

# Izrada android mobilne aplikacije za izradu i praćenje rezervacija auto servisa

---

**Bosak, Ivan**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:110:156915>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-22**



*Repository / Repozitorij:*

[Polytechnic of Međimurje in Čakovec Repository -  
Polytechnic of Međimurje Undergraduate and  
Graduate Theses Repository](#)



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU  
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Ivan Bosak, 0313024180

IZRADA ANDROID MOBILNE APLIKACIJE ZA  
IZRADU I PRAĆENJE  
REZERVACIJA AUTO SERVISA

ZAVRŠNI RAD

Čakovec, ožujak 2024.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU  
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Ivan Bosak, 0313024180

IZRADA ANDROID MOBILNE APLIKACIJE ZA  
IZRADU I PRAĆENJE  
REZERVACIJA AUTO SERVISA

DEVELOPMENT OF A ANDROID MOBILE  
APPLICATION FOR CREATION AND MONITORING  
OF RESERVATIONS FOR A CAR SERVICE

ZAVRŠNI RAD

Mentor:  
mr. sc. Željko Knok, v. pred.

Čakovec, ožujak 2024.



## MEDIMURSKO VELEUČILIŠTE U ČAKOVCU

### PRIJAVA TEME I OBRANE ZAVRŠNOG/DIPLOMSKOG RADA

Stručni prijediplomski studij:

Računarstvo       Održivi razvoj       Menadžment turizma i sporta

Stručni diplomski studij Menadžment turizma i sporta:

Pristupnik: Ivan Bosak, JMBAG: 0313024180  
(ime i prezime)

Kolegij: Baze podataka I  
(na kojem se piše rad)

Mentor: mr.sc. Željko Knok, v. pred.  
(ime i prezime, zvanje)

Naslov rada: IZRADA ANDROID MOBILNE APLIKACIJE ZA IZRADU I PRAĆENJE REZERVACIJA AUTO SERVISA

Naslov rada na engleskom jeziku: DEVELOPMENT OF A ANDROID MOBILE APPLICATION FOR CREATION AND MONITORING OF RESERVATIONS FOR A CAR SERVICE

Članovi povjerenstva: 1. mr.sc. Ivan Hegeduš, v. pred., predsjednik  
(ime i prezime, zvanje)  
2. Nenad Breslauer, v. pred., član  
(ime i prezime, zvanje)  
3. mr. sc. Željko Knok, v. pred., mentor  
(ime i prezime, zvanje)  
4. Marija Miščančuk, v. pred., zamjenski član  
(ime i prezime, zvanje)

Broj zadatka: 2021-RAC-R-40

Kratki opis zadatka: \_\_\_\_\_

Tema ovog završnog rada je izrada mobilne aplikacije za izradu i praćenje rezervacija auto servisa. Aplikacija je izrađena u razvojnom okruženju Android studio-u gdje se koristi programski jezik Java. Baza podataka je izrađena u SQL-u koristeći program MySQL Workbench. Aplikacija je podjeljena na dvije korisničke uloga, administrator i korisnik. Korisnik ima ograničen pristup podacima, dok za razliku od njega administrator ima sva prava.

Datum: 12.3.2024.

Potpis mentora: 

## **ZAHVALA**

*Ovim putem se želim zahvaliti mom mentoru mr. sc. Željku Knoku, v. pred. na pomoći oko izrade završnog rada. Također se želim zahvaliti majci i sestri na povjerenju i strpljenju.*

## SAŽETAK

Tema ovog završnog rada je izrada mobilne aplikacije za izradu i praćenje rezervacija auto servisa. Aplikacija je izgrađena u razvojnom okruženju Android studio-u gdje se koristi programski jezik Java. Baza podataka je izgrađena u SQL-u koristeći program MySQL Workbench. Za spajanje Android aplikacije i baze podataka potreban je API. API je kreiran u Springboot framework-u koji je baziran na Javi u razvojnom okruženju IntelliJ IDEA. Kod izrade API-a koriste se dvije značajke Springboot-a a to su Spring Data JPA gdje se koristi Hibernate kao davatelj usluge i Spring Security za sigurnost aplikacije. Za spajanje Android aplikacije sa serverom koristi se Retrofit biblioteka. Izgled mobilne aplikacije to jest UI layout aplikacije izgrađen je u programskom jeziku XML.

Aplikacija je podijeljena na 2 korisničke uloge. Korisnik može kreirati rezervaciju i ima uvid koje rezervacije su dovršene i koje nisu. Korisnik također može vidjeti sve podatke nedovršenih rezervacija kao što su broj rezervacije, datum izrade rezervacije, datum kada se rezervirao za servis, datum predviđenog završetka servisa itd. Isto tako korisnik može vidjeti i podatke gotovih rezervacija kao što su ukupna cijena, cijenu dijelova, radnik koji je radio na servisu, koliko radnih sati je potrošeno na servis automobila te cijenu rada. Korisnik može poslati recenzije o autoservisu, a recenzije se mogu pogledati bez registracije. Administrator ima opciju upisivanja novih zaposlenika te brisanje i ažuriranje postojećih zaposlenika. Administrator ima uvid na sve dovršene servise i nedovršene rezervacije, ne samo svoje. Administrator također može izbrisati rezervaciju i ima opciju upisivanja podataka koje korisnik ne može upisivati. Korisnik se mora registrirati kako bi koristio aplikaciju i prijavljuje se u aplikaciju sa e-mailom i lozinkom koju je upisao. Administrator i korisnik dobivaju obavijesti putem e-maila o stanju rezervacije. Korisnik može dobiti obavijest da je rezervacija odbijena i zašto je odbijena ili da je servis gotov dok administrator dobiva obavijest da ima novu rezervaciju za pregledati.

**Ključne riječi:** *Java, Springboot, Android studio, IntelliJ IDEA, SQL, MySQL Workbench, Retrofit.*

## **ABSTRACT**

The topic of this final thesis is the development of a mobile application for the creation and monitoring of car service reservations. The application is built in Android Studio development environment, where the programming language Java is used. The database is built in SQL using the program MySQL Workbench. An API is required to connect the Android app to the database. The API is created in the Springboot framework, which is based on Java in the IntelliJ IDEA development environment. When creating the API, two features of Springboot are used, namely Spring Data JPA, where Hibernate is used as a service provider, and Spring Security for application security. The Retrofit library is used to connect the Android application to the server. The appearance of the mobile application, that is, the UI layout of the application, is built in the programming language XML.

The application is divided into 2 user roles. The user can create a reservation and see which reservations are completed and which are not. The user can also see all the details of the incomplete reservations such as the reservation number, the date the reservation was made, the date the service was booked, the scheduled end date of the service, etc. The user can also see the details of the completed reservations such as the total price, the price of the parts, the worker who worked on the service, how many working hours were spent on the car service and the cost of the work. The user can send reviews about the car service, and the reviews can be viewed without registration. The administrator has the option of adding new employees and deleting and updating existing employees. The administrator has insight into all completed services and incomplete reservations, not only his own. The administrator can also delete the reservation and has the option of entering data that the user cannot. The user must register in order to use the application and logs into the application with the e-mail address and password he entered. The administrator and the user receive notifications via e-mail about the status of the reservation. The user can get a notification that the reservation has been rejected and why it was rejected or that the service is done while the administrator gets a notification that they have a new reservation to review.

**Keywords:** *Java, Springboot, Android studio, IntelliJ IDEA, SQL, MySQL Workbench, Retrofit.*

## **POPIS KORIŠTENIH KRATICA**

**SQL** Strukturirani jezik upita

**API** Aplikacijsko programsko sučelje

**JPA** Java uporan API

**UI** Korisnično sučelje

**XML** Proširivi označeni jezik

**JVM** Java virtualna mašina

**ORM** Objetno-Relacijsko Mapiranje

**IDE** Integrirano razvojno okruženje

**HTTP** Protokol prijena hiperteksta

**REST** Reprezentativni prijenos stanja

**JSON** JavaScript zapis objekta

**POJO** Običan satri Java objekt

**URL** Jednostavni lokator resursa

**DDL** Jezik za definiranje podataka

**DML** Jezik za manipulaciju podataka

**ER** Enitet-odnos

**SDK** Oprema za razvoj softvera

**PK** Primarni ključ

**FK** Vanjski ključ



# SADRŽAJ

1.	UVOD .....	1
2.	RAZVOJNI ALATI I TEHNOLOGIJE.....	2
2.1.	SPRINGBOOT .....	2
2.1.1.	SPRING DATA JPA.....	3
2.1.2.	SPRING SECURITY.....	3
2.1.3.	HIBERNATE.....	3
2.2.	JAVA.....	4
2.3.	INTELIJ IDEA .....	4
2.4.	ANDROID STUDIO .....	5
2.5.	RETROFIT .....	5
2.6.	SQL (Structured Query Language).....	7
2.7.	MySQL WORKBENCH.....	8
3.	IZRADA BAZE PODATAKA .....	9
4.	IZRADA API.....	12
5.	IZRADA ANDROID MOBILNE APLIKACIJE.....	16
6.	IZGLED I FUNKCIONALNOST ANDROID MOBILE APLIKACIJE.....	20
7.	ZAKLJUČAK .....	52
8.	POPIS LITERATURE .....	54
9.	POPIS SLIKA.....	55

## 1. UVOD

U današnje doba upotrebljavanje tehnologije je veoma bitno kako za velike tvrtke tako i za mala poslovanja. Jedna jednostavna i mala mobilna aplikacija može povećati učinkovitost i produkciju malog poslovanja. Nema potrebe za ručnim pisanjem i evidentiranjem podataka. Aplikacija može automatizirati te pojednostaviti poslovanje.

Upravo to je cilj završnog rada. Izgraditi full-stack mobilnu aplikaciju za izradu i praćenje rezervacija auto servisa korištenjem programskih okvira kao što su Springboot, SQL, Android studio, Retrofit, Hibrnate. Korisnicima aplikacije omogućuje jednostavnu izradu rezervacija za servis svojih automobila, a administratoru omogućuje praćenje svih rezervacija te izmjenjivanje podataka.

Aplikacija radi tako da se korisnik mora prvo registrirati. Nakon registracije korisnik se može prijaviti te kreirati novu rezervaciju. Kada odluči napraviti novu rezervaciju njegovi osobni podaci se uzmu iz baze podataka (ime, prezime, broj telefona) te se neki podaci automatski generiraju (broj rezervacije i datum kada je rezervacija izrađena). Korisnik potom mora upisati podatke u slobodna polja te može poslati rezervaciju samo kada su sva slobodna polja ispunjena. Nakon toga administrator je obaviješten na e-mail da je zaprimio novu rezervaciju, ako nešto nije u redu sa rezervacijom, administrator ima opciju odbaciti rezervacije te se ona briše iz baze podataka i korisnik zaprima e-mail da je rezervacija odbijena i dobiva razlog zašto je rezervacija odbijena. Ako je sve dobro sa rezervacijom administrator će rezervaciju potvrditi i korisnik će zaprijeti e-mail gdje će pisati da je rezervacija potvrđena i dobiti će informaciju kada treba dostaviti auto i predviđeni datum kada će servis biti gotov. Nakon što je servis odrađen administrator upisuje ostale podatke i pošalje rezervaciju koja se u bazi podataka ažurira te je korisnik obaviješten e-mailom da je rezervacija gotova i može doći podići svoj auto.

## 2. RAZVOJNI ALATI I TEHNOLOGIJE

Kod izrade Android mobilne aplikacije koristi se više programa i alata kao što su Springboot, Java, IntelliJ IDEA, Android Studio, Retrofit, XML, SQL i MySQL Workbench.

### 2.1. SPRINGBOOT

Spring Boot je okvir (engl. *framework*) otvorenog koda (engl. *open source*) koji je baziran na programskom jeziku Java. Razvio ga je Pivotal Team i koristi se za izradu samostalnih i produkcijskih Spring aplikacija. Konfiguracijom bilješki (engl. *annotation*) i zadanim kodovima Spring Boot skraćuje vrijeme potrebno za razvoj web aplikacije. Spring Boot pomaže u stvaranju samostalne aplikacije s manje ili gotovo nikakvom konfiguracijom. Autokonfiguracija je posebna značajka (engl. *feature*) u Spring Boot-u. Automatski konfigurira klasu na temelju zahtjeva.

Prednosti Spring Boota:

- Pruža fleksibilan način konfiguriranja Java Beans, XML konfiguracija i transakcija baze podataka.
- U Spring Boot-u sve je automatski konfigurirano, nisu potrebne ručne konfiguracije.
- Jednostavnije prilagođavanje i upravljanje.
- Jednostavnije pokretanje.
- Nudi značajke spremne za proizvodnju.
- Olakšava upravljanje ovisnostima (engl. *dependency management*).
- Pomaže izravno ugraditi Tomcat, Jetty ili Undertow.

### **2.1.1. SPRING DATA JPA**

Spring Data JPA, je dio veće obitelji Spring Data, olakšava implementaciju repozitorija koji se temelju na JPA-u. Ovaj modul bavi se poboljšanom podrškom za slojeve pristupa podacima temeljene na JPA-u. Olakšava izradu Spring aplikacija koje koriste tehnologije pristupa podacima. Implementacija sloja pristupa podacima aplikacije bila je glomazna već neko vrijeme. Previše standardnog koda mora biti napisano za izvršavanje jednostavnih upita, kao i za izvođenje paginacije i revizije. Cilj Spring Data JPA je značajno poboljšati implementaciju slojeva pristupa podacima smanjenjem napora na količinu koja je stvarno potrebna. Programer piše sučelja repozitorija, uključujući prilagođene metode traženja, a Spring će automatski osigurati implementaciju.

### **2.1.2. SPRING SECURITY**

Spring Security je moćan i vrlo prilagodljiv okvir (engl. *framework*) za autentifikaciju i kontrolu pristupa te je standard za osiguranje Spring aplikacija. Spring Security je framework koji se fokusira na pružanje provjere autentičnosti i autorizacije Java aplikacija. Kao i svi Spring projekti, stvarna snaga Spring Security-a leži u tome kako se lako može proširiti kako bi zadovoljio prilagođene zahtjeve.

### **2.1.3. HIBERNATE**

Hibernate je ORM visokih performansi, koja je licencirana pod GNU Lesser General Public License otvorenog koda i može se besplatno preuzeti. Hibernate se brine o preslikavanju iz Java klase u tablicu baze podataka te pretvaranje Java tipova podataka u SQL tipove podataka također pruža mogućnost upita i dohvaćanja podataka.

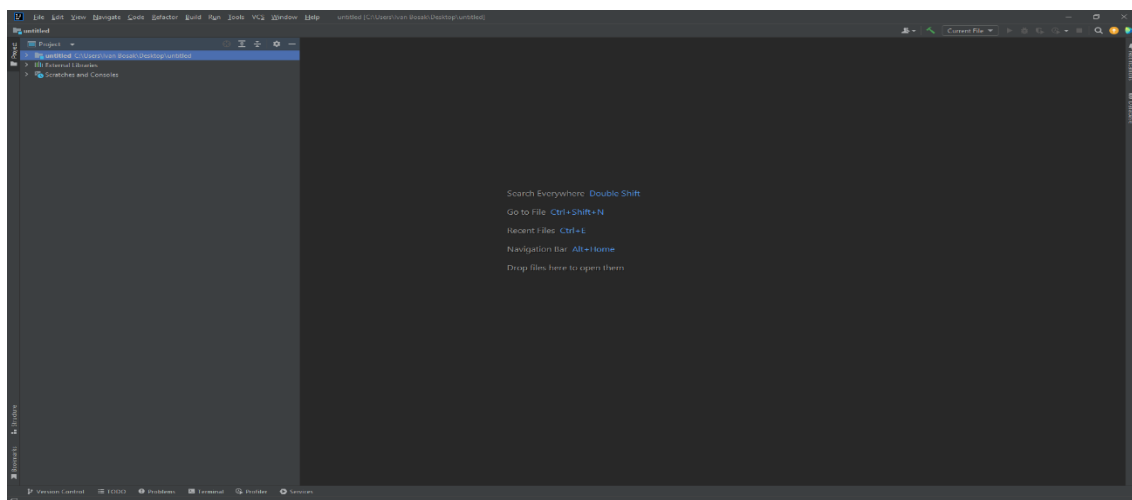
## 2.2. JAVA

Java je klasno zasnovan objektno orijentirani programski jezik koji je dizajniran da ima što manje implementacija ovisnosti (engl. *dependencies*). Namijenjen je omogućiti programerima aplikacija da pišu jednom i izvode bilo gdje, što znači da se kompilirani Java kod može izvoditi na svim platformama koje podržavaju Javu bez potrebe za ponovnom kompilacijom. Java je prvi put objavljena 1995. i naširoko se koristi za razvoj aplikacija za stolna računala, web i mobilne uređaje. Java je poznata po svojoj jednostavnosti, robusnosti i sigurnosnim značajkama, što je čini popularnim izborom za aplikacije na razini poduzeća. Java aplikacije kompajliraju se u byte code koji se može izvoditi na bilo kojem JVM-u.

## 2.3. INTELIJ IDEA

IntelliJ IDEA je posebno programsko okruženje ili integrirano razvojno okruženje uglavnom namijenjeno za razvoj programa u programskom jeziku Java. Razvila ga je tvrtka pod nazivom JetBrains, koja se formalno zvala IntelliJ. IntelliJ IDEA dolazi s jednim od najmoćnijih uređivača koda u industriji. Razumije detalje koda zahvaljujući početnom indeksiranju, što mu omogućuje otkrivanje pogrešaka u hodu, predlaganje opcija dovršetka koda uz preciznu svijest o kontekstu, izvođenje sigurnog refaktoriranja i još mnogo toga.

Slika 1. IntelliJ IDEA



Izvor: Autor

## 2.4. ANDROID STUDIO

Android Studio je službeno integrirano razvojno okruženje za razvoj Android aplikacija. Na temelju snažnog uređivača koda i alata za razvojne programere od IntelliJ IDEA.

Prednosti Android Studia:

- Fleksibilan sustav izrade temeljen na Gradleu
- Brz emulator bogat značajkama
- Jedinstveno okruženje za razvijanje za sve Android uređaje
- Live Edit za ažuriranje komponenata u emulatorima i fizičkim uređajima u stvarnom vremenu
- Predlošci koda i GitHub integracija za pomoć izrade zajedničkih značajki aplikacije i uvezivanje uzoraka koda
- Opsežni alati i okviri za testiranje

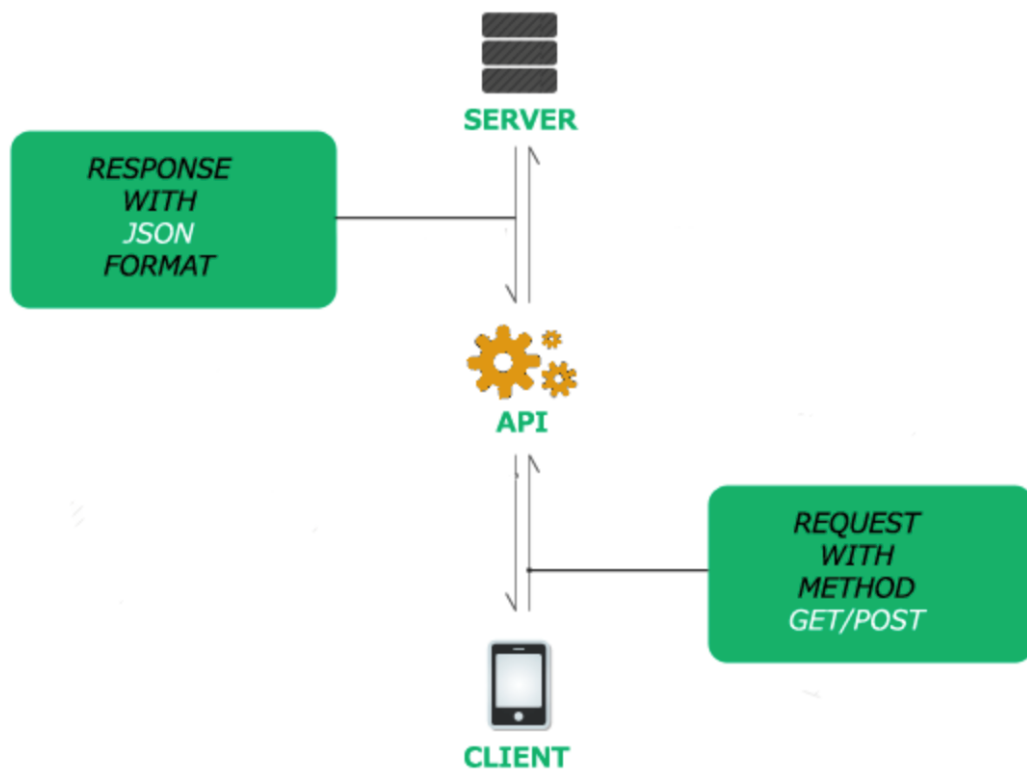
## 2.5. RETROFIT

Biblioteka Retrofit je type-safe REST klijent za Android, Javu i Kotlin, razvio ga je Square. Uz pomoć biblioteke Retrofit, ostvaruje se pristup moćnom okviru koji pomaže u autentifikaciji i interakciji s API-jima te slanju mrežnih zahtjeva s OkHttp-om. Uz pomoć Retrofit biblioteke, preuzimanje JSON ili XML podataka s web API-ja postaje jednostavno. U biblioteci Retrofit, kada se podaci preuzmu, analiziraju se u Plain Old Java Object koji mora biti definiran za svaki "resurs" u odgovoru. Retrofit je jednostavna i brza biblioteka za dohvaćanje i učitavanje podataka putem web usluge temeljene na REST-u. Retrofit upravlja procesom primanja, slanja i stvaranja HTTP zahtjeva i odgovora. Rješava probleme prije slanja pogreške i rušenja aplikacije. Udružuje veze kako bi smanjio kašnjenje te se koristi za spremanje odgovora u predmemoriju kako bi se izbjeglo slanje dvostrukih zahtjeva.

Značajke Retrofit:

- Retrofit je vrlo brz.
- Omogućuje izravnu komunikaciju s web servisom.
- Podržava dinamičke URL-ove.
- Jednostavan je za korištenje i razumijevanje.
- Podržava sinkrone i asinkrone mrežne zahtjeve.
- Podržava pretvarače.
- Podržava otkazivanje zahtjeva.
- Podržava zahtjeve za objavljivanje i višedjelna učitavanja.

Slika 2. Retrofit



Izvor: <https://www.topcoder.com/thrive/articles/retrofit-library-in-android>  
pristup: 23.02.2024.

Slika 3. Primjer Retrofitita u kodu

```
133 HttpLoggingInterceptor httpLoggingInterceptor = new HttpLoggingInterceptor();
134 httpLoggingInterceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
135
136 OkHttpClient okHttpClient = new OkHttpClient.Builder().addInterceptor(httpLoggingInterceptor).build();
137
138 Retrofit.Builder builder = new Retrofit.Builder().baseUrl(LoginActivity.SERVER)
139     .addConverterFactory(GsonConverterFactory.create());
140
141 Retrofit retrofit = builder.client(okHttpClient).build();
142
143 AutoservisAPI autosrvisAPI = retrofit.create(AutoservisAPI.class);
144 Call<User> provjera = autosrvisAPI.Login(user);
```

Izvor: Autor

## 2.6. SQL (Structured Query Language)

SQL je standardizirani programski jezik koji se koristi za upravljanje relacijskim bazama podataka i izvođenje različitih operacija nad podacima u njima. Razvijen je u 1970-ima, SQL redovito koriste administratori baza podataka, programeri koji pišu skripte za integraciju podataka i analitičari podataka koji žele postaviti i pokrenuti analitičke upite.

SQL naredbe podijeljene su u nekoliko različitih vrsta:

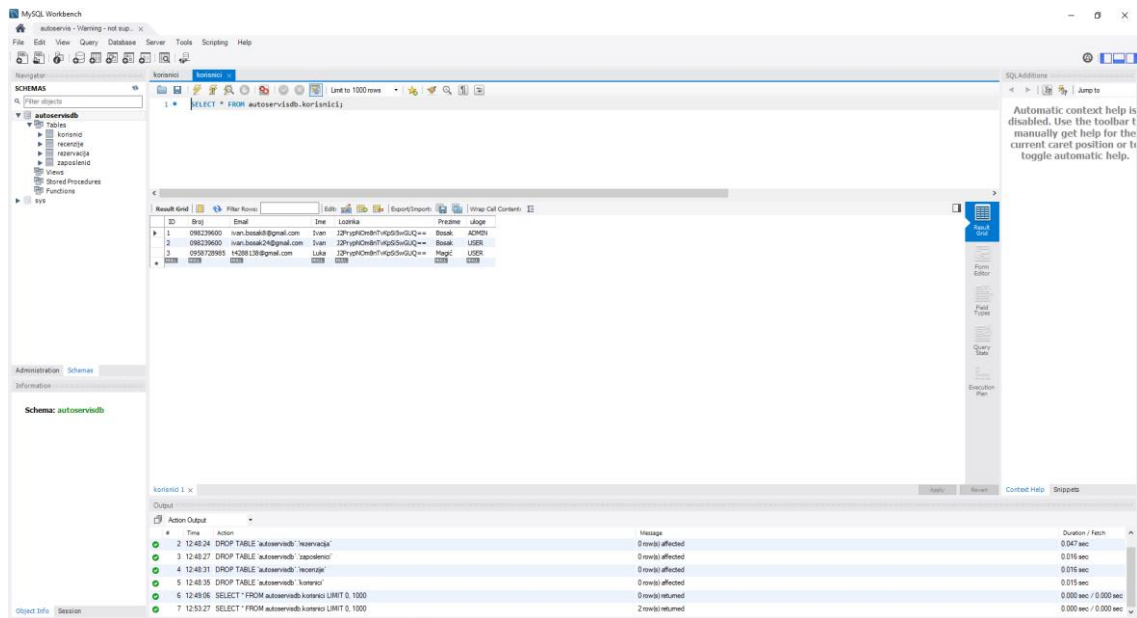
- Data Definition Language - nazivaju se i naredbama za definiranje podataka jer se koriste za definiranje podatkovnih tablica.
- Data Manipulation Language - koriste se za manipulaciju podacima u postojećim tablicama dodavanjem, mijenjanjem ili uklanjanjem podataka. Za razliku od DDL naredbi koje definiraju kako se podaci pohranjuju, DML naredbe djeluju u tablicama definiranim DDL naredbama.
- Data Control Language - koriste se za dodjelu ili opoziv korisničkih povlastica pristupa.
- Transaction Control Language - naredbe se koriste za promjenu stanja nekih podataka -- na primjer, za COMMIT transakcijske promjene ili za ROLLBACK transakcijske promjene.



## 2.7. MySQL WORKBENCH

MySQL Workbench je objedinjeni vizualni alat za arhitekta baza podataka, programere i administratore baza podataka. MySQL Workbench pruža modeliranje podataka, razvoj SQL-a i opsežne administrativne alate za konfiguraciju poslužitelja, korisničku administraciju, sigurnosno kopiranje i još mnogo toga.

Slika 4. Baza podataka u MySQL Workbench-u

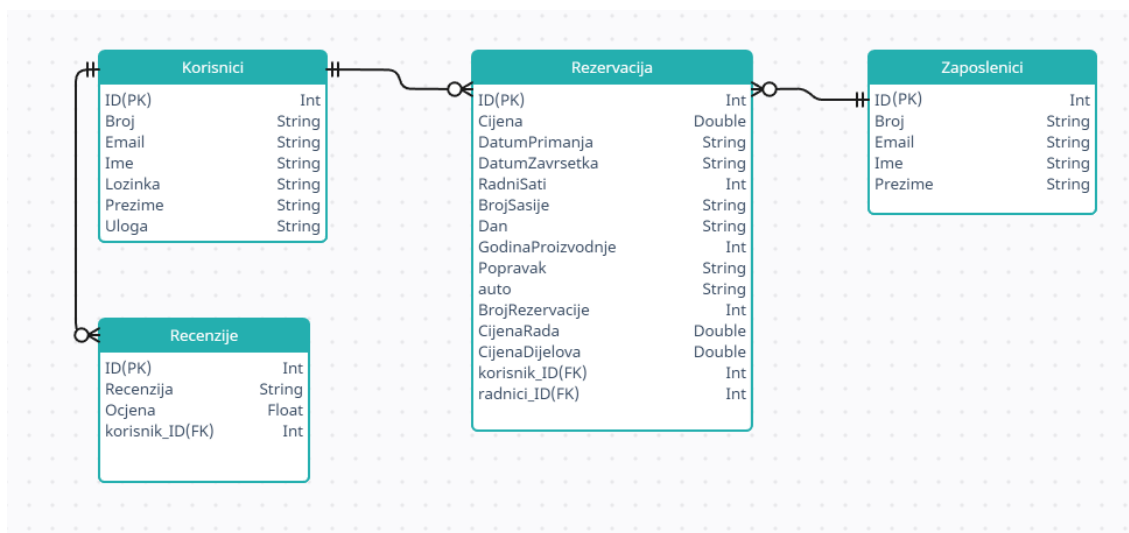


Izvor: Autor

### 3. IZRADA BAZE PODATAKA

Prije izrade baze podataka u MySQL Workbench-u treba izraditi ER model baze podataka. U ovom projektu baza podataka se sastoji od 4 entiteta a to su korisnici, rezervacija, zaposlenici i recenzija. Primarni ključ (PK) u tablici korisnici je ID. U tablici rezervacija primarni ključ je isto ID ali također ima i 2 vanjska ključa (FK) koji su korisnik\_ID i radnici\_ID. Nadalje, u tablici recenzije primarni ključ isto ID uz 1 vanjski ključ korisnik\_ID dok je u tablici zaposlenici primarni ključ ID.

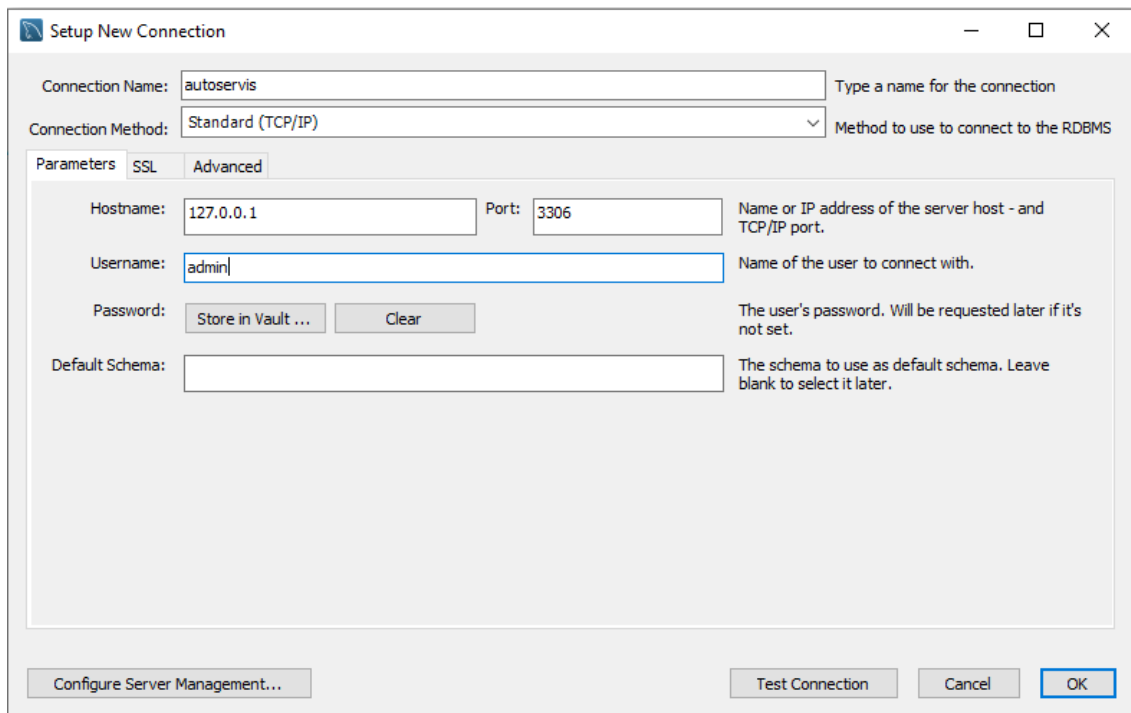
Slika 5. ER model baze podataka



Izvor: Autor

Nakon izrade ER modela u MySQL Workbench-u se izradi nova konekcija pod imenom „autoservis“. Konekcija mora imati korisničko ime (engl. *username*) i lozinku (engl. *password*) da bi se kasnije API mogao spojiti sa bazom podataka. U ovom slučaju korisničko ime je „admin“ a lozinka je „password“. Konekcija također mora imati navedeni port koji će se koristiti za spajanje sa API-em a to je port 3306.

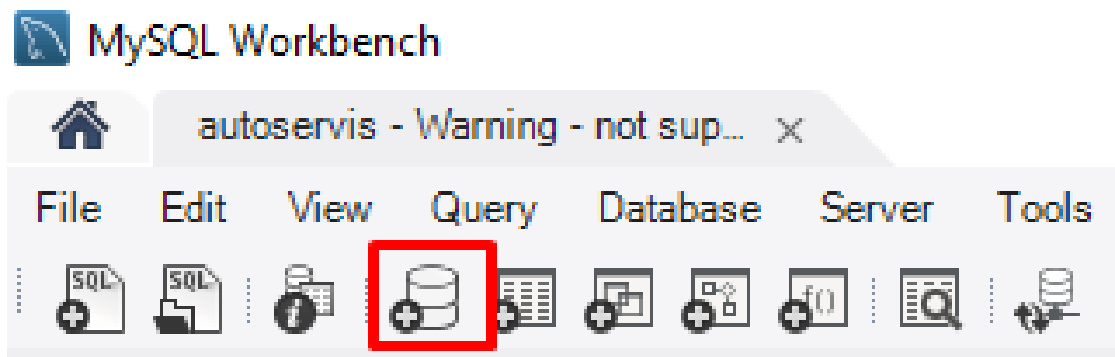
Slika 6. Kreiranje nove konekcije u MySQL Workbench-u



Izvor: Autor

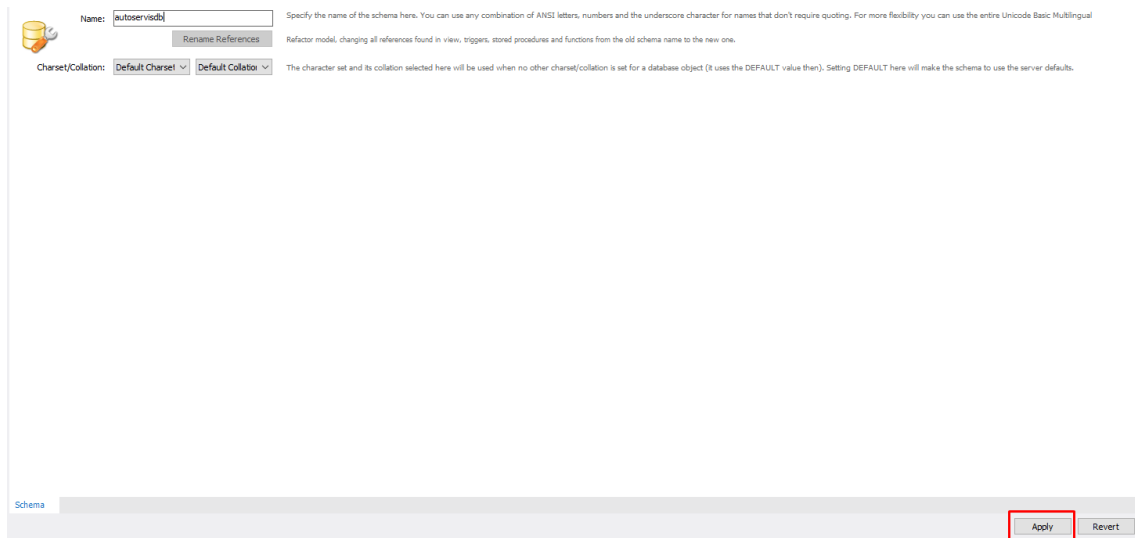
Nakon kreiranja konekcije treba se izraditi nova shema. U MySQL Workbench-u ispod opcije „file“ pritisne se četvrta opcija „create a new schema in the connected server“, nova shema nazove se autoservisdb i pritisne se „Apply“ te će se nova shema kreirati.

Slika 7. Opcija „create a new schema in the connected server“ za izradu nove sheme



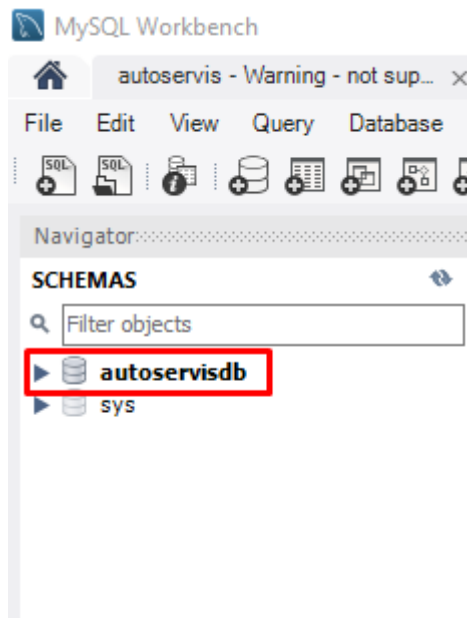
Izvor: Autor

Slika 8. Nakon pritiska opcije „create a new schema in the connected server“



Izvor: Autor

Slika 9. Kreirana nova shema

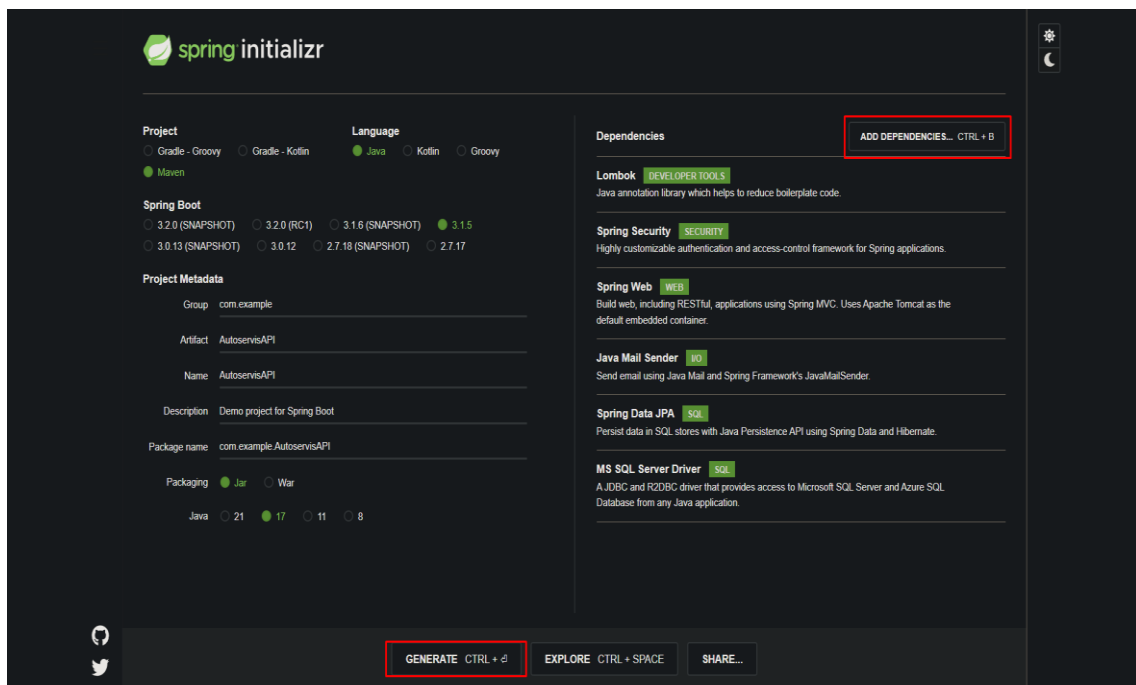


Izvor: Autor

## 4. IZRADA API

Za izradu API koristi se web alat Spring initalizer gdje se odaberu sve ovisnosti (engl. *dependencies*) koje su potrebne za izradu API. Za korištenje Spring initalizer-a treba otvoriti web stranici <https://start.spring.io/> . Za vrstu projekta odabere se opcija „Maven“ a za programski jezik odabere se opcija „Java“, verzija Springboot- a je 3.1.5. Nakon toga upiše se ime projekta u ovom slučaju ime projekta je „AutoservisAPI“. Odabere se verzija Jave „17“ te se pritisne opcija dodaj ovisnosti (engl. *Add dependencies*). Odaberu se ovisnosti kao što su Spring Web, Spring Security, Spring Data JPA, Java Mail Sender, Lombok i MS SQL Server Driver. Nakon toga pritisne se opcija generiraj (engl. *Generate*) te se projekt skida na računalo.

Slika 10. Web alat Spring initalizr



Izvor: Autor

Prva stvar koja se treba napraviti je definirati u `application.properties` na kojem priključaku (engl. *port*) će se nalaziti server a u ovom slučaju to je port 8090. Do `application.properties` dolazi se putanjom `src>main>resources>application.properties` te se upiše linija koda 2 vidljiva na slici 11. Iduće treba definirati konekciju na bazu podataka, a to se postiže linijom koda 1 vidljiva na slici 11. Nakon toga aplikacija se treba spojiti na bazu podataka kako bi se podaci u bazi mogli izmjenjivati. To postiže linijama koda 3 i 4 vidljive na slici 11. Ove dvije linije koda definiraju korisničko ime i lozinku koje su zadane u MySQL Workbench-u.

Slika 11. Kod za definiranje priključka i konekciju na bazu podataka

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/autoservisdb
2 server.port = 8090
3 spring.datasource.username=admin
4 spring.datasource.password=password
```

Izvor: Autor

Također ovdje se treba definirati koji priključak će se koristiti za slanje poruka putem emaila te email koji će se koristiti i lozinka toga emaila. Linije koda koje ovo definiraju su vidljive na slici 12.

Slika 12. Kod za definiranje email servisa

```
12 spring.mail.host=smtp.gmail.com
13 spring.mail.port=587
14 spring.mail.username=danas.autoservis@gmail.com
15 spring.mail.password=qmjz bruf egef dzki
16 spring.mail.properties.mail.smtp.auth=true
17 spring.mail.properties.mail.smtp.starttls.enable=true
```

Izvor: Autor

Nakon toga u API-u treba izraditi business logiku a to se postiže izradom 4 paketa (engl. *package*). Pritiskom desnog klika miša na com.Autoservis.Autoservis pokaže se opcija new i pritisne se package. Nazovu se controllers, enteties, repositories i services. U paketu entiteti (engl. *enteties*) izrade se 4 klase u kojima se definiraju objekti user, rezervacija, zaposlenik i recenzije. U paketu repozitorij (engl. *repositories*) se izrade 4 interface-a za izradu prilagođenih SQL upita. U paketu kontroleri (engl. *controllers*) se izrade 5 klasi, 4 klase služe za definiranje url-a na kojima se mogu dobivati podaci iz baze podataka (engl. *get*), slati podatke u bazu podataka (engl. *post*), ažurirati podatke u bazi podataka (engl. *put*) i brisnje podatka (engl. *delete*) dok 5 klasa služi za definiranje sigurnosne konfiguracije. I u zadnjem paketu servisi (engl. *services*) se definiraju sve metode koje se koriste u paketu kontroleri (engl. *controllers*).

Slika 13. Metode za slanje lozinke korisniku i za enkripciju i dekripciju lozinke

```
72     public void posaljilozinku(String email) throws Exception{
73         Optional<User> user = userRepository.findByEmail(email);
74         if (user.isPresent()){
75             String sifriranaLozinka = user.get().getLozinka();
76             emailService.posaljilozinku(email,dekripcija(sifriranaLozinka));
77         }
78
79         user.orElseThrow()-> new FindException("Korisnik sa ovom email poštom ne postoji");
80
81     }
82     1 usage
83     @ private String dekripcija(String password) throws Exception{
84         SecretKeySpec key = generirajKljuc();
85         Cipher c = Cipher.getInstance("AES");
86         c.init(Cipher.DECRYPT_MODE, key);
87         byte [] value= Base64.getMimeDecoder().decode(password);
88         byte [] dekodiraj= c.doFinal(value);
89         return new String(dekodiraj);
90     }
91     1 usage
92     @ private SecretKeySpec generirajKljuc() throws Exception{
93         final MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
94         byte[] bytes ="Autoservis123#".getBytes(StandardCharsets.UTF_8);
95         messageDigest.update(bytes, offset: 0, bytes.length);
96         byte[] key = messageDigest.digest();
97         return new SecretKeySpec(key, "AES");
98     }
```

Izvor: Autor

Slika 14. Get, Post i Put metode za rezervacije

```
55 public Rezervacija getRezervacijaByUser(int id){
56     Rezervacija rezervacijaObj= new Rezervacija();
57     Optional<Rezervacija> rezervacija= rezervacijaRepository.findById(id);
58     if (rezervacija.isPresent()){
59         rezervacija.filter(rezervacija ->!rezervacija.getRazlog().equals("nema"));
60         rezervacijaObj=rezervacija.get();
61     }
62     rezervacija.orElseThrow(()-> new NoSuchElementException("Rezervacija ne postoji."));
63     return rezervacijaObj;
64 }
1 usage
65 @ public void addRezervacija(Rezervacija rezervacija){
66     Optional<User> user = userRepository.findById(rezervacija.getKorisnik().getID());
67     if (user.isPresent()){
68         if (user.get().getUloge().equals("USER")){
69             List<User> adminList= new ArrayList<>(userRepository.findByUloge("ADMIN"));
70             for (User admin:adminList){
71                 emailService.posaljiEmailZaDodatnuRezervaciju(admin.getEmail());
72             }
73         }
74     }
75     user.orElseThrow(()-> new NoSuchElementException("Korisnik ne postoji."));
76     rezervacijaRepository.save(rezervacija);
77 }
1 usage
79 @ public void updateRezervacija(Rezervacija rezervacija){
80     Optional<Rezervacija> rez = rezervacijaRepository.findById(rezervacija.getID());
81     if(rez.isPresent()){
82         User korisnik = rez.get().getKorisnik();
83         emailService.posaljiEmailZaGotovuRezervaciju(korisnik.getEmail());
84     }
85     rez.orElseThrow(() -> new NoSuchElementException("Rezervacija ne postoji."));
86     rezervacijaRepository.save(rezervacija);
87 }
```

Izvor: Autor

Slika 15. Metoda za slanje lozinke korisniku putem e-maila

```
36 public void posaljiLozinku(String email, String lozinka) {
37     simpleMailMasage.setTo(email);
38     simpleMailMasage.setFrom("danas.autoservis@gmail.com");
39     simpleMailMasage.setSubject("VAŠA LOZINKA");
40     simpleMailMasage.setText("Vaša lozinka je: " + lozinka);
41
42     javaMailSender.send(simpleMailMasage);
43
44 }
```

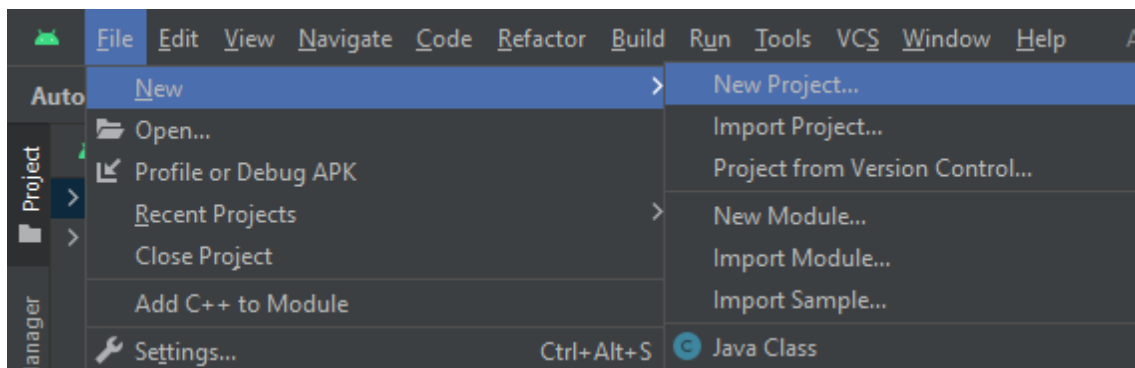
Izvor: Autor



## 5. IZRADA ANDROID MOBILNE APLIKACIJE

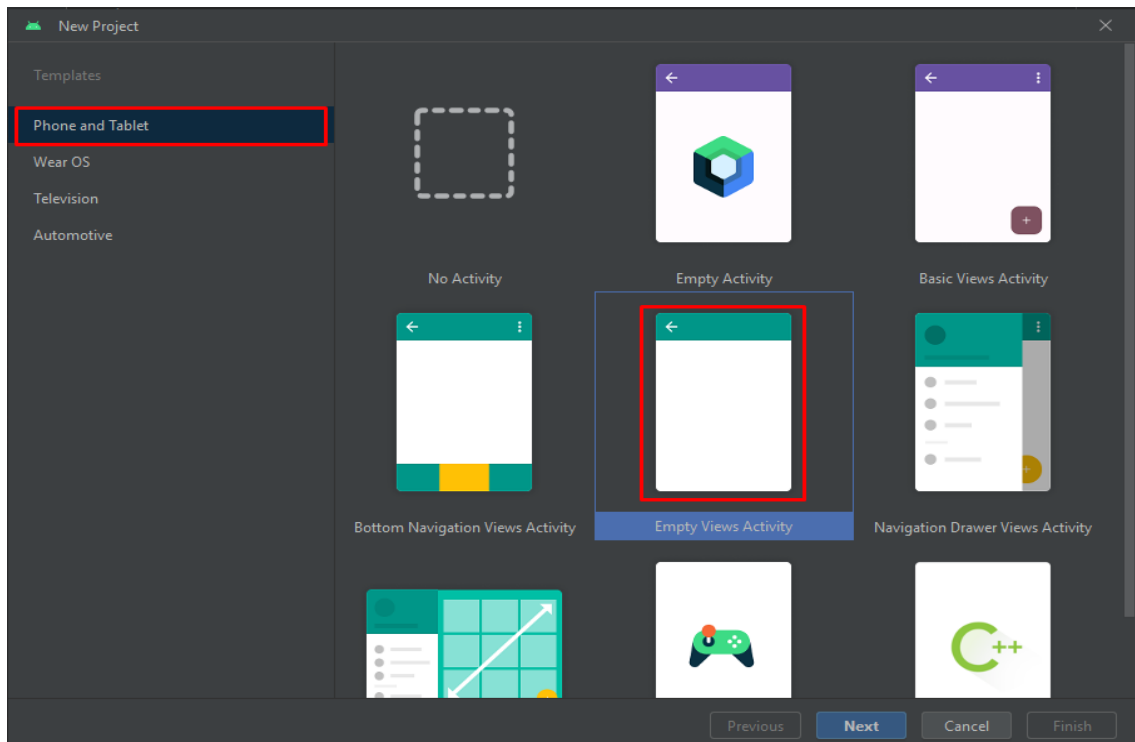
Za izradu aplikacije otvori se Android Studio, pritisne se opcija `file>new>New Project`. Nakon pritiska na opciju `New Project` otvori se novi prozor gdje se može odabrati na koji platformi će se aplikacija pokretati. Ovdje se odabere opcija „Phone and Tablet“ te opcija `Empty Views Activity`. Odaberu se ove dvije opcije i pritisne se gumb „Next“. Nakon pritiska gumba otvorit će se novi prozor gdje se mora upisati ime aplikacije u ovom slučaju je to „Autoservis“. Za programski jezik odabere se opcija „Java“, za Minimum SDK odabere se opcija „Android 7.0“ i za opciju `Build Configuration Language` se odabere „Kotlin DSL“. Na kraju se pritisne gumb `finish` i novi projekt je kreiran.

Slika 16. Kreiranje novog projekta u Android Studiu



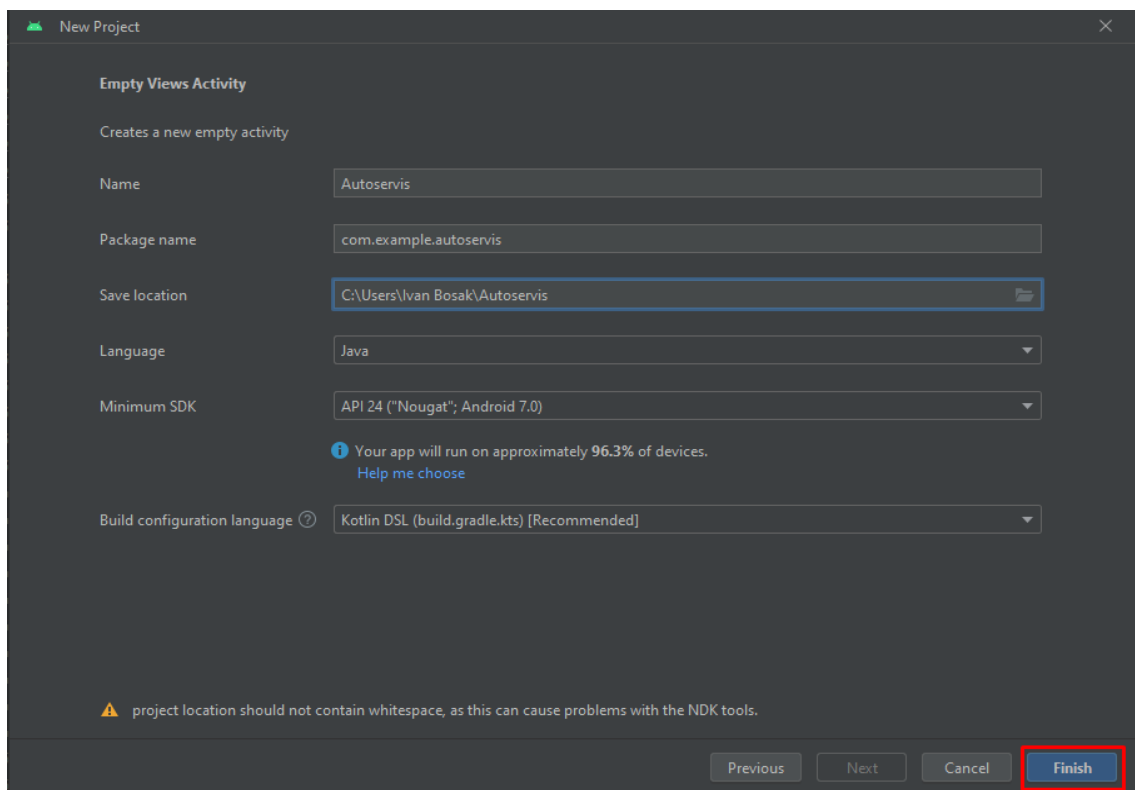
Izvor: Autor

Slika 17. Odabir platforme za mobitel



Izvor: Autor

Slika 18. Konfiguracija novog projekta



Izvor: Autor

Prva stvar koja se treba napraviti je uvesti ovisnosti (engl. *dependencies*) u Gradle Scripts > build.gradle(Module: app). Ovdje se upišu 3 linije koda za uvoz ovisnosti Retrofit koja šalje i prima HTTP zahtjeve.

Slika 19. Ovisnost Retrofit

```
39 implementation("com.squareup.retrofit2:retrofit:2.9.0")
40 implementation("com.squareup.retrofit2:converter-gson:2.9.0")
41 implementation("com.squareup.okhttp3:logging-interceptor:4.8.1")
```

Izvor: Autor

Slika 20. Linija koda za definiranje pristupa internetu u mapi Android Manifest.xml

```
5 <uses-permission android:name="android.permission.INTERNET" />
```

Izvor: Autor

Kada se aplikacija kreirala, automatski se izgenerirala Java klasa MainActivity i u mapi layout Layout Resorce File koja se zove activity\_main.xml. Java klasu MainActivity preimenovana je u LoginActivity te se u njoj definira logika koja pokreće aplikaciju, dok se activity\_main.xml definira izgled aplikacije te je to prva stvar koju korisnik vidi kada pokrene aplikaciju.

Slika 21. Primjer koda u klasi LoginActivity

```
32 private void provjeriLogin(User user) {
33     HttpLoggingInterceptor httpLoggingInterceptor = new HttpLoggingInterceptor();
34     httpLoggingInterceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
35
36     OkHttpClient okHttpClient = new OkHttpClient.Builder().addInterceptor(httpLoggingInterceptor).build();
37
38     Retrofit.Builder builder = new Retrofit.Builder().baseUrl(LoginActivity.SERVER)
39         .addConverterFactory(GsonConverterFactory.create());
40
41     Retrofit retrofit = builder.client(okHttpClient).build();
42
43     AutoservisAPI autoservisAPI = retrofit.create(AutoservisAPI.class);
44     Call<User> provjera = autoservisAPI.login(user);
45
46     provjera.enqueue(new Callback<User>() {
47         @Override
48         public void onResponse(Call<User> call, Response<User> response) {
49             if (response.isSuccessful()) {
50                 assert response.body() != null;
51                 User trenutni = response.body();
52                 Intent intent;
53
54                 if (trenutni.getUloge().equals("USER")) {
55
56                     intent = new Intent(packageContext, LoginActivity.this, UserHomeUI.class);
57                     intent.putExtra("trenutniUser", trenutni);
58
59                 } else {
60
61                     intent = new Intent(packageContext, LoginActivity.this, AdminHomeUI.class);
62                     intent.putExtra("trenutniUser", trenutni);
63                 }
64                 startActivity(intent);
65             } else {
66                 findViewById(R.id.loading).setVisibility(View.INVISIBLE);
67                 Toast.makeText(context, LoginActivity.this, "E-mail ili lozinka nisu točni.\nPokušajte ponovo...", Toast.LENGTH_LONG).show();
68             }
69         }
70     });
}
```

Izvor: Autor

Slika 22. Primjer koda u activity\_main.xml

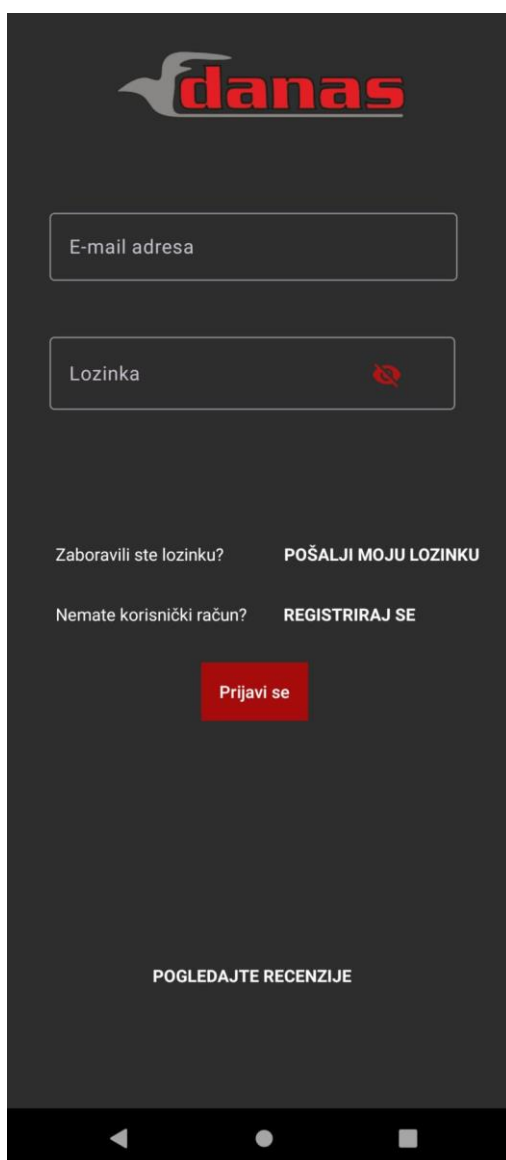
```
25 <LinearLayout
26     android:layout_width="match_parent"
27     android:layout_height="wrap_content"
28     android:layout_marginTop="20dp"
29     android:orientation="vertical">
30
31     <com.google.android.material.textfield.TextInputLayout
32         android:layout_width="match_parent"
33         android:layout_height="wrap_content"
34         android:paddingStart="35dp"
35         android:paddingTop="20dp"
36         android:paddingEnd="10dp"
37         app:errorEnabled="true">
38
39
40         <com.google.android.material.textfield.TextInputEditText
41             android:id="@+id/edittext_username"
42             android:layout_width="336dp"
43             android:layout_height="56dp"
44             android:hint="E-mail adresa"
45             android:textColor="@color/white"
46             android:inputType="textEmailAddress" />
47
48
49
50
51     </com.google.android.material.textfield.TextInputLayout>
```

Izvor: Autor

## 6. IZGLED I FUNKCIONALNOST ANDROID MOBILE APLIKACIJE

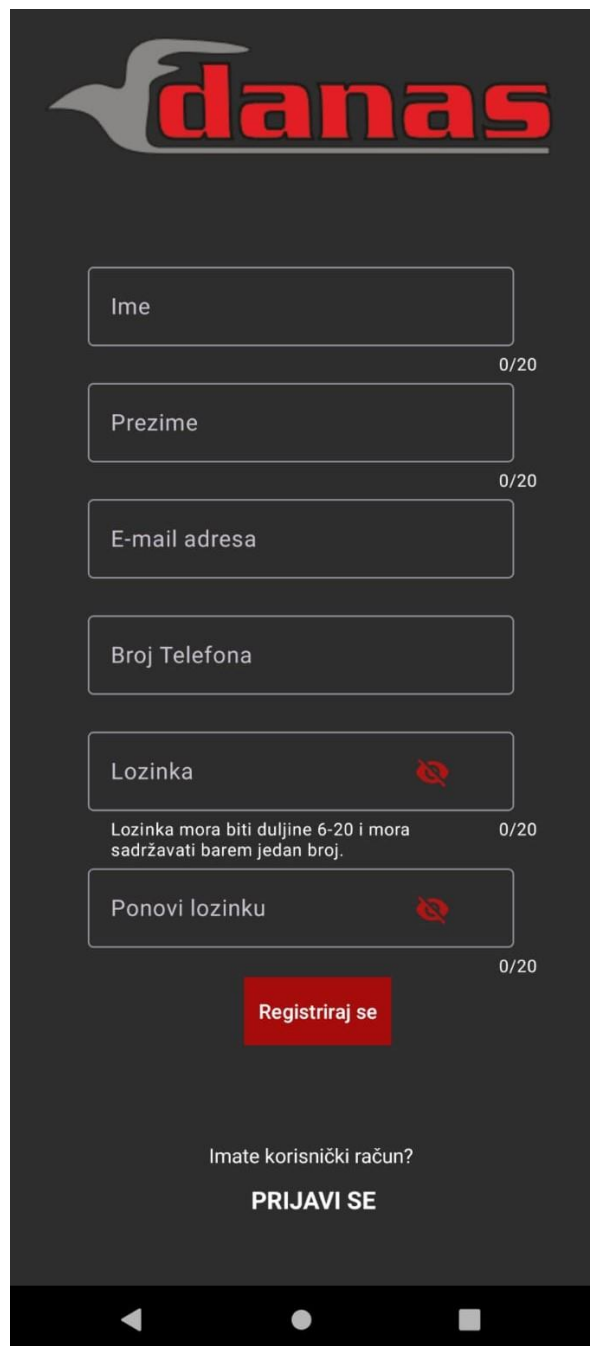
Prva stvar koju korisnik vidi kada otvori aplikaciju je LoginActivity to jest forma za prijavu. Ovdje se korisnik može prijaviti ili ako nema korisnički račun može pritisnuti opciju za registraciju. Ako korisnik ima korisnički račun, ali je zaboravio svoju lozinku, može pritisnuti opciju pošalji moju lozinku gdje može upisati svoju e-mail adresu te mu na e-mail dolazi poruka koja je njegova lozinka. Ovdje je i opcija pogledajte recenzije gdje korisnici mogu pogledati recenzije koje su oni ili neko drugi poslali.

Slika 23. Forma za prijavu



Izvor: Autor

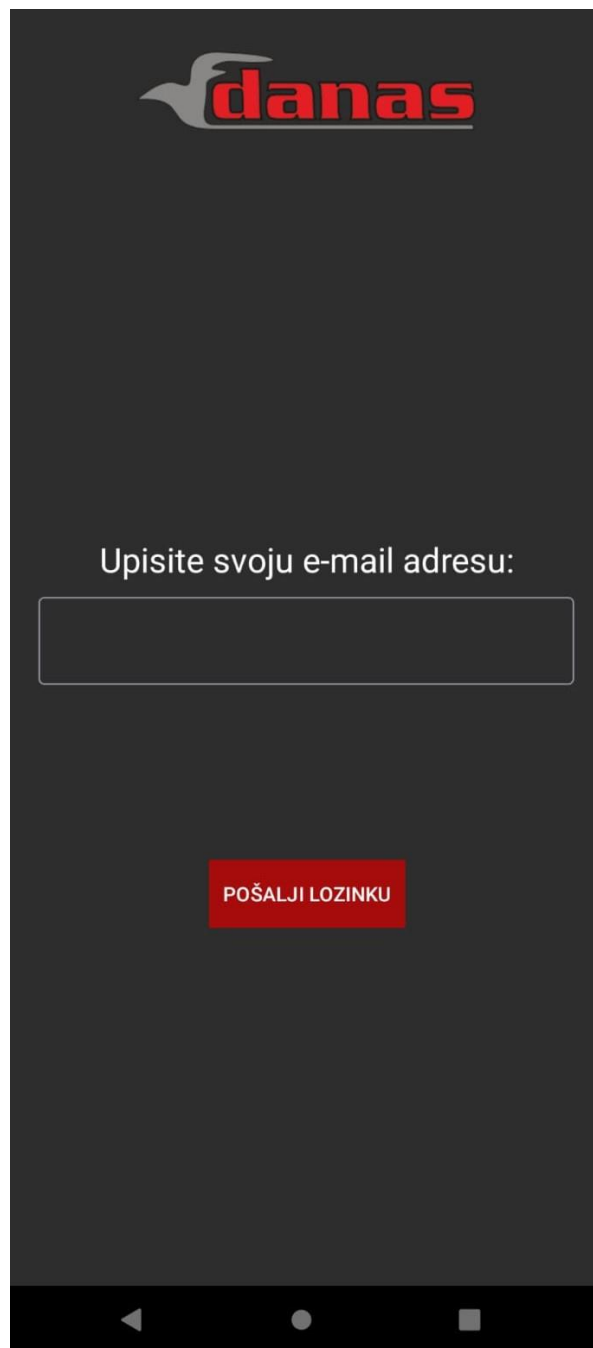
Slika 24. Forma za registraciju



The image shows a mobile registration form for the 'danas' website. At the top is the 'danas' logo in red and white. Below the logo are several input fields: 'Ime' (0/20), 'Prezime' (0/20), 'E-mail adresa', 'Broj Telefona', 'Lozinka' (0/20), and 'Ponovi lozinku' (0/20). The password field has a red eye icon and a note: 'Lozinka mora biti duljine 6-20 i mora sadržavati barem jedan broj.' Below the fields is a red 'Registriraj se' button. At the bottom, there is a link 'Imate korisnički račun?' and a 'PRIJAVI SE' button. The form is set against a dark background.

Izvor: Autor

Slika 25. Forma za slanje lozinke korisniku



**danas**

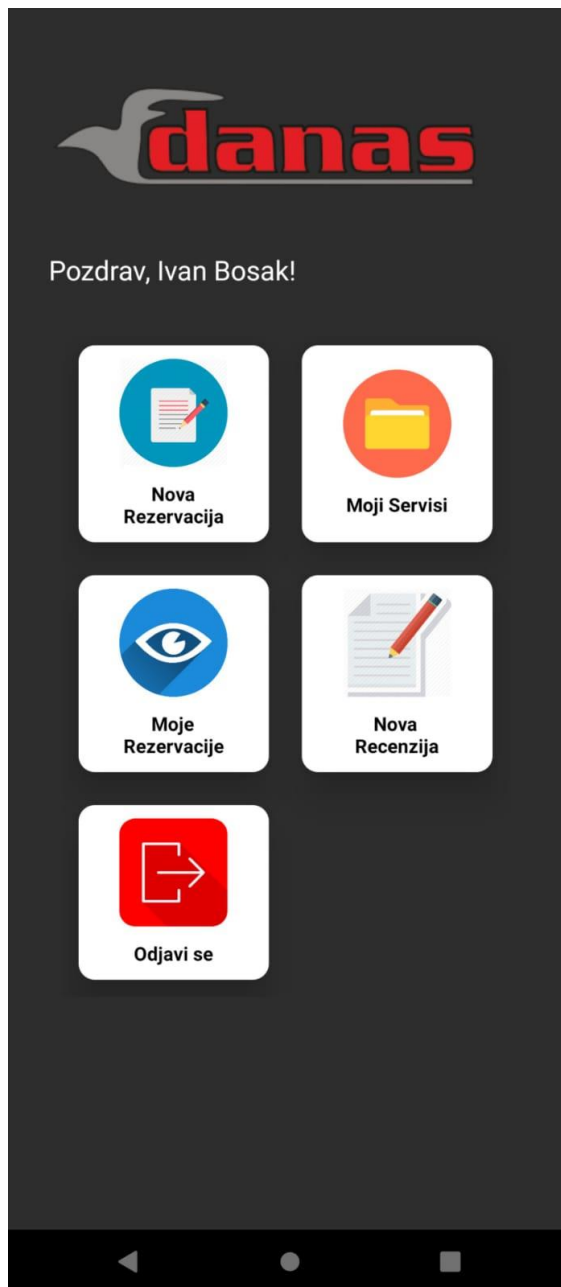
Upisite svoju e-mail adresu:

**POŠALJI LOZINKU**

Izvor: Autor

Nakon prijave aplikacija će preusmjeriti korisnika ovisno o tome je li on administrator ili samo obični korisnik. Korisnik nakon prijave ima samo 5 opcija, a to su Nova rezervacija gdje može rezervirati svoj auto za servis. Kada korisnik pošalje rezervaciju administrator dobiva e-mail da ima novu rezervaciju za pregledati. U opciji Moji servisi mogu se pogledati svi podaci o servisima koji su gotovi, dok u opciji Moje rezervacije mogu se pogledati podaci o rezervacijama. Opcija Nova recenzija šalje recenziju auto servisa i naravno opcija za odjavu.

Slika 26. Korisničko sučelje za običnog korisnika



Izvor: Autor



Slika 27. Upis podataka za običnog korisnika

**danas**

**Ivan Bosak**  
**Datum:** 23.02.2024  
**Broj Telefona:** 098239600  
**Datum primanja:**  
ODABERITE DATUM

**Auto:**  
Upišite marku i ime automobila

**Godina Proizvodnje:**  
Upišite godinu proizvodnje automobila

**Broj šasije:**  
Upišite broj šasije automobila

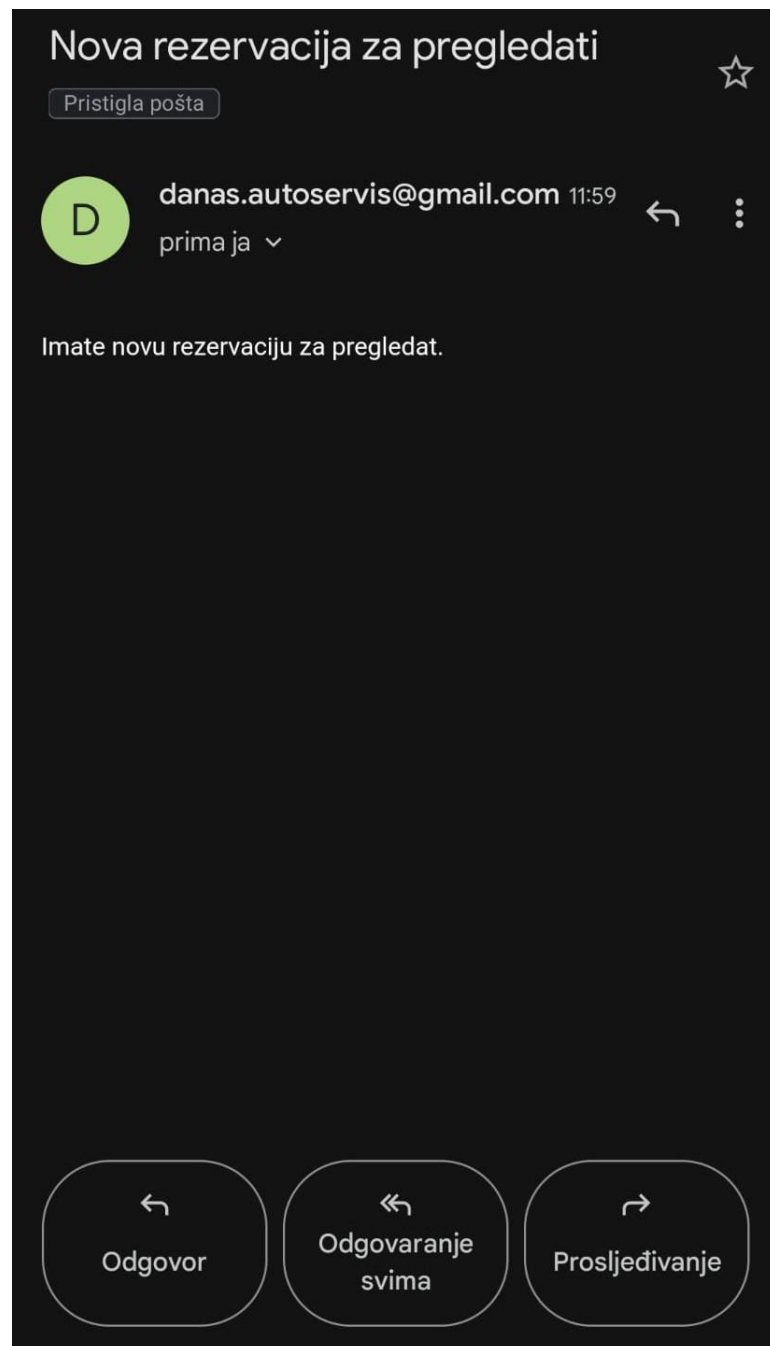
**Što treba popraviti?**  
Upišite što treba popraviti na automobilu

Pošalji Rezervaciju

Izađi

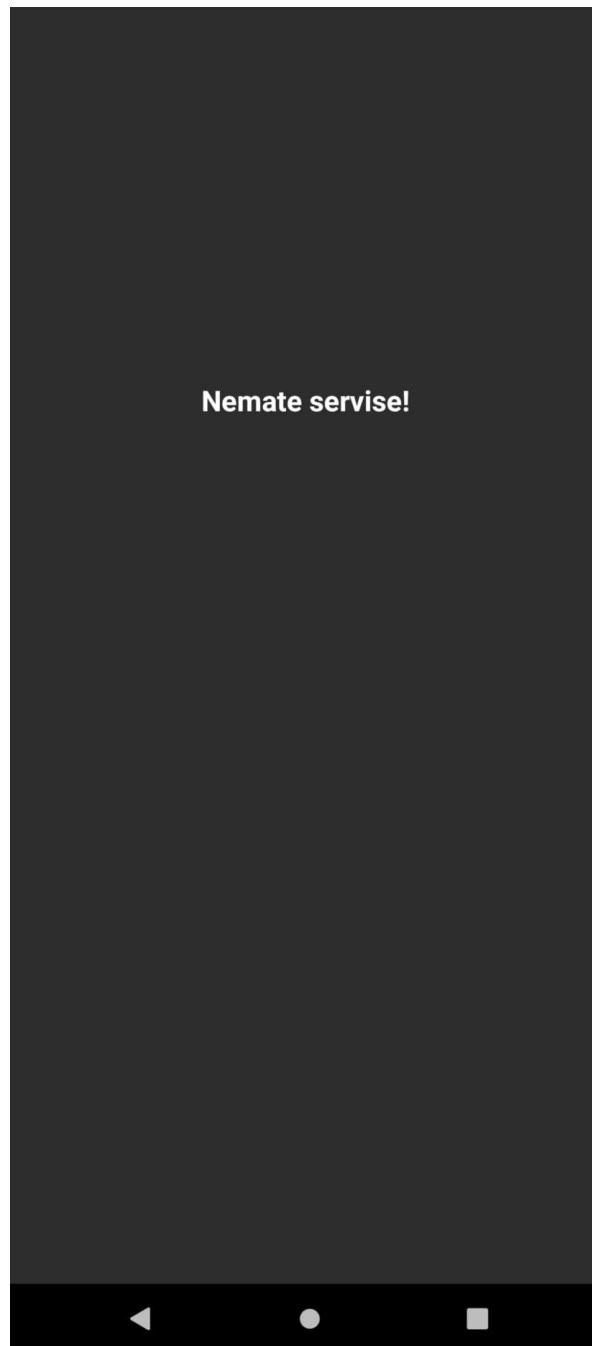
Izvor: Autor

Slika 28. E-mail poslan administratoru kada korisnik pošalje novu rezervaciju



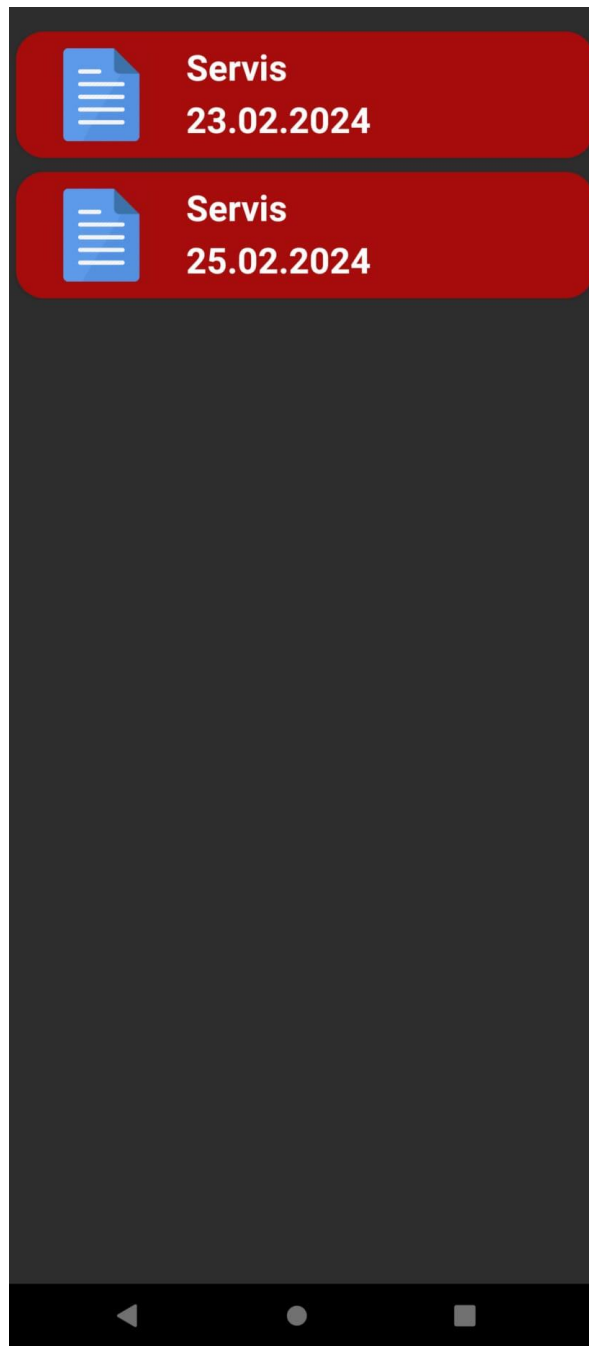
Izvor: Autor

Slika 29. Opcija Moji servisi ako korisnik nema niti jedan servis



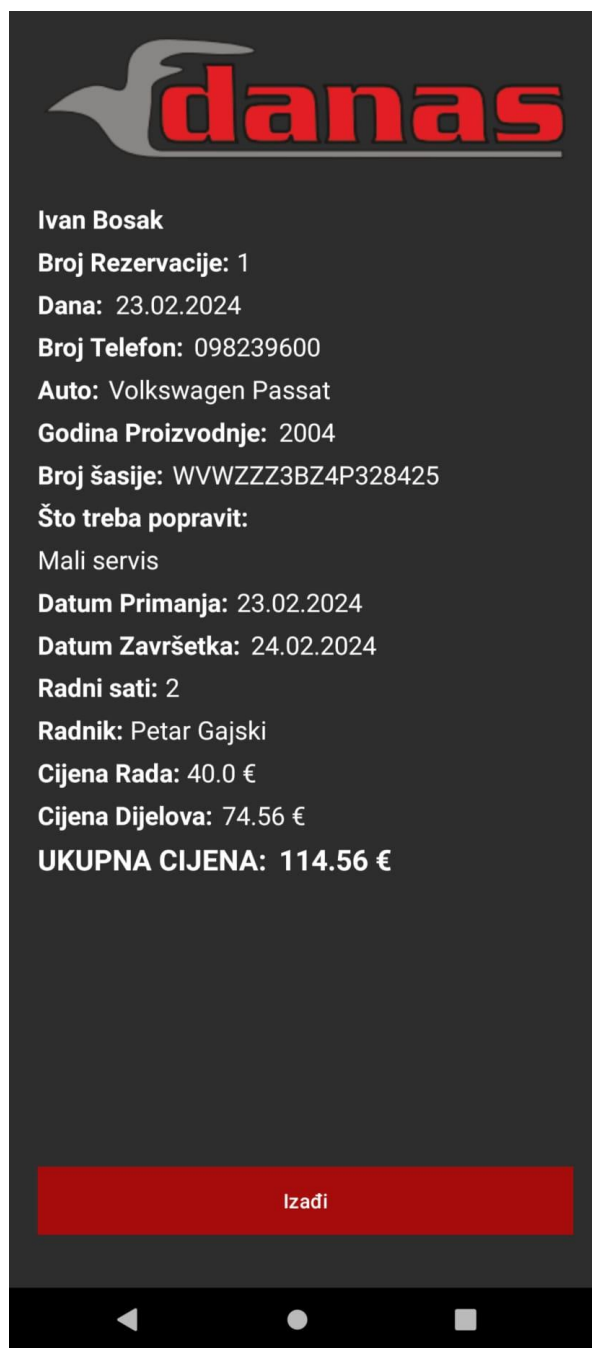
Izvor: Autor

Slika 30. Opcija Moji servisi ako korisnik ima gotove servise



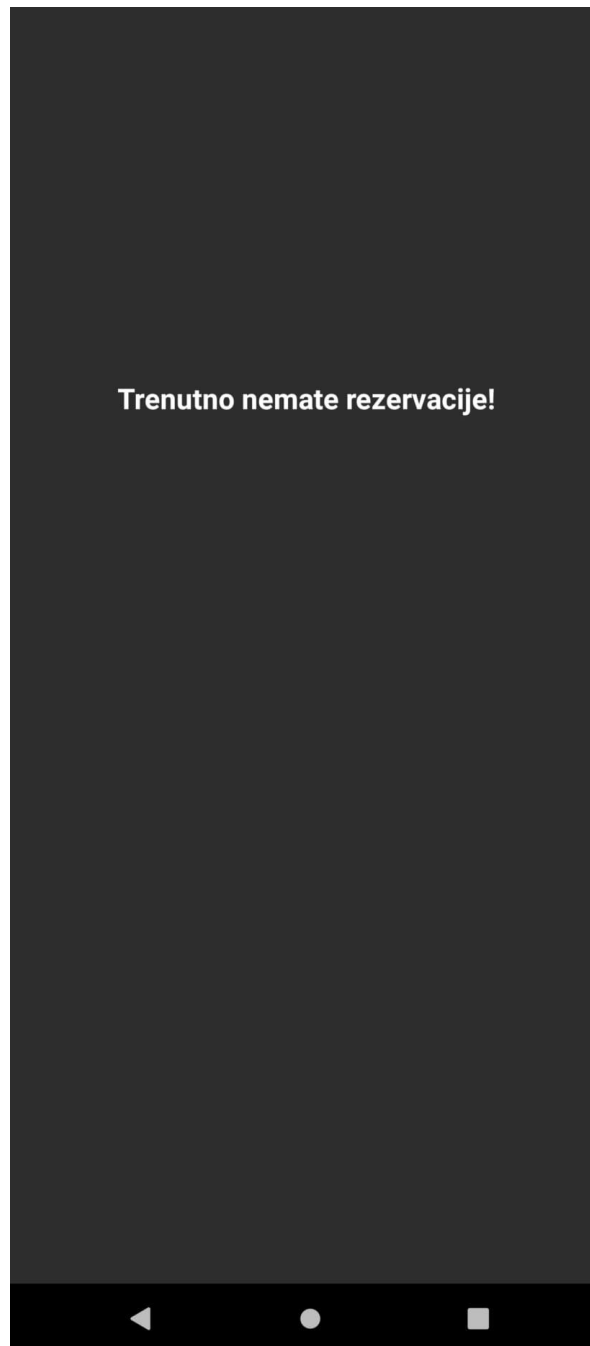
Izvor: Autor

Slika 31. Detalji odabranog gotovog servisa



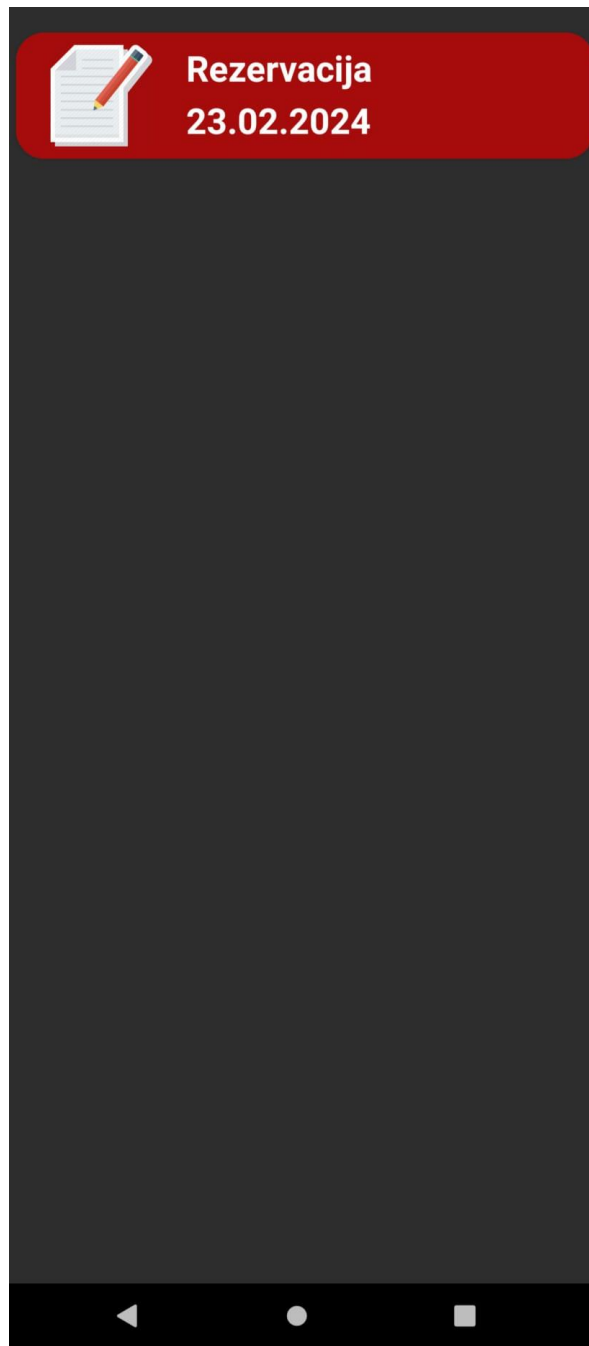
Izvor: Autor

Slika 32. Opcija Moje rezervacije ako korisnik nema niti jednu rezervaciju



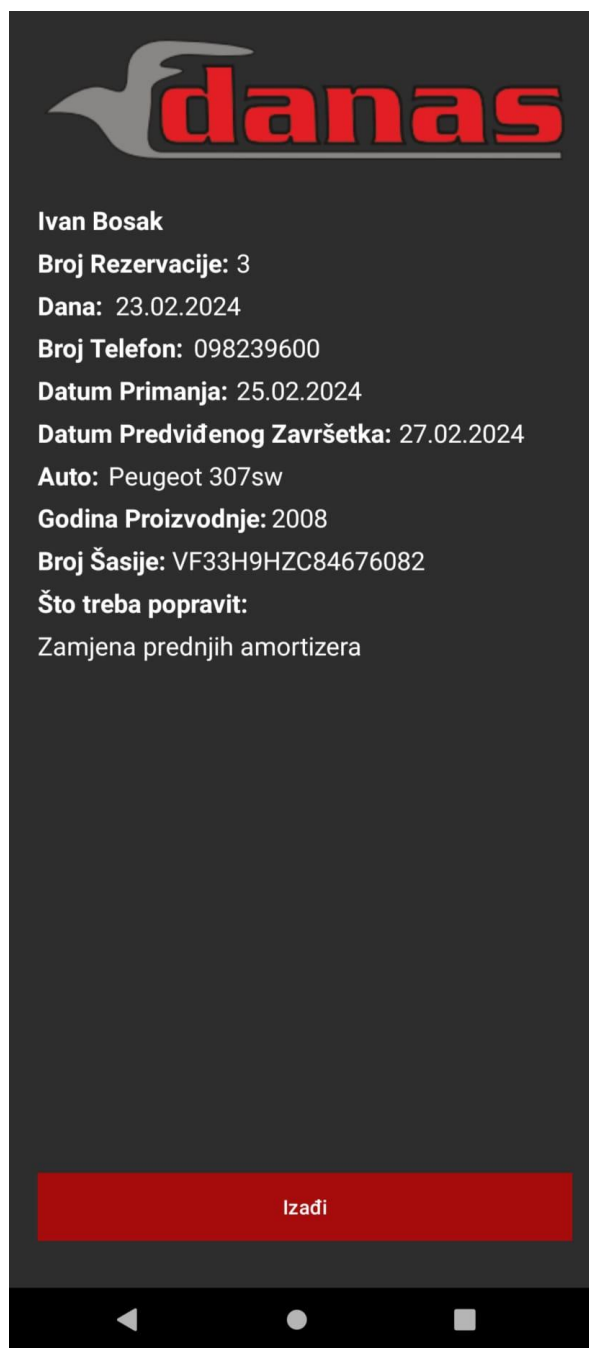
Izvor: Autor

Slika 33. Opcija Moje rezervacije kada korisnik ima rezervacije



Izvor: Autor

Slika 34. Detalji odabrane rezervacije



Izvor: Autor



Slika 35. Opcija Nova recenzija

**danas**

**Recenzija:**

Upišite svoju recenziju

**Ocijenite od 1 do 5:**

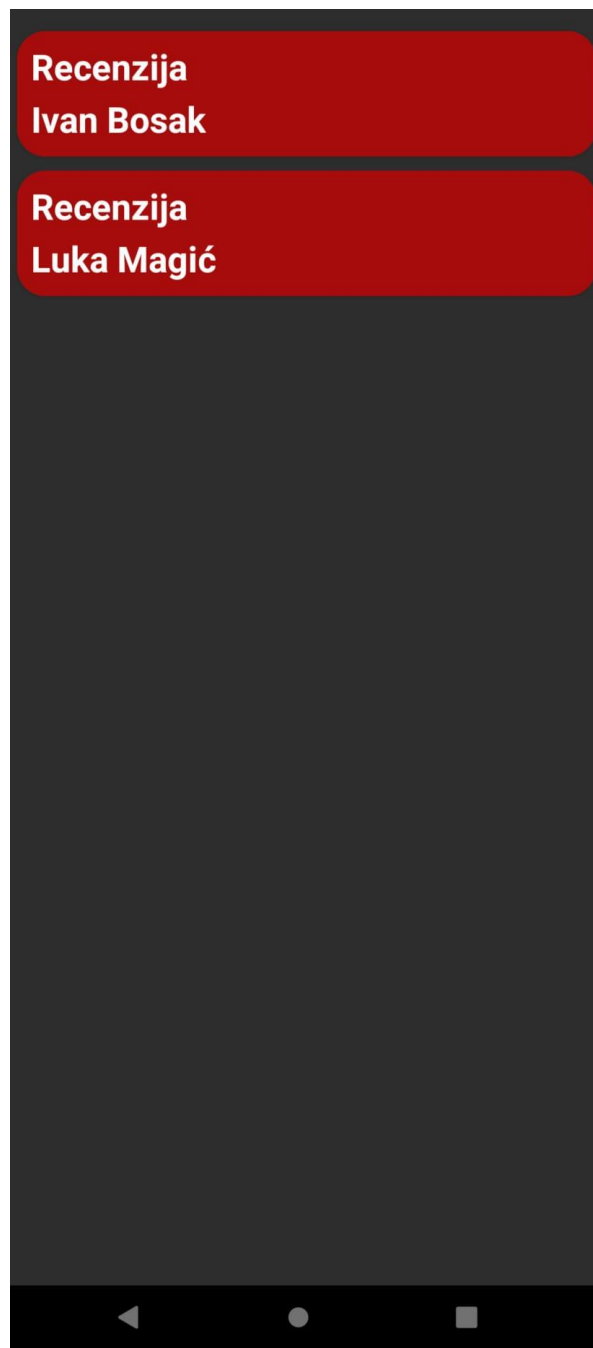
★ ★ ★ ★ ★

Pošalji recenziju

Izađi

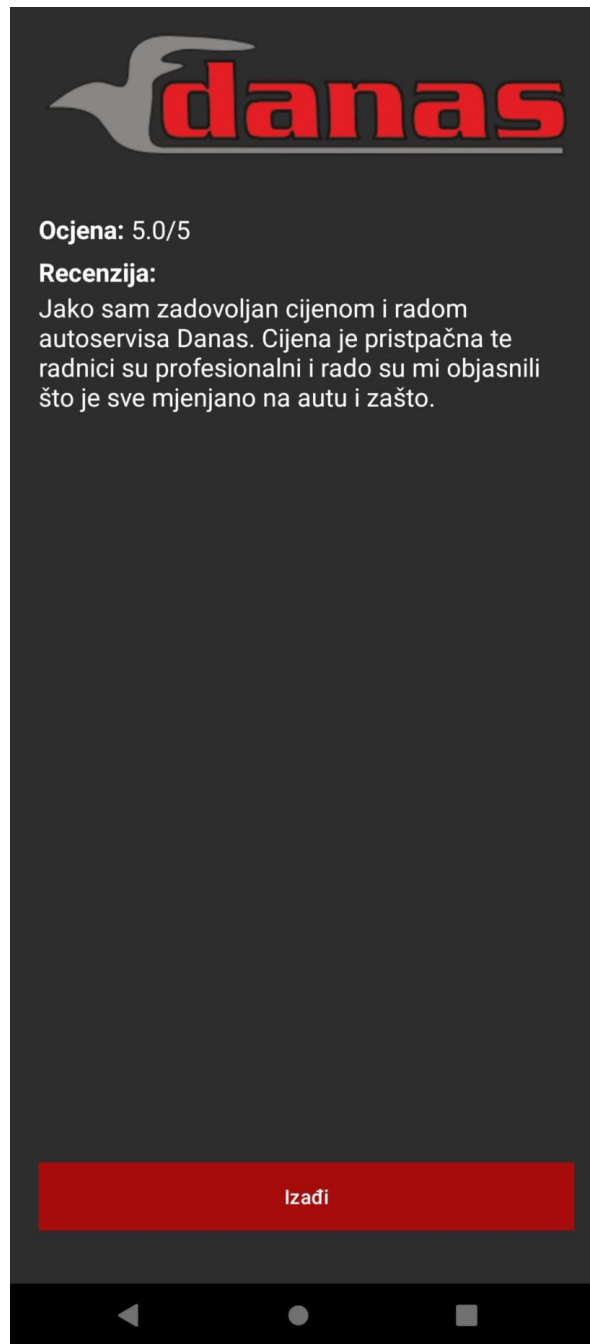
Izvor: Autor

Slika 36. Opcija Pogledajte recenzije u activity\_main.xml



Izvor: Autor

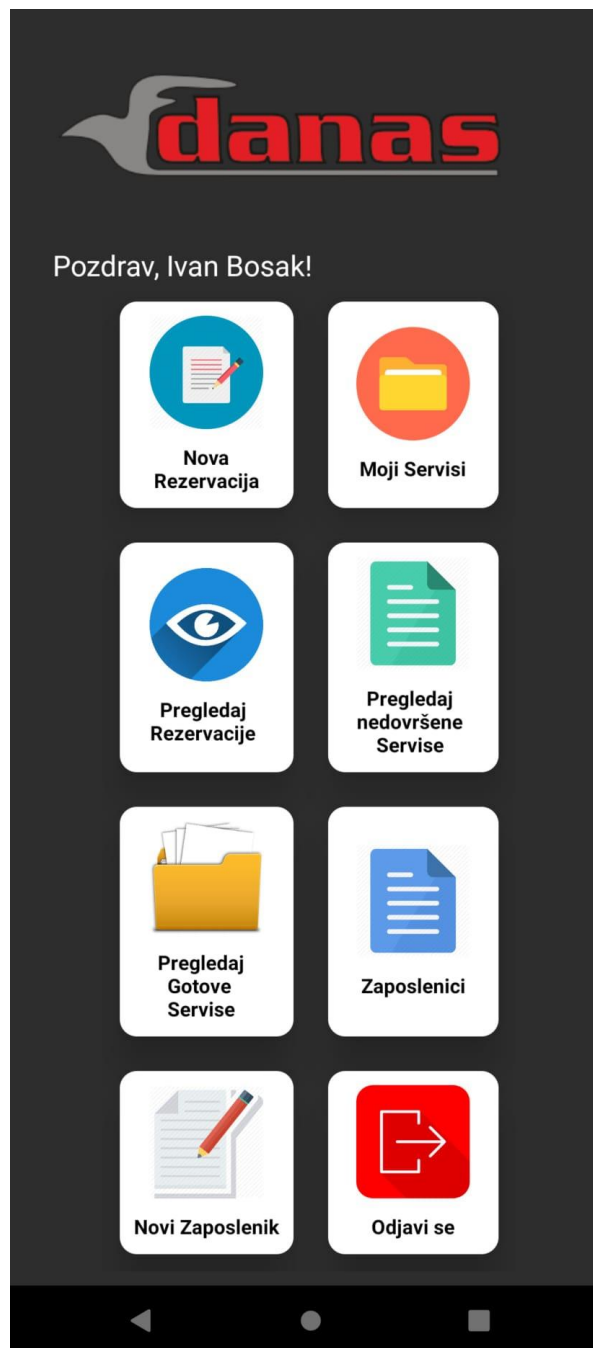
Slika 37. Recenzija jednog od korisnika



Izvor: Autor

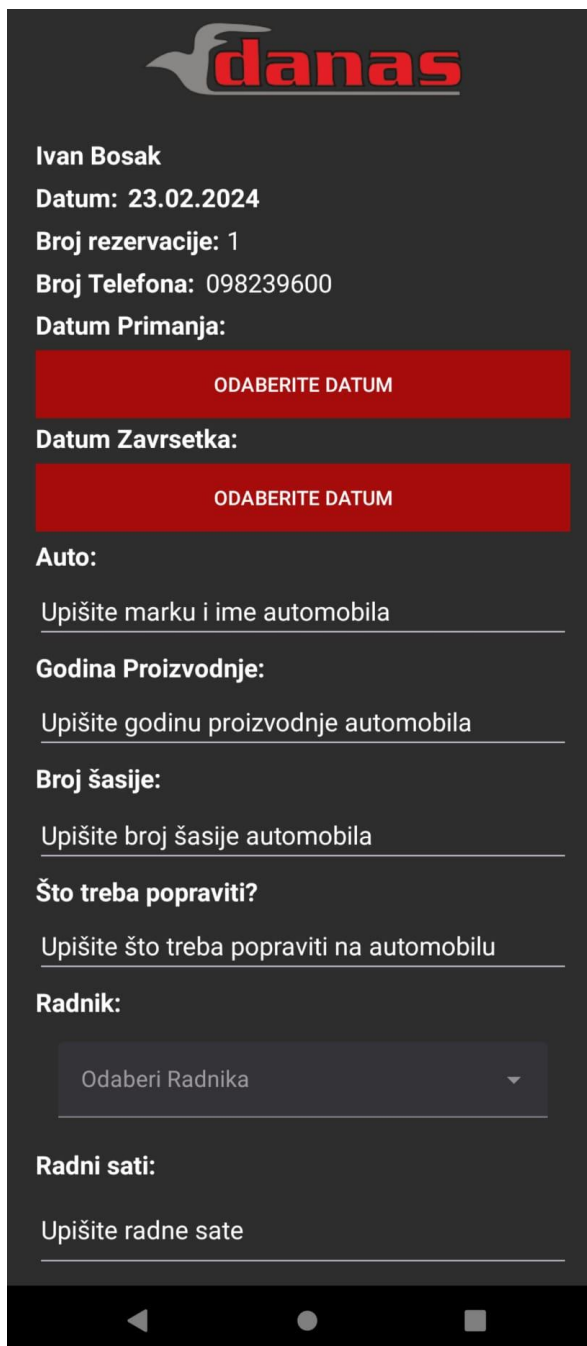
Administrator ima više opcija od običnog korisnika dok su neke stvari ipak iste. Na primjer, administrator može kreirati rezervaciju, ali on ima opciju upisa svih podataka kao što su datum završetka, radnik, radni sati i cijena. Administrator također može vidjeti sve rezervacije koje je on kreirao, ali ima i opciju pregledati sve rezervacije koje se nalaze u bazi podataka. Isto tako administrator može vidjeti sve rezervacije običnih korisnika te ima opciju dopuniti ih i potvrditi ili odbaciti te se onda ažurira u bazi podataka i korisnik dobiva informaciju ovisno o tome da li je rezervacija odbijena ili potvrđena. Ako je potvrđena korisnik dobiva e-mail gdje će pisati kada da dostavi auto i predviđeni datum kada će servis biti gotov, a ako je rezervacija odbijena korisnik dobiva email gdje će pisati da je odbijena te razlog zašto je odbijena. Kada je servis gotov administrator upisuje podatke kao što su cijena dijelova, cijena rada, radne sati i radnika koji je radio na servisu te potvrđuje servis. Servis se spremi u bazu podataka i korisnik dobiva e-mail da je njegov servis gotov te može doći podići auto. Administrator ima i opciju upisivanja novih zaposlenika, ažuriranje i brisanje postojećih zaposlenika.

Slika 38. Korisničko sučelje administratora



Izvor: Autor

Slika 39. Upis podataka za administratora (1.dio)



**danas**

Ivan Bosak  
Datum: 23.02.2024  
Broj rezervacije: 1  
Broj Telefona: 098239600  
Datum Primanja:  
ODABERITE DATUM  
Datum Zavrsetka:  
ODABERITE DATUM  
Auto:  
Upišite marku i ime automobila  
Godina Proizvodnje:  
Upišite godinu proizvodnje automobila  
Broj šasije:  
Upišite broj šasije automobila  
Što treba popraviti?  
Upišite što treba popraviti na automobilu  
Radnik:  
Odaberi Radnika  
Radni sati:  
Upišite radne sate

Izvor: Autor

Slika 40. Upis podataka za administratora (2.dio)

**Auto:**  
Upišite marku i ime automobila

**Godina Proizvodnje:**  
Upišite godinu proizvodnje automobila

**Broj šasije:**  
Upišite broj šasije automobila

**Što treba popraviti?**  
Upišite što treba popraviti na automobilu

**Radnik:**  
Odaberi Radnika

**Radni sati:**  
Upišite radne sate

**Cijena radova:**

**Cijena dijelova:**  
Upišite cijenu dijelova

**Ukupna cijena:**

Pošalji Rezervaciju

Izađi

Izvor: Autor

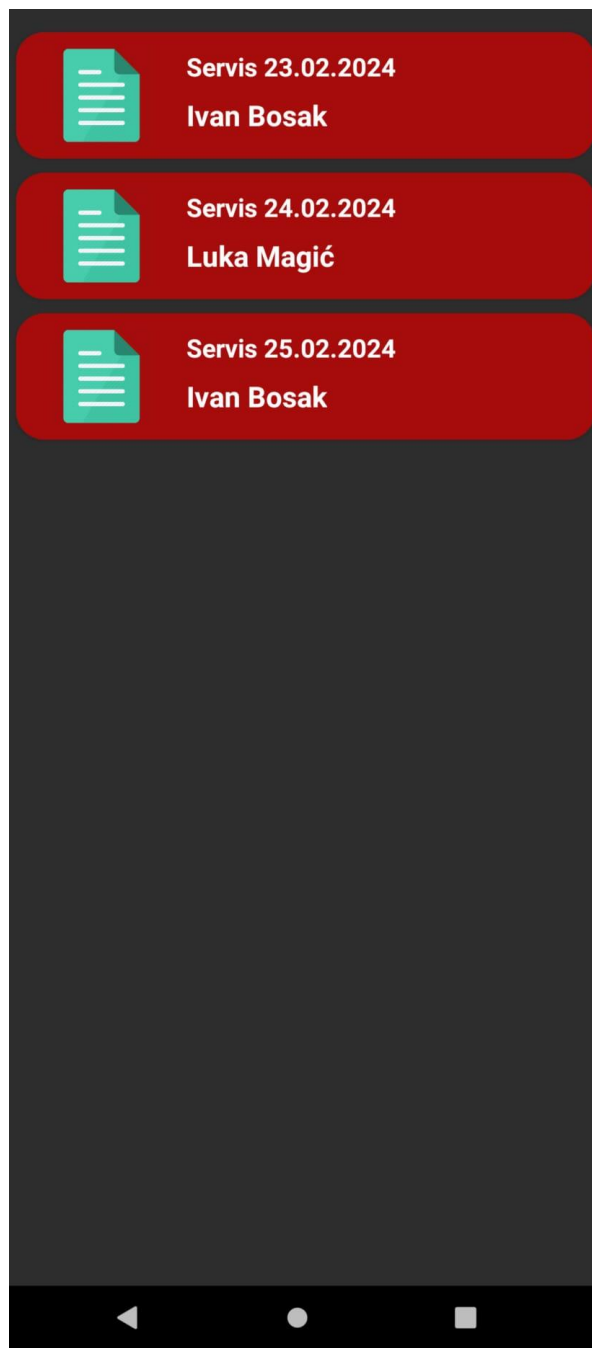
Slika 41. Opcija Pregledaj gotove servise ako u bazi podataka nema gotovih servisa



Izvor: Autor



Slika 42. Opcija Pregledaj gotove servise ako u bazi podataka postoje gotovi servisi



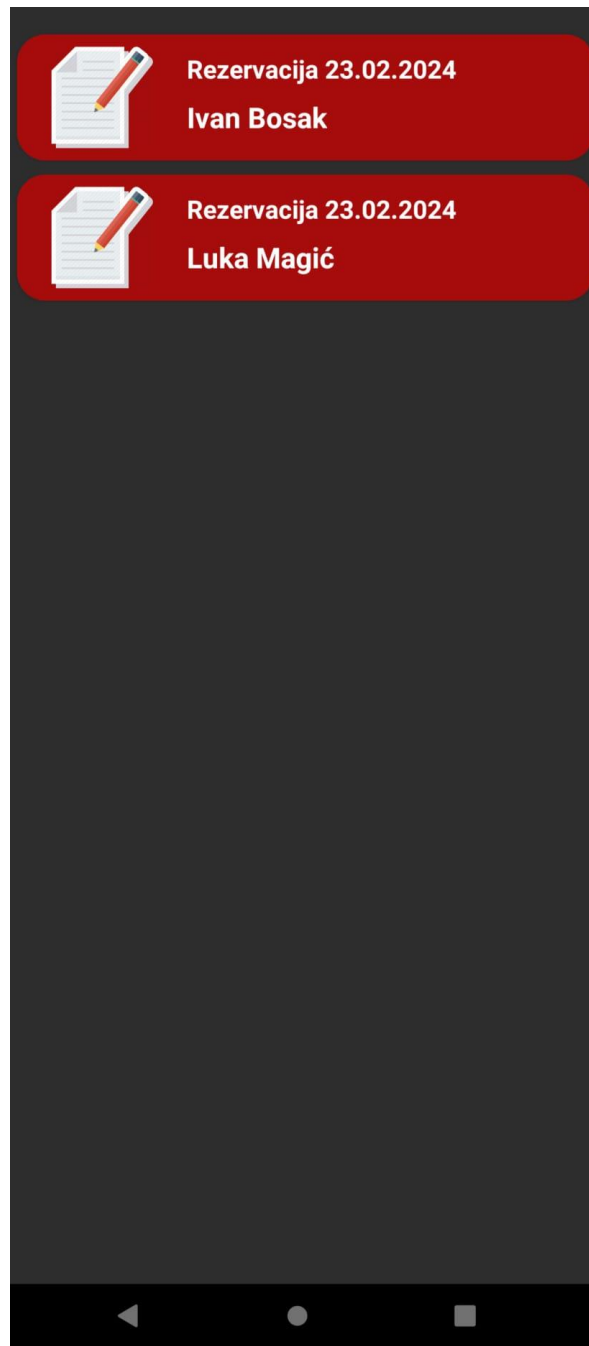
Izvor: Autor

Slika 43. Opcija Pregledaj rezervacije ako u bazi podataka nema rezervacija



Izvor: Autor

Slika 44. Opcija Pregledaj rezervacije ako u bazi podataka postoje rezervacije



Izvor: Autor

Slika 45. Upis podataka za potvrđivanje ili odbijanje rezervacije

The screenshot shows a mobile application interface with a dark grey background. At the top, there is a logo for 'danas' in red and white. Below the logo, the user's name 'Luka Magić' is displayed. The reservation details are listed: 'Broj rezervacije: 2', 'Dana: 23.02.2024', and 'Datum Primanja: 24.02.2024'. There are two red buttons with the text 'ODABERITE DATUM' (Select Date) for the reservation date and the end date. The car details are: 'Broj Telefona: 0958728985', 'Auto: Mazda Miata', 'Godina Proizvodnje: 2004', and 'Broj šasije: JM1NB3538Y0153757'. The section 'Što treba popraviti?' (What needs to be fixed?) contains the text 'Zamjena diskova i pločica' (Wheel and brake pad replacement). Below this is a dropdown menu for 'Razlog:' (Reason) with the text 'Odaberi Razlog Odbijanja' (Select Reason for Rejection). At the bottom, there are three buttons: 'Pošalji Rezervaciju' (Send Reservation), 'Odbaci Rezervaciju' (Reject Reservation), and 'Izađi' (Exit).

**danas**

Luka Magić

Broj rezervacije: 2  
Dana: 23.02.2024  
Datum Primanja: 24.02.2024

ODABERITE DATUM

Datum Zavrsetka:

ODABERITE DATUM

Broj Telefona: 0958728985  
Auto: Mazda Miata  
Godina Proizvodnje: 2004  
Broj šasije: JM1NB3538Y0153757

Što treba popraviti?  
Zamjena diskova i pločica

Razlog:

Odaberi Razlog Odbijanja

Pošalji Rezervaciju

Odbaci Rezervaciju

Izađi

Izvor: Autor

Slika 46. Upis podataka za nedovršeni servis (1.dio)

**danas**

**Ivan Bosak**  
Broj rezervacije: 1  
Dana: 23.02.2024  
Datum Primanja: 23.02.2024  
ODABERITE DATUM  
Datum Zavrsetka: 24.02.2024  
ODABERITE DATUM  
Broj Telefona: 098239600  
Auto: Volkswagen Passat  
Godina Proizvodnje: 2004  
Broj šasije: WWZZZ3BZ4P328425  
Što treba popraviti?  
Mali servis  
Radnik:  
Odaberi Radnika  
Radni sati:  
Upišite radne sate  
Cijena radova:  
Cijena dijelova:  
Upišite cijenu dijelova  
Ukupna cijena:

Izvor: Autor

Slika 47. Upis podataka za nedovršeni servis (2.dio)

Datum Zavjete: 24.02.2024

**ODABERITE DATUM**

Broj Telefona: 098239600

**Auto:** Volkswagen Passat

**Godina Proizvodnje:** 2004

**Broj šasije:** WVWZZZ3BZ4P328425

**Što treba popraviti?**

Mali servis

**Radnik:**

Odaberi Radnika

**Radni sati:**

Upišite radne sate

**Cijena radova:**

**Cijena dijelova:**

Upišite cijenu dijelova

**Ukupna cijena:**

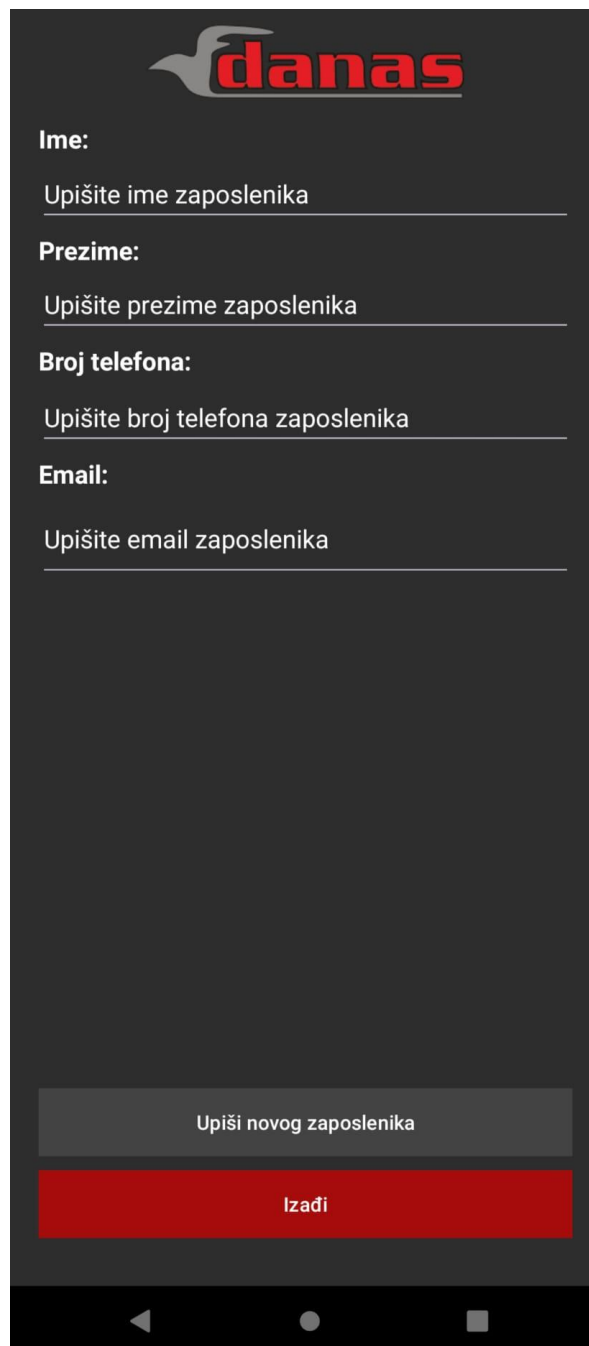
Pošalji Servis

Odbaci Servis

Izađi

Izvor: Autor

Slika 48. Opcija Novi zaposlenik gdje se može upisat novi zaposlenik



**danas**

**Ime:**  
Upišite ime zaposlenika

**Prezime:**  
Upišite prezime zaposlenika

**Broj telefona:**  
Upišite broj telefona zaposlenika

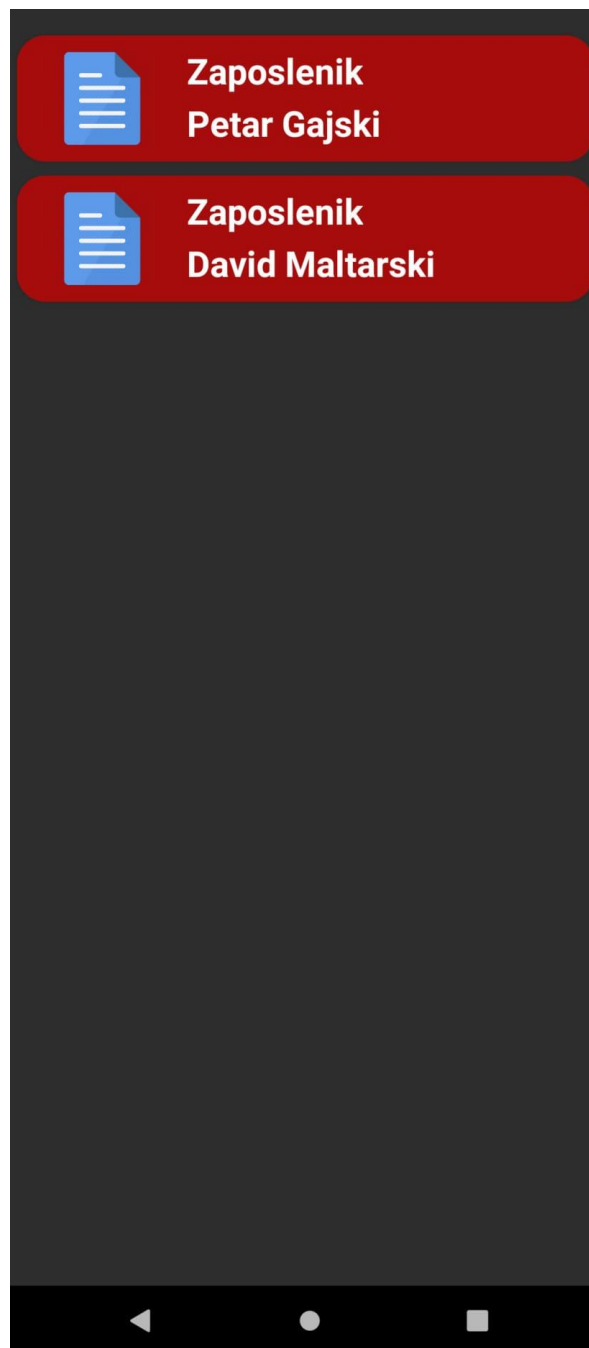
**Email:**  
Upišite email zaposlenika

Upiši novog zaposlenika

Izađi

Izvor: Autor

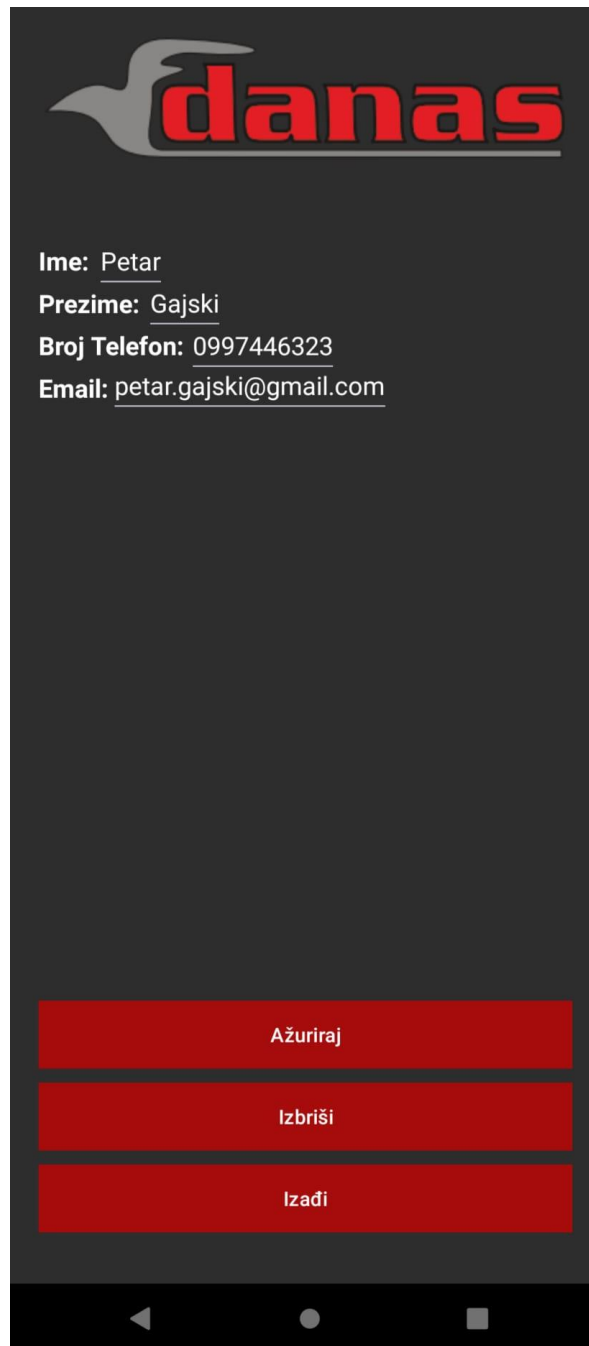
Slika 49. Opcija Zaposlenici kada u bazi podataka postoje zaposlenici



Izvor: Autor

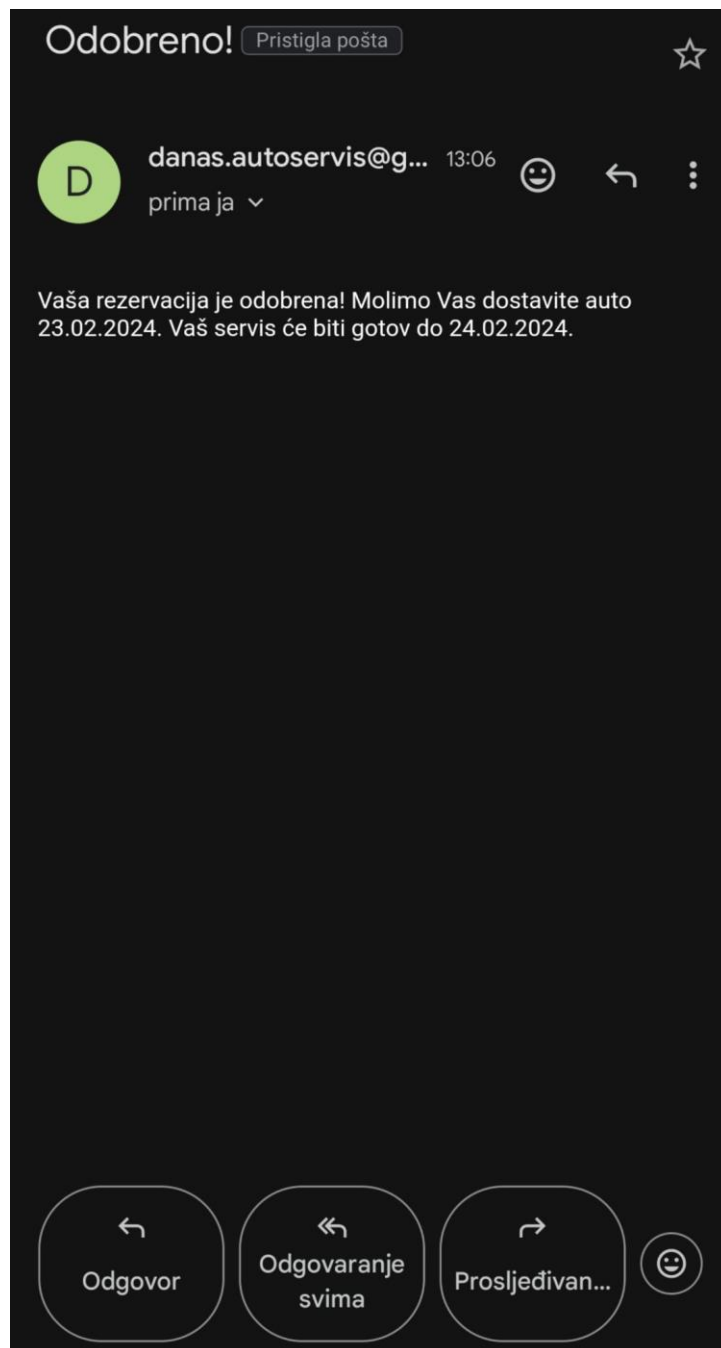


Slika 50. Detalji odabranog zaposlenika



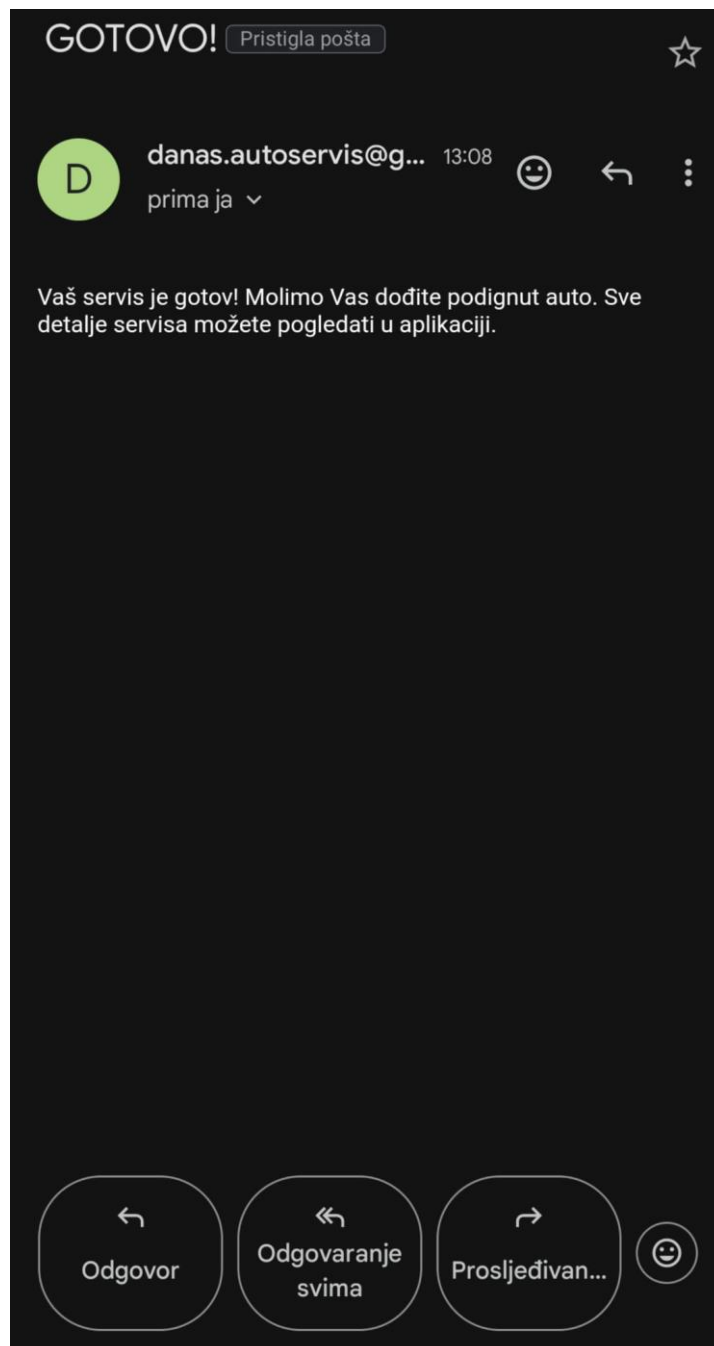
Izvor: Autor

Slika 51. E-mail poslan korisniku kada administrator potvrdi rezervaciju



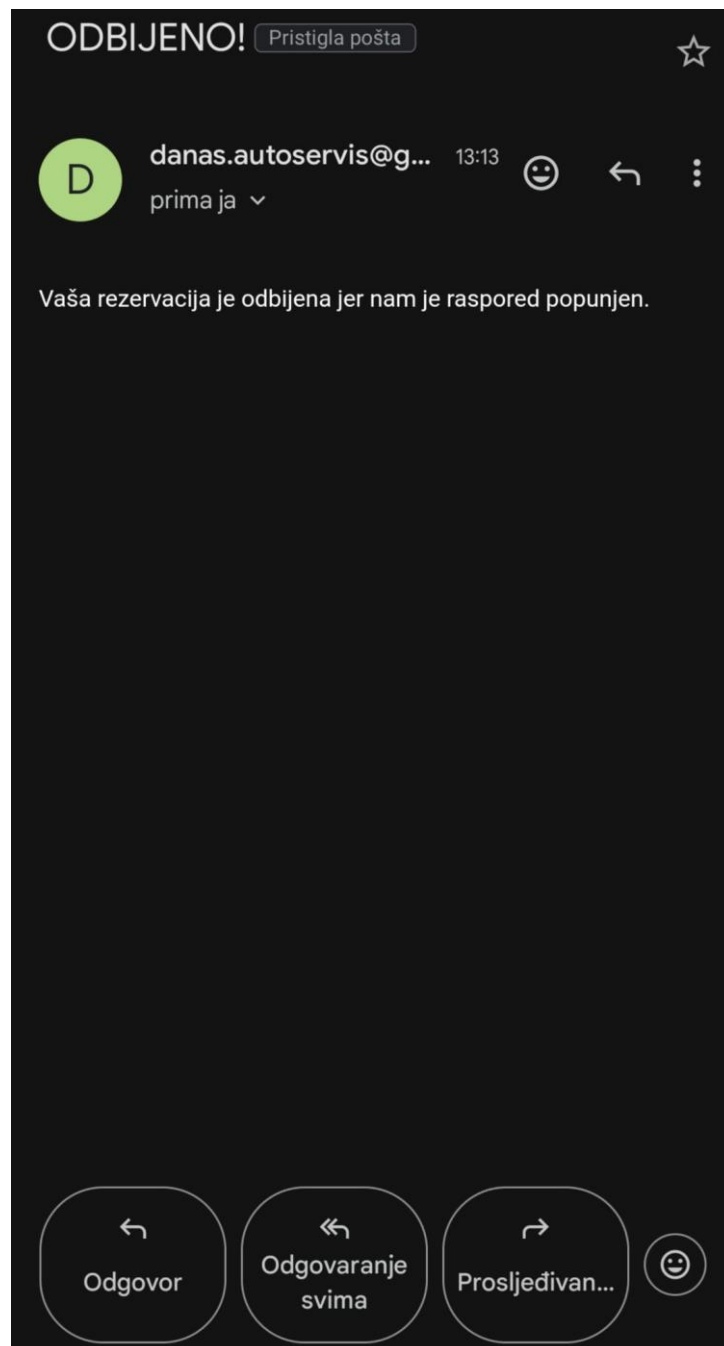
Izvor: Autor

Slika 52. E-mail poslan korisniku kada administrator pošalje servis



Izvor: Autor

Slika 53. E-mail poslan korisniku kada administrator odbije rezervaciju



Izvor: Autor

## 7. ZAKLJUČAK

Mobilna aplikacija izgrađena je uz pomoć Android studia i Retrofita sa kojima se veoma lagano koristiti. Programski jezik Java ima dosta jednostavnu sintaksu za koristiti te je uvelike pomogao u brzom i praktičnoj izradi mobilne aplikacije. API je izgrađen uz pomoć IntelliJ IDEA i Springboot programskog okruženja gdje su veliku pomoć ponudile njegove značajke Spring Data JPA i Hibernate za lak rad sa bazom podataka, a Spring Security je dao odlično rješenje za zaštitu API. Baza podataka je izgrađena u MySQL Workbenchu za jednostavno upravljanje sa podacima. Za testiranje aplikacije koristio se programski alat za slanje i primanje HTTP zahtjeva zvan Postman.

Aplikacija je praktična za koristiti te nudi veliku pomoć u učinkovitosti rada auto servisa. Isto tako i korisnicima koji ju koriste je odličan način za rezerviranje auta za servis u jako kratkom roku. Svi korisnici imaju pregled svojih rezervacije i obaviješteni su kada je rezervacija potvrđena, gotova ili odbijena. Isto tako administrator je obaviješten kada ima novu rezervaciju za pregledati.

**MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU**

**Bana Josipa Jelačića 22/a, Čakovec**

**IZJAVA O AUTORSTVU**

Završni/diplomski rad isključivo je autorsko djelo studenta te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, internetskih i drugih izvora) bez pravilnog citiranja. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom i nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, IVAN BOSAK (ime i

prezime studenta) pod punom moralnom, materijalnom i kaznenom odgovornošću,

izjavljujem da sam isključivi autor/ica završnog/diplomskog rada pod naslovom

IZRADA ANDROID MOBILNE APLIKACIJE ZA IZRADU I PRAĆENJE REZERVACIJA AUTO SERVISA

te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:

Ivan Bosak

(vlastoručni potpis)

## 8. POPIS LITERATURE

1. Springboot  
Dostupno na: [https://www.tutorialspoint.com/spring\\_boot/spring\\_boot\\_introduction.htm](https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm)  
(19.10.2023)
2. Springboot  
Dospuno na: <https://spring.io/projects/spring-boot> (19.02.2024)
3. Spring Data JPA  
Dostupno na: <https://spring.io/projects/spring-data-jpa> (19.02.2024)
4. Spring Security  
Dostupno na: <https://spring.io/projects/spring-security> (22.02.2024)
5. Hibernate  
Dostupno na: <https://www.tutorialspoint.com/hibernate/index.htm> (22.02.2024)
6. Java  
Dostupno na: <https://www.geeksforgeeks.org/introduction-to-java/> (22.02.2024)
7. IntelliJ IDEA  
Dostupno na: <https://www.jetbrains.com/idea/features/> (22.02.2024)
8. IntelliJ IDEA  
Dostupno na: <https://www.techopedia.com/definition/7755/intellij-idea>  
(19.10.2023)
9. Android Studio  
Dostupno na: <https://developer.android.com/studio/intro> (23.02.2024)
10. Retrofit  
Dostupno na: <https://www.topcoder.com/thrive/articles/retrofit-library-in-android> (23.02.2024)
11. SQL  
Dostupno na: <https://www.techtarget.com/searchdatamanagement/definition/SQL>  
(24.02.2024)
12. MySQL Workbench  
Dostupno na: <https://www.mysql.com/products/workbench/> (24.02.2024)

## 9. POPIS SLIKA

Slika 1. IntelliJ IDEA .....	4
Slika 2. Retrofit .....	6
Slika 3. Primjer Retrofita u kodu.....	7
Slika 4. Baza podataka u MySQL Workbench-u .....	8
Slika 5. ER model baze podataka .....	9
Slika 6. Kreiranje nove konekcije u MySQL Workbench-u .....	10
Slika 7. Opcija „create a new schema in the connected server“ za izradu nove sheme .	10
Slika 8. Nakon pritiska opcije „create a new schema in the connected server“ .....	11
Slika 9. Kreirana nova shema .....	11
Slika 10. Web alat Spring initializr .....	12
Slika 11. Kod za definiranje priključka i konekciju na bazu podataka .....	13
Slika 12. Kod za definiranje email servisa .....	13
Slika 13. Metode za slanje lozinke korisniku i za enkripciju i dekripciju lozinke .....	14
Slika 14. Get, Post i Put metode za rezervacije .....	15
Slika 15. Metoda za slanje lozinke korisniku putem e-maila .....	15
Slika 16. Kreiranje novog projekta u Android Studiu .....	16
Slika 17. Odabir platforme za mobitel.....	17
Slika 18. Konfiguracija novog projekta.....	17
Slika 19. Ovisnost Retrofit .....	18
Slika 20. Linija koda za definiranje pristupa internetu u mapi Android Manifest.xml ..	18
Slika 21. Primjer koda u klasi LoginActivity .....	18
Slika 22. Primjer koda u activity_main.xml .....	19
Slika 23. Forma za prijavu.....	20
Slika 24. Forma za registraciju .....	21
Slika 25. Forma za slanje lozinke korisniku .....	22
Slika 26. Korisničko sučelje za običnog korisnika.....	23
Slika 27. Upis podataka za običnog korisnika.....	24
Slika 28. E-mail poslan administratoru kada korisnik pošalje novu rezervaciju.....	25
Slika 29. Opcija Moji servisi ako korisnik nema niti jedan servis .....	26
Slika 30. Opcija Moji servisi ako korisnik ima gotove servise .....	27
Slika 31. Detalji odabranog gotovog servisa .....	28
Slika 32. Opcija Moje rezervacije ako korisnik nema niti jednu rezervaciju .....	29
Slika 33. Opcija Moje rezervacije kada korisnik ima rezervacije .....	30
Slika 34. Detalji odabrane rezervacije .....	31
Slika 35. Opcija Nova recenzija .....	32
Slika 36. Opcija Pogledajte recenzije u activity_main.xml.....	33
Slika 37. Recenzija jednog od korisnika .....	34
Slika 38. Korisničko sučelje administratora .....	36
Slika 39. Upis podataka za administratora (1.dio).....	37
Slika 40. Upis podataka za administratora (2.dio).....	38
Slika 41. Opcija Pregledaj gotove servise ako u bazi podataka nema gotovih servisa ..	39
Slika 42. Opcija Pregledaj gotove servise ako u bazi podataka postoje gotovi servisi ..	40
Slika 43. Opcija Pregledaj rezervacije ako u bazi podataka nema rezervacija.....	41
Slika 44. Opcija Pregledaj rezervacije ako u bazi podataka postoje rezervacije.....	42



Slika 45. Upis podataka za potvrđivanje ili odbijanje rezervacije .....	43
Slika 46. Upis podataka za nedovršeni servis (1.dio).....	44
Slika 47. Upis podataka za nedovršeni servis (2.dio).....	45
Slika 48. Opcija Novi zaposlenik gdje se može upisat novi zaposlenik.....	46
Slika 49. Opcija Zaposlenici kada u bazi podataka postoje zaposlenici .....	47
Slika 50. Detalji odabranog zaposlenika .....	48
Slika 51. E-mail poslan korisniku kada administrator potvrdi rezervaciju .....	49
Slika 52. E-mail poslan korisniku kada administrator pošalje servis .....	50
Slika 53. E-mail poslan korisniku kada administrator odbije rezervaciju .....	51