

# Sustav za informiranje turista putem aplikacije pametnog TV-a

---

**Farina, Toni**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:640703>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2025-02-05**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Sveučilišni diplomski studij računarstva

Diplomski rad

**SUSTAV ZA INFORMIRANJE TURISTA  
PUTEM APLIKACIJE PAMETNOG TV-A**

Rijeka, rujan 2024.

Toni Farina  
0069078594

SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Sveučilišni diplomski studij računarstva

Diplomski rad

**SUSTAV ZA INFORMIRANJE TURISTA PUTEM  
APLIKACIJE PAMETNOG TV-A**

Mentor: izv. prof. dr. sc. Sandi Ljubić

Rijeka, rujan 2024.

Toni Farina  
0069078594

**Umjesto ove stranice umetnuti zadatak  
za završni ili diplomski rad**

## **Izjava o samostalnoj izradi rada**

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2024.

---

Toni Farina

## **Zahvala**

Najveću zahvalnost izražavam svojim roditeljima na pruženoj mogućnosti obrazovanja i podršci da završim fakultet.

Zahvaljujem isto svojim prijateljima Matiji, Mauru, Nikoli, Rencu, Toniju i Vedranu na pomoći i lijepim uspomenama tijekom svih godina studija.

Hvala izv. prof. dr. sc. Sandiju Ljubiću na mentoriranju, pokazanom strpljenju i razumijevanju pri izradi rada.

# Sadržaj

Popis slika.....	viii
Popis isječaka programskog koda .....	ix
<b>1. UVOD .....</b>	<b>1</b>
<b>2. PREGLED POSTOJEĆIH APLIKACIJA S TURISTIČKIM INFORMACIJAMA U FOKUSU.....</b>	<b>3</b>
<b>3. RAZVOJ SUSTAVA ZA INFORMIRANJE TURISTA PUTEM APLIKACIJE PAMETNOG TV-A.....</b>	<b>6</b>
<b>3.1. Arhitektura sustava.....</b>	<b>6</b>
<b>3.2. Tehnološki stog .....</b>	<b>8</b>
<b>3.3. Implementacija klijentske strane sustava .....</b>	<b>10</b>
3.3.1. Implementacija pametne TV aplikacije .....	10
3.3.2. Implementacija web aplikacije .....	13
<b>3.4. Implementacija poslužiteljske strane sustava.....</b>	<b>14</b>
3.4.1. Firebase Cloud Firestore .....	15
3.4.2. Firebase Storage.....	17
3.4.3. Integracija Android i web aplikacija s Firebaseom.....	18
3.4.4. Sigurnost i kontrola pristupa .....	20
<b>4. PREGLED SLUČAJEVA KORIŠTENJA .....</b>	<b>21</b>
<b>4.1. Slučajevi korištenja u pametnoj TV aplikaciji .....</b>	<b>21</b>
4.1.1. Postavljanje aplikacije za unajmljivače .....	22
4.1.2. Prikaz kategorija i stavaka .....	25
4.1.3. Prikaz detaljnih informacija o stavci.....	27
4.1.4. Prikaz i slanje recenzija.....	30
<b>4.2. Slučajevi korištenja u web aplikaciji .....</b>	<b>31</b>

4.2.1.	Promjena lozinke i brisanje korisničkog računa .....	32
4.2.2.	Dodavanje, uređivanje i brisanje nekretnina.....	33
4.2.3.	Dodavanje, uređivanje i brisanje kategorija.....	34
4.2.4.	Dodavanje, uređivanje i brisanje stavaka.....	35
4.2.5.	Blokiranje i brisanje iznajmljivača .....	36
4.2.6.	Pregled informacija korisnika i nekretnina .....	36
<b>5. ZAKLJUČAK.....</b>		<b>38</b>
<b>LITERATURA .....</b>		<b>39</b>
<b>POPIS OZNAKA I KRATICA .....</b>		<b>41</b>
<b>SAŽETAK.....</b>		<b>42</b>



## Popis slika

3.1.	Arhitektura sustava .....	7
3.2.	Zaglavlje s tekstom dobrodošlice i gumbovima .....	11
3.3.	Matrica sadržaja 1 x 1 .....	12
3.4.	Matrica sadržaja 1 x 3 .....	12
3.5.	Matrica sadržaja 2 x 3 .....	12
3.6.	Widget za prikupljanje i slanje slika na Firebase .....	14
3.7.	Relacije između Firestore kolekcija .....	16
3.8.	Primjer sadržaja direktorija unutar Firebase Storagea.....	18
4.1.	Dijagram navigacije po aktivnostima .....	22
4.2.	Aktivnost za prijavu .....	23
4.3.	Mogućnosti personalizacije na aktivnosti za prikaz nekretnina .....	24
4.4.	Traka za pretraživanje i gumb za navigaciju po stranicama.....	25
4.5.	Autentifikacija potrebna za povratak na aktivnost s prikazom smještajnih objekata.....	27
4.6.	Minimalni predložak za prikaz detaljnih informacija o stavci .....	28
4.7.	Prikaz dodatnih informacija ovisno o kontekstu (u prikazanom slučaju - događaj) .....	29
4.8.	Kompaktni predložak za prikaz detaljnih informacija o stavci .....	29
4.9.	Potpuni predložak za prikaz detaljnih informacija o stavci.....	30
4.10.	Vertikalna lista zadnjih 10 recenzija .....	31
4.11.	Sučelje za unos vlastite recenzije .....	31
4.12.	Dijagram navigacije po stranicama web aplikacije .....	32
4.13.	Prikaz padajućeg izbornika u web aplikaciji .....	33
4.14.	Kartica za prikaz i upravljanje kategorijom .....	35
4.15.	Prikaz iznajmljivača u web sučelju .....	36
4.16.	Prikaz registriranih smještajnih objekata na karti.....	37

## **Popis isječaka programskog koda**

3.1. Firebase config objekt .....	19
-----------------------------------	----

## 1. UVOD

Uvođenje digitalnih rješenja u ugostiteljstvu je značajno promijenilo način na koji se informacije dostavljaju gostima. Danas se u hotelima i kućama za odmor sve češće koriste pametni televizori, pogotovo oni opremljeni Android TV operacijskim sustavom [1]. Normalno, glavna ideja iza nabave pametnog televizora je zabava poput gledanja filmova i serija. Međutim, u ovom radu primarno se govori o drugom aspektu korištenja pametnog televizora: korištenje aplikacija za dohvaćanje informacija o smještajnom objektu i lokalitetima.

U radu je implementiran sustav za prikaz informacija vezanih za smještajne objekte i lokalne sadržaje koristeći Android operacijski sustav i pametni televizor. Cijeli sustav sastoji se od dvije aplikacije i jednog poslužitelja. Prva aplikacija (web aplikacija) je osmišljena kako bi se vlasnicima smještajnih objekata omogućilo učinkovito upravljanje i prikazivanje bitnih informacija o svojim nekretninama i obližnjim turističkim sadržajima. Ideja druge aplikacije je, korištenjem pametnog televizora, prijelaz iz tradicionalnih metoda prijenosa informacija prema gostima (brošure ili direktna komunikacija) na moderniji i digitalni način komunikacije, nudeći unajmljivačima jednostavan način pristupa korisnim informacijama o svom smještaju i obližnjim atrakcijama.

Potreba za ovom aplikacijom proizlazi iz izazova s kojima se suočavaju vlasnici smještajnih objekata. Za učinkovitu komunikaciju važnih pojedinosti o kućanskim uređajima, značajkama pametnog doma i lokalnim pogodnostima, tradicionalne metode poput brošura ili direktne komunikacije često ne pružaju pristupačnost i pogodnosti koje bi se gostima u današnje vrijeme moglo omogućiti. Tiskanje brošura uvijek nosi određene troškove, a vrlo brzo mogu i zastarjeti. Brošure su također ograničene svojom veličinom i time se ograničava količina informacija koje se mogu prikazati. S druge strane, kada se informacije komuniciraju verbalno, to zahtijeva od osoblja provođenje vremena u interakciji s gostima, koje bi mogli posvetiti izvršavanju drugih zadataka. Osim toga, pri korištenju verbalne komunikacije ljudi imaju tendenciju krivo protumačiti ili samo zaboraviti dane upute. Konačno, treba također uračunati vjerojatnost da će gost odmah moći postaviti pitanja. Nudeći digitalno rješenje, ovaj rad cilja na prevladavanje ograničenja, pružajući ažurirane i lako dostupne informacije koje poboljšavaju cjelokupno iskustvo gosta.

Primarni cilj ovog rada je modernizirati način dijeljenja informacija u turističkim smještajima, te premostiti tradicionalne izazove stvaranjem platforma prilagođene za iznajmljivače i unajmljivače. Korištenje Firebasea [2] (koji predstavlja poslužiteljsku stranu) za centralizirano upravljanje podacima osigurava da se ažuriranja lako izrađuju i dosljedno prikazuju na web i Android TV sučeljima. Ova aplikacija ne samo da pojednostavljuje isporuku informacija, već i poboljšava cjelokupno iskustvo gosta čineći ključne detalje lako dostupnima.

## 2. PREGLED POSTOJEĆIH APLIKACIJA S TURISTIČKIM INFORMACIJAMA U FOKUSU

Razvoj turizma u digitalnom dobu je jednim dijelom obilježen širokim prihvaćanjem mobilnih i web aplikacija, ali i pametnih TV aplikacija, koje su promijenile način na koji putnici pristupaju informacijama. Takve aplikacije su postale nezamjenjivi alati u usavršavanju turističkog iskustva nudeći metode za detaljno planiranje putovanja te prikaz bitnih informacija u prostori. S time, u ovom poglavlju pregledavaju se trenutna aplikacijska rješenja za pristup turističkim informacijama, kako bi se istaknule njihove prednosti, ograničenja i rastuća potreba za integriranim rješenjem.

Aplikacije za turističko informiranje kao što su TripAdvisor, Google Travel i Expedia postale su glavne za putnike koji traže sveobuhvatne upute i preporuke. Na primjer, TripAdvisor [3] se ističe velikom količinom recenzija, ocjena i fotografija koje su objavili drugi korisnici. TripAdvisor također pokriva široki raspon hotela, restorana, parkova, povijesnih mjesta te ostalih uslužnih objekata i atrakcija. Ova platforma omogućuje korisnicima da iskoriste kolektivna iskustva globalne zajednice, što olakšava donošenje informiranih odluka o putovanju.

Google Travel [4] također nudi mogućnosti pristupa sličnim informacijama, međutim, koristeći drugačiji pristup. Integracijom različitih Googleovih usluga kao što su Google Maps, Google Flights i Google Hotels, Google Travel stvara jednu kohezivnu platformu koja pomaže u planiranju putovanja, te koristi bitne značajke svakog navedenog web servisa. Osim toga, velika prednost Google Travela leži u sposobnosti pružanja personalizirane preporuke bazirane na korisničkim podacima i povijesti pretraživanja, što je čini vrlo prilagođenim alatom, te korisnik u kratko vrijeme uspije pronaći točno one informacije koje ga interesiraju.

Expedia [5] je još jedna ključna aplikacija u turizmu i planiranju putovanja. Poznata po svojoj sposobnosti prikupljanja podataka iz više izvora kako bi se korisnicima ponudilo što veći raspon opcija putovanja. Pod time, smatra se ponuda letova, smještaja i iznajmljivanje automobila. Njeno intuitivno sučelje i sveobuhvatne funkcionalnosti čine ju omiljenom aplikacijom za mnoge putnike.

Iako su iznimno uspješne i konkurentne u svom "teritoriju", svaka navedena aplikacija ima isti problem, a to je da su prvenstveno optimizirane za rad na mobilnim uređajima i računalima, što znači da je fokus stavljen na praktičnost izvršavanja zadataka, a ne na imerzivna iskustva. Ove aplikacije olakšavaju pronalaženje željenog smještajnog objekta, mjesta za jelo ili aktivnosti kojima se može

baviti, međutim, one nisu stvorene za maksimiziranje vizualnog doživljaja i angažiranja koji se može postići na većem zaslonu poput onog na pametnom TV-u. To je dovelo do razvoja pametnih TV aplikacija posebno implementiranih za sektor turizma i ugostiteljstva, čiji je cilj pružiti sveobuhvatnije i društveno angažiranije iskustvo.

Tijekom proučavanja različitih programskih rješenja za dostavu informacija na pametnom TV-u, alat Prezi se najviše istaknuo po svojoj sličnosti programskom rješenju ovoga rada. Iako su dostupna i druga programska rješenja, Prezi se istaknuo zbog svojih jedinstvenih značajki i popularnosti na tržištu.

Prezi [6] je programski alat za izradu prezentacija temeljen na oblaku koji nudi jedinstven pristup vizualnoj komunikaciji. Za razliku od tradicionalnih linearnih prezentacija, Prezi nudi mogućnost stvaranja strukturiranih, tj. nelinearnih prezentacija [7]. Dok linearne prezentacije imaju striktno definiran slijed slikokaza (engl. slide) koje će se prikazati, nelinearne omogućuju preskakanje i vraćanje na određene slikokaze prezentacije, ovisno o strukturi i mogućnostima navigiranja. Prezi je ostvario nelinearnost prezentacija tako da korisnicima omogućuje povećavanje i smanjivanje različitih dijelova svoje prezentacije dajući efekt navigiranja različitim slikokazima što čini prezentaciju daleko dinamičnijom i zanimljivijom. Ovaj alternativni pristup prezentacijama može pomoći u spajanju različitih koncepata ili tema na vizualno intrigantniji način.

Prezi podržava različite interaktivne elemente, poput videozapisa, slika i hiperveza, koji mogu poboljšati cjelokupno iskustvo prezentacije. Dodatno, Prezi nudi širok raspon prilagodljivih predložaka koji korisnicima pomažu u brzom i jednostavnom stvaranju prezentacija profesionalnog izgleda.

Prezi nudi još jednu ključnu značajku, a to je omogućavanje suradnje. Korisnicima omogućuje rad na prezentacijama s različitih uređaja u stvarnom vremenu što ga čini pogodnim alatom za timove ili tvrtke koje trebaju surađivati na prezentacijama.

Aplikacija za pametni TV implementirana u ovome radu, kao i alat Prezi, ciljaju na ostvarenje svog punog potencijala u isporučivanju sadržaja na vizualno privlačan i sveobuhvatan način. Obje aplikacije u potpunosti iskorištavaju veću veličinu zaslona i veću razlučivost koju nude pametni televizori. To ih čini posebno učinkovitima u turističkim smještajima, gdje gosti često dijele životne prostore i cijene zajedničko iskustvo gledanja.

Osim rješavanja ograničenja tradicionalnih metoda dostave informacija, kao što su brošure ili usmene upute, aplikacija za unajmljivače, također, smanjuje potrebu osoblja da provodi vrijeme prenoseći informacije kojima se lako može pristupiti putem TV-a. Sposobnost aplikacije da isporuči personalizirani sadržaj izravno na zaslon pametnog TV-a čini ju vrijednim alatom za poboljšanje iskustva gostiju, nudeći razinu pogodnosti i pristupačnosti koja je u skladu s očekivanjima današnjih turista.

Općenito, ovaj pregled dostupnih aplikacija s turističkim informacijama u fokusu naglašava potrebu za integriranijom i svestranijom aplikacijom za pametne televizore koja može zadovoljiti pojedinačne i zajedničke slučajeve uporabe u sektoru turizma i ugostiteljstva. Android TV aplikacija razvijena u ovom radu predstavlja napredak u ovom polju, nudeći sveobuhvatno rješenje koje se bavi ograničenjima postojećih aplikacija, a istovremeno pruža zadovoljavajuće iskustvo i jednostavnost korištenja.

### **3. RAZVOJ SUSTAVA ZA INFORMIRANJE TURISTA PUTEM APLIKACIJE PAMETNOG TV-A**

U ovom poglavlju raspravlja se o implementaciji ciljane aplikacije. Govori se o izvedbi nekih funkcionalnosti, arhitekturi cjelokupnog sustava, korištenim tehnologijama, te preprekama koje je bilo potrebno savladati.

#### **3.1. Arhitektura sustava**

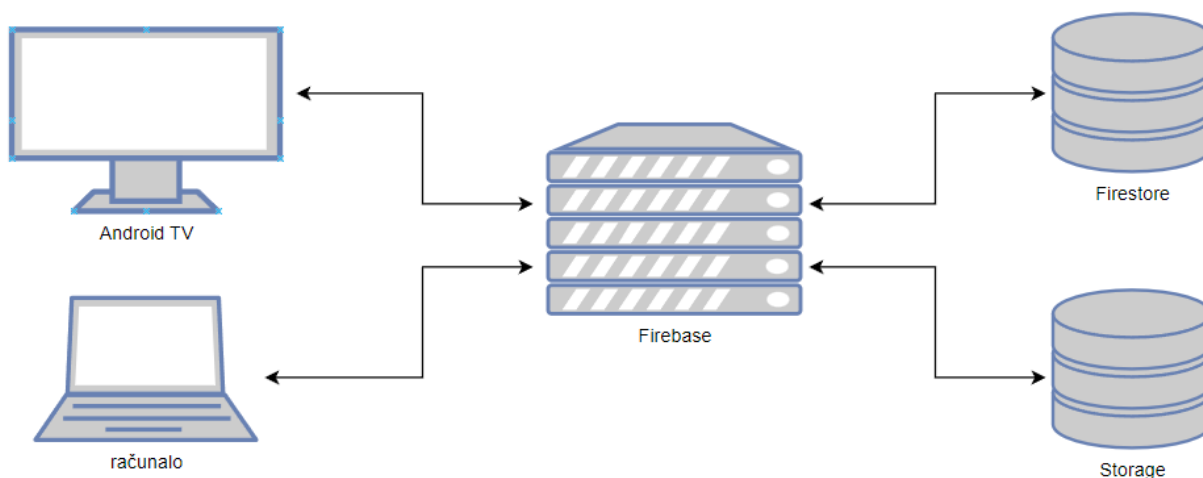
Arhitektura sustava odnosi se na konceptualni model koji definira strukturu, ponašanje i druge poglede na sustav. To se omogućuje definiranjem različitih komponenti, njihovih odnosa i načina na koji međusobno djeluju kako bi ispunili zahtjeve sustava.

Arhitektura ovog rada temelji se na tri dijela: aplikacija za iznajmljivače, aplikacija za unajmljivače i Googleov Firebase, kao što je prikazano na slici 3.1. Aplikacije za iznajmljivače i unajmljivače čine klijentsku stranu sustava. Klijentska strana odnosi se na sve što se događa na korisnikovom uređaju (kao što je računalo, pametni telefon ili tablet) kada komunicira sa sustavom.

S druge strane sustava nalazi se Firebase koji čini poslužiteljsku stranu sustava. Poslužitelj prima i obrađuje zahtjeve klijenata, izvodi potrebne radnje kao što su upiti prema bazama podataka, provjera autentičnosti ili složena izračunavanja, te šalje odgovarajući povratni odgovor.

Firestore platforma je odabrana zbog svoje svestranosti i mogućnosti integracije s različitim rješenjima, posebno u mobilnom i web razvoju. Firestoreova gotova rješenja za jednostavnu integraciju aplikacijsko-programskog sučelja (API-a) [8], baze podataka [9] u stvarnom vremenu i pohranu u oblaku učinila su ga atraktivnim izborom za ovaj rad.





*Slika 3.1. Arhitektura sustava*

Kada aplikacijsko-programsko sučelje komunicira s Firestoreom, aplikacija šalje zahtjev prema poslužitelju koristeći HTTP protokol (engl. Hypertext Transfer Protocol). Zahtjev navodi radnju koja se želi izvršiti (npr. stvaranje, čitanje, ažuriranje ili brisanje) i po potrebi uključene podatke. Nakon što Firestore primi zahtjev, provjeravaju se vjerodajnice za autentifikaciju te ima li aplikacija potrebna dopuštenja za izvođenje radnje na temelju definiranih sigurnosnih pravila. Ako je zahtjev autoriziran, Firestore ga obrađuje i sukladno tome ažurira bazu podataka. Baza podataka zatim šalje odgovor natrag aplikaciji, koji sadrži rezultate operacije. Ovaj odgovor može uključivati ažurirane ili novostvorene podatke. Aplikacija tada može koristiti te podatke za ažuriranje korisničkog sučelja ili izvođenje drugih radnji.

U aplikaciji za iznajmljivače (Flutter [10] i Dart [11] aplikacija), Firebase se prvenstveno koristi za autentifikaciju korisnika, upravljanje bazom podataka i sinkronizaciju podataka. Cloud Firestore [12] koristi se kao primarna baza podataka, pohranjujući korisničke podatke, podatke o njihovim turističkim smještajima i druge relevantne informacije. Aplikacija također koristi Firebaseovu značajku ažuriranja aplikacije u stvarnom vremenu radi omogućavanja odražavanja svih promjena u bazi podataka u istom trenutku na svim uređajima.

U aplikaciji za unajmljivače (Android TV aplikacija), Firebase se koristi za dohvaćanje i prikaz podataka koji se odnose na korisnikov odabrani smještajni objekt. Ova aplikacija primarno komunicira s Firebaseom kako bi pristupila informacijama o nekretninama pohranjenim u Cloud Firestoreu i Firebase Storageu [13], omogućujući unajmljivaču pregled pojedinosti vezane za

unajmljeni smještajni objekt i njegovu okolicu. Firebase osigurava da su podaci prikazani u Android TV aplikaciji ažurni i dosljedni informacijama pohranjenim putem aplikacije za iznajmljivače.

Općenito, Firebase igra ključnu ulogu u obje aplikacije pružajući pouzdano i skalabilno pozadinsko rješenje koje upravlja autentifikacijom i pohranom podataka i slika, osiguravajući glatko i dosljedno iskustvo na svim uređajima.

### **3.2. Tehnološki stog**

Odabir odgovarajućeg tehnološkog stoga bio je ključan za postizanje željenih funkcionalnosti. Odabrani alati i jezici učinili su proces razvoja značajno učinkovitijim i osigurali da konačna aplikacija ispunjava specifične zahtjeve zadatka. Ovo potpoglavlje pruža pregled odabranih tehnologija korištenih u aplikacijama za iznajmljivače i unajmljivače, ističući razloge iza njihova odabira i specifične uloge koje su imale u radu.

Integrirano razvojno okruženje (engl. Integrated development environment, IDE) [14] je aplikacija koja pomaže programerima u razvoju programskog koda. Cilj integrirano razvojnih okruženja je povećanje produktivnosti programera usklađujući značajke kao što su uređivanje programa, njegova izgradnja i testiranje, isticanje i otklanjanje pogrešaka, označavanje sintakse, inteligentno dovršavanje koda, automatizacija izgradnje aplikacije, itd.

Android Studio [15] je integrirano razvojno okruženje projektirano posebno za razvoj aplikacija za Android operacijski sustava. Kao primarna platforma za razvoj Android aplikacija, to je bio očiti izbor za implementaciju aplikacije za unajmljivače zbog opsežnog skupa alata i emulatora prilagođenih Android uređajima. Unatoč tome što je najbolji izbor za razvoj Android aplikacija, Android Studio povremeno zna pokazati svoje nedostatke. Poznato je da Android Studio ima neke svoje greške koje povremeno uzrokuju "zastajkivanje" tijekom korištenja. „Television (Android S) (4K) API 31“ je odabran kao glavni virtualni uređaj kako bi se aplikacija testirala u okruženju koje je vrlo slično ciljnim uređajima. Osim toga, program za ispravljanje pogrešaka ponekad je preskakao prijelomne točke, što je otežavalo praćenje stanja varijabli i cjelokupno izvršenja koda. Zadnji se problem mogao izbjeći ispisivanjem sadržaja varijabli u Logcatu. To je alat koji prikazuje zapisnik sistemskih poruka uključujući poruke koje je sam programer ispisao koristeći svoju aplikaciju. Međutim, ovo rješenje je daleko od idealnog, pogotovo jer bi dodavalo nepotrebnu složenost procesu

otklanjanja pogrešaka.

Java [16] je sekundarni programski jezik (Kotlin je primarni jezik od 2019. godine) koji se koristi za implementaciju temeljne funkcionalnosti Android aplikacija. U kontekstu aplikacije za unajmljivače, Java se koristi za nekoliko ključnih aspekata kao što su upravljanje aktivnostima, rukovanje događajima i iznimkama, upravljanje podacima, integracija s Firebaseom i mnogo drugih. U ovoj aplikaciji Java se doista pokazala kao dobar izbor u odnosu na Kotlin zbog svoje široke primjene, te opsežne i dostupne dokumentacije, što je olakšalo pronalaženje resursa i podrške tijekom razvoja.

XML [17] (engl. eXtensible Markup Language) se prvenstveno koristi za definiranje izgleda i oblikovanje korisničkog sučelja Android aplikacije. U aplikaciji za unajmljivače XML se koristi u nekoliko ključnih područja poput definiranje izgleda, upravljanje resursima, stilovima i temama te konfiguracijama i postavkama.

Korištenjem Jave za implementaciju funkcionalnosti i XML-a za oblikovanje korisničkih sučelja, proces razvoja aplikacije za unajmljivače znatno je pojednostavljen jer se tako odvajaju problemi vizualne prezentacije i logike i funkcionalnosti aplikacije. Ovaj pristup poboljšava organizaciju koda, a istovremeno olakšava razvoj korisniku vizualno privlačne aplikacije.

S druge strane, Visual Studio Code [18] (poznatiji pod kraticom "VS Code") je odabran kao glavno integrirano razvojno okruženje za razvoj aplikacije za iznajmljivače. Odabran je prvenstveno zbog svoje jednostavne integracije s Flutterom i Dartom. Također VS Code uključuje mnoge druge značajke poput isticanja sintakse, podršku za ispravljanje pogrešaka i refaktoriranje koda, inteligentno dovršavanje koda, preuzimanje ekstenzija i kontrolu verzije s Gitom (više o Gitu kasnije).

Tijekom razvoja aplikacije za iznajmljivače koristili su se Flutter i Dart programski jezici. Flutter je glavni alat za izradu korisničkog sučelja aplikacije. Poznat je po svom bogatom skupu unaprijed oblikovanih widgeta i svojim mogućnostima izrade aplikacije za više platformi. Dart, s druge strane, pruža temeljnu logiku i strukturu potrebnu za implementaciju značajki aplikacije. Dart se može smatrati kao programski jezik koji radi u pozadini kako bi osigurao ispravnu funkcionalnost aplikacije.

Git [19] je besplatni raspodijeljeni sustav otvorenog koda projektiran za praćenje verzija datoteka. Uglavnom se koristi za kontrolu izvornog koda od strane programera koji zajednički razvijaju programska rješenja. Ciljevi Gita uključuju integritet podataka, brzinu i podršku za distribuirane, nelinearne tijekove rada s tisućama paralelnih grana (engl. branch) koje se izvode na različitim računalima. Iako je Git korišten kao sustav kontrole verzija za rad, prvenstveno je korišten kao rezervno rješenje za pohranjivanje koda.

### **3.3. Implementacija klijentske strane sustava**

U ovom potpoglavlju obrađena je implementacija klijentske strane sustava, koja uključuje izradu aplikacija namijenjenih iznajmljivačima i unajmljivačima. Prikazani su ključni koraci u razvoju korisničkog sučelja i logike aplikacija. Osim toga, detaljno su opisane tehničke odluke i izazovi koji su bili neizbježni tijekom implementacije, kao i rješenja koja su omogućila da aplikacije budu funkcionalne, intuitivne i jednostavne za korištenje.

#### **3.3.1. Implementacija pametne TV aplikacije**

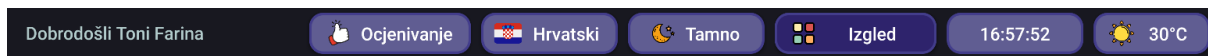
Aplikacija za unajmljivače razvijena je i testirana na Android emulatoru pomoću virtualnog uređaja Television (4K) (Android S) API 31. Iako je razvoj bio ograničen na emulator, uloženi su napor i trud da se optimizira izvedba aplikacije, osiguravajući nesmetan rad unutar ograničenja emulatora.

Aplikacija za unajmljivače razvijena je za rad na pametnom televizoru koji pokreće Android operacijski sustav, a namijenjena je turistima koji borave u smještajnim objektima. Cilj je bio stvoriti intuitivno i jednostavno sučelje koje će turistima omogućiti lak pristup informacijama o lokalnim atrakcijama, događanjima, uslugama i sl. Posebna je pozornost posvećena prilagodbi navigacije daljinskim upravljačem. Za razliku od tradicionalnih aplikacija, TV aplikacija zahtijevala je navigacijski sustav temeljen na komponentama sučelja. Komponentama sučelja smatraju se dijelovi kojima korisnik aplikacije može upravljati, kao što su gumbi, polja za unos teksta, dijaprojeksije i sl. Takva vrsta upravljanja izvedena je koristeći navigacijske matrice. Aplikacija ima 22 navigacijske matrice, te svaka opisuje mogućnosti kretanja po aktivnosti ili prozoru, ovisno o određenom stanju

aplikacije. Redovi navigacijske matrice predstavljaju trenutnu komponentu koja je u fokusu, dok stupci predstavljaju 4 smjera kretanja (gore, dolje, lijevo i desno). Svaka ćelija sadrži identifikacijski broj sljedeće komponente koja bi trebala biti u fokusu. U slučaju da je vrijednost identifikacijskog broja  $\emptyset$ , to znači da daljnje kretanje u željenom smjeru nije moguće, te fokus ostaje na trenutno aktivnoj komponenti. Svaka fokusirana komponenta ističe se od ostalih različitom bojom pozadine (tamnija). Time se korisnicima aplikacije olakšava predviđanje daljnjeg kretanja.

Na vrhu korisničkog sučelja nalaze se gumbovi koji omogućuju korisniku personaliziranje aplikacije kao što je prikazano na slici 3.2. Korisnik može:

- Odabrati željeni jezik između troje ponuđenih: engleski, njemački i hrvatski;
- Odabrati između svijetle i tamne teme;
- Postaviti prikaz vremena na 12-satni ili 24-satni format;
- Odrediti matricu prikaza nekretnina, kategorija i stavki (entiteta).

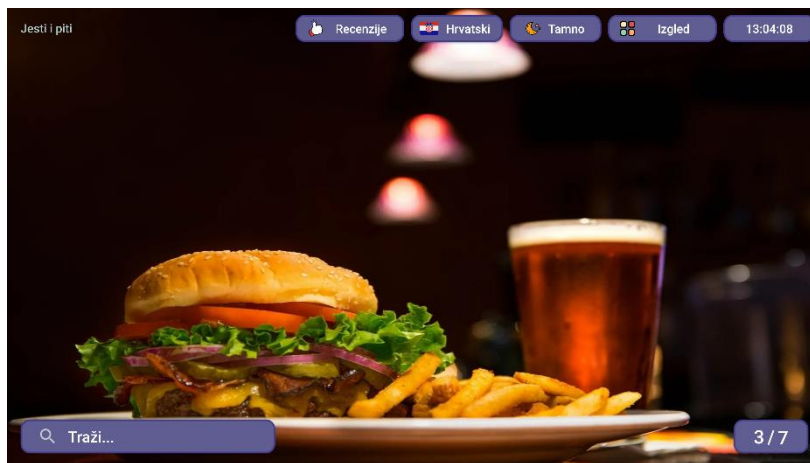


*Slika 3.2. Zaglavlje s tekstom dobrodošlice i gumbovima*

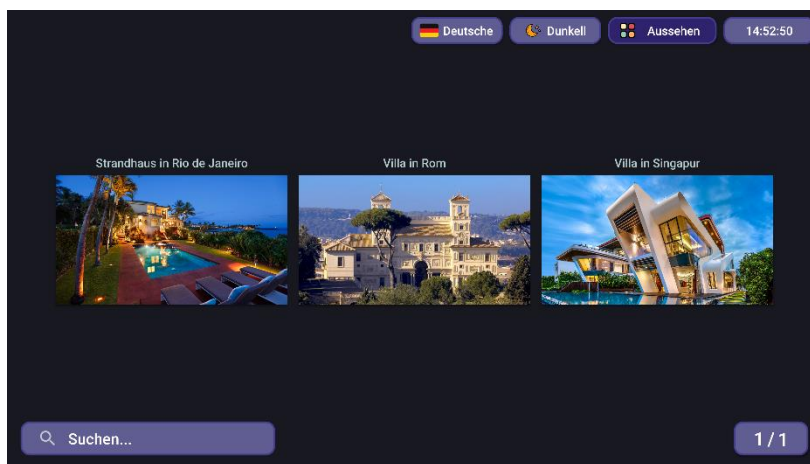
Kao što je navedeno, korisniku je omogućeno da sam odredi kako će mu se podatci prikazivati. Postoje tri matrice za izbor vizualizacijskog formata:

- $2 \times 3$ : Prikazuje šest entiteta u matrici s dva retka i tri stupca;
- $1 \times 3$ : Tri entiteta u matrici s jednim redom i tri stupca;
- $1 \times 1$ : Samo jedan entitet u matrici.

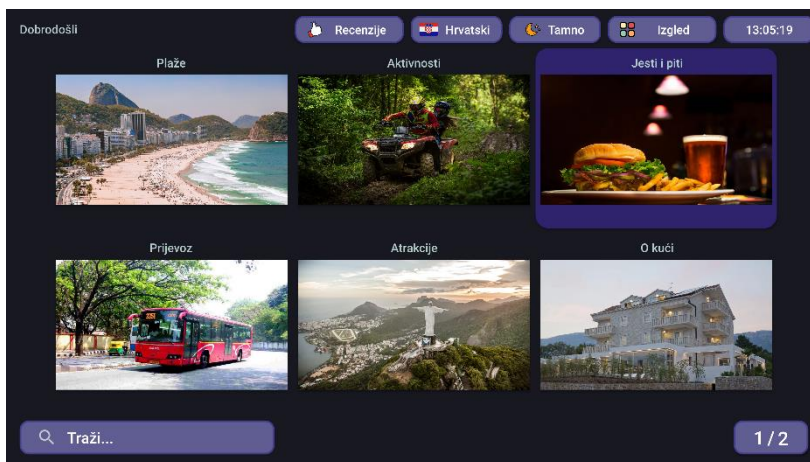
Razlog za implementaciju matrica je da se korisniku aplikacije omogući opušteno listanje i pregledavanje svih navedenih entiteta kako bi se mogao upoznati s opcijama vezanim za svoj smještajnim objekt i/ili okolicu. To omogućuje matrica  $1 \times 1$  (slika 3.3.). Drugi razlog je brzo i efikasno pretraživanje, što omogućuju matrice  $1 \times 3$  (slika 3.4.) i  $2 \times 3$  (slika 3.5.). Uz to, svaka aktivnost ima traku za pretraživanje i gumb za preskakanje stranica. Ove dvije značajke mogu drastično ubrzati pretraživanje entiteta, bez obzira koja se matrica za vizualizaciju koristila.



Slika 3.3. Matrica sadržaja 1 x 1



Slika 3.4. Matrica sadržaja 1 x 3



Slika 3.5. Matrica sadržaja 2 x 3

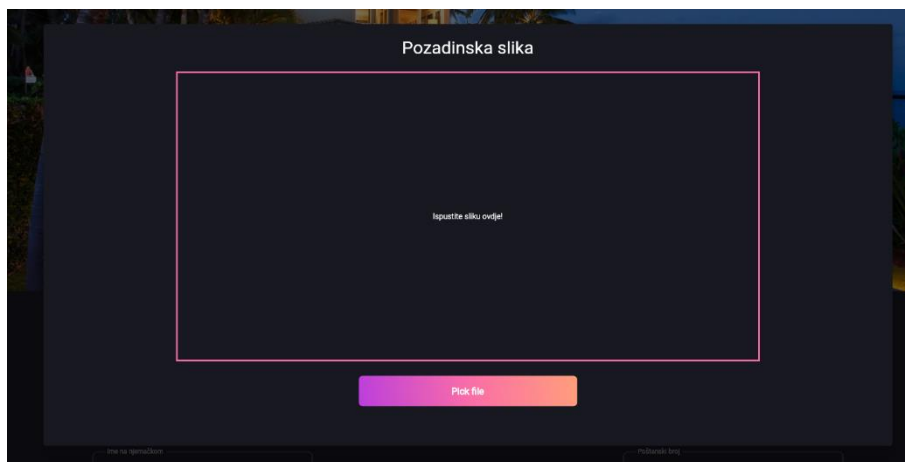
### 3.3.2. Implementacija web aplikacije

Aplikacija za iznajmljivače razvijena je koristeći Visual Studio Code i testirana u Google Chrome pregledniku. Aplikacija se pokreće u terminalu naredbom “flutter run -d chrome”.

Sučelje web aplikacije je implementirano s fokusom na funkcionalnost. Jedna od značajki koja se koristi u aplikaciji je prikaz smještajnih objekata na karti. Iako se nekretnine mogu prikazivati u vertikalnoj listi, prikaz istih na karti predstavlja zanimljivu alternativu. Preuzimanjem paketa "flutter\_map" [20] i korištenjem widgeta "FlutterMap" dodana je karta svijeta, te svaki smještajni objekt, koji ima postavljene koordinate, ima i odgovarajući prikaz na karti. Bitno je napomenuti da korisnici aplikacije ne mogu direktno zapisati koordinate smještajnog objekta, već se moraju ispuniti polja koja zahtijevaju adresu, poštanski broj, grad i državu navedenog objekta. Nakon pritiska na potvrdni gumb, pokreće se proces geokodiranja [21] (engl. geocoding) što pretvara predane vrijednosti adrese smještajnog objekta u fizičke koordinate istoga. Te koordinate se spremaju u bazu podataka na Firebaseu, te se ubuduće koriste za prikaz lokacije nekretnine na karti. Integracijom karte može se znatno ubrzati potragu za željenom nekretninom.

Aplikacija za iznajmljivače projektirana je sa specifičnim administrativnim funkcijama koje omogućuju administratoru učinkovito upravljanje korisnicima. Za razliku od iznajmljivača, administrator ima pristup naprednim značajkama koje pružaju sveobuhvatan pregled podataka sustava i kontrolu nad korisnicima. Administrator ima pristup stranici koja mu omogućava pregled svih korisničkih računa iznajmljivača. Osim toga, administrator može sortirati prema tekstualnim vrijednostima unutar objekta koji predstavlja iznajmljivača, poput elektroničke pošte, imena i prezimena te telefonskog broja. Moguće je filtriranje prema istim tekstualnim vrijednostima, statusu blokiranoosti i zabranjenosti, vrsti iznajmljivača (individualna osoba ili tvrtka) i količini smještajnih objekata. Administrator može pristupiti privatnim informacijama iznajmljivača, njegovim nekretninama, kategorijama i stavkama. Bitno je napomenuti da, iako administrator može pregledavati sve tuđe informacije, administrator ne može dodavati, uređivati ili brisati tuđe podatke. Jedina dužnost administratora je privremeno blokirati korisnika ili trajno zabraniti pristup aplikaciji. Jednom kada se korisniku zabrani pristup aplikaciji, svi podatci vezani uz njegov korisnički račun brišu se osim tri vrijednosti: elektronička pošta iznajmljivača, te zastave blokiranoosti i zabranjenosti. Time se sprječava daljnja prijava na isti korisnički račun, a u isto vrijeme i stvaranje novog koji ima istu elektroničku poštu.

Zadnja značajka, koja je nedvojbeno bila i najteža za implementiranje, omogućava prikupljanje lokalnih slikovnih datoteka s elektroničkih uređaja koji pokreću aplikaciju i spremanje na Firebase (prikazano na slici 3.6.). Slike koje se spremaju u oblak mogu predstavljati korisnike aplikacije, njihove nekretnine, kategorije i stavke. Inicijalni dio zadatka bio je omogućiti preuzimanje lokalnih slika i privremeno spremanje u aplikaciji. Osim tradicionalnog biranja datoteka koje se prikazuju u malom prozoru, korisniku je, također, omogućeno slanje datoteka koristeći metodu "povuci i ispusti" (engl. "drag-and-drop").



*Slika 3.6. Widget za prikupljanje i slanje slika na Firebase*

### **3.4. Implementacija poslužiteljske strane sustava**

Ovo potpoglavlje obrađuje implementaciju i konfiguraciju poslužiteljske strane sustava, tj. Firebasea. Firebase je Googleova platforma bazirana na "oblaku", a nudi niz alata za izradu i upravljanje web i mobilnim aplikacijama. Firebase omogućava korištenje mnogih značajki kako bi se pomoglo programerima u što bržem i efikasnijem razvoju aplikacija. Od svih značajki koje nudi Firebase, u ovom će se potpoglavlju obraditi dvije: Cloud Firestore i Storage. Također, opisan će se ključni koraci u postavljanju Firebasea i organiziranju podataka. Osim toga, opisan će se tehničke odluke i izazovi koji su bili neizbježni tijekom implementacije, kao i rješenja koja su omogućila da poslužitelj bude funkcionalan i jednostavan za korištenje.



### 3.4.1. Firebase Cloud Firestore

Firestore je baza podataka temeljena na NoSQL modelu, što znači da je projektirana za prilagodljivo pohranjivanje i upravljanje nestrukturiranim podacima. NoSQL baze podataka [22] često koriste modele poput ključa i vrijednosti, grafova ili stupaca, ali Firestore koristi model dokumenata. Takve baze podataka omogućuju horizontalno skaliranje i visoku dostupnost, čime se lakše prilagođavaju velikim količinama podataka i visokom prometu. Dokumenti unutar Firestorea organizirani su u nekoliko kolekcija [23], što znači da su različite vrste dokumenata grupirane u logičke cjeline, što je neophodno za pravilno funkcioniranje sustava. Firestore, također, podržava stvaranje potkolekcija unutar pojedinačnih dokumenata, što dodatno pojednostavljuje modeliranje složenih odnosa između podataka. Ova organizacija podataka omogućava preglednu i intuitivnu navigaciju kroz podatke, olakšavajući razvoj i održavanje aplikacije.

Jedna od ključnih prednosti Firestorea je njegova sposobnost sinkronizacije podataka u stvarnom vremenu, što znači da se promjene u bazi podataka odmah reflektiraju na klijentskim aplikacijama. To je posebno korisno u slučajevima gdje je potrebna trenutna reakcija na promjene. Ova funkcionalnost se koristila u Flutter aplikaciji za iznajmljivače i pokazala se praktičnom ne samo zbog toga što uvijek garantira korisniku prikaz najnovijih podataka, već je implementacija takve značajke bila jednostavna. Firestore, također, podržava napredne upite, omogućujući filtriranje i sortiranje podataka na način koji odgovara specifičnim potrebama aplikacije. Zahvaljujući tome, obje aplikacije mogu učinkovito dohvaćati samo one podatke koji su im potrebne, čime se smanjuje opterećenje mreže i ubrzava rad aplikacija.

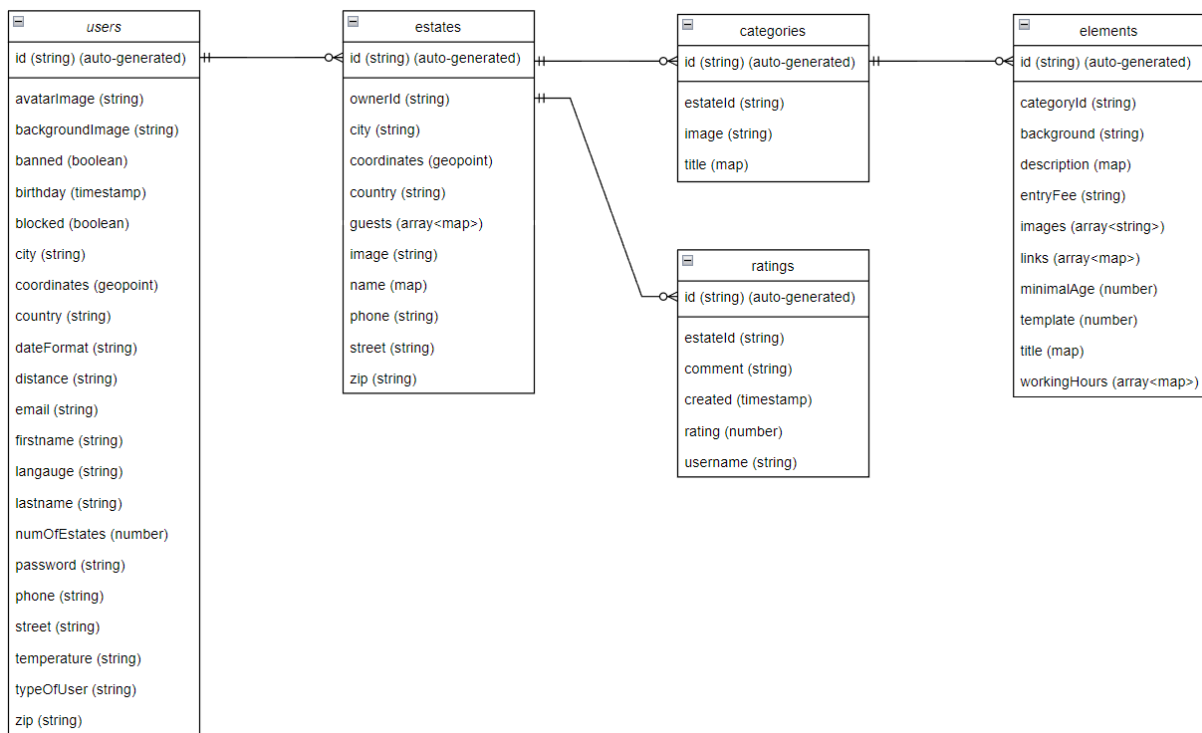
U ovom radu, Firestore je postavljen tako da koristi pet kolekcija, pri čemu svaka kolekcija predstavlja različiti skup entiteta. Bitno je napomenuti da svaki nadređeni entitet ima relaciju [24] s podređenim u obliku 1:n, što znači da bilo koji nadređeni entitet smije imati proizvoljan broj podređenih, dok podređeni entiteti smiju imati samo jedan nadređeni (prikazano na slici 3.7.). U Firestoreu su postavljene sljedeće kolekcije:

- korisnici (engl. "users");
- nekretnine (engl. "estates");
- kategorije (engl. "categories");

- stavke (engl. "elements");
- recenzije (engl. "ratings").

Kolekcija korisnika sadrži sve bitne informacije o korisnicima sustava, uključujući iznajmljivače i administratora (postoji samo jedan administratorski korisnički račun). Svaki dokument unutar ove kolekcije sadrži jedinstvene podatke, prikazane u tablici, te mu se automatski dodjeljuje jedinstveni identifikator. Ovaj pristup omogućuje jednostavno praćenje i upravljanje korisnicima te lako povezivanje korisnika s odgovarajućim dokumentima unutar kolekcije nekretnina.

Slika 3.7. Relacije između Firestore kolekcija



Kolekcija nekretnina pohranjuje sve relevantne informacije o nekretninama koje iznajmljivači unose u sustav. Svaki dokument unutar ove kolekcije predstavlja pojedinačnu nekretninu i sadrži detalje navedene na slici 3.7. Ova kolekcija igra ključnu ulogu u organizaciji podataka, omogućujući izravnu povezanost s kategorijama, te posrednu povezanost sa stavkama.

U kolekciji kategorija pohranjuju se dokumenti koji predstavljaju različite kategorije, koje služe za grupiranje i organiziranje stavaka u logičke cjeline. Svaka kategorija sadrži informacije prikazane na slici 3.7. Ovaj koncept omogućava korisnicima brzu i jednostavnu navigaciju kroz različite aspekte

smještajnih objekata, olakšavajući pregled i pretraživanje relevantnih informacija.

Kolekcija stavaka pohranjuje pojedinačne elemente koji su povezani sa smještajnim kategorijama. Svaki dokument unutar ove kolekcije sadrži informacije navedene na slici 3.7. Stavke su ključne za pružanje detaljnih informacija o smještajnim objektima i njihovim značajkama, kao i o lokalnim uslugama, atrakcijama, itd. Organizacija stavaka unutar Firestorea omogućuje lako dohvaćanje i prikazivanje podataka unutar aplikacija.

Kolekcija recenzija sadržava dokumente koji predstavljaju svaku recenziju koju je gost ostavio za smještajni objekt. Svaki dokument sadrži informacije prikazane na slici 3.7. Ova kolekcija omogućuje transparentno i učinkovito upravljanje povratnim informacijama korisnika, čime se olakšava praćenje i analiza iskustava gostiju.

Firestoreova struktura koja se temelji na dokumentima omogućava ne samo jednostavno dohvaćanje podataka već i fleksibilnost u načinu na koji se podatci mogu organizirati i koristiti. Svaka kolekcija sadrži dokumente, a odnosi između dokumenata uspostavljaju se pomoću jedinstvenih identifikatora. Ova vrsta organizacije podržava različite scenarije upotrebe i omogućava skalabilnost sustava sukladno povećanju broja korisnika i nekretnina.

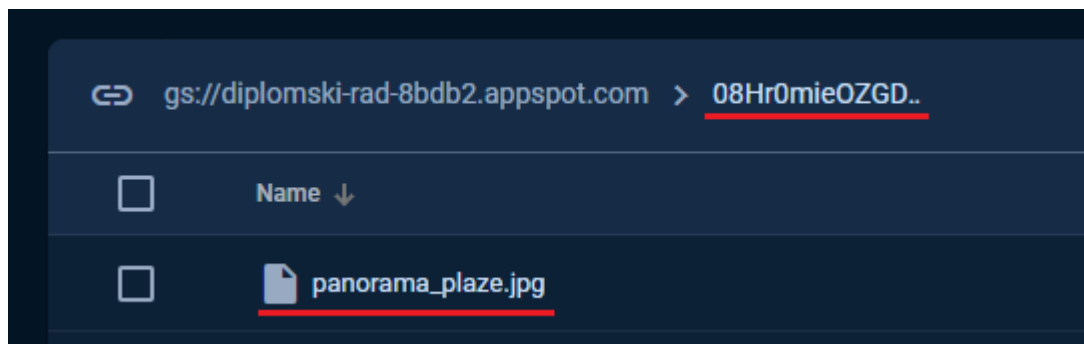
Uz to, Firestore nudi mogućnost indeksiranja podataka [25]. Indeksiranje omogućava brže pretraživanje i filtriranje informacija unutar velikih kolekcija podataka. Međutim, indeksi nisu dodani u Firestore samo kako bi ubrzali procese unutar baza podataka, već ako indeks nije prethodno stvoren za postavljeni upit, on se neće izvršiti. U tom slučaju, zahtjev prema Firestoreu će vratiti grešku i povratnu informaciju u kojoj je zadana poveznica. Ta poveznica navodi korisnika direktno na Firestore kako bi potvrdio generiranje već spremnog upita za stvaranje novog indeksa (uz uvjet da korisnik ima takve ovlasti). Ova funkcionalnost ne samo da dodatno poboljšava performanse sustava, već i poboljšava njegovu sigurnost osiguravajući da se neće pokretati upiti koje programer nije dozvolio.

### 3.4.2. Firebase Storage

Firebase Storage se u ovom radu koristi za pohranu slikovnih dokumenata, a poveznice su u Firestoreu spremljene tako da svaki dokument kojemu pripada slika ima u sebi zapisanu direktnu poveznicu na istu sliku. Slikovne datoteke su organizirane u direktorije, a direktoriji su povezani s

odgovarajućim dokumentima u Firestoreu tako da identifikator dokumenta odgovara nazivu direktorija. Tako se osigurava uredna i logična struktura podataka. Firebase Storage, također, pruža visoku skalabilnost, što omogućuje jednostavno upravljanje velikim količinama datoteka bez gubitka performansi.

Bilo koja od razvijenih aplikacija može jednostavno pristupiti slici koristeći identifikator dokumenta i korištenjem određenog polja unutar dokumenta s tekstualnim zapisom poveznice. Kao primjer, može se navesti slika "panorama\_plaze.jpg" koja se nalazi u direktoriju "08Hr0mieOZGDYBjkRik1". Preko aplikacijsko-programskog sučelja, aplikacija može poslati zahtjev za slikom tako da definira točnu lokaciju slike unutar Storagea u obliku "/". Koristeći vrijednosti definirane u primjeru, lokacija slike izgledat će kao "08Hr0mieOZGDYBjkRik1/panorama\_plaze.jpg" (slika 3.8.). Na ovaj način Firebase prepoznaje u kojem direktoriju će pronaći sliku, koju na kraju prosljeđuje aplikaciji.



*Slika 3.8. Primjer sadržaja direktorija unutar Firebase Storagea*

### 3.4.3. Integracija Android i web aplikacija s Firebaseom

Kada je riječ o integraciji aplikacija s Firebaseom, ključna komponenta ovog procesa je uspostavljanje ispravnog aplikacijsko-programskog sučelja. To zahtijeva pravilno postavljanje svih potrebnih konfiguracijskih datoteka kako bi se omogućila uspješna povezanost između aplikacija i Firebase platforme. U ovom radu, bilo je potrebno integrirati dvije vrste aplikacija: Android aplikaciju i web aplikaciju razvijenu u Flutteru. Svaka od ovih aplikacija zahtijeva specifične korake u postavljanju konfiguracije koje rezultira pravilnim povezivanjem s Firebaseom.

Za Android aplikaciju potrebno je preuzeti konfiguracijsku datoteku pod nazivom „google-services.json“. Ova datoteka sadrži ključne vjerodajnice koje omogućuju aplikaciji da se prvo predstavi Firebaseu i naknadno komunicira s njim i zahtijeva razne usluge. Datoteku je potrebno dodati u odgovarajući direktorij unutar aplikacije (u ovom slučaju u app/ direktorij) kako bi se omogućila integracija i sinkronizacija podataka s Firebaseom. Ovaj korak osigurava da Android aplikacija može koristiti Firebaseove značajke. Dodatno, trebaju se preuzeti i instalirati paketi za korištenje Firebasovih usluga.

Za web aplikaciju se koristi firebaseConfig objekt koji se generira prilikom stvaranja nove web aplikacije u Firebase projektu (prikazano na ispisu 3.1.). Podaci iz firebaseConfig objekta trebaju se uključiti u programski kod web aplikacije kako bi se omogućila uspješna povezanost s Firebaseom. Nakon uspješne konfiguracije, potrebno je bilo preuzeti i instalirati dodatne Flutter pakete kako bi se mogle koristiti prethodno definirane Firebaseove značajke. Paket firebase\_core [26] služi za inicijalizaciju Firebaseovog kompleta za razvoj programa (engl. software development kit, SDK) omogućavajući povezivanje s Firebaseovim uslugama. Paketi cloud\_firestore i firebase\_storage [27] nadovezuju se na firebase\_core, pružajući pristup Firestore bazi podataka i Firebase Storageu za rad s pohranom podataka i datoteka.

---

```
const firebaseConfig = {
  apiKey: "AIzaSyCr1D00mTiEk3G7G31ir462s61f9kiZqEQ",
  authDomain: "diplomski-rad-8bdb2.firebaseio.com",
  databaseURL: "https://diplomski-rad-8bdb2-default-
  rtdb.europe-west1.firebaseio.com",
  projectId: "diplomski-rad-8bdb2",
  storageBucket: "diplomski-rad-8bdb2.appspot.com",
  messagingSenderId: "236389351601",
  appId: "1:236389351601:web:66c1f13569d93cc056a290",
  measurementId: "G-4KSSRHL1KF"
};
```

---

*Ispis 3.1. Firebase config objekt*

Time se zaokružuje cijeli proces povezivanja aplikacija na Firebase koristeći Firebaseove API-je. Važno je napomenuti da je cijeli proces integracije detaljno dokumentiran na službenoj Firebase web stranici. Svi koraci potrebni za preuzimanje i konfiguriranje ovih datoteka jasno su opisani, što znatno olakšava implementaciju i pomaže u izbjegavanju grešaka. Obje dokumentacije (za web i Android platforme) pružaju korisne upute i savjete koji pomažu pri pravilnom postavljanju aplikacija kako bi se osigurala učinkovita komunikacija s Firebase platformom.

#### 3.4.4. Sigurnost i kontrola pristupa

Sigurnost i kontrola pristupa bile su važne stavke u ovom radu, iako nisu implementirani svi mogući mehanizmi za autentifikaciju koji su dostupni kroz Firebase platformu. Umjesto korištenja Firebase Authentication [28] usluge, koristi se jednostavniji pristup provjere korisnika putem izravnih upita prema Firestore bazi podataka.

Prilikom prijave korisnika, neovisno je li riječ o Android ili web aplikaciji, prvo se traži unos adrese e-pošte i lozinke. Nakon unosa, šalje se upit prema Firestoreu s točno definiranim kriterijima. Putem upita traži se korisnika s istom adresom e-pošte i istom vrijednosti šifrirane zaporke. Za šifriranje zaporke koristi se SHA-256 algoritam [29], koji je poznat po svojoj sigurnosti i nepovratnosti, što znači da se šifrirana lozinka ne može dešifrirati natrag u izvornu formu. Ako se pronađe korisnik koji zadovoljava te uvjete, sustav smatra da je prijava uspješna.

Nakon uspješne prijave, korisnički identifikator se sprema lokalno u aplikaciji. Ovaj identifikator se može ponovno koristiti tijekom daljnjeg rada s aplikacijom, time omogućavajući pristup relevantnim podacima i značajkama bez potrebe za ponovnim unosom vjerodajnica. Ovaj pristup je jednostavan, ali dovoljno učinkovit za potrebe ovog rada, gdje složeniji sustavi autentifikacije jesu poželjni, ali nisu bili nužni.

Iako ovaj sustav ne uključuje napredne značajke sigurnosti poput dvofaktorske autentifikacije ili dinamičkog upravljanja sjednicama, implementirani postupak šifriranja i provjere podataka osigurava osnovnu razinu zaštite korisničkih podataka i sprječava neovlašteni pristup sustavu.

## **4. PREGLED SLUČAJEVA KORIŠTENJA**

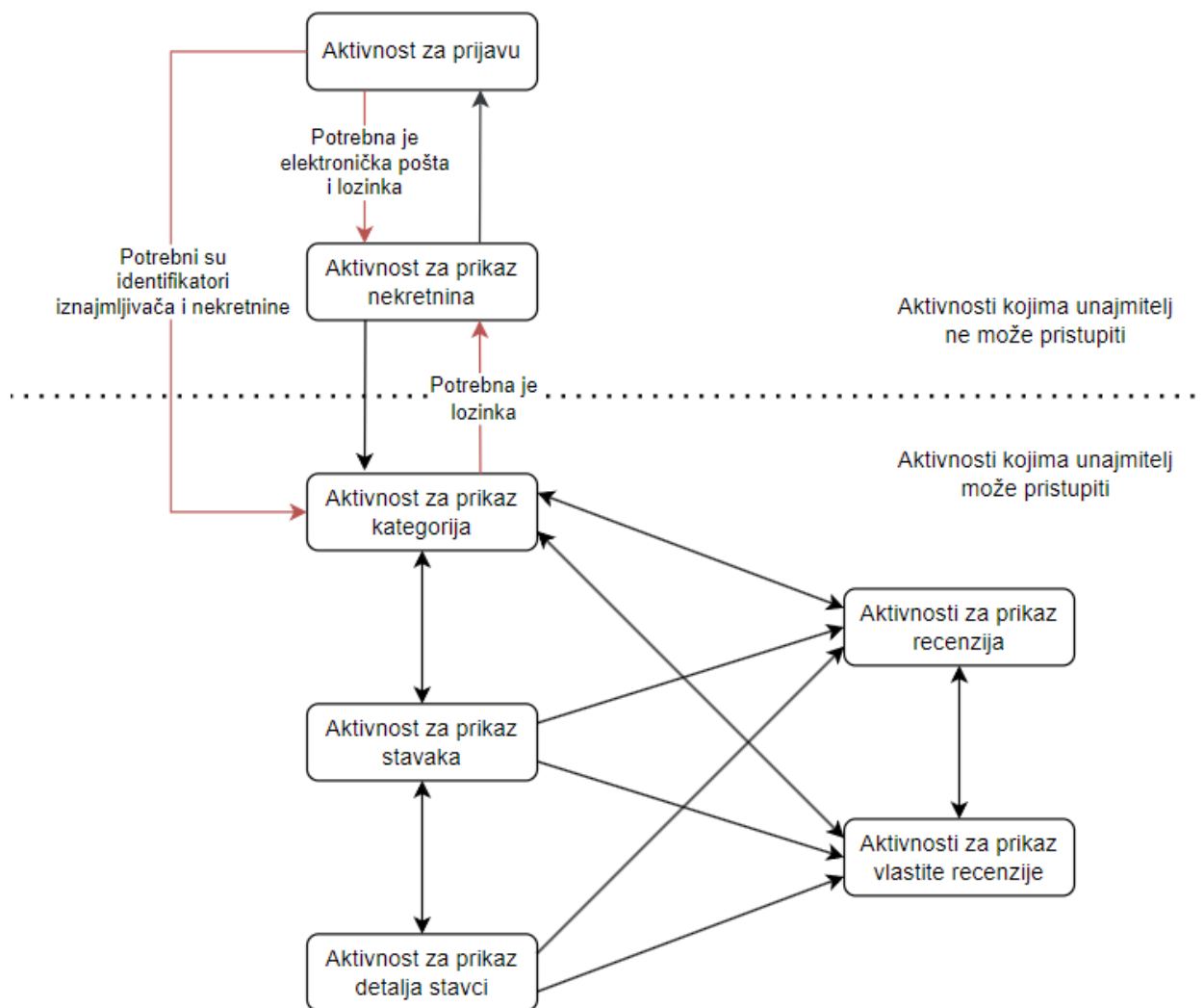
Ovo poglavlje bavi se specifičnim slučajevima korištenja sustava, istražujući kako različiti korisnici stupaju u interakciju s cjelokupnim sustavom i raznim funkcijama koje su im dostupne. Slučajevi korištenja ključni su za razumijevanje praktične primjene sustava i načina na koji on zadovoljava potrebe svojih korisnika. Ovo poglavlje ocrta primarne scenarije u kojima se sustav koristi, s detaljima radnji koje korisnici mogu izvesti, očekivanim ishodima i tijekom interakcije unutar aplikacije.

Kroz ove slučajeve korištenja, ilustrirat će se praktične vrijednosti sustava, naglašavajući kako on ispunjava zahtjeve administratora, iznajmljivača i unajmljivača. Ovo poglavlje, također, istražuje rubne slučajeve i iznimke, pružajući sveobuhvatan pogled na to kako sustav rješava različite situacije. Predstavljanjem ovih slučajeva korištenja, cilj je jasno prikazati mogućnosti sustava i problema iz stvarnog svijeta koje sam sustav rješava.

### **4.1. Slučajevi korištenja u pametnoj TV aplikaciji**

Aplikacija za unajmljivače posebno je oblikovana da bude intuitivna i laka za korištenje, što unajmljivačima (i iznajmljivačima) olakšava interakciju sa sustavom pomoću daljinskog upravljača za Android TV. Sučelje aplikacije optimizirano je za velike zaslone, s fokusom na jednostavnost i pristupačnost, osiguravajući da se korisnici mogu kretati kroz različite značajke uz minimalan napor. U ovom potpoglavlju će se govoriti o slučajevima korištenja (engl. "use cases") u vezi s aplikacijom za pametni TV, ističući različite načine na koje korisnici mogu imati interakciju sa sustavom.

U razvoju Android aplikacija, aktivnost je temeljna komponenta koja upravlja korisničkim sučeljem i interakcijama aplikacije. Aktivnosti služe kao osnovni sastavni blokovi Android aplikacija i predstavljaju žarišnu točku u kojoj korisnici stupaju u interakciju s aplikacijom. Na slici 4.1. prikazan je dijagram navigiranja po aktivnostima unutar aplikacije.



Slika 4.1. Dijagram navigacije po aktivnostima

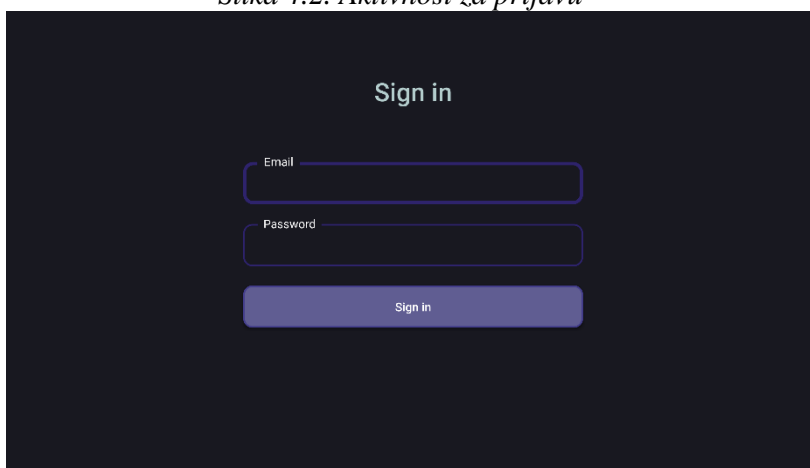
#### 4.1.1. Postavljanje aplikacije za unajmljivače

Odmah prilikom pokretanja aplikacije, korisniku će se prvo prikazati "aktivnost za prijavu", koja od iznajmljivača traži da se prijavi koristeći iste vjerodajnice koje koristi na drugoj aplikaciji (slika 4.2.). Ovaj postupak prijave ključan je za određivanje korisnikova pristupa značajkama aplikacije i određenim smještajnim objektima povezanim s njegovim računom. Odluka o izbacivanju mogućnosti registracije za unajmljivače iz ove aplikacije donesena je jer se od iznajmljivača očekuje da ispuni sve tražene vrijednosti kako bi svojim gostima pružili što ugodnije i jednostavnije iskustvo i time izbjegli potencijalne komplikacije. Ovaj pristup osigurava da aplikacija za pametni TV ostaje



usredotočena na pružanje ugodnog iskustva za postojeće korisnike, dopuštajući im brz i jednostavan pristup značajkama relevantnim za njihove potrebe.

*Slika 4.2. Aktivnost za prijavu*

The image shows a dark-themed sign-in form. At the top, the text "Sign in" is centered in a light color. Below it are two input fields: the first is labeled "Email" and the second is labeled "Password". Both fields have a light blue border. At the bottom of the form is a solid blue button with the text "Sign in" in white.

Nakon prijave, iznajmljivač se usmjerava na "aktivnost za prikaz nekretnina" ili "aktivnost za prikaz smještajnih objekata". Ova je aktivnost implementirana kako bi se iznajmljivaču pružio jasan pregled svih svojih smještajnih objekata, mogućnost da odabere jedan i time se prilagodi funkcionalnost ostatka aplikacije kako bi odgovarala specifičnim potrebama unajmljivača povezanog s određenom nekretninom. Aktivnost nudi iznajmljivaču nekoliko mogućnosti personalizacije aplikacije, dostupnih putem gumba koji se nalaze u zaglavlju, na desnoj strani aktivnosti (kao što je prikazano na slici 4.3.). Neke od tih mogućnosti su:

- Odabir jezika: Aplikacija podržava više jezika, uključujući engleski, njemački i hrvatski. Ova je značajka osobito važna za iznajmljivače koji upravljaju smještajnim objektima u regijama s različitim jezičnim podrijetlom, čime omogućuje korisnicima komuniciranje s aplikacijom na jeziku koje preferiraju.
- Odabir teme: Korisnici mogu birati između tamne ili svijetle teme, što im omogućuje da prilagode izgled aplikacije na temelju svojih osobnih preferencija. Ova značajka je posebno korisna za poboljšanje vidljivosti u različitim uvjetima osvjetljenja.
- Matrica sadržaja: Aplikacija nudi tri različite opcije raspodjele sadržaja zasnovane na matricama: 1 x 1, 1 x 3 i 2 x 3. To omogućuje iznajmljivaču da kontrolira koliko će stavaka biti prikazano na zaslonu u nekom trenutku. Ova fleksibilnost je osobito vrijedna

za upravljanje s većim brojem entiteta (pod time smatraju se nekretnine, kategorije i stavke), osiguravajući da zaslona ostane organiziran i jednostavan za navigaciju.

- Format vremena: Korisnici mogu birati između 24-satnog i 12-satnog (AM/PM) formata vremena, osiguravajući da je prikazano vrijeme usklađeno s lokalnim konvencijama ili osobnim preferencijama. Ova značajka poboljšava upotrebljivost aplikacije, posebno za međunarodne goste koji su možda navikli na različite standarde prikaza vremena i koji dolaze iz različitih vremenskih zona.



*Slika 4.3. Mogućnosti personalizacije na aktivnosti za prikaz nekretnina*

Uz navedene mogućnosti personalizacije, aktivnost za prikaz smještajnih objekata, također, uključuje pomagala za brže pronalaženje željenih nekretnina. Postoje dvije značajke, a to su traka za pretraživanje i gumb za navigaciju po stranica.

- Traka za pretraživanje: Smještena u donjem lijevom kutu zaslona, traka za pretraživanje omogućuje iznajmljivaču filtriranje popisa nekretnina unosom željenog teksta (lijeva komponenta na slici 4.4.). Nakon što iznajmljivač upiše traženi tekst, lista nekretnina se ažurira i prikazuje samo nekretnine čiji nazivi na odabranom jeziku odgovaraju ili barem sadrže uneseni tekst, dok sve ostale ostaju skrivene. Ako se ne unese nikakav tekst u tražilicu, aplikacija prikazuje cijeli popis smještajnih objekata. Ova je značajka osobito korisna iznajmljivačima koji upravljaju s više nekretnina. Takva im funkcionalnost omogućuje brzo pronalaženje željene nekretnine, smanjujući ili čak u potpunosti poništavajući vrijeme potrebno za listanje kroz popis.
- Navigacija po stranicama: U donjem desnom kutu zaslona nalazi se gumb koji prikazuje trenutni broj stranice uz ukupni broj stranica (desna komponenta na slici 4.4.). Korisnici mogu pretraživati tako da pritisnu na gumb, unesu broj stranice na koju žele doći, te pošalju zahtjev. Ako iznajmljivač unese broj stranice manji od 1, aplikacija automatski prikazuje prvu stranicu. S druge strane, ako uneseni broj premašuje ukupan broj stranica,

aplikacija prikazuje posljednju stranicu. Navigacija po stranicama ključna je značajka za upravljanje velikim popisima smještajnih objekata, budući da rastavlja sadržaj na dijelove kojima se može upravljati te omogućuje jednostavnost korištenja korisničkog sučelja, čak i kada se radi sa značajnom količinom podataka.

*Slika 4.4. Traka za pretraživanje i gumb za navigaciju po stranicama*



Ove značajke uvelike poboljšavaju korisničko iskustvo zato što dopuštaju personalizaciju te poboljšavaju učinkovitost pretraživanja. Odabir jezika i teme, s prilagodljivim izgledima sadržaja, čine aplikaciju pristupačnijom. Filtriranje sadržaja i pretraživanje stranica čine sam proces pronalazanja određenih nekretnina bržim i lakšim.

#### 4.1.2. Prikaz kategorija i stavaka

Nakon što je smještajni objekt odabran, aplikacija prelazi na "aktivnost za prikaz kategorija". Ova je aktivnost prvenstveno osmišljena za unajmljivače kojima pruža strukturirani način za istraživanje različitih tema poput atrakcija, restorana, plaža, muzeja, informacija o odabranom smještajnom objektu. Bitno je napomenuti da od ovog trenutka, aplikacija je stvorena kako bi gosti upravljali njome. Prethodne aktivnosti očekivale su od korisnika, tj. iznajmljivača da obavi proces autentifikacije i odabere smještajni objekt koji želi da je aplikacijom prezentira. Od iznajmljivača se ništa drugo ne očekuje, te je svaka sljedeća aktivnost dizajnirana s ciljem informiranja unajmitelja, tj. gosta o smještajnom objektu i njegovoj okolini. Aktivnost za prikaz kategorija ne služi samo kao organizacijski alat, već također poboljšava iskustvo unajmljivača nudeći dodatne interaktivne značajke koje su prikazane na slici 3.2. Te značajke uključuju:

- Gumb za ocjenjivanje: Gumb se pojavljuje na krajnje lijevom rubu zaglavlja i omogućuje unajmljivaču slanje svojih recenzija i pregled recenzija prethodnih gostiju.
- Gumb za vrijeme: Ako su dostupne geografske koordinate smještajnog objekta, aplikacija može prikazati trenutno vrijeme i temperaturu na toj lokaciji slanjem zahtjeva

prema OpenWeatherMap stranici [30]. Ove informacije prikazuju se putem gumba na krajnje desnom rubu zaglavlja. Ova značajka je korisna gostima koji žele planirati svoje dnevne aktivnosti na temelju vremenskih uvjeta u stvarnom vremenu. Tako se poboljšava cjelokupno iskustvo tijekom boravka. Gumb također ima funkciju mijenjanja stupnjeva iz Fahrenheita u Celzijuse i obrnuto, u skladu sa željama unajmljivača.

- Poruka dobrodošlice: Ovisno o odabranoj matrici sadržaja i odabranom jeziku, aplikacija može prikazati personaliziranu poruku dobrodošlice. Takva razina personalizacije pomaže u stvaranju ugodnog iznenađenja i osjećaja cijenjenosti, te se time unajmljivač osjeća više kao kod kuće.

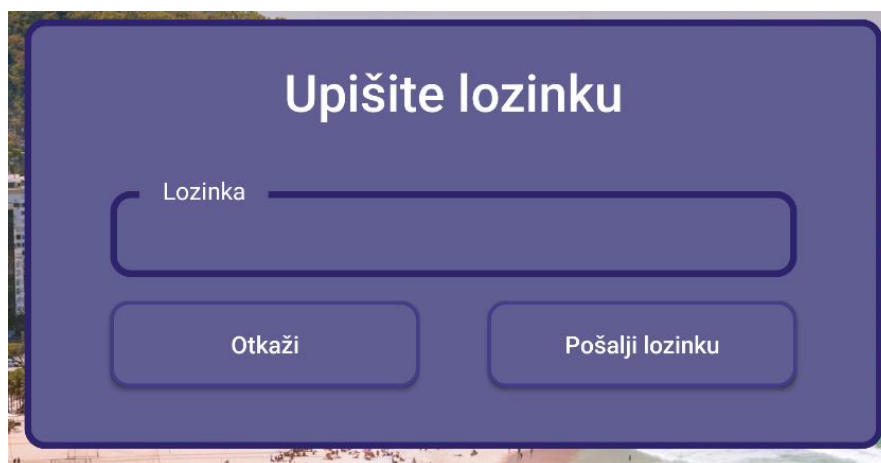
Bitno je napomenuti da ove značajke nisu prisutne u aktivnosti za prikaz smještajnih objekata zbog jednostavnog razloga: aplikacija još uvijek nema kontekst o kojoj se točno nekretnini radi. Jednom kad se odabere smještajni objekt, aplikacija ima pristup identifikatoru nekretnine, koordinatama (bitno za prikaz lokalnih vremenskih prilika i temperature) i listi gostiju (bitno za prikaz imena trenutnog gosta).

Poput aktivnosti za prikaz smještajnih objekata, aktivnost za prikaz kategorija također uključuje značajke filtriranja i navigacije po stranicama. Traka za pretraživanje uvijek se nalazi u donjem lijevom kutu zaslona, ali u ovom slučaju traka omogućuje unajmljivaču filtriranje kategorija unosom određenog teksta. Nakon što unajmitelj upiše željeni tekst, popis kategorija se ažurira, prikazujući samo one kategorije čiji naslov na odabranom jeziku odgovara ili barem sadržava uneseni tekst. Isto vrijedi i za gumb za navigaciju po stranica. U donjem desnom kutu zaslona nalazi se gumb za navigaciju po stranica koji prikazuje broj trenutne stranice i ukupan broj, ali u kontekstu kategorija.

Aktivnost za prikaz kategorija služi iznajmljivačima kao ulazna točka za detaljnije istraživanje stavaka unutar svake kategorije. Funkcionalnost kategorija osmišljena je kako bi pružile skup informacija zaokružene u jednu logičku cjelinu, što gostima olakšava pristup najrelevantnijim informacijama za vrijeme njihovog boravka.

Posljednja značajka koju je vrijedno spomenuti na aktivnosti za prikaz kategorija je nemogućnost vraćanja unajmljivača na aktivnost za prikaz smještajnih objekata. Ova značajka je implementirana kako bi se zaštitili podaci o nekretninama iznajmljivača. Jedini način na koji se korisnik u ulozi iznajmljivača može vratiti na aktivnost za prikaz smještajnih objekata je klikom na gumb "Natrag" (engl. "Back button") na daljinskom upravljaču koji uzrokuje pojavljivanje prozora (prikazano na slici 4.5.). Taj prozor traži od korisnika unos iste lozinke koja je korištena u procesu prijave. Očito, ako se lozinka podudara, korisnik će biti preusmjeren na aktivnost za prikaz nekretnina.

*Slika 4.5. Autentifikacija potrebna za povratak na aktivnost s prikazom smještajnih objekata*



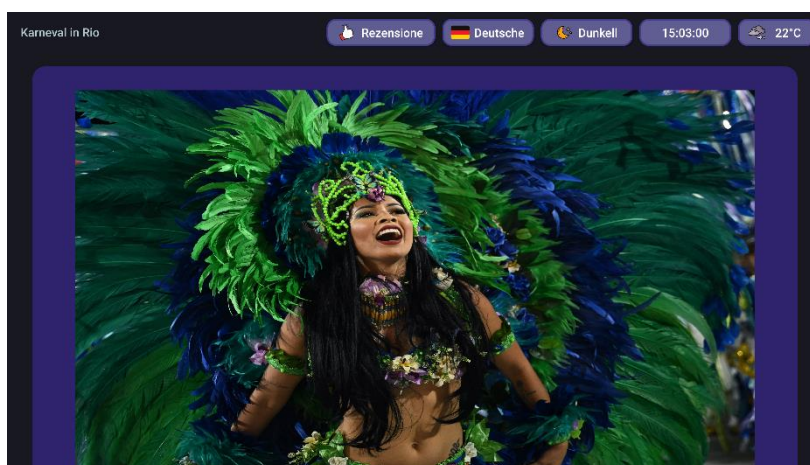
Odabirom dotične kategorije, unajmljivač se usmjerava na "aktivnost za prikaz stavaka", gdje može vidjeti pojedinačne stavke vezane uz odabranu kategoriju. Ova aktivnost nadograđuje se na organizacijsku strukturu kategorija, pružajući detaljniji prikaz specifičnih usluga, pogodnosti, sadržaja, atrakcija ili bilo koje druge teme koja je na neki način povezana sa smještajnim objektom.

Aktivnost za prikaz stavaka uglavnom je ista kao aktivnost za prikaz kategorija, u pogledu izgleda i funkcionalnosti. Jedina očita promjena je da se prikazuje lista stavaka, a ne kategorija. Sukladno tome, modificirane su funkcionalnosti trake za pretraživanje i gumba za navigaciju po stranicama kako bi se ažurirala lista koja prikazuje stavke.

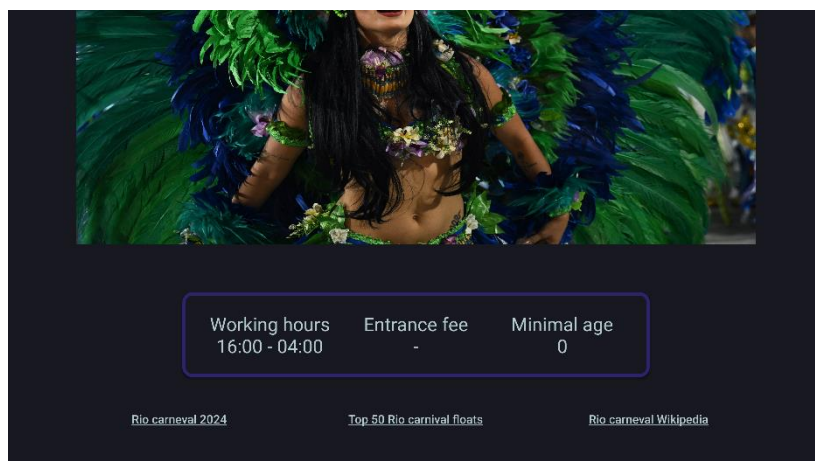
#### 4.1.3. Prikaz detaljnih informacija o stavci

Završna faza u prikazu informacija smještajnog objekta i lokaliteta događa se unutar "aktivnosti za prikaz detalja stavke", gdje unajmljivač može vidjeti sve dostupne pojedinosti o odabranoj stavci. Ova je aktivnost vrlo prilagodljiva, pri čemu iznajmljivač određuje koji će se predložak koristiti za prikaz sadržaja. Mogućnost odabira između različitih predložaka omogućeno je u aplikaciji za iznajmljivače te gost ne može odabrati drugi predložak. Time iznajmljivač osigurava da su informacije prikazane unajmljivaču na način koji smatra da najbolje opisuju stavku. S obzirom na to da pod "stavke" možemo smatrati bilo koji turistički sadržaj, aplikacija nudi tri prilagodljiva predloška za prikaz informacija o svakoj stavki: minimalni, kompaktni i potpuni predložak.

Minimalni predložak je oblikovan za jednostavnost i učinkovitost, prikazuje naslov stavke i horizontalnu dijaprojeksiju slika (slika 4.6.). Podatci poput radnog vremena, cijene ulaznica i minimalna dob za ulaz također se prikazuju kada su relevantni i primjenjivi (slike 4.7.). Za kraj, na dnu aktivnosti prikazuju se do tri poveznice na vanjske web stranice (neki od primjera su Google Maps, TripAdvisor, Wikipedia, YouTube, ali može biti i bilo koja druga web stranica). Vanjske poveznice su dodane kako bi iznajmljivač mogao pristupiti dodatnim informacijama o određenoj temi u slučaju da nisu dovoljne one informacije koje su već prikazane na aktivnosti. Minimalni predložak idealan je za stavke u kojima je vizualni sadržaj najvažniji, omogućujući unajmljivaču brz pristup ključnim informacijama bez nepotrebnog ometanja.

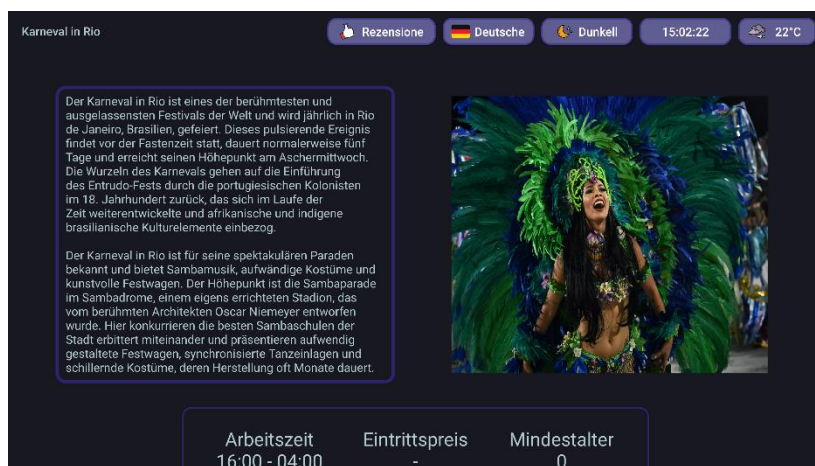


*Slika 4.6. Minimalni predložak za prikaz detaljnih informacija o stavci*



Slika 4.7. Prikaz dodatnih informacija ovisno o kontekstu (u prikazanom slučaju - događaj)

Kompaktni predložak uključuje detaljan opis stavke na odabranom jeziku, prikazan na lijevoj polovici zaslona, dok prva slika s popisa zauzima desnu polovicu (slika 4.8.). Klikom na opis otvara se novi prozor koji popunjava zaslon s opisom stavke. Tako se korisniku omogućuje čitanje cijelog opisa u slučaju kada je isti duži od 20 redaka, što bi u suprotnome bilo nemoguće. U slučaju da korisnik klikne na prvu sliku, otvara se novi prozor koji popunjava zaslon i omogućuje dijaprojekciju slika. Ovaj izgled uspostavlja ravnotežu između vizualne privlačnosti i detaljnog sadržaja, što ga čini prikladnim za stavke koje zahtijevaju i vizualnu i narativnu prezentaciju.

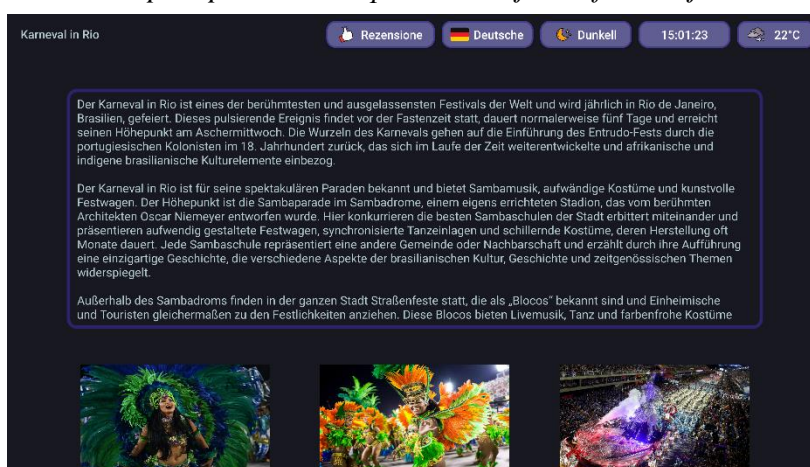


Slika 4.8. Kompaktni predložak za prikaz detaljnih informacija o stavci



Konačno, potpuni predložak prikazuje istu količinu informacija kao i kompaktni, ali sve je drugačije raspoređeno po zaslonu te svaka komponenta ima dovoljno mjesta za uredno prikazivanje (slika 4.9.). Sadrži opis stavke u odabranom jeziku širine zaslona (u početku prikazuje 16 redaka, proširivo na 32), 3 x 3 matricu slika i sve značajke prethodnih predložaka. Pritiskom na bilo koju sliku, aktivira se dijaprojeksija. U slučaju da lista sadrži više od devet slika, početne se prikazuju u matrici, a ostale su dostupne putem dijaprojeksije. Ovaj je predložak idealan za stavke koje nude detaljan opis ili višestruke vizualne elemente, kao što su povijesna mjesta, muzeji ili detaljni vodiči do lokalnih atrakcija.

Slika 4.9. Potpuni predložak za prikaz detaljnih informacija o stavci

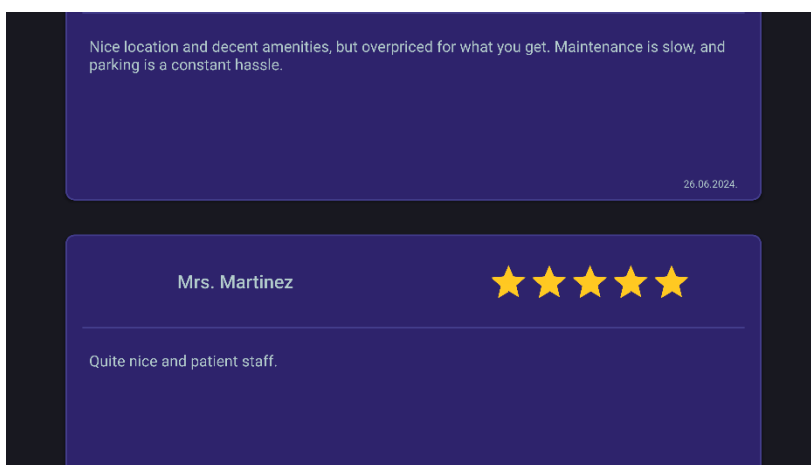


Svaki je predložak osmišljen kako bi predstavio informacije na način koji je i privlačan i lak za navigaciju po komponentama aktivnosti te osigurava unajmljivaču brz pristup informacijama koje su mu potrebne bez da se osjeća preopterećeno. Iznajmljivaču je omogućen odabir predloška kako bi se prezentacija prikazala u najboljem mogućem svjetlu bez da se preopteret korisnika s nepotrebnim komponentama, rezultirajući poboljšanjem cjelokupnog korisničkog iskustva.

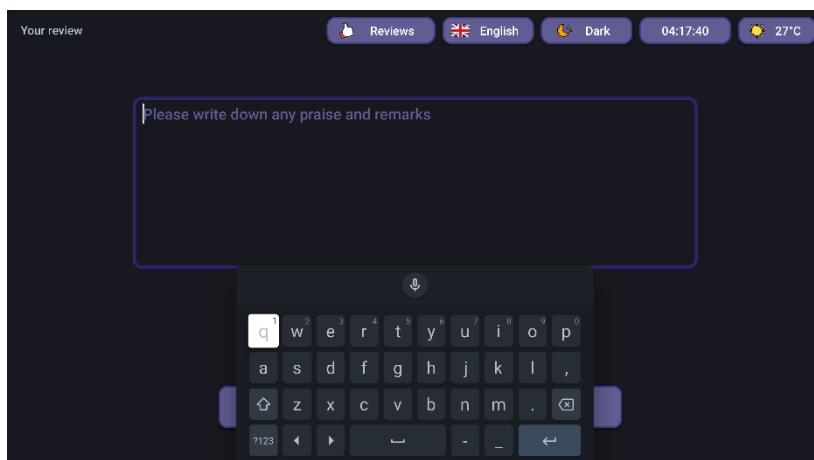
#### 4.1.4. Prikaz i slanje recenzija



Posljednje dvije aktivnosti odnose se na slanje i čitanje recenzija. "Aktivnost za prikaz recenzija" prikazuje 10 najnovijih recenzija (prikazano na slici 4.10.). Time unajmljivač ima kratki uvid u reputaciju smještajnog objekta. Ako unajmljivač želi ostaviti vlastitu recenziju, to može učiniti putem "aktivnosti za prikaz vlastite recenzije" (prikazano na slici 4.11.), koja mu omogućuje da ostavi komentar i ocijeni smještajni objekt i obližnje turističke sadržaje na ljestvici od 1 do 5 zvjezdica. Ova značajka potiče osjećaj zajedništva među gostima i istovremeno pruža vrijedne povratne informacije iznajmljivačima, pomažući im da održi visoke standarde usluge.



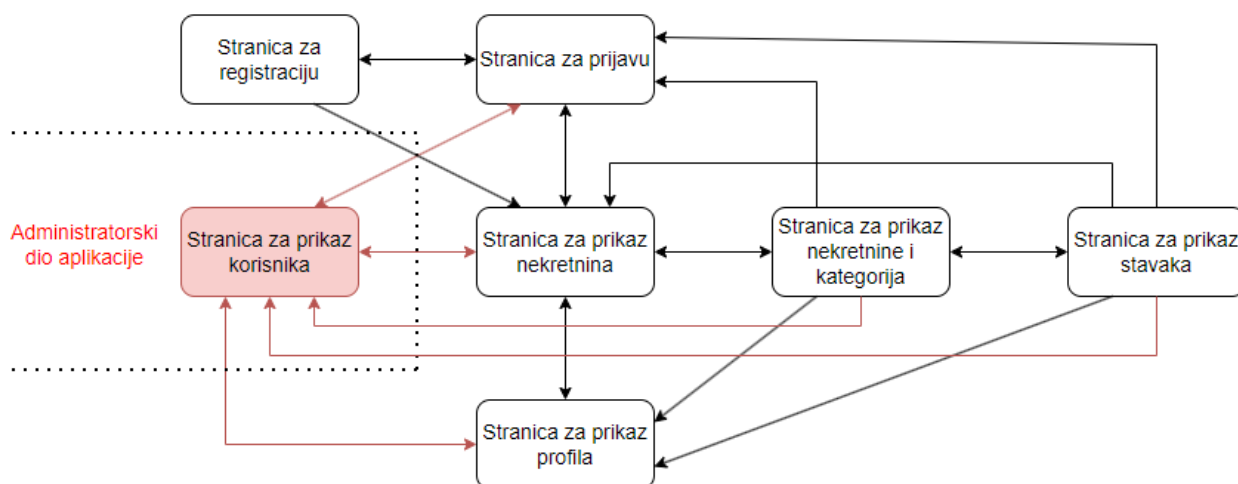
Slika 4.10. Vertikalna lista zadnjih 10 recenzija



Slika 4.11. Sučelje za unos vlastite recenzije

## 4.2. Slučajevi korištenja u web aplikaciji

Aplikacija za iznajmljivače implementirana je s ciljem ostvarivanja praktičnosti i jednostavnosti korištenja. Sučelje je optimizirano za rad na računalu kako bi korisnicima olakšalo i ubrzalo unos informacija i slika za svaku temu, smanjujući pritom potrebno vrijeme. U ovom potpoglavlju bit će obrađeni slučajevi korištenja koji se odnose na aplikaciju za iznajmljivače i time predstaviti različiti načini na koje korisnici mogu upravljati sustavom. Na slici 4.12. prikazan je dijagram navigiranja po stranicama unutar razvijene web aplikacije.



Slika 4.12. Dijagram navigacije po stranicama web aplikacije

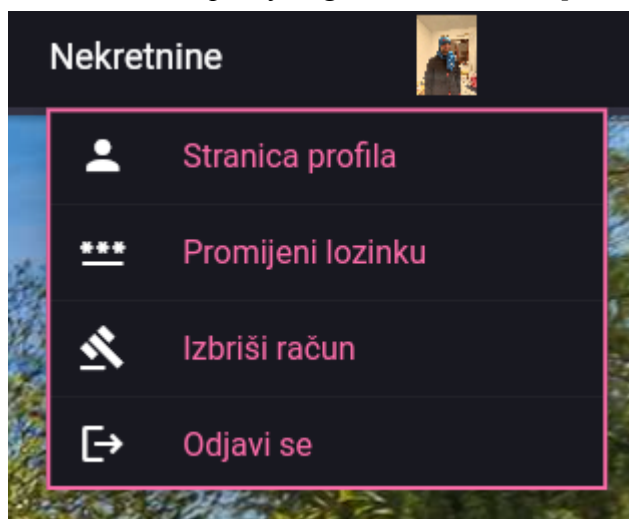
#### 4.2.1. Promjena lozinke i brisanje korisničkog računa

Korisnik može promijeniti lozinku ili izbrisati svoj korisnički račun putem zadnjeg desnog gumba u zaglavlju aplikacije. Klikom na ovaj gumb otvara se padajući izbornik s opcijama prikazanim na slici 4.13. Odabirom opcije "Promijeni lozinku," korisniku se otvara prozor koji zahtijeva unos stare lozinke, nove lozinke i ponovnu potvrdu nove lozinke. Da bi promjena lozinke bila uspješna, korisnik mora ispuniti sljedeće uvjete:

- Stara lozinka mora biti ispravna, a njezina ispravnost provjerava se slanjem upita prema Firestoreu;
- Nova lozinka mora se podudarati s ponovnim unosom;
- Nova lozinka mora biti duža od 8 znakova.

Ako su ovi uvjeti ispunjeni, zahtjev se šalje Firestoreu, a korisniku se prikazuje obavijest o uspješnoj promjeni lozinke.

Slika 4.13. Prikaz padajućeg izbornika u web aplikaciji



Ako korisnik želi izbrisati svoj korisnički račun, može to učiniti odabirom opcije "Izbriši račun." Pojavljuje se prozor koji traži unos lozinke za trenutni korisnički račun. Ako je lozinka točno unesena, sustav pokreće metode koje brišu sve slike i informacije povezane s korisnikom, njegovim smještajnim objektima, kategorijama i stavkama. Time se osigurava privatnost korisnika i sprječava nepotrebno zauzimanje prostora u Firebaseu. Nakon brisanja računa, korisnik se automatski preusmjerava na stranicu za prijavu.

Važno je istaknuti da administrator, za razliku od ostalih korisnika, nema mogućnost brisanja svog korisničkog računa. Međutim, administrator i dalje ima mogućnost mijenjanja vlastite lozinke.

#### 4.2.2. Dodavanje, uređivanje i brisanje nekretnina

Nakon prijave, iznajmljivaču se na stranici "Nekretnine" nudi mogućnost stvaranja novog smještajnog objekta. Odabirom te opcije, iznajmljivač se preusmjerava na stranicu za unos detalja o novoj nekretnini. Kao što je očekivano, sva polja koja zahtijevaju informacije su prazna. Iznajmljivač mora unijeti ime nekretnine na engleskom, njemačkom i hrvatskom jeziku kako bi stvorio novu nekretninu. Također je moguće unijeti dodatne podatke poput broja kućnog telefona, adrese,

poštanskog broja, grada i države smještajnog objekta. Klikom na gumb "Stvorite nekretninu" pokreće se geokodiranje, koje određuje koordinate na temelju unesenih podataka o adresi, poštanskom broju, gradu i državi. Svi uneseni podaci, zajedno s koordinatama, šalju se na Firestore.

U slučaju uspješne pohrane, stranica se ažurira kako bi omogućila dodatne opcije, kao što su ažuriranje (gumb "Spremi promjene") i brisanje (gumb "Izbriši nekretninu"). Uz to, iznajmljivač sada može dodati sliku pozadine, podatke o gostima i kategorije.

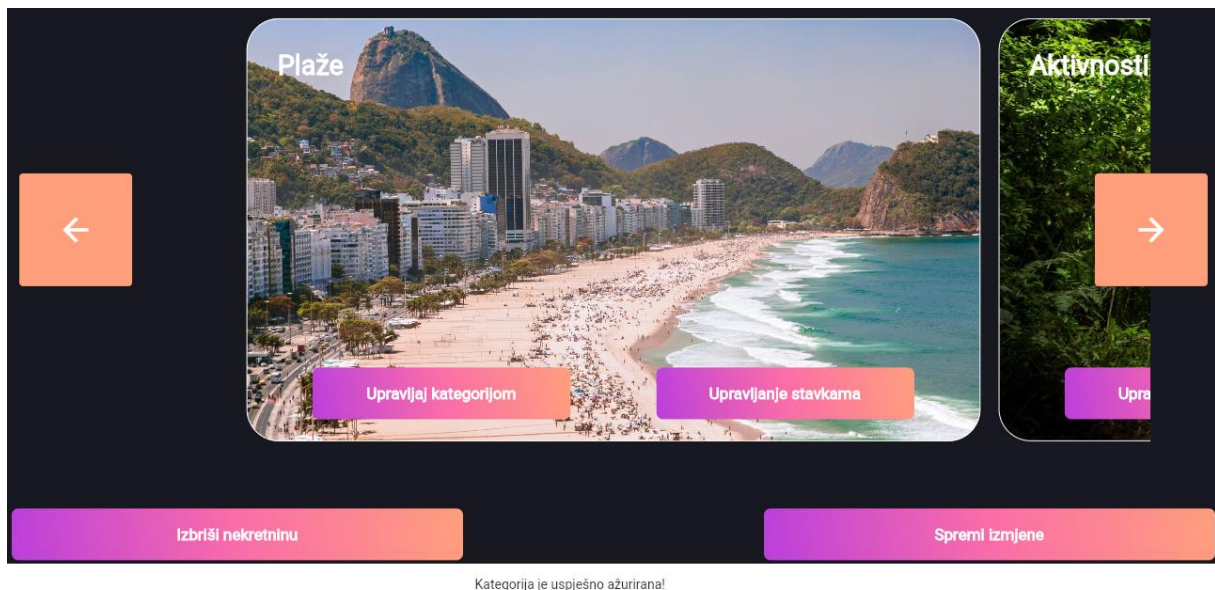
S druge strane, ako iznajmljivač zbog nekog razloga ne može stvoriti, ažurirati ili izbrisati smještajni objekt, prikazat će mu se adekvatna povratna informacija.

#### 4.2.3. Dodavanje, uređivanje i brisanje kategorija

Proces dodavanja, uređivanja i brisanja kategorija započinje nakon što iznajmljivač odabere nekretninu na kojoj želi vršiti takve radnje. Sve postojeće kategorije prikazuju se u obliku kartica, pri čemu svaka kartica sadrži naslov i sliku pozadine - ako je postavljena (prikazano na slici 4.14). Posljednja kartica u nizu uvijek je "prazna" i ne predstavlja nijednu ranije stvorenu kategoriju. Na toj kartici piše "Dodaj kategoriju", a njena svrha je upravo dodavanje novih kategorija. Klikom na ovu karticu otvara se prozor u kojem se unosi naziv kategorije na engleskom, njemačkom i hrvatskom jeziku. Korisnik također može odmah dodati sliku pozadine.

Nakon što je kategorija stvorena, korisnik može ponovno otvoriti isti prozor klikom na gumb "Upravljaj kategorijom", koji sada nudi opcije za ažuriranje ili brisanje te kategorije.

Kao i u prethodnom slučaju, implementirane su povratne informacije u slučaju neuspješnog upravljanja kategorijama, a odgovarajuće obavijesti prikazuju se na dnu sučelja. Primjer takve povratne informacije može se vidjeti na slici 4.14.



*Slika 4.14. Kartica za prikaz i upravljanje kategorijom*

#### 4.2.4. Dodavanje, uređivanje i brisanje stavaka

Nakon odabira kategorije čije stavke želi uređivati, iznajmljivač treba kliknuti na gumb "Upravljanje stavkama" kako je prikazano na slici 4.14. Pritiskom na gumb otvara se nova stranica na kojoj iznajmljivač može pregledati prethodno stvorene stavke te ih po potrebi ažurirati ili obrisati. Kao i kod dodavanja kategorija, posljednja stavka na listi je prazna i služi za unos informacija koje će stvoriti novu stavku i zatim je pohraniti u Firestore. Iznajmljivač mora unijeti naziv stavke na tri prethodno definirana jezika. Po želji može odmah dodati poveznice, opise i radne sate za stavku, ukoliko su kontekstualno primjenjivi.

Nakon što je nova stavka stvorena, korisniku se nude i opcije za odabir predloška koji će se koristiti u Android TV aplikaciji za prikaz informacija, kao i mogućnost unosa cijene ulaznice i minimalne dobi za ulaz, ako su te informacije primjenjive. Također, korisnik može dodati sliku pozadine stavke te ispuniti galeriju slikom koje će gosti moći pregledavati putem Android TV aplikacije.

Kao i u prethodnim slučajevima, ako dođe do problema s povezivanjem na Firebase ili s upravljanjem stavkama, aplikacija će prikazati povratnu informaciju kako bi korisnik bio obaviješten

o nastalom problemu i mogao donijeti odluku o daljnjim koracima.

#### 4.2.5. Blokiranje i brisanje iznajmljivača

Kao što je navedeno u poglavlju 3.3.2., samo administrator ima ovlasti blokiranja i brisanja iznajmljivača kako bi se osigurao prikaz primjerenih i stvarnih informacija. Nakon uspješne prijave, administratora se vodi na stranicu za prikaz iznajmljivača. Iznajmljivači su prikazani u tablici u kojoj svaki stupac predstavlja dotične informacije o iznajmljivaču. Kao što je prikazano na slici 4.15., zadnja dva gumba služe za blokiranje (lijevi gumb) i za brisanje (desni gumb). U slučaju kada administrator želi potpuno izbrisati korisnika, pojavljuje se prozor kojim se administratora pita je li siguran u svoj odabir, te želi li nastaviti s brisanjem iznajmljivača. U slučaju da se administrator odluči na brisanje, jedino iznajmljivačeva adresa e-pošte ostaje spremljena u sustavu, a zastavice blokiranja i brisanja potvrđuju da je iznajmljivač izbrisan.

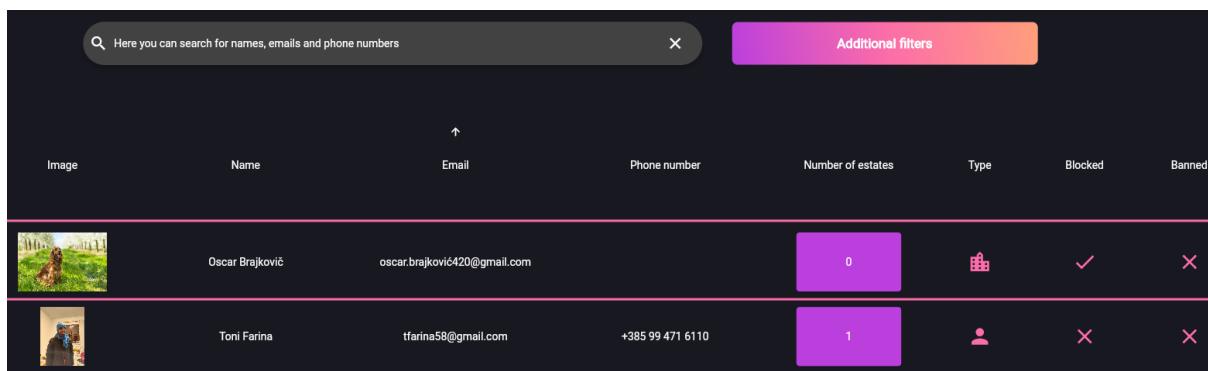






Image	Name	Email	Phone number	Number of estates	Type	Blocked	Banned
	Oscar Brajković	oscar.brajkovic420@gmail.com		0		<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Toni Farina	tfarina58@gmail.com	+385 99 471 6110	1		<input type="checkbox"/>	<input type="checkbox"/>

Slika 4.15. Prikaz iznajmljivača u web sučelju

#### 4.2.6. Pregled informacija korisnika i nekretnina

Administrator može jednostavno pregledati osobne podatke iznajmljivača. Klikom na redak koji predstavlja ciljanog iznajmljivača, administrator je preusmjeren na stranicu s detaljnim informacijama o tom korisniku, uključujući sve osobne podatke koje je prethodno unio.

Ako administrator klikne na ljubičasti gumb, bit će preusmjeren na stranicu koja prikazuje sve nekretnine iznajmljivača, organizirane u vertikalnu listu ili prikazane na karti (slika 4.16.).

Administrator nije ograničen samo na tu stranicu te može dalje pregledavati informacije o pojedinim kategorijama i stavkama vezanim uz tu nekretninu.



*Slika 4.16. Prikaz registriranih smještajnih objekata na karti*

## 5. ZAKLJUČAK

Sustav koji je implementiran u ovom radu uključuje web aplikaciju za iznajmljivače, Android TV aplikaciju za unajmljivače i Firebase kao poslužitelja, a razvijen je s ciljem izrade funkcionalnog rješenja za osobe koje žele poboljšati iskustvo gostiju u vlastitim smještajnim objektima. Iako aplikacije imaju ograničenja po pitanju značajki i oblikovanja, svejedno učinkovito služe svojoj glavnoj svrsi: omogućuju iznajmljivačima upravljanje svojim smještajnim objektima i informacijama koje će se prikazivati, a unajmljivači mogu pregledavati detalje o smještajnim objektima u kojima borave i turističkim sadržajima u okolici.

Web aplikacija za iznajmljivače pruža jednostavno sučelje administratoru i iznajmljivaču. Administrator može upravljati korisničkim računima i nadzirati iznajmljivačeve smještajne objekte, dok iznajmljivači mogu upravljati svojim nekretninama, njenim kategorijama i stavkama, te time definirati prezentaciju svih bitnih informacija u smislenim cjelinama.

Android TV aplikacija projektirana je posebno za goste (turiste) u apartmanima, nudeći jednostavan način za odabir i pregled informacija koje se odnose na zanimljive turističke sadržaje u okolici, kao i o samom smještaju u kojem borave. Aplikacija je projektirana s ciljem da bude jednostavna i intuitivna za korištenje. Aplikacija učinkovito zadovoljava potrebe korisnika pružajući jednostavno sučelje za pristup relevantnim informacijama.

Tijekom razvoja rješenja bilo je određenih izazova, posebice s integracijom Firebasea i prilagodbom korisničkog sučelja različitim platformama. No odabir tehnološkog stoga koji uključuje Flutter, Dart i Firebase pokazao se prikladnim pristupom.

Razvijeni sustav ima još puno potencijalnih funkcionalnosti koje bi se mogle implementirati, dok trenutna verzija ipak nije na razini komercijalne upotrebe.



## LITERATURA

- [1] Android, “An easier way to enjoy the entertainment”, s interneta, <https://www.android.com/tv/>, 20.07.2024.
- [2] Firebase, “Make your app the best”, s interneta, <https://firebase.google.com/>, 26.08.2024.
- [3] TripAdvisor, “Where to?”, s interneta, <https://www.tripadvisor.com/>, 27.08.2024.
- [4] Google, “Pronađite povoljne letove za bilo koje odredište”, s interneta, <https://www.google.com/travel/flights>, 26.08.2024.
- [5] Expedia, “Discover your new favorite stay”, s interneta, <https://www.expedia.com/>, 26.08.2024.
- [6] Prezi, “Presentations that move audiences”, s interneta, <https://prezi.com/>, 25.08.2024.
- [7] Prezi, “Welcome to the Prezi Gallery”, s interneta, <https://prezi.com/gallery/>, 12.08.2024.
- [8] Wikipedija, “Aplikacijsko programsko sučelje (eng. application programming interface, API)”, s interneta, <https://hr.wikipedia.org/wiki/API>, 04.08.2024.
- [9] Wikipedija, “Baza podataka (eng. database) je organizirani skup podataka.”, s interneta, [https://hr.wikipedia.org/wiki/Baza\\_podataka](https://hr.wikipedia.org/wiki/Baza_podataka), 04.08.2024.
- [10] Flutter, “Build for any screen”, s interneta, <https://flutter.dev/>, 11.06.2024.
- [11] Dart, “An approachable, portable, and productive language”, s interneta, <https://dart.dev/>, 11.06.2024.
- [12] Google, “Use our flexible, scalable NoSQL cloud database”, s interneta, <https://firebase.google.com/docs/firestore>, 27.08.2024.
- [13] Google, “Cloud Storage for Firebase is built on fast and secure”, s interneta, <https://firebase.google.com/docs/storage>, 27.08.2024.
- [14] Wikipedija, “An integrated development environment (IDE) is a software application”, s interneta, [https://en.wikipedia.org/wiki/Integrated\\_development\\_environment](https://en.wikipedia.org/wiki/Integrated_development_environment), 17.08.2024.
- [15] Android, “Get the official Integrated Development Environment”, s interneta, <https://developer.android.com/studio>, 17.08.2024.
- [16] Wikipedija, “Java je objektno orijentirani programski jezik koji su razvili”, s interneta, [https://hr.wikipedia.org/wiki/Java\\_\(programski\\_jezik\)](https://hr.wikipedia.org/wiki/Java_(programski_jezik)), 10.08.2024.

- [17] Wikipedija, “XML je kratica za EXtensible Markup Language, odnosno jezik”, s interneta, <https://hr.wikipedia.org/wiki/XML>, 01.07.2024.
- [18] Visual Studio, “Code Editing. Redefined.”, s interneta, <https://code.visualstudio.com/>, 27.07.2024.
- [19] Git, “Git is a distributed version control system”, s interneta, <https://en.wikipedia.org/wiki/Git>, 24.08.2024.
- [20] Flutter, “A versatile mapping package for Flutter.”, s interneta, [https://pub.dev/packages/flutter\\_map](https://pub.dev/packages/flutter_map), 24.08.2024.
- [21] Wikipedija, “Address geocoding, or simply geocoding”, s interneta, [https://en.wikipedia.org/wiki/Address\\_geocoding](https://en.wikipedia.org/wiki/Address_geocoding), 24.08.2024.
- [22] Wikipedija, “NoSQL (originally referring to "non-SQL" or "non-relational")”, s interneta, <https://en.wikipedia.org/wiki/NoSQL>, 22.08.2024.
- [23] Wikipedija, “A document-oriented database, or document store, is ”, s interneta, [https://en.wikipedia.org/wiki/Document-oriented\\_database](https://en.wikipedia.org/wiki/Document-oriented_database), 19.07.2024.
- [24] Lucidchart, “What is an Entity Relationship Diagram (ERD)?”, s interneta, <https://www.lucidchart.com/pages/er-diagrams>, 19.07.2024.
- [25] Wikipedija, “Indeksiranje dokumenata na računalu predstavlja postupak”, s interneta, <https://hr.wikipedia.org/wiki/Indeksiranje>, 22.07.2024.
- [26] Flutter, “A Flutter plugin to use the Firebase Core API”, s interneta, [https://pub.dev/packages/firebase\\_core](https://pub.dev/packages/firebase_core), 31.07.2024.
- [27] Flutter, “A Flutter plugin to use the Firebase Cloud Storage API.”, s interneta, [https://pub.dev/packages/firebase\\_storage](https://pub.dev/packages/firebase_storage), 22.06.2024.
- [28] Google, “Most apps need to know the identity of a user.”, s interneta, <https://firebase.google.com/docs/auth>, 27.08.2024.
- [29] Wikipedija, “SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic”, s interneta, <https://en.wikipedia.org/wiki/SHA-2>, 27.08.2024.
- [30] OpenWeatherMap, “Weather forecasts, nowcasts and history”, s interneta, <https://openweathermap.org/>, 11.06.2024.

## **POPIS OZNAKA I KRATICA**

**API** Application Programming Interface. 6, 20

**HTTP** Hypertext Transfer Protocol. 7

**SDK** JavaScript Object Notation. 19

**XML** Extensible Markup Language. 9

## **SAŽETAK**

U ovome radu implementiran je i predstavljen raspodijeljeni sustav koji omogućava stvaranje i uređivanje turističkih informacija putem odgovarajućeg web sučelja, a kojima se može pristupiti putem aplikacije na pametnom TV-u. Sustav je namijenjen vlasnicima turističkih apartmana kako bi unaprijedili iskustvo gostiju pružanjem relevantnih informacija, kako o samom smještajnom objektu tako i o turističkim sadržajima u okolici, putem pametnog TV-a. Razvijeno rješenje uključuje web aplikaciju za upravljanje sadržajem, odgovarajuće Firebase usluge i aplikaciju za pametni TV s operacijskim sustavom Android. Svi slučajevi korištenja testirani su na emulatoru u integriranom razvojnom okruženju Android Studio.

Ključne riječi — sustav za informiranje turista, aplikacija za pametni TV, Firebase

## **ABSTRACT**

In this thesis, a distributed system was implemented and presented that enables the creation and editing of tourist information via a suitable web interface that can be accessed via an application on a Smart TV. The system is intended for vacation rental owners to improve the guest experience by providing relevant information about both the accommodation itself and tourist facilities in the area via Smart TV. The developed solution includes a web application for content management, corresponding Firebase services, and an application for a Smart TV with an Android operating system. All use cases are tested on an emulator in the integrated development environment Android Studio.

Keywords — tourist information system, Smart TV app, Firebase