezervacije. Status rezervacije može biti tri stanja. Rezervirano se odnosi na rezervacije koju kojim datumima tek trebaju doći. Rezervacija u tijeku se odnosi na rezervacije koje su trenutno tijeku. Znaći gdje je datum dolaska prošao, ali datum odlaska još nije. Prošla rezervacija je status za rezervacije kojima je datum odlaska prošao. Isto tako se nudi mogućnosti brisanja rezervacija s liste, pritiskom na crvenu tipku. Tipka info će otvoriti manji prozor sa svim podatcima o rezervaciji

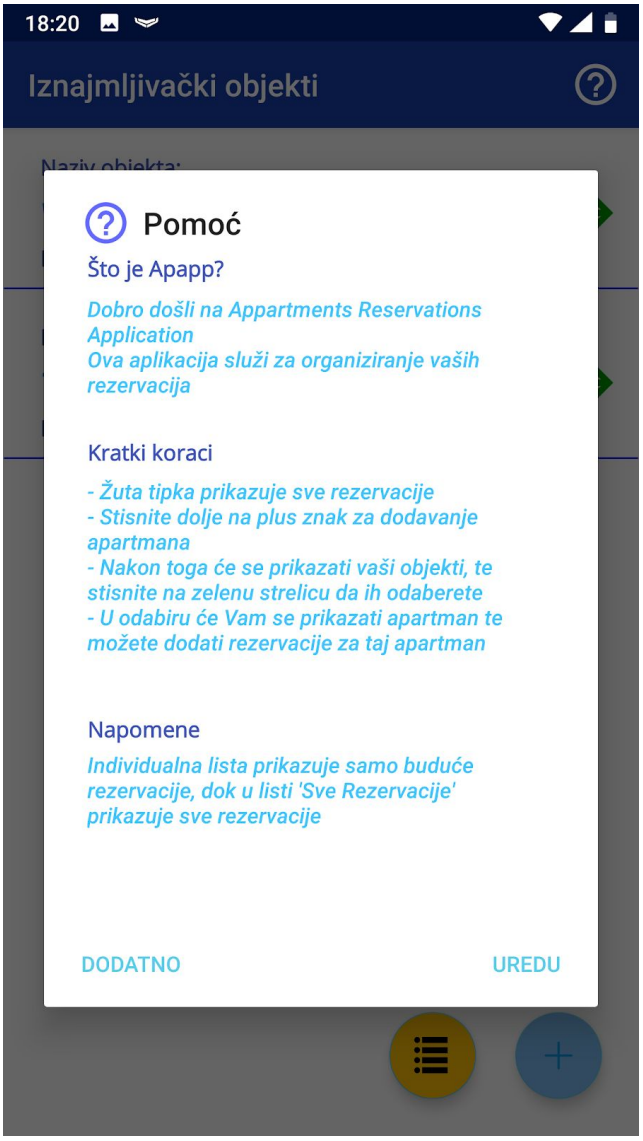
4.20 Prikaz liste svih rezervacija

4.21 AlertDialog s prikazom svih informacija o rezervaciji

Pritiskom na desno gornju tipku, otvara se opcija flitriranja. Filtriranje omogućava filtrirati sve rezervacije ovisno o nazivu iznajmljivačkog objekta i godine rezervacije. Može se ili jedno ili drugo polje ispuniti, ili oba polja.

4.22 AlertDialog za filtriranje rezervacija

Povratkom na listu objekata, može se stisnuti upitnik u desnom gornjem kutu, za dodatne informacije, koji će se pojaviti u malom prozoru.



4.23 Pomoć

Pritiskom na tipku dodatno, otvara se novi aktivitet koji prikazuje dodatne informacije o aplikaciji.

4.24 Dodatne informacije o aplikaciji

5. Zaključak

Mobilna aplikacija Apartments Reservation Application (Apapp) pokazala je kako sadržava većinu potrebnu funkcionalnosti za skladištenje te organiziranje rezervacija za iznajmljivače. S mogućnosti spremanja, uređivanja te brisanja iznajmljivačkih objekata i rezervacija, pokazuje koliko je prilagođena određenim situacijama. Mogućnost filtriranja spremljenih rezervacija uveliko pomaže u organiziranju i prisjećanja starih rezervacija, u slučaju da je iznajmljivaču poprijeko potrebna određena informacija. Dizajn mobilne aplikacije je namjerno jednostavna. Kako je prije spomenuto, aplikacija je namijenjena za hrvatske iznajmljivače. Kako su većina iznajmljivača srednje i starije dobi, poprijeko je bilo potrebno da aplikacija bude što jednostavnija moguće. Uzmivši u obzir da teže barataju s modernom tehnologijom. Radi toga je dizajn aplikacije jednostavan, tako da se brže i lakše nauče koristiti njome. Iako je napravljena da izgleda što jednostavnija, aplikacija nudi mogućnosti da prikaže kratke upute za korištenje aplikacije, te napomena i o aplikaciji općenito.

Za razvoj mobilne aplikacije je korišten Android Studio. Gledajući statistiku, jasno se vidi kako je velika većina pametnih mobitela bazirana na Android-u. Sama ta činjenica objašnjava zašto je mobilna aplikacija napravljena za Android uređaje. Android Studio je odličan IDE, napravljen sa strane Google Inc., izričito napravljen za razvoj Android aplikacija. Korisničko sučelje Android Studio-a je veoma jasan, te je lako za naučiti koristiti. Android Studio nudi sve mogućnosti koji bi svaki moderni IDE trebao imati, pa čak i više. Dva su standardna programska jezika koji Android Studio nudi za korisnika za koristiti, a to su Kotlin i Java. Programski jezik koji je korišten za razvoj ove mobilne aplikacije je Kotlin. Kotlin je opći (General Purpose), statično tipkan programski jezik za razvoj multiplatformskih aplikacija. Kotlin je poprilično nov jezik, stavljen je u razvoj 2011. godine. Ruska tvrtka JetBrains je službeno izdala Kotlin 2013. godine i otada su redovito nadogradili isti programski jezik. S jednostavnom sintaksom, brzo su ju programeri prihvatili. Od 2017. godine, Google je proglasio Kotlin kao preporučeni jezik za razvoj Android aplikacije. Danas se Kotlin često pojavljuje u modernim aplikacijama jer je kompatibilan s Javom.

Iako je turizam i dalje jedna od najrazvijenijih grana industrije u Hrvatskoj, itekako još treba dosta napretka da bude jedna potpuno funkcionalna cjelina. Cilj ove aplikacije je da se olakša hrvatskim iznajmljivačima barem malo što se tiče njihovog posla.

6. Literatura

https://developer.android.com/

https://kotlinlang.org

https://gs.statcounter.com/os-market-share/mobile/worldwide

7. Sažetak

Aplikacija Apapp je aplikacija koja služi za organiziranje rezervacija u iznajmljivačkim objektima, te je isključivo za Android uređaje. Ciljani korisnici su iznajmljivači. Aplikacija je pisana cijela u Kotlinu, a Android Studio je bio program koji je služio kao razvojna okolina. S jednostavnim dizajnom, Apapp je lako shvatljiv svim korisnicima, te još lakši za koristiti.

Ključne riječi: Android, aplikacija, iznajmljivački objekti, rezervacije

Application Apapp is a mobile application which main purpose is to organise reservations in apartments, and is exclusively for Android. Main targeted users for this application are the renters. The mobile application is written wholly in Kotlin, and Android Studio was used a development environment. With a simple design, Apapp is easy to understand and even easier to use.

Keywords: Android, application, apartments, reservations