

# RAZVOJ SUSTAVA ZA REZERVACIJU SPORTSKIH TERENA, NA PRIMJERU TENISA

---

Županović Mustapić, Ivana

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:225:197094>

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

Download date / Datum preuzimanja: **2024-11-09**



Repository / Repozitorij:

[Algebra University - Repository of Algebra University](#)



**VISOKO UČILIŠTE ALGEBRA**

ZAVRŠNI RAD

**RAZVOJ SUSTAVA ZA REZERVACIJU  
SPORTSKIH TERENA, NA PRIMJERU  
TENISA**

Ivana Županović Mustapić

Zagreb, veljača 2023.



*„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristila sam tuđe materijale navedene u popisu literature, ali nisam kopirala niti jedan njihov dio, osim citata za koje sam navela autora i izvor, te ih jasno označila znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.*

*U Zagrebu, 28. 2. 2023.*

# Predgovor

Zahvaljujem se mentoru Danielu Beleu na iskazanom povjerenju, motivaciji i prenesenom znanju i iskustvu.

Zahvaljujem se svome suprugu Petru Rafaelu na pruženoj ljubavi, podršci, motivaciji i strpljenju tijekom cijelog školovanja. Također, zahvaljujem se svojim roditeljima i sestri koji su mi bili velika potpora za vrijeme školovanja. Veliku zahvalnost iskazujem i svojoj pokojnoj baki Cviti koja mi je bila velika podrška i inspiracija tijekom cijelog života.

Temeljem članka 8. Pravilnika o završnom radu i završnom ispitu na preddiplomskom studiju Visokog učilišta Algebra sačinjena je ova

## Potvrda o dodjeli završnog rada

kojom se potvrđuje da studentica Ivana Županović Mustapić, JMBAG 0177053921, OIB 35619151976 u šk. godini 2021./2022., studij: Primjenjeno računarstvo - Preddiplomski studij, smjer: Programsko inženjerstvo, od strane povjerenstva za provedbu završnog ispita, dana 23.02.2022. godine, ima odobrenu izradu završnog rada

s temom: **Razvoj sustava za rezervaciju sportskih terena, na primjeru tenisa**

i sažetkom rada: Studentica će svojim radom prikazati napredno korištenje inovativnih tehnologija u kontekstu razvoja sustava za rezervaciju sportskih terena. Rezultat ovog završnog rada je sustav koji omogućuje interaktivno i napredno korištenje informacijskog sustava u svrhu organizacije rezervacije teniskih terena.

Mentor je: Daniel Bele.

Odobrenjem završnog rada studentici je omogućen upis kolegija "Izrada završnog projekta/Praksa" te je sukladno članku 8. Pravilnika o završnom radu i završnom ispitu dužan najkasnije do početka nastave ljetnog semestra u sljedećoj školskoj godini, uspješno obraniti završni rad uspješnim polaganjem završnog ispita.

U protivnom studentica može zatražiti novog mentora/icu i temu te ponovo upisati kolegij "Izrada završnog projekta/Praksa" budući da rad koji nije predan i obranjen na završnom ispitu u roku određenom Pravilnikom završnom radu i završnom ispitu prestaje vrijediti. Izrada novog završnog rada se izvodi sukladno rokovima određenima za školsku godinu u kojoj je studentici određen novi mentor/ica i dodijeljen novi završni rad.

Potpis studentice:

Potpis mentora:

Potpis predsjednika  
povjerenstva:

Ova potvrda izdaje se u 4 (četiri) primjerka od kojih 3 (tri) idu kao prilog završnom radu.

## Sažetak

Tema ovoga završnog rada je izrada programskog rješenja za rezervaciju sportskih terena na primjeru tenisa. Mobilna je aplikacija namijenjena korisnicima teniskih terena, a putem nje mogu rezervirati teniske terene. Također, mobilna je aplikacija namijenjena i zaposlenicima sportskih centara koji su zaduženi za rezervacije teniskih terena, putem koje mogu upravljati tim rezervacijama. Prototip aplikacije je izrađen u Android Studio razvojnom okruženju sa svim pripadajućim funkcionalnostima. Mobilna aplikacija za obične korisnike sastoji se od šest ekrana te ekrana za prijavu i registraciju. Mobilna aplikacija za administratore sportskih centara sastoji se od četiri ekrana i ekrana za prijavu. U završnom radu opisana je implementacija rješenja, njegove funkcionalnosti te korištene tehnologije i alati.

**Ključne riječi:** mobilna aplikacija, Android, baza podataka, teniski teren, rezervacija, prototip, korisnici, zaposlenici

## Summary

The topic of this final work is the development of a software solution for booking sports fields on the example of tennis. The mobile application is intended for users of tennis courts through which they can book tennis courts. Also, the mobile application is intended for employees of sports centers who are in charge of booking tennis courts, through which they can manage these reservations. The prototype of the application was created in the Android Studio development environment with all the associated functionalities. The mobile application for ordinary users consists of six screens and screens for login and registration. The mobile application for sports center administrators consists of four screens and a login screen. The final paper describes the implementation of the solution, its functionality and the technologies and tools used.

**Keywords:** mobile application, Android, database, tennis court, reservation, prototype, users, employees

# Sadržaj

1. Uvod .....	3
2. Problematika i prijedlog rješenja za rezervaciju teniskog terena .....	4
2.1. Opis problematike .....	4
2.2. Prijedlog rješenja .....	4
2.3. Provođenje i rezultati ankete o potrebitosti mobilne aplikacije za rezervaciju teniskih termina u Splitu.....	5
2.3.1. Provođenje ankete.....	5
2.3.2. Istraživačka pitanja .....	6
2.3.3. Rezultati ankete .....	6
2.4. SWOT analiza.....	9
2.4.1. Snage .....	10
2.4.2. Prilike .....	10
2.4.3. Slabosti .....	11
2.4.4. Prijetnje.....	11
3. Implementacija softverskog rješenja .....	13
3.1. Baza podataka.....	13
3.1.1. Tablice .....	17
3.1.2. ER Dijagram .....	18
3.2. Web API .....	18
3.3. Mobilna aplikacija .....	21
3.3.1. Arhitektura sustava .....	21
3.3.2. Opis korisnika.....	22
4. Funkcionalnosti implementirane aplikacije.....	23



4.1.	Prijava korisnika .....	23
4.2.	Registracija korisnika .....	24
4.3.	Administrator – Pregled teniskih terena .....	25
4.4.	Administrator – Pregled i otkazivanje rezervacija .....	26
4.5.	Pregled sportskog centra.....	27
4.5.1.	Detalji sportskog centra .....	28
4.5.2.	Rezervacija terena.....	29
4.5.3.	Pregled i otkazivanje rezervacija .....	30
4.5.4.	Podsjetnik o nadolazećem terminu .....	31
4.6.	Pregled profila .....	31
4.6.1.	Podsjetnik o nadolazećem terminu .....	32
4.6.2.	Uređivanje profila.....	32
5.	Korištene tehnologije za implementaciju programskog rješenja.....	33
5.1.	Android Studio razvojno okruženje.....	33
5.2.	Tehnologije i programski jezici.....	34
5.2.1.	Mocky .....	34
5.2.2.	Json2Kotlin.....	35
5.2.3.	SQLite.....	36
5.2.4.	Retrofit.....	37
5.2.5.	Kotlin .....	39
	Zaključak .....	40
	Popis kratica .....	42
	Popis slika.....	43
	Popis kôdova .....	45
	Literatura .....	46

# 1. Uvod

Napredak tehnologije doveo je do širenja digitalizacije u poslovanju. Pružatelji usluga sve se više okreću aplikacijama koje nude svojim korisnicima jednostavan i brz pregled usluga. Jedno od rješenja, koje se sve više pojavljuje u svijetu, pa tako i kod nas, mobilne su aplikacije. U svijetu, gdje većina ljudi posjeduje mobilne uređaje, preko mobilnih aplikacija korisnicima se nudi pregled i korištenje raznih vrsta usluga, kao što su bankarstvo, kupovina, slušanje glazbe, upravljanje kućanskim uređajima i slično. Također, sve je veća upotreba mobilnih aplikacija za rezervaciju usluga, kao što su kozmetički tretmani, ulaznice za događanja, sportski sadržaji i slično.

Unatoč brojnim internetskim i mobilnim aplikacijama za rezervacije, uočeno je nepostojanje mobilne aplikacije koja bi na jednome mjestu obuhvatila sve teniske terene s mogućnošću njihove rezervacije. Naime, rješenja koja trenutno postoje nisu uspjela riješiti postojeće probleme s kojima se korisnici teniskih terena susreću. Rješenja, koja se nude korisnicima odnose se na samo određene sportske centre ili je potrebno članstvo u nekom sportskom centru kako bi se mogla koristiti.

Cilj je ovog završnog rada izrada rješenja koje bi pojednostavnilo proces rezervacije termina tenisa za sve sudionike: zaposlenike zadužene za rezervaciju termina i igrače.

Kroz rad je obrađen problem koji se želi riješiti, opis mogućeg softverskog rješenja, njegove funkcionalnosti i opis najvažnijih tehnologija koje su se koristile pri procesu njegove izrade. Rad se sastoji od sedam cjelina. Prva je cjelina uvod u kojem se opisuje tema i cilj završnog rada. U drugoj cjelini opisani su neki od glavnih i najčešćih problema s kojima se svakodnevno susreću korisnici teniskih terena na području Splita. Zatim se daje prijedlog rješenja kojima bi se ti problemi otklonili i umanjili. Nakon toga, opisana je anketa koja se provela kao dio istraživanja problema za završni rad i njeni rezultati. Treća cjelina opisuje arhitekturu i implementaciju softverskog rješenja. U četvrtoj cjelini opisane su same funkcionalnosti implementirane aplikacije s prikazima najvažnijih ekrana mobilne aplikacije. Peta cjelina opisuje najvažnije tehnologije i alate koji su korišteni za implementaciju ovog programskog rješenja. Završna je cjelina zaključak završnog rada u kojem se analizira problem, rješenje te dobre i loše strane mobilne aplikacije koje smo dobili SWOT analizom u drugom poglavlju.

## **2. Problematika i prijedlog rješenja za rezervaciju teniskog terena**

U ovom poglavlju opisuje se problematika rezervacije teniskih terena, prijedlog kako bi se ona mogla riješiti, anketa koja je provedena, rezultati ankete te SWOT analiza ove mobilne aplikacije.

### **2.1. Opis problematike**

Ideja za razvojem mobilne aplikacije za rezervaciju sportskih sadržaja na primjeru tenisa proizlazi iz potrebe za sučeljem koje na brz i jednostavan način omogućava rezervaciju sportskih terena. Trenutno u gradu Splitu ne postoji mobilna aplikacija za rezervaciju terena u teniskim klubovima. Jedini način na koji igrači mogu rezervirati termin je kontaktirati klub putem telefona. U Hrvatskoj postoje neka rješenja, primjerice Sportbooking.hr, međutim, neki od nedostataka su npr. što je za samu registraciju i korištenje sustava potrebno kontaktirati voditelja sportskog centra. Proces rezervacije termina u većini gradova Hrvatske, pa tako i u Splitu, veoma je spor, zastario i zamoran. Naime, igrači se moraju prvo međusobno dogovoriti koji dan i koje vrijeme bi rezervirali, a potom jedan od njih mora kontaktirati teniski klub u kojem žele igrati. U većini slučajeva se događa da taj dan ili u to vrijeme nema slobodnih termina u klubu, gdje nastaje problem. Dakle, igrač tada treba ponovno nazvati svog suigrača i pokušati dogovoriti drugo vrijeme (nakon čega ponovno mora kontaktirati klub) ili dalje zvati druge klubove i tako ponavljati postupak dok ne pronađe odgovarajući termin. Također, osobe koje rade na samoj rezervaciji termina imaju uz to i druge poslove koji često zahtijevaju boravak izvan ureda. Time se učestalo događa da se igračima nitko ne javi na telefon. Sve to čini jednu radnju, kao što je rezervacija termina, veoma sporom i kompliciranom za same igrače.

### **2.2. Prijedlog rješenja**

Rješenje je sustav koji bi se sastojao od mobilne aplikacije čija je primarna funkcionalnost rezervacija termina teniskih terena, baze podataka te web-aplikacijskoga programskog

sučelja (engl. *application programming interface*, skraćeno API) koji će omogućiti komunikaciju komponenti sustava.

Sustav prepoznaje dvije vrste korisnika: zaposlenike, koji su zaduženi za rezervaciju termina, i igrače. Zaposlenicima je omogućeno praćenje rezervacija i obavještanje igrača ako rezervirani teren iz nekog razloga nije dostupan (npr. zbog vremenskih uvjeta). Igrači imaju mogućnost rezervacije i otkazivanja termina u odabranom klubu, pregleda aktivnih rezervacija, korištenja podsjetnika na dan rezervacije, uređivanja osobnog profila i dobivanja obavijesti o terminu. Prednost ovog sustava je mobilna aplikacija koja će olakšati i ubrzati rezervaciju termina za igrače, ali i za zaposlenike koji će imati više vremena za ostale poslove.

## **2.3. Provođenje i rezultati ankete o potrebitosti mobilne aplikacije za rezervaciju teniskih termina u Splitu**

U ovom poglavlju opisana je anketa koja se koristila za istraživanje problema ovoga završnog rada, odnosno za dobivanje uvida u načine rezervacije korisnika teniskih terena. Također, poglavlje sadrži pitanja koja su se postavljala ispitanicima, rezultate istraživanja i zaključak dobiven iz tih rezultata.

### **2.3.1. Provođenje ankete**

U svrhu izrade završnog rada provedeno je istraživanje kojim se željelo ustanoviti postoji li potreba za uvođenjem mobilne aplikacije za rezervaciju teniskih terena na području Splita. Za ovo istraživanje korištena metoda istraživanja je internet anketa. Anketa je izrađena pomoću Google formi.

Istraživanje je provedeno na 30 ispitanika, a ciljana skupina ispitanika bili su članovi splitske tenis lige „Glavno da se igra“ i ljudi koji povremeno igraju tenis na području Splita. Ispitanicima je naglašeno kako je anketa anonimna te da se provodi u svrhu izrade završnog rada. Kroz nekoliko se pitanja u anketi željelo dobiti informaciju koliko ispitanici često koriste i na koji način rezerviraju teniske terene.

## 2.3.2. Istraživačka pitanja

U anketi je postavljeno šest istraživačkih pitanja:

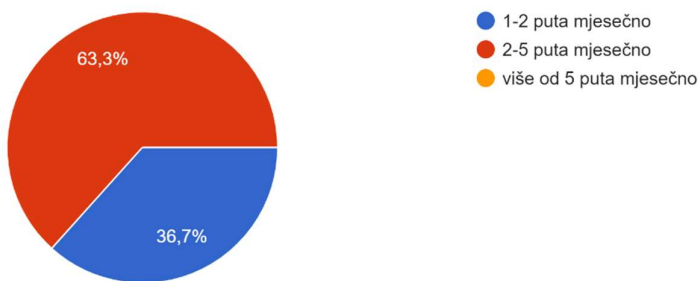
- Koliko često koristite teniske terene na području Splita?
- Na koji način rezervirate teniski teren?
- Ako je vaš prethodni odgovor bio „putem telefona“, koliko vam je vremena za to otprilike potrebno?
- Kada kontaktirate sportski centar, događa li se da vam se nitko ne javlja na telefon? Ako da, koliko često?
- Kako biste dogovorili termin koji odgovara vama i vašem/oj partneru/ici, trebate li zbog dogovora s partnerom/icom više puta kontaktirati sportski centar kako biste rezervirali teren? Ako da, koliko često?
- Smatrate li da bi mobilna aplikacije putem koje biste mogli rezervirati teniski teren ubrzala i olakšala proces rezervacije?

## 2.3.3. Rezultati ankete

Rezultati istraživanja pokazali su da najveći broj ispitanika (63,3%) koristi terene na području Splita dva do pet puta mjesečno. Njih 36,7% koristi terene jedan do dva puta mjesečno, dok nitko od ispitanika (0%) ne koristi terene više od pet puta mjesečno.

Koliko često koristite teniske terene na području Splita?

30 odgovora

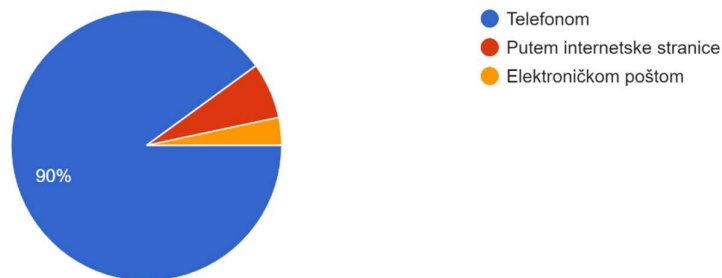


Slika 2.1 Grafički prikaz učestalosti korištenja teniskih terena

Istraživanje je pokazalo da najveći broj ispitanika (90%) rezervira teniske terene putem telefona. Njih 6,7% rezervaciju obavlja putem internetske stranice, dok 3,3% ispitanika rezervira putem elektroničke pošte.

Na koji način rezervirate teniski teren?

30 odgovora

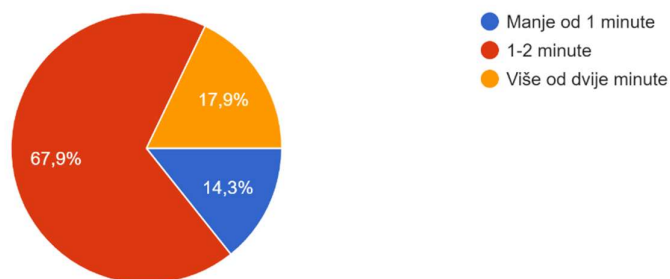


Slika 2.2 Grafički prikaz rezultata načina rezerviranja teniskih terena

Prema istraživanju, ispitanicima je najčešće potrebna jedna do dvije minute za rezervaciju teniskih terena putem telefona (67,9%). Više od dvije minute za rezervaciju putem telefona potrebno je za 17,9% ispitanika. Za njih 14,3% potrebno je manje od minute.

Ako je vaš prethodni odgovor bio „putem telefona“, koliko vam je vremena za to otprilike potrebno?

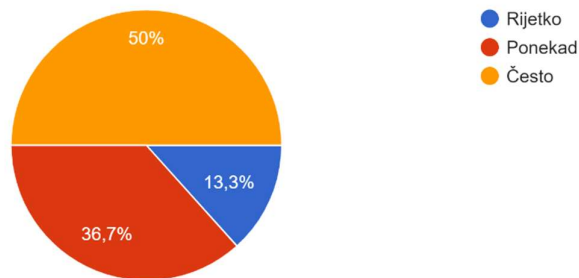
28 odgovora



Slika 2.3 Grafički prikaz rezultata okvirnog trajanja procesa rezervacije

Prilikom kontaktiranja sportskog centra, najveći broj ispitanika (50%) izjasnio se da im se često događa da im se nitko ne javlja na telefon. 36,7% ispitanika izjasnilo se da im se to ponekad događa, dok 13,3% ispitanika izjasnilo se da im se to rijetko događa.

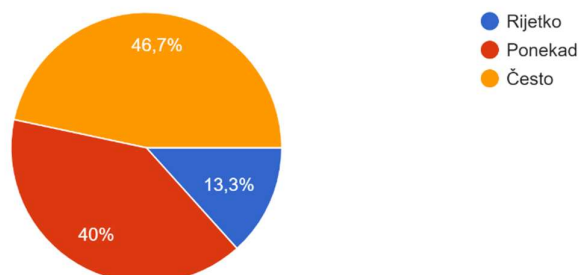
Kada kontaktirate sportski centar, događa li se da vam se nitko ne javlja na telefon? Ako da, koliko često?  
30 odgovora



Slika 2.4 Grafički prikaz rezultata učestalosti nedostupnosti sportskog centra putem telefona

Kada klijenti rezerviraju teniski teren putem telefona, može se dogoditi da moraju više puta kontaktirati sportski centar kako bi mogli provjeriti koji termini su slobodni te se dogovoriti s partnerom/icom koji termin im odgovara. Najveći broj ispitanika (46,7%) izjavio je da im se to često događa. Malo manje ispitanika, tj. njih 40% izjasnilo se da im se to ponekad događa. Najmanji broj ispitanika (13,3%) izjasnio se da im se to rijetko događa.

Kako biste dogovorili termin koji odgovara vama i vašem/oj partneru/ici, trebate li zbog dogovora s partnerom/icom više puta kontaktirati sportski centar...ako biste rezervirali teren? Ako da, koliko često?  
30 odgovora

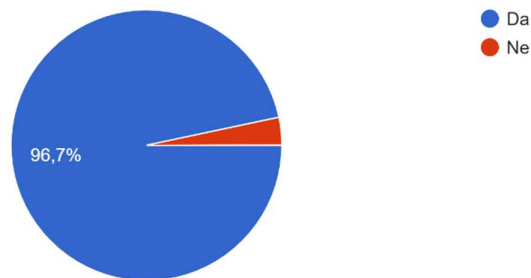


Slika 2.5 Grafički prikaz rezultata učestalosti uspjeha rezervacije terena u jednom pozivu

Istraživanje je pokazalo da čak 96,7% ispitanika smatra da bi mobilna aplikacija bila korisna kako bi se ubrzao i olakšao proces rezervacije, dok 3,3% ispitanika smatra kako mobilna aplikacija ne bi bila korisna.

Smatrate li da bi mobilna aplikacije putem koje biste mogli rezervirati teniski teren ubrzala i olakšala proces rezervacije?

30 odgovora



Slika 2.6 Grafički prikaz rezultata o potrebitosti mobilne aplikacije za rezervaciju teniskih terena  
Rezultati ankete pokazali su da najveći broj korisnika rezervira teniske terene putem telefona, što je često sporo jer se više puta mora kontaktirati sportski centar. Također, gotovo svi ispitanici smatraju da bi mobilna aplikacija bila korisno rješenje ovog problema.

## 2.4. SWOT analiza

SWOT analiza je tehnika strateškog planiranja koja pruža pogled na slabosti i snage organizacije, proizvoda ili slično. Riječ SWOT kratica je za četiri kategorije: snage (engl. *strengths*), slabosti (engl. *weaknesses*), prilike (engl. *opportunities*) i prijetnje (engl. *threats*). SWOT analiza ispituje vanjske i unutarnje čimbenike koji mogu utjecati na sami razvoj organizacije, proizvoda i sl. Ovakvom analizom mogu se pravovremeno otkriti neke slabosti i prijetnje, ali isto tako otkriti koje su snage i prilike te ih iskoristiti u budućem razvoju. [10] Također, identifikacijom i analizom tih čimbenika, moguće je bolje razumjeti svoje poslovanje, kao i njegovu poziciju na tržištu. [13]



### 2.4.1. Snage

Snage, tj. prednosti, odnose se na unutarnje čimbenike koji imaju pozitivan utjecaj za postizanje poslovnih ciljeva. Uglavnom su to faktori kojih nas ističu od konkurencije, a oni mogu biti tehnologije koje se koriste, kvaliteta proizvoda, tehničko znanje, vještine i slično. [12]

Neka od pitanja koja bi se mogla postaviti pri analizi snaga su:

- „Što nas čini boljima od konkurencije?
- Koje su naše prednosti u odnosu na konkurenciju?
- Po čemu smo posebni na tržištu?
- Koje kvalitete imaju naši zaposlenici?
- Imamo li snažnu bazu lojalnih kupaca?“ [11]

Snage ove mobilne aplikacije bi bile:

- Pregled i rezervacija svih teniskih terena, na području grada Splita, na jednome mjestu.
- Brz i jednostavan način rezervacije teniskih terena.
- Brz i jednostavan način otkazivanja rezervacija.
- Jednostavno upravljanje rezervacijama.
- Intuitivno i jednostavno korištenje.

### 2.4.2. Prilike

Prilike se odnose na vanjske čimbenike na koje nemamo utjecaja, ali mogu pozitivno djelovati na nas i pomoći u razvoju i stvaranju prednosti nad konkurencijom. Uglavnom su to faktori poput tehnoloških, ekonomskih, ekoloških, političkih i sličnih trendova. [12]

Neka od pitanja koja bi se mogla postaviti pri analizi prilika su:

- „Koje vanjske promjene nam mogu pružiti nove prilike?
- Koji su trendovi u našoj djelatnosti ili u ostalima?
- Što trenutno nedostaje na tržištu, a mi možemo ponuditi?“ [11]

Prilike ove mobilne aplikacije bi bile:

- Proširenje ponude teniskih terena i na druge gradove.
- Proširenje ponude i na druge sportske sadržaje.

- Proširenje mogućnosti upravljanja teniskim terenima, kao što su brisanje, dodavanje, uređivanje terena.

### 2.4.3. Slabosti

Slabosti se odnose na unutarnje čimbenike koji imaju negativan utjecaj na poslovanje. Uglavnom su to faktori poput nedostatka tehničkih vještina, loša kvaliteta proizvoda, loša organizacijska kultura, nedostatak financijskih sredstava i slično. Pri definiranju slabosti potrebno je definirati slabosti koje su ključne za uspješno poslovanje i trebaju se odmah ukloniti, slabosti koje treba što više smanjiti ili one koje se lako mogu riješiti. [12]

Neka od pitanja koja bi se mogla postaviti pri analizi slabosti su:

- *„Koja područja u poslovanju moramo unaprijediti?*
- *Što trebamo izbjeđavati?*
- *Imamo li dovoljno likvidnih sredstava?*
- *Jesmo li dovoljno profitabilni?*
- *Prestiže li nas konkurencija?*
- *Trebamo li promijeniti poslovni model?“ [11]*

Slabosti ove mobilne aplikacije bi bile:

- Potrebno je preuzeti mobilnu aplikaciju.
- Trenutno se može koristiti jedino za područje Grada Splita.
- Mobilna aplikacija je dostupna samo za Android mobilne uređaje.

### 2.4.4. Prijetnje

Prijetnje se odnose na vanjske čimbenike nad kojima nemamo kontrolu, ali mogu imati negativan utjecaj na poslovanje. Uglavnom su to faktori iz područja ekonomije, društva, politike, tehnologije, ekologije i slično. Ako se prijetnje dobro i na vrijeme analiziraju, moguće je smanjiti njihov utjecaj na poslovanje ili ih pretvoriti u prilike. [12]

Neka od pitanja koja bi se mogla postaviti pri analizi prijetnji su:

- *„Koji su negativni utjecaji promjene na tržištu?*
- *Javljaju li se na tržištu novi konkurenti koji mogu pružiti nižu cijenu, bolju uslugu?*
- *Mijenjaju li se navike naših kupaca?*
- *Jesu li naši ključni zaposlenici zadovoljni s uvjetima rada (plaće i druge beneficije)?*
- *Postoji li opasnost odlaska ključnih zaposlenika u konkurentsko poduzeće?“ [11]*

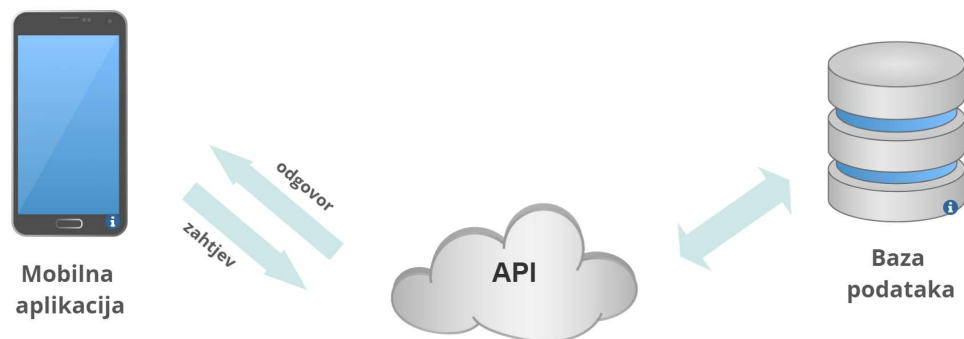
Prijetnje ove mobilne aplikacije su:

- Širenje ponude aplikacija za rezervacije sportskih sadržaja.

### 3. Implementacija softverskog rješenja

U ovom poglavlju opisana je implementacija softverskog rješenja za rezervaciju sportskih terena na primjeru tenisa. Ovo softversko rješenje može se podijeliti na tri komponente:

- Bazu podataka
- API
- Mobilnu aplikaciju



Slika 3.1 Prikaz arhitekture softverskog rješenja

#### 3.1. Baza podataka

Baza podataka je skup povezanih podataka koji su organizirani u strukturiranom formatu. Ona korisnicima omogućava dohvaćanje, spremanje, i manipuliranje podacima na dosljedan i učinkovit način. Baze podataka u današnjem svijetu, punom informacija, postaju sve važniji faktor jer su neophodne za pohranu i upravljanje podacima u različitim aplikacijama. [6] Za izradu ove mobilne aplikacije korištena je SQLite baza podataka koja sadrži podatke o sportskim centrima, korisnicima i samim rezervacijama. U nastavku teksta opisan je proces izrade klasa (engl. *classes*) i samih tablica u bazi na primjeru `CourtType` i `Court` klasa. Kako bi se preuzeti podaci o sportskim centrima mogli spremiti u bazu, potrebno je prvo definirati klase. Kod (Kod 3.1) prikazuje klasu `CourtType.kt`, dok kod (Kod 3.2) prikazuje klasu `Court.kt`.

```

data class CourtType (
    var _id: Long?,
    var name : String,
    var isOpen : Boolean
)

```

### Kôd 3.1 Prikaz CourtType klase

```

data class Court (
    var _id: Long?,
    var name : String,
    var price : Double,
    var nightPrice : Double,
    var courtType : CourtType
)

```

### Kôd 3.2 Prikaz Court klase

Zatim se definira *interface* `TennisRepository` koji definira metode za ažuriranje, brisanje, dodavanje i dohvaćanje podataka iz baze. *Content provider* zatim može pozivati te metode za rad s podacima, a da ne zna kako je `TennisSqlHelper` klasa definirana.

```

fun updateCourtType(
    values: ContentValues?, selection: String?,
    selectionArgs: Array<String>?
): Int

```

### Kôd 3.3 Prikaz metode za uređivanje CourtType entiteta

`TennisSqlHelper` klasa nasljeđuje `TennisRepository` sučelje te implementira njegove metode. Sučelje u programiranju označava koncept koji opisuje objekte i metode bez njihovog implementiranja. To je važno zbog toga što ono omogućava da više klasa može naslijediti jedno sučelje, ali svaka klasa ga na različite načine može implementirati, čime se povećava sama fleksibilnost koda.

```

override fun updateCourtType(
    values: ContentValues?,
    selection: String?,
    selectionArgs: Array<String>?) =
    writableDatabase.update(COURT_TYPE_TABLE_NAME,
        values, selection, selectionArgs
    )

```

#### Kôd 3.4 Prikaz implementacije metode za uređivanje CourtType entiteta

Nasljeđivanje označava koncept objektno orijentiranog programiranja koji omogućava izradu nove klase na temelju već postojeće klase. Nova klasa time preuzima, odnosno nasljeđuje svojstva i metode od postojeće klase i ima mogućnost mijenjanja njihova ponašanja.

Uz `TennisRepository` sučelje, `TennisSqlHelper` klasa nasljeđuje i `SQLiteOpenHelper` apstraktnu klasu, koja vodi brigu o postojanju i verziji baze, npr. brine se da se baza kreira ako ne postoji ili ažurira ako postoji.

Zatim se u klasi `TennisSqlHelper.kt` definiraju varijable za izradu tablica, pri čemu je važno definirati potrebna ograničenja, poput dužine teksta, mogućnost praznog (engl. *null*) unosa i, najvažnije, same odnose tablica.

```

private val CREATE_COURT_TYPE_TABLE = "create table
    $COURT_TYPE_TABLE_NAME ( " +
    "${CourtType::_id.name} integer primary key autoincrement, "+
    "${CourtType::_name.name} text not null, " +
    "${CourtType::_isOpen.name} integer not null" +
    ")"

```

#### Kôd 3.5 Prikaz varijable za izradu CourtType tablice

```

private val CREATE_COURT_TABLE = "create table
\$COURT_TABLE_NAME( " +
    "\${Court::_id.name} integer primary key
autoincrement, "+
    "\${Court::name.name} text not null," +
    "\${Court::price.name} double not null," +
    "\${Court::nightPrice.name} double not null," +
    "\$FOREIGN_KEY_COURT_TYPE INTEGER NOT NULL, " +
    "\$FOREIGN_KEY_TENNIS_PLACE INTEGER NOT NULL, " +
    "FOREIGN KEY (\$FOREIGN_KEY_COURT_TYPE) REFERENCES
\$COURT_TYPE_TABLE_NAME(\${CourtType::_id.name})," +
    "FOREIGN KEY (\$FOREIGN_KEY_TENNIS_PLACE) REFERENCES
\$TENNIS_PLACE_TABLE_NAME(\${TennisPlace::_id.name})" +
    ")"

```

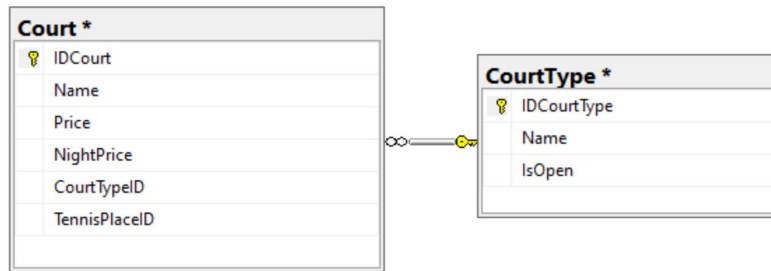
### Kôd 3.6 Prikaz varijable za izradu Court tablice

U bazi podataka postoje tri vrste relacija:

- jedan-prema-jedan (1-1)
- jedan-prema-više (1-N)
- više-prema-više (M-N)

„Jedan na jedan“ relacija koristi se kada za svaki redak u jednoj tablici postoji samo jedan povezani redak u drugoj tablici. Relacija „jedan na više“ koristi se kada u jednoj tablici jedan redak može imati više povezanih redaka u drugoj tablici. Relacija „više na više“ koristi se kada više redaka u jednoj tablici mogu biti povezani s više redaka u drugoj tablici. Tada je potrebno napraviti treću tablicu koja će služiti kao poveznica tih dviju tablica. [6]

Tablice CourtType i Court u odnosu su „jedan prema više“, tj. jedan teren može biti samo jednog tipa terena, ali više terena može biti istog tipa terena. Iz tog razloga na tablici se Court definira strani ključ (engl. *foreign key*) CourtTypeId koji je veza na tablicu CourtType. Odnos tih tablica prikazan je slikom (Slika 3.2).



Slika 3.2 Prikaz relacije 1-N na primjeru Court i CourtType tablica

### 3.1.1. Tablice

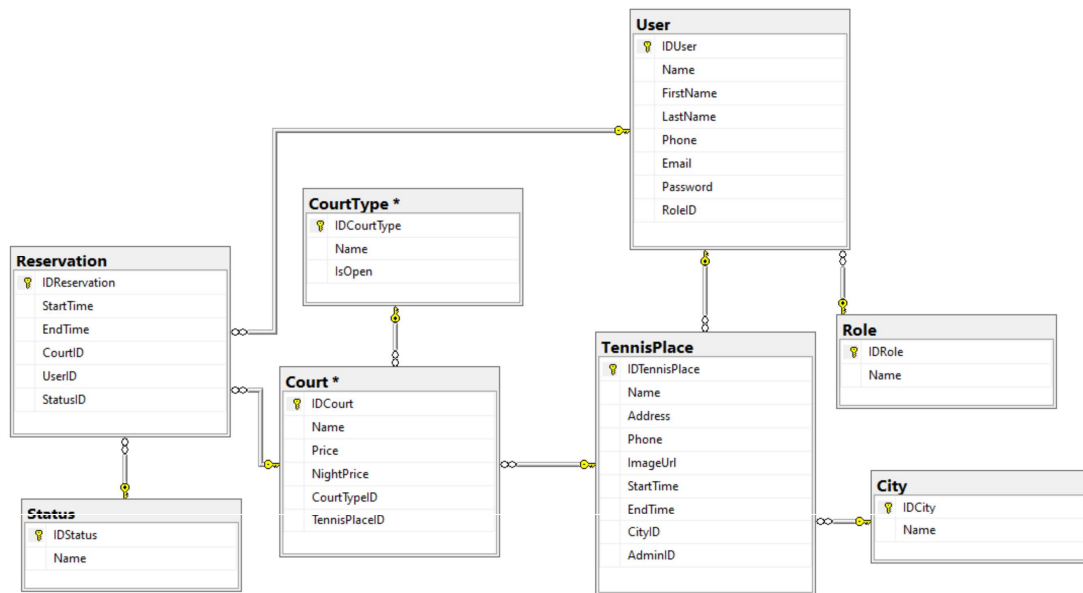
Baza podataka ove aplikacije sastoji se od osam tablica:

- Role - sadrži nazive korisničkih rola u aplikaciji i jedinstveni identifikacijski broj.
- User - sadrži informacije o korisnicima aplikacije kao što su ime, prezime, telefon, e-mail, lozinka, tip role i jedinstveni identifikacijski broj.
- City - sadrži nazive gradova u kojima se nalaze sportski centri i jedinstveni identifikacijski broj.
- TennisPlace - sadrži informacije o sportskim centrima kao što su naziv, adresa, telefon, grad, putanju slike, administratora, početno radno vrijeme, krajnje radno vrijeme i jedinstveni identifikacijski broj.
- Court type - sadrži podatke o tipovima teniskih terena, informaciju je li otvorenog tipa i jedinstveni identifikacijski broj.
- Court - sadrži podatke o teniskim terenima kao što su naziv, cijena, sportski centar, tip terena i jedinstveni identifikacijski broj.
- Status - sadrži podatke o tipu statusa rezervacije i jedinstveni identifikacijski broj.
- Reservation - sadrži podatke o rezervacijama, kao što su početno vrijeme rezervacije, krajnje vrijeme rezervacije, teren, korisnik, status rezervacije i jedinstveni identifikacijski broj.



### 3.1.2. ER Dijagram

Dijagram odnosa entiteta (engl. *entity relationship diagram*, skraćeno ER dijagram) je grafički prikaz entiteta, atributa i relacija baze podataka. ER dijagrami su veoma korisni za vizualizaciju i modeliranje strukture baze podataka. Na slici (Slika 3.3 ER dijagram baze podataka) prikazan je ER dijagram entiteta u bazi podataka za rezervaciju teniskih terena.



Slika 3.3 ER dijagram baze podataka

### 3.2. Web API

Sučelje za programiranje aplikacija (engl. *Application Programming Interface*, skraćeno API) je vrsta API-ja dizajnirana da se pristupa putem interneta koristeći protokol prijenosa hiperteksta (engl. *Hyper Text Transfer Protocol*, skraćeno HTTP protokol) s ciljem pružanja standardiziranog načina komunikacije i razmjene podataka za različite softverske aplikacije. Web API-ji mogu se dizajnirati za upotrebu s različitim klijentima, poput web-preglednika, mobilnih i desktop aplikacija. Moguće ih je implementirati koristeći različite tehnologije poput JSON-a, XML-a (engl. *EXtensible Markup Language*) i REST-a (engl. *Representational State Transfer*). [3]

Podaci o sportskim centrima preuzimaju se s javnog API-ja. S obzirom na to da trenutno ne postoji javni API s podacima o svim sportskim centrima na području Splita, putem aplikacije Mocky izrađen je prilagođeni, tj. testni API kako bi se olakšala sama izrada mobilne aplikacije.

Kako bi se moglo pristupiti API-ju, potrebno je u aplikaciji, tj. u `AndroidManifest.xml` datoteci dodati dozvolu za pristupanje internetu. `AndroidManifest.xml` datoteka sadrži informacije o aplikaciji poput verzije aplikacije, popisa aktivnosti, servisa, biblioteka, dozvola i slično. Vrlo je važna jer se u njoj nalaze informacije koje su potrebne za pokretanje aplikacije, ali i da aplikacija ima pristup potrebnim resursima i funkcionalnostima.

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

Kôd 3.7 Prikaz dodavanja dozvole za pristup internetu u `AndroidManifest.xml` datoteci

Nakon slanja upita s prikladnom putanjom za dohvat podataka o svim sportskim centrima na području Splita, podaci se pohranjuju u prethodno definirane varijable. S obzirom na to da sportski centri sadrže i putanju za sliku, napravljen je `ImagesHandler` koji u sebi sadrži metode za njezino preuzimanje i spremanje slika. Preuzeta slika ima prilagođenu dimenziju, novu putanju i naziv te se takva sprema u bazu.

```

fun downloadImageAndStore(context: Context,url: String, fileName:
String) : String? {
val extension = url.substring(url.lastIndexOf("."))
val file: File = getFile(context, fileName, extension)
try {
    val con: HttpURLConnection = createGetHttpURLConnection(url)
    con.inputStream.use { `is` ->
        FileOutputStream(file).use{ fos ->
            val bitmap = BitmapFactory.decodeStream(`is`)
            val resizedBitmap : Bitmap = getResizedBitmap(
            bitmap,
            TENNIS_PLACE_PICTURE_WIDTH,
            TENNIS_PLACE_PICTURE_HEIGHT)
            val buffer: ByteArray = getBytesFromBitmap(resizedBitmap)
            fos.write(buffer)
            return file.absolutePath}}
    } catch (e: Exception){
        Log.e(TAG, e.message, e)
        return null}
}

```

### Kôd 3.8 Prikaz spremanja slika u ImageHandleru

Unuatr metode `fetchItems` dohvaćaju se podaci o sportskim centrima s API-ja i pretvaraju se u definirane objekte, koji se zatim spremaju u bazu podataka.

```

fun fetchItems() {
    val request = tennisApi.fetchItems()

    request.enqueue(object: Callback<List<TennisPlaceItem>>{
        override fun onResponse(
            call: Call<List<TennisPlaceItem>>,
            response: Response<List<TennisPlaceItem>>
        ) {
            if (response.body() != null) {
                populateItems(response.body()!!)
            }
        }
        override fun onFailure(call:
            Call<List<TennisPlaceItem>>, t: Throwable) {
                Log.d(javaClass.name, t.message, t)
            }
        })
    }
}

```

Kôd 3.9 Prikaz metode za dohvaćanje i spremanje podataka

### 3.3. Mobilna aplikacija

U ovom je poglavlju ukratko opisana arhitektura mobilne aplikacije, odnosno na koje dijelove je možemo podijeliti te čemu koji dio služi. Također, poglavlje sadrži i opise dviju vrsta korisnika, koji mogu koristiti mobilnu aplikaciju, i opisuje funkcionalnosti koje im mobilna aplikacija pruža.

#### 3.3.1. Arhitektura sustava

Arhitektura mobilne aplikacije može se podijeliti na Android korisničko sučelje (engl. *user interface*, skraćeno UI), sloj repozitorija te *Content Providera*.

Android UI sadrži prikaze i komponente poput aktivnosti (engl. *activities*), fragmenata i obavijesti (engl. *notifications*). Aktivnosti obično sadrže elemente korisničkog sučelja za prikaz neke funkcionalnosti poput teksta, polja za unos, gumbova i slično. Fragmenti se koriste za stvaranje fleksibilnijeg i modularnijeg korisničkog sučelja. S njima se korisničko sučelje može „razbiti“ na manje dijelove i koristiti ih za više različitih aktivnosti, što olakšava razvoj i kasnije održavanje aplikacije.

Obavijesti su način prikazivanja nekih poruka korisniku bilo da se radi o porukama upozorenja, podsjetnika ili nekim drugim porukama. U ovom softverskom rješenju koriste se dvije vrste obavijesti, jedna koja korisnika obavještava dva sata prije početka rezerviranog termina i ona koja se poziva kada administrator otkaže rezervaciju. Prva vrsta obavijesti koristi *BroadcastReceiver* i *AlarmManager*. *AlarmManager* koristi se za postavljanje alarma koji se ponavlja nakon zadanog vremena. Pri aktivaciji alarma poziva se *BroadcastReceiver* pa on pokreće *IntentService*. *BroadcastReceiver* je komponenta koja služi za reagiranje na događaje poput poruka, promjena stanja i slično. *IntentService* je servis koji se izvršava u pozadini, a služi za obavljanje neke radnje poput slanja podatka, preuzimanja slika i slično. Nakon što *IntentService* obavi zadanu akciju, šalje rezultat *BroadcastReceiveru* ili aktivnosti te se automatski zaustavlja. Za drugu vrstu obavijesti nije potrebno koristiti *AlarmManager*.

Repozitorij je odgovoran za rukovanje pristupom podacima, a sadrži logiku za dohvaćanje podataka iz lokalne baze podataka i Web API-ja. *Content Provider* upravlja mrežnim pozivima za dohvaćanje i ažuriranje podataka s udaljenog poslužitelja. Također, komunicira s repozitorijem kako bi pružio tražene podatke. Ovim načinom razdvajaju se odgovornosti i omogućuje se fleksibilnost u pohrani i dohvaćanju podataka.

### 3.3.2. Opis korisnika

Postoje dvije vrste korisnika mobilne aplikacije:

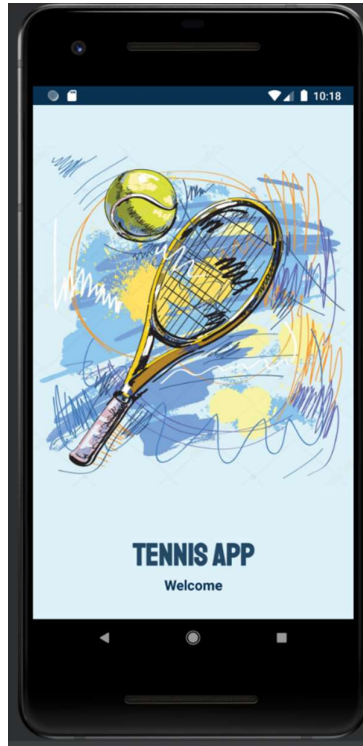
- administratori teniskih terena
- obični korisnici

Administratori teniskih terena su osobe koje rade u sportskom centru i zadužene su za upravljanje rezervacijama terena. Administratori imaju uvid u rezervacije za teniske terene sportskog centra za koje su zaduženi. Također, imaju mogućnost otkazivanja rezervacija teniskih terena koje nisu započele i uređivanja profila.

Obični korisnici aplikacije su osobe koje rezerviraju teniske terene. Oni imaju mogućnost pregleda sportskih centara na području Splita, rezerviranja teniskih terena, otkazivanja rezervacija i uređivanja profila.

## 4. Funkcionalnosti implementirane aplikacije

U ovom su poglavlju opisane funkcionalnosti mobilne aplikacije za rezervaciju teniskih terena. Otvaranjem mobilne aplikacije za rezervaciju teniskih terena, svim korisnicima prvo se prikazuje `SplashScreenActivity`, koji sadrži sliku i tekst s animacijom, nakon čega se korisnika automatski prebacuje na sljedeći ekran.

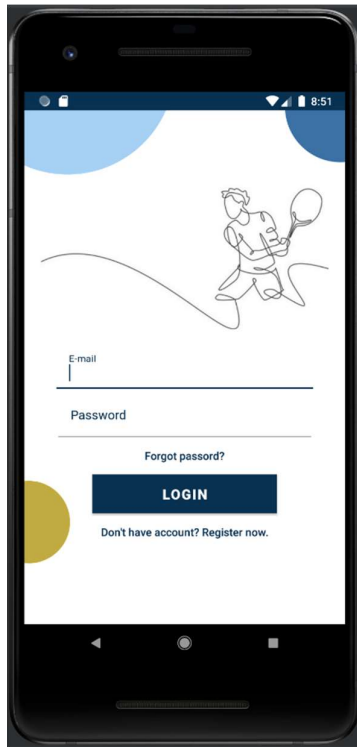


Slika 4.1 Prikaz početnog ekrana s animacijom

### 4.1. Prijava korisnika

Nakon početnog ekrana s animacijom, otvara se `LoginActivity`, tj. ekran za prijavu korisnika u aplikaciju. Na polje `E-mail` korisnik unosi naziv svoje elektroničke pošte kojom se registrirao u aplikaciju. Na polje `Password` korisnik unosi svoju lozinku. Pritiskom na gumb `Login`, poziva se metoda koja dohvaća naziv elektroničke pošte i lozinku korisnika te prvo provjerava postoji li korisnik s tom elektroničkom poštom. Ako ne postoji, korisniku se prikazuje poruka da korisnik nije pronađen. Ako postoji, provjerava podudara li se upisana lozinka sa spremljenom lozinkom. Nakon uspješne prijave, provjerava se rola korisnika, a

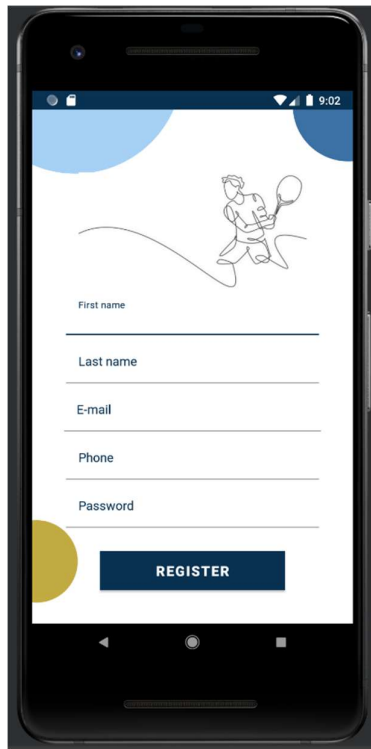
ovisno o roli korisnika, prebacuje se na sljedeći ekran. Administratore terena prebacuje se na `TennisCourtsActivity` , dok se obične korisnike prebacuje na `TennisPlacesActivity`.



Slika 4.2 Prikaz ekrana za prijavu korisnika

## 4.2. Registracija korisnika

Korisnici, koji nemaju registriran račun u mobilnoj aplikaciji, mogu se registrirati na ekranu za registraciju kojem se pristupa s ekrana za prijavu pritiskom na link `Don't have account? Register now.` Za registraciju je potrebno ispuniti polja za `FirstName`, `LastName`, `E-mail`, `Phone` te `Password`. Pritiskom na gumb `Register` poziva se metoda koja provjerava postoji li korisnik s istom elektroničkom poštom u bazi podataka. Ako korisnik postoji, prikazuje mu se poruka da postoji korisnik s istom elektroničkom poštom. Ako pak ne postoji, korisnika se ubacuje u bazu pa ga se šalje na ekran s prikazom sportskih centara. Svaki korisnik, koji se registrira, ima rolu običnog korisnika.



Slika 4.3 Prikaz ekrana za registraciju korisnika

### **4.3. Administrator – Pregled teniskih terena**

Ekran s popisom teniskih terena, za koje je prijavljeni administrator zadužen, početna je stranica nakon uspješne prijave administratora u aplikaciju. Na ovom ekranu nalazi se lista svih terena sportskog centra administratora. Svaki teren prikazan je u obliku kartice na kojoj se prikazuje naziv terena. Odabirom kartice terena, korisniku se otvara fragment `CourtReservationFragment`.



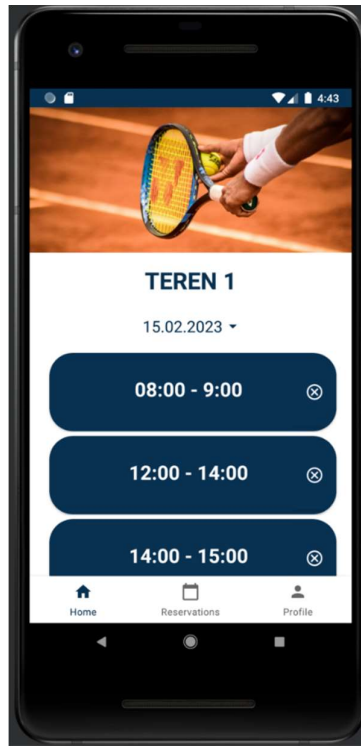


Slika 4.4 Prikaz ekrana za pregled teniskih terena

#### 4.4. Administrator – Pregled i otkazivanje rezervacija

Na ekran s popisom rezervacija korisnika, administrator dolazi odabirom ikone kalendara u izborniku. Na ovom ekranu administrator može vidjeti sve nadolazeće rezervacije za sve terene. Prilikom dolaska na ekran prikazuju se nadolazeće rezervacije za sve terene sportskog centra za trenutni datum. Ako korisnik želi vidjeti rezervacije za neki drugi dan u izborniku datuma (engl. *date picker*), može odabrati željeni datum. Rezervacije terena prikazane su kao kartice na kojima se nalazi naziv terena i vrijeme rezervacije i gumb za otkazivanje rezervacije. Korisnik rezervaciju može otkazati na način da pritisne ikonu „X“, koja se nalazi na svakoj kartici, te u dijaloškom prozoru pritisne gumb za otkazivanje. Na stranici se također nalazi i gumb za otkazivanje svih rezervacija za odabrani datum.

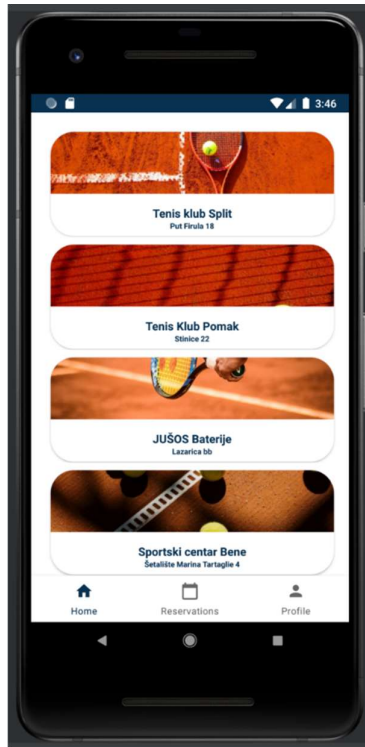
Na fragment s pregledom rezervacija po pojedinom terenu dolazi se na način da se na početnoj stranici s prikazom terena odabere kartica željenog terena. Na ovom ekranu također se nalazi izbornik datuma s čijom promjenom se prikazuju rezervacije za taj datum i samo za taj teren. Rezervacije terena na ovom ekranu su prikazane kao kartice na kojima se nalazi vrijeme rezervacije i gumb za otkazivanje rezervacije.



Slika 4.5 Prikaz ekrana za pregled rezervacija odabranog terena

## 4.5. Pregled sportskog centra

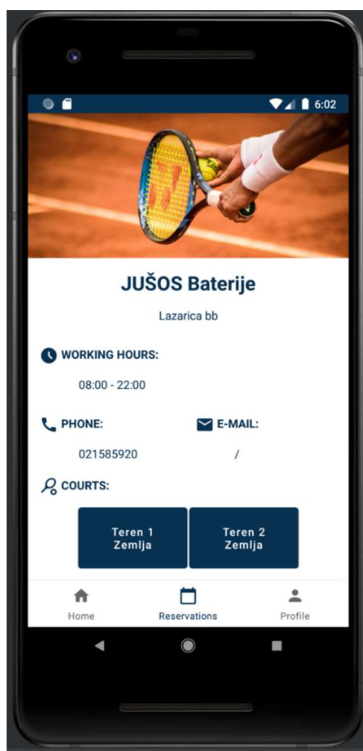
Nakon uspješne prijave u aplikaciju, korisniku se prikazuje stranica za pregled sportskih centara. Na ovoj stranici korisniku se prikazuje popis lokacija, odnosno sportskih centara za koje se može rezervirati teniski teren. Sportski centri prikazani su kao kartice sa slikom, imenom i adresom teniskog kluba. Pritiskom na bilo koji od ponuđenih sportskih centara, korisnika se prebacuje na fragment s informacijama o tom sportskome centru.



Slika 4.6 Prikaz ekrana s popisom sportskih centara

#### 4.5.1. Detalji sportskog centra

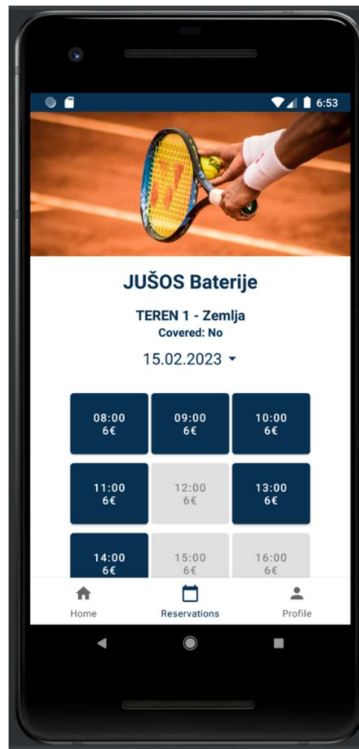
Na stranicu detalja, dolazi se odabirom sportskog centra na stranici za pregled sportskih centara. Na ovoj stranici nalaze se informacije o sportskom centru, kao što su naziv, adresa, telefon, radno vrijeme i popis terena. Teniski tereni prikazani su kao kartice s nazivom i vrstom terena. Odabirom nekog terena otvara se fragment za rezervaciju teniskog terena s prikazom termina.



Slika 4.7 Prikaz ekrana s detaljima sportskog centra

#### 4.5.2. Rezervacija terena

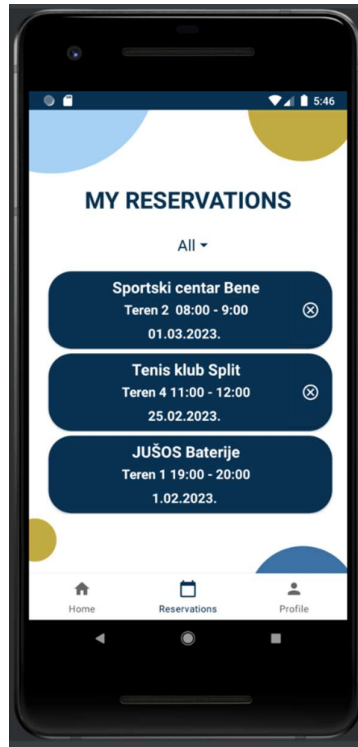
Dolaskom na ekran za rezervaciju, prikazuju se detalji teniskog terena poput naziva, vrste terena te je li teniski teren pokriven ili nije. Također, na stranici se nalaze termini rezervacija za trenutni datum. Datum je moguće mijenjati u izborniku datuma. Termini za rezervacije su prikazani u obliku kartica s početnim vremenom rezervacije i cijenom teniskog terena za to vrijeme. Termine je moguće rezervirati na sat vremena, a ako korisnik želi rezervirati teniski teren na više od sat vremena, moguće je napraviti više rezervacija. Termini koji su već rezervirani prikazani su u sivoj boji i ne mogu se odabrati. Odabirom slobodnog termina korisniku se prikazuje dijaloški prozor na kojem može potvrditi da želi rezervirati termin ili odustati. Ako korisnik potvrdi rezervaciju, kartica s odabranim terminom postaje rezervirana, tj. mijenja boju.



Slika 4.8 Prikaz ekrana za rezervaciju teniskog terena

### 4.5.3. Pregled i otkazivanje rezervacija

Ekranu za pregled rezervacija, korisnik pristupa putem izbornika odabirom ikone kalendara. Na ovom ekranu korisniku se prikazuju sve njegove rezervacije. Rezervacije su prikazane u obliku kartica s informacijama o datumu, terminu i nazivu sportskog centra. Korisnik ima mogućnost filtrirati rezervacije na sve, prošle te nadolazeće. Za nadolazeće rezervacije, korisnik ima mogućnost otkazati rezervaciju ako je do rezervacije ostalo još minimalno dva sata, i to pritiskom na gumb za otkazivanje rezervacije. Rezervacije koje su prošle ili ih više nije moguće otkazati nemaju gumb za otkazivanje rezervacije.



Slika 4.9 Prikaz ekrana s korisnikovim rezervacijama

#### 4.5.4. Podsjetnik o nadolazećem terminu

Odabirom ikone profila, korisniku se otvara ekran s pregledom korisnikova profila na kojem se nalazi i opcija kojom može uključiti ili isključiti dolazak obavijesti o rezervacijama. Korisnika se obavještava jedan sat prije same rezervacije da rezervacija počinje za jedan sat. Također, ako administratori teniskog kluba otkažu korisnikovu rezervaciju, korisnik dobiva obavijest o tome.

#### 4.6. Pregled profila

Odabirom ikone profila u izborniku, otvara se ekran s informacijama o korisniku, gumbom za postavke obavijesti i gumbom za odjavu iz korisnikova računa.

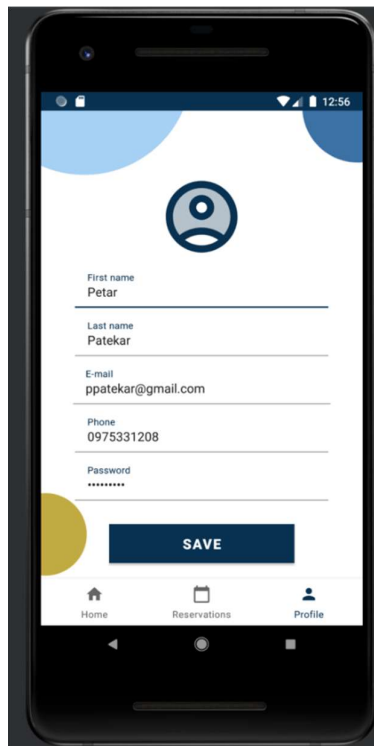
Ekran za pregled profila kod administratora teniskih terena nema gumb za obavijesti.

#### 4.6.1. Podsjetnik o nadolazećem terminu

Odabirom gumba za obavijesti, korisnik može uključiti ili isključiti dolazak obavijesti o rezervacijama. Korisnika se obavještava jedan sat prije same rezervacije da rezervacija počinje za jedan sat. Također, ako administratori teniskog kluba otkazu korisnikovu rezervaciju, korisnik dobiva obavijest o tome.

#### 4.6.2. Uređivanje profila

Korisnik ima pristup fragmentu za uređivanje profila odabirom gumba za uređivanje profila ispod naziva korisnika. Na ovom ekranu korisnik može uređivati podatke kao što su ime i prezime, telefon, lozinka te e-mail. Odabirom gumba za spremanje, spremaju se unesene promjene te se korisnika vraća na ekran za pregled profila.



Slika 4.10 Prikaz ekrana za uređivanje profila

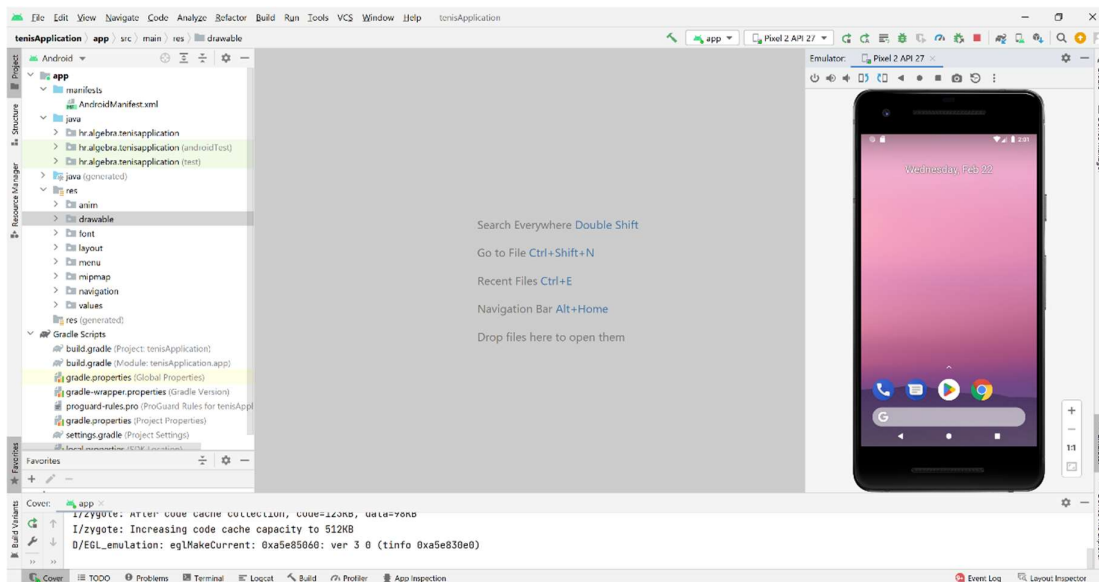
## 5. Korištene tehnologije za implementaciju programskog rješenja

U ovom je poglavlju opisano razvojno okruženje, alati i jezici koji su korišteni za implementaciju programskog rješenja za rezervaciju teniskih terena.

### 5.1. Android Studio razvojno okruženje

Android Studio je službeno integrirano razvojno okruženje (engl. *integrated development environment*, skraćeno IDE) za razvoj Android aplikacija koji je razvio Google. Prije Android Studija, kao službeni IDE za razvoj aplikacija, koristio se Eclipse koji je omogućavao razvoj Android aplikacija u Java programskom jeziku. Android Studio korisnicima nudi sučelje za pisanje i testiranje Android aplikacija pomoću programski jezika: Java, Kotlin i C++. [1]

IDE sadrži različite alate poput uređivača koda, sustava za izgradnju i kompiliranje aplikacija, vizualnog uređivača za dizajniranje korisničkih aplikacija, alata za ispravljanje grešaka u kodu, podršku za sustave za kontrolu verzija, kao što je Git te mnoge druge alate. [2] Sve to omogućava da se u ovom razvojnom okruženju razvijaju visokokvalitetne Android aplikacije. [5]



Slika 5.1 Prikaz korisničkog sučelja u Android Studiju



Neke od prednosti korištenja Android Studija su:

- razne funkcije i alati koji olakšavaju razvoj Android aplikacija
- Android Studio je besplatan za korištenje te se stalno ažurira i nadograđuje
- pruža podršku za više programskih jezika
- intuitivan dizajn sučelja pa omogućava brzo i jednostavno korištenje.

Neki od nedostataka Android Studija su:

- zbog brojnih funkcionalnosti i alata koje nudi. može biti kompleksan za početnike
- za korištenje Android Studija potrebno je imati računalo s dovoljno jakim procesorom i memorijom
- prilikom početnog učitavanja projekta, a pogotovo složenijih projekata, učitavanje projekta može biti sporo.

## 5.2. Tehnologije i programski jezici

Za izradu ovoga softverskog rješenja koristile su se različite tehnologije i alati od kojih su neke najvažnije opisane u ovom poglavlju. Uz objašnjenje što su i čemu služe, ukratko je objašnjeno i njihovo korištenje uz primjere korištenja za ovu mobilnu aplikaciju. U ovom poglavlju također je opisan i programski jezik koji se koristio za izradu ove mobilne aplikacije.

### 5.2.1. Mocky

Mocky je besplatna aplikacija za stvaranje prilagođenih API-ja, a uglavnom se koristi za razvoj i testiranje softvera. Omogućava razvojnim programerima simulaciju HTTP odgovora, bez potrebe za izgradnjom pune funkcionalnosti API-ja. Korištenje ovakvih API-ja može pomoći i pri testiranju različitih slučajeva, koje možda ne bi mogli testirati sa stvarnim API-jima. Prednost ovakvih API-ja je što mogu pomoći i ubrzati razvoj, ali i poboljšati krajnju kvalitetu proizvoda. [8]

Kako bi se izradio prilagođeni API, potrebno je odabrati vrstu HTTP statusa, vrstu sadržaja odgovora, vrstu kodne stranice (engl. *charset*). Opcionalno se mogu dodati HTTP zaglavlja (eng. *headers*), http odgovor te jedinstveni skriveni token. Ako korisnik ne unese token, generira se nasumični. Prikaz stranice za izradu API-ja prikazan je slikom (Slika 5.2).

HTTP Status REQUIRED

200 - OK

The HTTP Code of the HTTP response you'll receive.

Response Content Type REQUIRED      Charset REQUIRED

application/json      UTF-8

The Content-Type header that will be sent with the response.      The Charset used to encode/decode your payload.

HTTP Headers OPTIONAL

```
{
  "X-Foo-Bar": "Hello World"
}
```

Customize the HTTP headers sent in the response. Define the headers as a JSON object.

HTTP Response Body OPTIONAL

```
{
  "name": "Teren 3",
  "price": 8,
  "nightPrice": 11,
  "courtType": {
    "name": "Zemlja",
    "isOpen": true
  }
},
{
  "name": "Teren 4",
  "price": 8,
  "nightPrice": 11,
  "courtType": {
    "name": "Zemlja",
    "isOpen": true
  }
},
{
  "name": "Teren 5",
  "price": 8,
  "nightPrice": 11
}
```

Options to manage your mock after its creation OPTIONAL

Secret token      Mock identifier

Required to update/delete your mock. If blank, a random secret will be generated.      Just a name to identify this mock in your management console later.

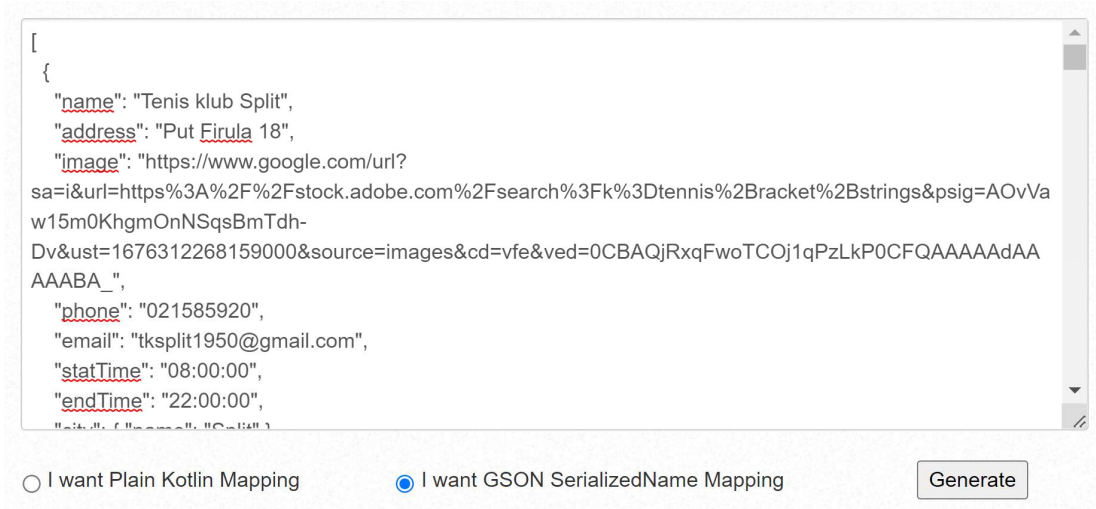
**GENERATE MY HTTP RESPONSE**      Never expire

Slika 5.2 Mocky – prikaz primjera izrade prilagođenog API-ja

### 5.2.2. Json2Kotlin

Java2Kotlin je web-alat koji pretvara JSON podatke u kotlin klase podataka. Alat omogućava programerima da na jednostavan i brz način dobiju generirane kotlin klase podataka iz JSON podataka. [7]

Java2Kotlin koristi se na način da se unesu JSON podaci i odabere opcija mapiranja podataka.



Slika 5.3 Json2Kotlin – prikaz djela za generiranje Kotlin klasa

Nakon toga potrebno je preuzeti datoteku s generiranim klasama i kopirati klase u svoj programski kod.

```

data class TennisPlace (

    @SerializedName("name") val name : String,
    @SerializedName("address") val address : String,
    @SerializedName("image") val image : String,
    @SerializedName("phone") val phone : Int,
    @SerializedName("email") val email : String,
    @SerializedName("startTime") val startTime : Date,
    @SerializedName("endTime") val endTime : Date
    @SerializedName("city") val city : City,
    @SerializedName("courts") val courts : List<Courts>
)

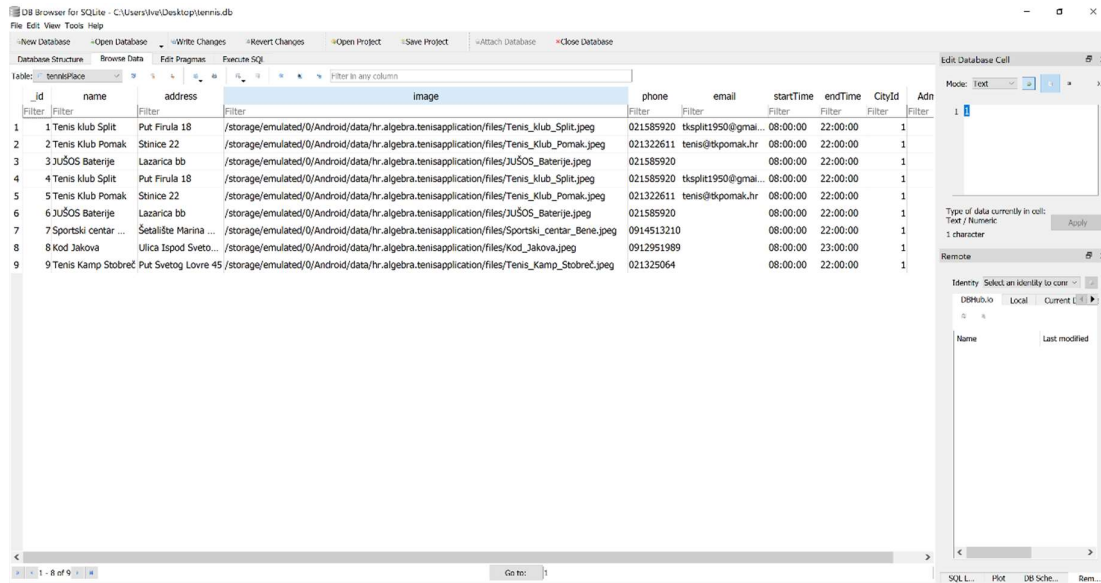
```

Slika 5.4 Json2Kotlin – Prikaz generirane klase za TennisPlace

### 5.2.3. SQLite

SQLite je samostojeća, neovisna i besplatna biblioteka kojoj nije potreban poslužitelj niti dodatna konfiguracija. Može se koristiti na raznim platformama, kao što su Linux, Windows i macOS, te na mobilnim uređajima s Androidom i iOS-om. Podržava popularne programske jezike poput Java, Python-a, PHP-A te mnoge druge. Operacije brisanja, čitanja, ažuriranja i stvaranja podataka obavljaju se izravno kroz API SQLite-a, što bi značilo da se aplikacije koje ga koriste mogu lako distribuirati bez potrebe za posebnim upravljanjem bazom

podataka. Glavna prednost SQLite-a je brzina i lakoća izvršavanja upita pa je veoma popularan za aplikacije koje imaju manje zahtjeve za baze podataka. [15]



Slika 5.5 Prikaz SQLite baze podataka

## 5.2.4. Retrofit

Retrofit je biblioteka (engl. *library*) koja programerima omogućava, na veoma jednostavan i brz način, dohvaćati i učitati podatke putem web-usluge temeljene na REST-u. „Retrofit upravlja procesom primanja, slanja i stvaranja HTTP zahtjeva i odgovora“. [14]

Kako bi se Retrofit mogao koristiti, potrebno je prvo dodati Retrofit i njegove ovisnike (engl. *dependencies*) u `build.gradle` datoteku unutar mobilne aplikacije.

`build.gradle` datoteka je konfiguracijska datoteka koja služi za definiranje izgradnje i postavki Android Studio projekta. U njoj se također definiraju biblioteke, moduli i drugi ovisnici koji se služe u projektu. Pri izgradnji projekta ta će se datoteka koristiti za stvaranje izvršne verzije aplikacije.

S obzirom na to da su podaci koje dobivamo iz prilagođenog API-ja u JSON formatu, potrebno je u `build.gradle` datoteci specificirati biblioteku koja će rukovati obradom podataka u tom formatu. Jedna od njih je GSON Java biblioteka koja pretvara Java objekte

u JSON format i obrnuto. Iz tog razloga se kod generiranja klasa na slici (Slika 5.4) odabrala opcija GSON mapiranja.

```
dependencies {
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
}
```

Kôd 5.1 Prikaz dodavanja ovisnika za korištenje Retrofit-a u `build.gradle` datoteci

Nakon toga potrebno je definirati protokol (engl. *interface*) koji sadrži metode koje su potrebne za rad s podacima, npr. *GET* metodu koja dohvaća podatke s definiranog API-ja.

```
package hr.algebra.tenisapplication.api

import retrofit2.Call
import retrofit2.http.GET

const val API_URL="https://run.mocky.io/"
interface TennisApi {
    @GET("v3/b32d18d3-3488-4fe0-a842-5c6f89427196")
    fun fetchItems(): Call<List<TennisPlaceItem>>
}
```

Kôd 5.2 Retrofit - prikaz definiranog protokola

Zatim se radi *builder* koji dohvaća podatke s API-ja te ih pretvara ih u GSON objekte. Također, Retrofit *builder* sam izgrađuje cijeli servis i njegovu implementaciju.

```
private val tennisApi: TennisApi
init {
    val retrofit = Retrofit.Builder()
        .baseUrl(API_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .build()
    tennisApi = retrofit.create(TennisApi::class.java)
}
```

Kôd 5.3 Retrofit – prikaz korištenja Retrofit *buildera*

## 5.2.5. Kotlin

„Kotlin je statički tipizirani programski jezik koji se izvršava na JVM-u ili se prevodi u JavaScript. Kotlin je izgrađen na Java Virtual Machine-u (JVM) i savršeno se uklapa u postojeću Java infrastrukturu." [4] Riječ je o modernome programskom jeziku koji kombinira snažne i stabilne značajke Java platforme, s novim i naprednim značajkama poput proširenja funkcija, lambda izraza i slično. Dizajniran je za korištenje kod različitih vrsta projekata od server-side aplikacija do mobilnih aplikacija.

Neke od ključnih značajki i prednosti Kotlin programskog jezika su:

- vrlo kompaktan jezik koji omogućava pisanje izražajnijeg, kraćeg i čitljivijeg koda u odnosu na kod napisan u drugim programskim jezicima
- interoperabilan je s Javom, pa se može koristiti uz postojeći Java kod
- kombinira najbolje značajke objektno orijentiranog i funkcionalno programiranja
- uključuje niz funkcionalnih značajki programiranja, kao što su lambda izrazi, proširenja funkcija, funkcije višeg reda i slično
- snažan sustav tipova koji pomaže u sprječavanju grešaka vezanih uz prazne (engl. *null*) vrijednosti, koje su česti izvori grešaka u Javi
- omogućava programerima pisanje koda bez eksplicitnog deklariranja tipova varijabli i izraza
- podržava korutine (engl. *coroutines*), koje su moćan alat za pisanje asinkronog i ne blokirajućeg koda
- sadrži brojne značajke koje olakšavaju razvoj mobilnih aplikacija. [9]

Sve prethodno navedene značajke čine Kotlin snažnim i popularnim programskim jezikom, pogotovo u području razvoja mobilnih aplikacija.

## Zaključak

Sve veća upotreba mobilnih uređaja dovela je do širenja mobilnih aplikacija. Zbog pristupačnosti i jednostavnosti koje mobilne aplikacije pružaju korisnicima, sve se češće pojavljuju i aplikacije za rezerviranje različitih usluga. Time se želi ubrzati, ali i pojednostaviti stari način rezerviranja usluga preko telefona.

Za uvid u način rezerviranja teniskih terena na području Splita, provedena je anonimna anketa. Rezultati ankete pokazali su da većina korisnika rezervira teniske terene putem telefona, što se pokazalo kao zamoran proces. Upravo taj problem bio je inspiracija za izradu mobilne aplikacije za rezervaciju teniskih terena.

Svrha ovog završnog rada bila je stvoriti jednostavnu aplikaciju za rezerviranje i upravljanje rezervacijama teniskih terena, koja će to ponuditi na jednome mjestu za sve sportske centre na području grada Splita koji iznajmljuju teniske terene.

Za pohranu podataka, korištena je SQLite baza podataka koja se pokazala kao vrlo jednostavno i praktično rješenje za ovu mobilnu aplikaciju. Android Studio koristio se kao razvojno okruženje mobilne aplikacije zbog mogućnosti koje nudi, ali i zbog toga što je službeni IDE za razvoj Android aplikacija. Samo je rješenje pisano u Kotlin programskom jeziku koji se pokazao kao odličan izbor za razvoj mobilnih aplikacija zbog brojnih funkcionalnosti koje nudi.

SWOT analizom definirane su snage, slabosti, prijetnje i prilike ovog softverskog rješenja. Nadogradnjom rješenja za korištenje na području cijele Hrvatske, uvođenjem mogućnosti ažuriranja i dodavanja teniskih terena, proširenje na rezervaciju drugih sadržaja, samo su neke od mogućih budućih prilika koje se mogu iskoristiti za stjecanje konkurentske prednosti na tržištu. Najbitnija je funkcionalnost ovog rješenja što korisnicima pruža uvid u sve teniske terene jednoga grada, u ovom slučaju Splita, te im nudi mogućnost rezervacije na vrlo jednostavan način uz pregled slobodnih termina teniskih terena. Prilike za stjecanjem konkurentske prednosti na tržištu su neke od mogućih nadogradnji aplikacije, poput mogućnosti korištenja na području cijele Hrvatske, uvođenjem mogućnosti ažuriranja i dodavanja teniskih terena i slično. Jedna od slabosti ove mobilne aplikacije je što se može koristiti jedino na Android uređajima. Ta se slabost može pretvoriti i u snagu tako što se napravi nadogradnja rješenja i za druge vrste uređaja. Također, ovo rješenje može se

prilagoditi i nadograditi za rezervaciju nekih drugih sadržaja. Najveća prijetnja ovoj mobilnoj aplikaciji je pojava sve većeg broja mobilnih aplikacija za rezervaciju nekih usluga, ali to se može umanjiti nadogradnjom prethodno navedenih funkcionalnosti, kontinuiranim poboljšanjima i kvalitetnim održavanjem rješenja.



## Popis kratica

API	<i>Application Programming Interface</i>	aplikacijsko programsko sučelje
HTTP	<i>Hypertext Transfer Protocol</i>	protokol za prijenos hiperteksta
IDE	<i>Integrated development environment</i>	integrirano razvojno okruženje
JSON	<i>JavaScript Object Notation</i>	JavaScript objektni zapis
JVM	<i>Java Virtual Machine</i>	java virtualna mašina
REST	<i>Representational State Transfer</i>	prijenos prikazanog stanja
SQL	<i>Structured query language</i>	strukturirani upitni jezik
SWOT	<i>Strengths, Weaknesses, Opportunities, and Threats</i>	snage, slabosti, prilike i prijetnje
UI	<i>User Interface</i>	korisničko sučelje
URI	<i>Uniform Resource Identifier</i>	jedinstveni resursni identifikator
URL	<i>Uniform Resource Locator</i>	jedinstveni resursni lokator

## Popis slika

Slika 2.1 Grafički prikaz učestalosti korištenja teniskih terena.....	6
Slika 2.2 Grafički prikaz rezultata načina rezerviranja teniskih terena.....	7
Slika 2.3 Grafički prikaz rezultata okvirnog trajanja procesa rezervacije.....	7
Slika 2.4 Grafički prikaz rezultata učestalosti nedostupnosti sportskog centra putem telefona .....	8
Slika 2.5 Grafički prikaz rezultata učestalosti uspjeha rezervacije terena u jednom pozivu.	8
Slika 2.6 Grafički prikaz rezultata o potrebitosti mobilne aplikacije za rezervaciju teniskih terena .....	9
Slika 3.1 Prikaz arhitekture softverskog rješenja .....	13
Slika 3.2 Prikaz relacije 1-N na primjeru Court i CourtType tablica.....	17
Slika 3.3 ER dijagram baze podataka.....	18
Slika 4.1 Prikaz početnog ekrana s animacijom .....	23
Slika 4.2 Prikaz ekrana za prijavu korisnika .....	24
Slika 4.3 Prikaz ekrana za registraciju korisnika.....	25
Slika 4.4 Prikaz ekrana za pregled teniskih terena .....	26
Slika 4.5 Prikaz ekrana za pregled rezervacija odabranog terena .....	27
Slika 4.6 Prikaz ekrana s popisom sportskih centara.....	28
Slika 4.7 Prikaz ekrana s detaljima sportskog centra .....	29
Slika 4.8 Prikaz ekrana za rezervaciju teniskog terena .....	30
Slika 4.9 Prikaz ekrana s korisnikovim rezervacijama.....	31
Slika 4.10 Prikaz ekrana za uređivanje profila.....	32
Slika 5.1 Prikaz korisničkog sučelja u Android Studiju.....	33
Slika 5.2 Mocky – prikaz primjera izrade prilagođenog API-ja.....	35
Slika 5.3 Json2Kotlin – prikaz djela za generiranje Kotlin klasa.....	36

Slika 5.4 Json2Kotlin – Prikaz generirane klase za TennisPlace .....	36
Slika 5.5 Prikaz SQLite baze podataka .....	37

## Popis kôdova

Kôd 3.1 Prikaz <code>CourtType</code> klase.....	14
Kôd 3.2 Prikaz <code>Court</code> klase .....	14
Kôd 3.3 Prikaz metode za uređivanje <code>CourtType</code> entiteta .....	14
Kôd 3.4 Prikaz implementacije metode za uređivanje <code>CourtType</code> entiteta .....	15
Kôd 3.5 Prikaz varijable za izradu <code>CourtType</code> tablice.....	15
Kôd 3.6 Prikaz varijable za izradu <code>Court</code> tablice .....	16
Kôd 3.7 Prikaz dodavanja dozvole za pristup internetu u <code>AndroidManifest.xml</code> datoteci .....	19
Kôd 3.8 Prikaz spremanja slika u <code>ImageHandleru</code> .....	20
Kôd 3.9 Prikaz metode za dohvaćanje i spremanje podataka.....	21
Kôd 5.1 Prikaz dodavanja ovisnika za korištenje Retrofit-a u <code>build.gradle</code> datoteci	38
Kôd 5.2 Retrofit - prikaz definiranog protokola.....	38
Kôd 5.3 Retrofit – prikaz korištenja Retrofit <i>buildera</i> .....	38

## Literatura

- [1] Android Studio, Dostupno na: <https://developer.android.com/studio/intro>, 16.2.2023.
- [2] Što je android studio? - definicija iz tehopedije, Dostupno na: <https://hr.theastrologypage.com/android-studio>, 16.2.2023.
- [3] What is Web API and why we use it ?, Dostupno na: <https://www.geeksforgeeks.org/what-is-web-api-and-why-we-use-it/>, 17.2.2023.
- [4] D. Jemerov, S. Isakova; Kotlin In Action; Manning Publications; 2017.
- [5] Darryl Bayliss, Tom Blankenship, Fuad Kamal, Namrata Bandekar: ANDROID APPRENTICE: Beginning Android Development with Kotlin; 2019.
- [6] T. J. Teorey, S. S. Lightstone, T. Nadeau; Database modeling and design: Logical design; Morgan Kaufmann; 2005.
- [7] Json2Kotlin, Dostupno na: <https://www.json2kotlin.com/>, 16.2.2023.
- [8] Mocky, Dostupno na: <https://designer.mocky.io/faq>, 16.2.2023.
- [9] Dawn Griffiths, David Griffiths: Head First Kotlin: A Brain-Friendly Guide; O'Reilly Media, 2019.
- [10] SWOT Analysis: How To With Table and Example, Dostupno na: <https://www.investopedia.com/terms/s/swot.asp>, 1.2.2023.
- [11] SWOT analiza – što je, primjeri i predložak, Dostupno na: <https://acj-aims.hr/blog/kako-nam-swot-analiza-pomaze-u-strateskom-planiranju>, 1.2.2023.
- [12] Što je SWOT analiza i koje su smjernice potrebne za njezinu izradu?, Dostupno na: <https://www.markething.hr/sto-je-swot-analiza-i-koje-su-smjernice-potrebne-za-izradu/>, 1.2.2023.
- [13] The SWOT Analysis: Develop Strengths to Decrease the Weaknesses of Your Business: Book by 50Minutes.com
- [14] RETROFIT LIBRARY IN ANDROID, Dostupno na: <https://www.topcoder.com/thrive/articles/retrofit-library-in-android>, 15.2.2023.
- [15] SQLite, Dostupno na: <https://www.sqlite.org/about.html>, 15.2.2023.