



UNIVERZITET U BIHAĆU  
TEHNIČKI FAKULTET  
ODSJEK: ELEKTROTEHNIKA  
SMJER: RAČUNARSTVO I INFORMATIKA

---

## Objektno orijentirane baze podataka

---

PROJEKTNI ZADATAK  
TEMA: DATING APLIKACIJA

**Profesor:** Doc. dr. Admir Midžić  
**Asistent:** Zinaid Kapić MA ing. el.

**Student:**  
Sajra Alijagić, 1279

Bihać, februar 2026. godine

## Sažetak

Većina savremenih frameworka za izradu web aplikacija zasnovana je na objektno-orientiranom pristupu i MVC arhitekturi, među kojima je i Laravel framework koji je korišten za izradu ove aplikacije. Laravel je objektno-orientirani PHP framework namijenjen razvoju modernih web aplikacija sa jasno organizovanom strukturu kodom i podrškom za rad sa bazama podataka. U ovom projektu razvijena je web aplikacija DatingApp koja omogućava registraciju korisnika, kreiranje i uređivanje korisničkih profila, pregled drugih korisnika, kao i ostvarivanje međusobnih podudaranja putem sistema lajkova i dislajkova. Kroz aplikaciju su prikazane osnovne mogućnosti rada sa Laravelom, uključujući CRUD operacije nad bazom podataka, autentifikaciju korisnika i upravljanje relacijskim podacima korištenjem objektno-relacionog modela. Aplikacija koristi relacijsku SQL bazu podataka čiji se podaci mapiraju u objekte pomoću Laravel Eloquent ORM sistema. Glavni entiteti sistema su korisnici, profili, slike profila, lajkovi i dislajkovi, koji zajedno omogućavaju funkcionisanje dating sistema. Rad aplikacije prikazan je kroz različite funkcionalnosti kao što su upravljanje profilima, galerija slika, statistički pregled korisnika i prikaz lokacija korisnika pomoću eksternog API-ja.

**Ključne riječi:** Web aplikacija, Laravel, PHP, Objektno-orientirano, Baza podataka

## Abstract

Most modern frameworks for web application development are object-oriented and based on the MVC architecture, including the Laravel framework, which was used to develop this application. Laravel is an object-oriented PHP framework designed for building modern web applications with a well-structured code organization and strong support for database operations. In this project, a web application called DatingApp was developed. The application allows user registration, profile creation and editing, browsing of other users, and establishing mutual matches through a like and dislike system. The application demonstrates the basic capabilities of working with Laravel, including CRUD operations on a database, user authentication, and management of relational data using the object-relational model. The main entities of the system are users, profiles, profile images, likes, and dislikes, which together enable the functionality of the dating system. The operation of the application is demonstrated through various features such as profile management, image galleries, statistical overview of users, and displaying user locations using an external API.

**Keywords:** Web Application, Laravel, PHP, Object-Oriented, Database

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Modeliranje aplikacije</b>	<b>2</b>
2.1	Opis aplikacije . . . . .	2
2.2	Statički UML dijagrami . . . . .	3
2.2.1	Klasni dijagram . . . . .	4
2.3	Dinamički UML dijagrami . . . . .	5
2.3.1	Dijagram slučajeva korištenja . . . . .	5
2.4	ER dijagram baze podataka . . . . .	8
<b>3</b>	<b>Implementacija</b>	<b>10</b>
3.1	Tehnologija izrade aplikacije . . . . .	10
3.1.1	Laravel . . . . .	10
3.1.2	SQL . . . . .	10
3.2	MVC arhitektura . . . . .	10
3.3	Objektno-relaciono mapiranje . . . . .	11
<b>4</b>	<b>Analiza rada aplikacije</b>	<b>12</b>
4.1	Opis slučajeva korištenja . . . . .	12
<b>5</b>	<b>Zaključak</b>	<b>22</b>

## **Popis slika**

1	Klasni dijagram . . . . .	4
2	Use-Case dijagram . . . . .	6
3	Entity Relationship dijagram . . . . .	9
4	Stranica za prijavu . . . . .	12
5	Stranica za registraciju . . . . .	13
6	Stranica za kreiranje računa . . . . .	14
7	Početna stranica . . . . .	16
8	Stranica za like i dislike . . . . .	17
9	Stranica vlastitog profila . . . . .	18
10	Stranica za sve profile . . . . .	19
11	Stranica jednog profila . . . . .	20
12	Stranica za match sistem . . . . .	21
13	Prikaz kada korisnik nema nijedan match . . . . .	21

# 1 Uvod

Projekat koji je razvijen predstavlja web aplikaciju za upoznavanje korisnika, s ciljem omogućavanja jednostavnog i modernog načina povezivanja ljudi putem interneta. U savremenom digitalnom okruženju, sve više korisnika koristi online platforme za upoznavanje novih osoba i ostvarivanje društvenih kontakata. Zbog toga je postojala potreba za razvojem aplikacije koja omogućava jednostavno kreiranje profila, pregled drugih korisnika i ostvarivanje međusobnih podudara-nja (match-eva), uz fokus na preglednost, sigurnost i jednostavnost korištenja. Aplikacija je razvijena korištenjem Laravel PHP frameworka, koji omogućava izradu stabilnih, sigurnih i skalabilnih web aplikacija. Laravel se zasniva na MVC (Model–View–Controller) arhitekturi, čime se postiže jasna podjela odgovornosti unutar aplikacije. Modeli su zaduženi za rad s bazom podataka, kontroleri obrađuju poslovnu logiku, dok su pogledi (views) odgovorni za prikaz korisničkog interfejsa. Ovakav pristup omogućava lakše održavanje, nadogradnju i testiranje aplikacije. Aplikacija omogućava korisnicima registraciju i prijavu na sistem, kreiranje i uređivanje vlastitog profila, kao i pregled profila drugih korisnika. Korisnici mogu označiti profile drugih korisnika opcijama like ili dislike, pri čemu sistem automatski prepozna međusobne lajkove i formira match između korisnika. Također, korisnici imaju mogućnost upravljanja galerijom slika, pregleda statističkih podataka i prikaza lokacija korisnika na interaktivnoj mapi. Za po-hranu podataka koristi se relaciona baza podataka, koja omogućava sigurno i efikasno upravljanje informacijama o korisnicima, profilima, slikama i interak-cijama između korisnika. Korištenjem Eloquent ORM-a, Laravel omogućava jednostavan rad sa bazom kroz objektno-orientisani pristup, čime se značajno smanjuje potreba za pisanjem kompleksnih SQL upita. Aplikacija podržava osnovne sigurnosne mehanizme, uključujući autentifikaciju korisnika, verifikaciju email adrese i zaštitu ruta. Time se osigurava da samo registrovani korisnici mogu pristupati funkcionalnostima aplikacije i upravljati vlastitim profilima. Cilj projekta je izrada funkcionalne, pregledne i sigurne web aplikacije za upoznavanje korisnika koja omogućava jednostavno povezivanje ljudi putem interneta. Korištenjem savremenih tehnologija i principa razvoja softvera, aplikacija predstavlja stabilno rješenje koje se može dalje nadograđivati dodatnim funk-cionalnostima kao što su napredni sistem pretrage ili real-time komunikacija između korisnika.

## 2 Modeliranje aplikacije

Modeliranje aplikacije predstavlja jedan od ključnih koraka u procesu razvoja softverskog sistema, jer omogućava jasno definisanje strukture aplikacije, njenih funkcionalnosti i međusobnih odnosa između pojedinih komponenti. Prije same implementacije bilo je neophodno analizirati potrebe korisnika, definisati osnovne funkcionalnosti sistema te predvidjeti način na koji će korisnici komunicirati s aplikacijom. Ovakav pristup omogućava stabilan razvoj, bolju organizaciju koda i lakše održavanje sistema. U okviru projekta DatingApp, modeliranje aplikacije obuhvata izradu UML dijagrama koji prikazuju strukturu sistema, odnose između entiteta i interakciju između korisnika i aplikacije. Korišteni su dijagrami slučajeva upotrebe (Use Case), dijagrami klase i sekvenički dijagrami, koji zajedno pružaju jasan uvid u funkcionisanje aplikacije. Ovi dijagrami opisuju ključne funkcionalnosti poput registracije i prijave korisnika, kreiranja i uređivanja korisničkog profila, pregleda profila drugih korisnika, sistema lajkova i dislajkova, kao i ostvarivanja međusobnih podudaranja (match-eva). Modeliranje sistema omogućava lakše planiranje razvoja aplikacije, smanjuje mogućnost grešaka tokom implementacije i olakšava buduća proširenja funkcionalnosti. Također, predstavlja važan dio tehničke dokumentacije koji omogućava bolje razumijevanje strukture i logike sistema. Na ovaj način postavljeni su čvrsti temelji za izgradnju stabilne, funkcionalne i skalabilne web aplikacije namijenjene upoznavanju korisnika putem interneta.

### 2.1 Opis aplikacije

Cilj ove aplikacije jeste prikazati rad sa objektno-orientiranim pristupom razvoju web aplikacija i rad sa podacima koji se čuvaju u relacijskoj bazi podataka, ali se u aplikaciji koriste u obliku objekata. Aplikacija je razvijena kao web aplikacija za upoznavanje korisnika (DatingApp) koja omogućava registraciju korisnika, kreiranje profila, pregled drugih korisnika i ostvarivanje međusobnih podudaranja putem sistema lajkova i dislajkova. U aplikaciji su definisane glavne klase koje predstavljaju osnovne entitete sistema: User, Profil, ProfilSlika, Like i Dislike. Klasa User predstavlja registrovanog korisnika aplikacije, dok klasa Profil sadrži detaljne informacije o korisniku koje su vidljive drugim korisnicima. Klasa ProfilSlika predstavlja slike koje pripadaju galeriji određenog profila. Klase Like i Dislike predstavljaju interakcije između korisnika i koriste se za određivanje međusobnih podudaranja (match-eva). U programu se kreiraju objekti svake klase, a zatim se nad njima izvršavaju odgovarajuće funkcije kao što su dodavanje, izmjena, brisanje i pregled podataka, čime se prikazuje rad sa podacima u obliku objekata korištenjem objektno-relacionog modela. Korištena tehnologija je Laravel, objektno-orientirani PHP framework za izradu web aplikacija koji omogućava organizaciju aplikacije prema MVC arhitekturi. Za čuvanje podataka koristi se SQL relacijska baza podataka čiji se entiteti mapiraju u objekte pomoću Laravel Eloquent ORM sistema. Glavni sadržaj aplikacije čine funkcionalnosti kao što su upravljanje korisničkim profilima, pregled drugih korisnika, like/dislike sistem, match sistem, galerija slika i

statistički pregled korisnika. Aplikacija sadrži navigacioni meni koji omogućava jednostavan pristup svim funkcionalnostima kao što su dashboard, profil korisnika, pregled profila drugih korisnika i match sekcija.

## 2.2 Statički UML dijagrami

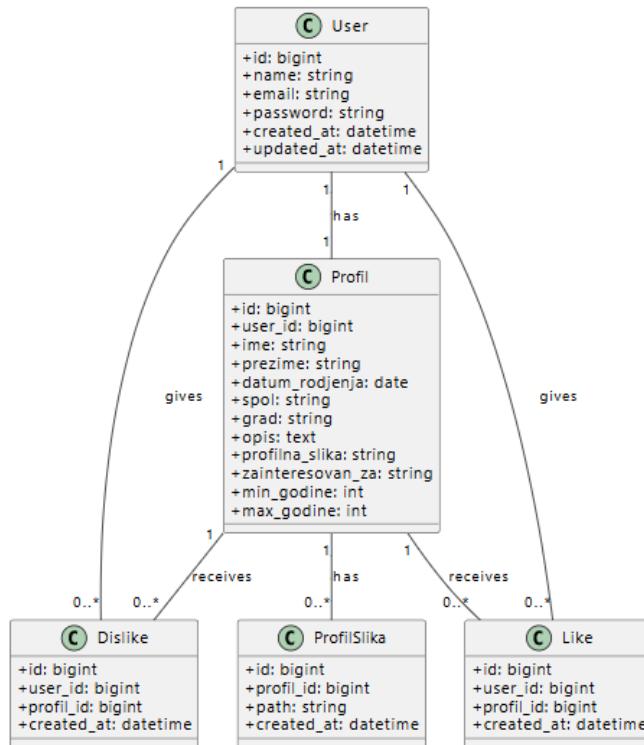
Statički UML (engl. *Unified Modelling Language*) dijagrami služe za opisivanje karakteristika ili dijela sistema [5]. Postoji 5 vrsta statičkih dijagrama:

- Dijagram strukture aplikacije
- Klasni dijagram
- Objektni dijagram
- Dijagram međusobnih odnosa objekata (kompozicioni dijagram)
- Dijagram pravila (ograničenja svakog dijela sistema)

Od navedenih statičkih dijagrama u nastavku je detaljnije opisan i prikazan klasni dijagram.

### 2.2.1 Klasni dijagram

Dijagram klasa opisuje klase koje su dio softverskog sistema. Termin klasa se koristi u objektno orijentiranom programiranju za definiranje strukture podataka koja se sastoji od svojstava (atributa) i funkcija (takođe se nazivaju metode). Klasni dijagram rađenog sistema je prikazan na slici 1.



Slika 1: Klasni dijagram

## 2.3 Dinamički UML dijagrami

Za modeliranje aplikacija, pored statičkih dijagrama, koriste se i dinamički dijagrami. Dinamički UML dijagrami opisuju ponašanje sistema, odnosno oni opisuju operacije, radnje i promjene koje se događaju u sistemu tokom rada. Postoji 5 vrsta dinamičkih dijagrama:

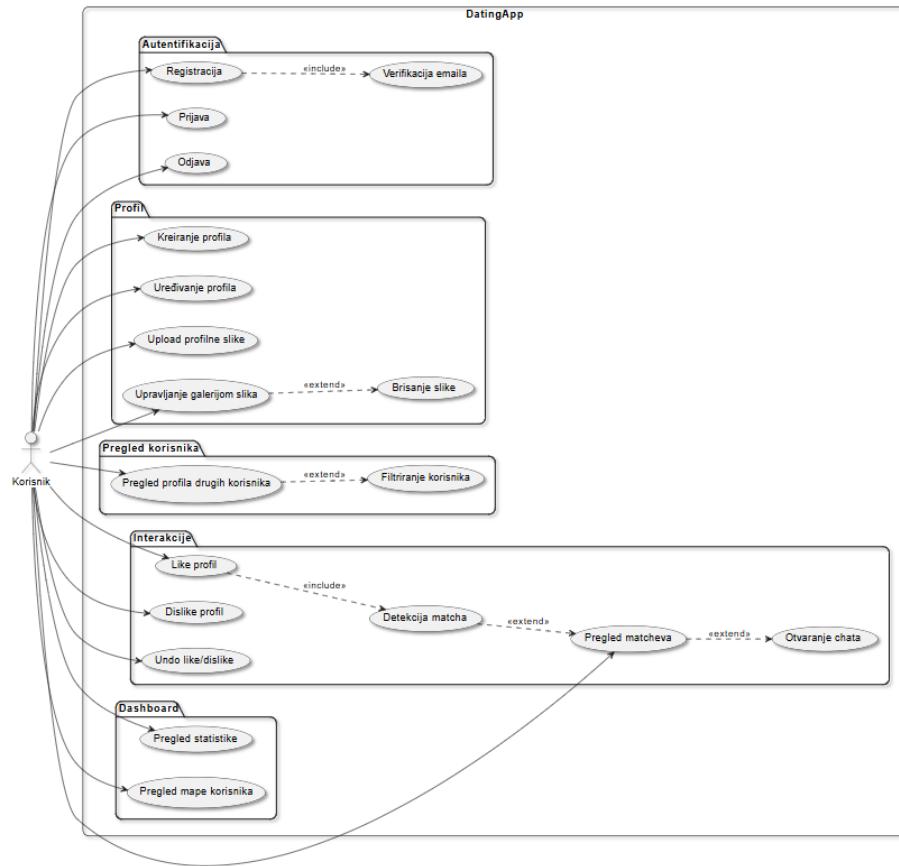
- Dijagram aktivnosti
- Dijagram stanja
- Dijagram slučajeva korištenja
- Sekvencijalni dijagram
- Komunikacioni dijagram

Od navedenih dinamičkih dijagrama u nastavku je detaljnije opisan i prikazan dijagram slučajeva korištenja.

### 2.3.1 Dijagram slučajeva korištenja

Dijagram slučajeva korištenja (engl. *Use Case Diagram*) je osnovni dinamički dijagram koji opisuje sve mogućnosti korištenja aplikacije. Dijagrami slučajeva korištenja opisuju kako korisnik može koristiti sistem, kako komponente sistema mogu koristiti druge komponente sistema, kako i tko upravlja ponašanjem sistema. U slučaju ovog sistema postoji samo jedna vrsta korisnika, odnosno svi korisnici imaju jednaka prava na korištenje svih funkcija sistema. Funkcije registracije i prijave dođu unaprijed postavljene zajedno sa postavkom Laravel projekta. Pored ovih funkcija funkcija pristupa podacima korisničkog profila i njihovom uređivanju također dolazi postavljena u sklopu Laravel projekta. Ostale funkcije koje su pravljene jesu pretraga i dodavanje svih entiteta. Entitet Film posjeduje i opcije uređivanja i brisanja, kao glavni entitet od interesovanja korisnika. Glavna funkcija aplikacije jeste prikaz posebno dizajniranih upita, kojoj se pristupa preko linka sa početne stranice nakon prijave korisnika.

Opisani dijagram slučajeva korištenja (Use-case dijagram) rađenog sistema je prikazan na slici 2.



Slika 2: Use-Case dijagram

Kao što se može vidjeti iz priloženog Use-Case dijagrama, aplikacija DatingApp ima jednu primarnu vrstu korisnika – registrovanog korisnika sistema. Sve funkcionalnosti aplikacije dostupne su nakon uspješne autentifikacije, dok korisnici bez naloga najprije prolaze kroz proces registracije. Ukoliko korisnik već posjeduje korisnički račun, vrši prijavu (login) na sistem. U suprotnom, korisnik se prvo registruje, nakon čega može izvršiti prijavu. Tokom registracije aplikacija koristi mehanizme autentifikacije (Laravel Jetstream), a po potrebi se vrši i verifikacija email adrese, čime se povećava sigurnost i pouzdanost korisničkih naloga. Nakon uspješne prijave, korisniku se prikazuju sve dostupne opcije kroz navigacioni meni, pri čemu se funkcionalnosti mogu posmatrati kroz nekoliko logičkih cjelina:

- Upravljanje profilom: korisnik kreira svoj dating profil, unosi lične podatke (ime, prezime, godine, spol, grad, opis, interesovanja i preferencije), te može vršiti naknadno uređivanje profila. U okviru profila omogućeno je i dodavanje profilne slike, kao i upravljanje galerijom slika (upload i brisanje slika).
- Pregled korisnika: korisnik može pregledati profile drugih korisnika. Sistem može prikazivati profile u skladu sa preferencijama korisnika (npr. interesovanje, raspon godina), čime se olakšava pronađazak odgovarajućih profila.
- Interakcije (like/dislike): prilikom pregleda profila drugih korisnika, korisnik može izvršiti like ili dislike. Također je omogućena opcija undo, kojom se prethodna odluka može poništiti. Ove interakcije se evidentiraju u bazi podataka i predstavljaju osnovu za rad match sistema.
- Match sistem: aplikacija automatski prepoznaće match u trenutku kada se desi obostrani like između dva korisnika. Nakon toga korisnik može u posebnoj sekciji pregledati sve svoje match-eve, a predviđena je i mogućnost otvaranja chata (demo verzija), što predstavlja osnovu za eventualno buduće proširenje aplikacije.
- Dashboard (statistika i mapa): korisnik ima pristup dashboard prikazu koji sadrži statistiku o korisnicima (npr. broj korisnika, starosna struktura, procenat muškaraca i žena, prosječna starost, najčešći gradovi), kao i prikaz lokacije korisnika na interaktivnoj mapi koja je realizovana korištenjem eksternog API-ja (Leaflet.js).

Za razliku od aplikacija koje su zasnovane na klasičnom CRUD konceptu (dodavanje, uređivanje i brisanje više različitih entiteta kroz administrativne forme), u DatingApp aplikaciji fokus je na korisničkom iskustvu: korisnik primarno upravlja svojim profilom, dok se interakcija sa drugim profilima odvija kroz pregled i donošenje odluka (like/dislike), a sistem zatim automatski implementira logiku prepoznavanja match-eva. Na ovaj način Use-Case dijagram jasno prikazuje glavne tokove rada korisnika i sve ključne funkcionalnosti koje aplikacija pruža.

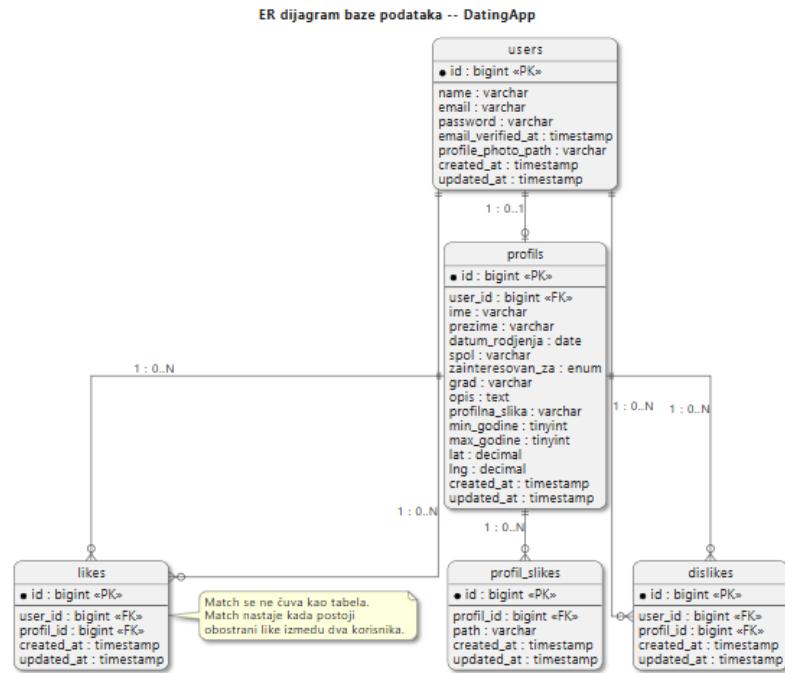
## 2.4 ER dijagram baze podataka

Baza podataka aplikacije *DatingApp* opisana je ER (engl. *Entity Relationship*) dijagrom. Ovaj dijagram prikazuje glavne entitete sistema, njihove atributе, kao i veze (relacije) koje postoje između entiteta. ER dijagram je važan dio dokumentacije jer omogućava jasan uvid u strukturu baze, način čuvanja podataka i odnose koji su potrebni da bi aplikacija ispravno funkcionalisala.

Baza podataka korištena za realizaciju ovog sistema sastoji se od pet osnovnih tabela: *users*, *profils*, *profil\_slikes*, *likes* i *dislikes*. Tabela *users* sadrži osnovne informacije o registrovanim korisnicima aplikacije i koristi se za autentifikaciju. Svakom korisniku pripada tačno jedan profil, zbog čega je između tabela *users* i *profils* uspostavljena veza 1:1. Profil sadrži detaljnije informacije koje su relevantne za dating aplikaciju (lični podaci, opis, preferencije, raspon godina i sl.). Tabela *profil\_slikes* služi za čuvanje galerije slika jednog profila. Jedan profil može imati više slika, pa je između tabela *profils* i *profil\_slikes* ostvarena veza 1:N. Na ovaj način aplikacija omogućava korisniku da pored profilne slike ima i dodatne slike koje su prikazane u galeriji. Interakcije između korisnika i profila realizovane su kroz tabele *likes* i *dislikes*. Ove tabele evidentiraju da je određeni korisnik označio neki profil kao like ili dislike. Iz tog razloga postoji veza 1:N između *users* i *likes* (korisnik može lajkovati više profila), kao i veza 1:N između *profils* i *likes* (jedan profil može biti lajkovan od više korisnika). Identična logika važi i za tabelu *dislikes*.

Ovakva struktura omogućava jednostavno praćenje interakcija i implementaciju glavne funkcionalnosti aplikacije – match sistema. Match u aplikaciji ne mora biti posebna tabela, jer se može izračunati iz postojećih podataka: match nastaje kada postoji obostrani like između dva korisnika (korisnik A lajkuje profil korisnika B, i korisnik B lajkuje profil korisnika A). Na taj način, ER model podržava automatsko prepoznavanje podudaranja bez dodatnog čuvanja u zasebnoj tabeli, što pojednostavljuje dizajn baze.

Opisani ER dijagram rađenog sistema prikazan je na slici 3.



Slika 3: Entity Relationship dijagram

## 3 Implementacija

### 3.1 Tehnologija izrade aplikacije

Aplikacija je implementirana u programskom jeziku PHP i koristeći Laravel framework. Laravel je baziran na MVC modelu (engl. *Model View Controller*) gdje je prezentacijska, funkcionalna i podatkovna strana međusobno odvojena. DBMS (engl. *Database Management System*) sistem korišten za kreiranje baze podataka je MySQL.

#### 3.1.1 Laravel

Laravel je besplatni open-source (otvorenog koda) PHP framework kojeg je kreirao Taylor Otwell i namjenjen je razvoju web aplikacija koristeći MVC arhitekturu. Ovaj framework temelji se na Symphony PHP framework-u koji je dosta stariji. Neke od osobina Laravela su modularni sistem pakiranja s namjenskim upraviteljem ovisnosti, različiti načini za pristup relacijskim bazama podataka i uslužni programi koji pomažu u implementaciji i održavanju aplikacija. Izvorni kod Laravela nalazi se na GitHubu i licenciran je pod uvjetima MIT licence [4].

#### 3.1.2 SQL

MySQL je open-source sistem za upravljanje relacijskim bazama podataka. Relacijska baza podataka organizira podatke u jednu ili više tablica podataka u kojima tipovi podataka mogu biti međusobno povezani, a ti odnosi pomažu strukturirati podatke. SQL je jezik koji programeri koriste za stvaranje, modifikaciju i izdvajanje podataka iz relacijske baze podataka, kao i kontrolu pristupa korisnika bazi podataka. Uz relacijske baze podataka i SQL (engl. *Structured Query Language*), RDBMS kao što je MySQL radi s operativnim sistemom za implementaciju relacijske baze podataka u sistem za pohranu računara, upravlja korisnicima, omogućuje pristup mreži i olakšava testiranje integriteta baze podataka i stvaranje sigurnosnih kopija [2].

### 3.2 MVC arhitektura

Model–View–Controller je šablon softverske arhitekture. Koristi se u softverskom inženjeringu za odvajanje pojedinih dijelova aplikacije u komponente ovisno o njihovoј namjeni. Model se sastoji od podataka, poslovnih pravila, logike, i funkcija ugrađenih u programsku logiku. View odnosno pogled je bilo kakav prikaz podataka kao što je obrazac, tablica ili dijagram. Moguće je prikaz podataka kroz više različitih pogleda. Controller ili upravitelj prihvata ulazne podatke i zahtjeve (od korisnika prema sistemu) i pretvara ih u zahtjeve modelu ili pogledu. Ovakva arhitektura olakšava nezavisan razvoj, testiranje i održavanje određene aplikacije [1].

### **3.3 Objektno-relaciono mapiranje**

Objektno-relaciono mapiranje (ORM) u računarskoj nauci je tehnika programiranja za pretvaranje podataka različitih tipova u objekte i obrnuto, korištenjem objektno orijentiranih programske jezika. Time se zapravo stvara "baza podataka virtualnih objekata" koja se može koristiti unutar programske jezike. Dostupni su i besplatni i komercijalni paketi koji izvode objektno-relacijsko mapiranje, iako se neki programeri odlučuju za izradu vlastitih ORM alata.

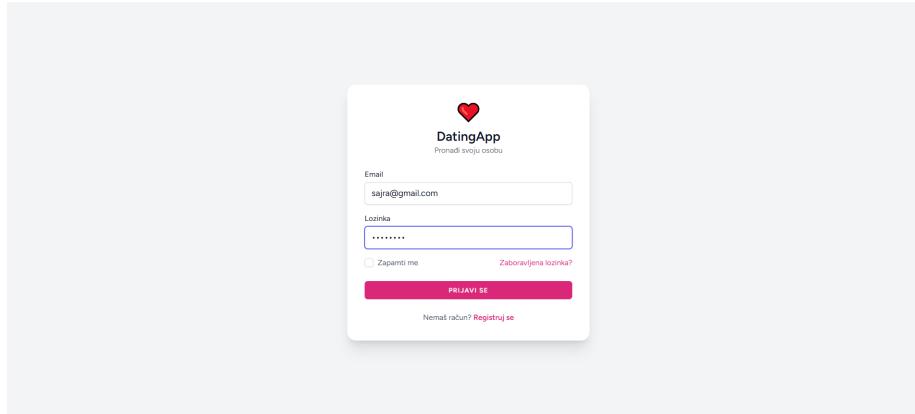
## 4 Analiza rada aplikacije

### 4.1 Opis slučajeva korištenja

#### Slučaj korištenja 1: Prijava na sistem

- Naziv SK: Prijava na sistem
- Akteri SK: Korisnik
- Učesnici SK: Korisnik i sistem
- Preduslov: Korisnik je prethodno registrovan u sistemu i posjeduje važeći korisnički račun
- Osnovni scenarij SK:
  - Korisnik otvara stranicu za prijavu na sistem.
  - Korisnik unosi email adresu i lozinku.
  - Sistem provjerava ispravnost unesenih podataka.
  - Sistem provjerava da li korisnik postoji u bazi podataka.
  - U koliko su podaci ispravni, sistem omogućava pristup aplikaciji.
  - Sistem prikazuje početnu stranicu (Dashboard) sa dostupnim opcijama.
- Alternativni scenarij:
  - U koliko uneseni podaci nisu ispravni, sistem prikazuje poruku o grešci i zahtjeva ponovni unos podataka.

Prikaz stranice za prijavu na sistem prikazan je na slici 4.



Slika 4: Stranica za prijavu

## Slučaj korištenja 2: Registracija korisnika

- Naziv SK: Registracija korisnika
- Akteri SK: Korisnik
- Učesnici SK: Korisnik i sistem
- Preduslov: Korisnik nema kreiran korisnički račun
- Osnovni scenarij SK:
  - Korisnik otvara stranicu za registraciju.
  - Korisnik unosi potrebne podatke (ime, email i lozinku).
  - Sistem provjerava ispravnost unesenih podataka.
  - Sistem kreira novi korisnički račun u bazi podataka.
  - Sistem omogućava korisniku prijavu na aplikaciju.

Prikaz stranice za registraciju na sistem prikazan je na slici 5.

The screenshot shows the registration form for the 'DatingApp'. The title 'DatingApp' is at the top, followed by a heart icon. Below it is the heading 'Registracija' and a sub-instruction 'Napravi račun i kreni s upoznavanjem.' The form consists of several input fields: 'Ime i prezime' (Nejra Mahmutović), 'Email' (nejra@gmail.com), 'Lozinka' (\*\*\*\*\*), and 'Potvrdi lozinku' (\*\*\*\*\*). At the bottom is a pink 'REGISTRUJ SE' button. Below the button, there's a link 'Ved' imasi račun? Prijavi se'.

Slika 5: Stranica za registraciju

- Sporedni scenarij SK:
  - U nešeni podaci nisu ispravni ili korisnik već postoji u sistemu, te sistem prikazuje poruku o grešci.

### Slučaj korištenja 3: Kreiranje profila

- Naziv SK: Kreiranje profila
- Akteri SK: Korisnik
- Učesnici SK: Korisnik i sistem
- Preduslov: Korisnik je prijavljen na sistem
- Osnovni scenarij SK:
  - Korisnik otvara stranicu za kreiranje profila.
  - Korisnik unosi podatke o profilu (ime, prezime, datum rođenja, spol, grad i opis).
  - Korisnik postavlja profilnu sliku.
  - Sistem sprema podatke u bazu podataka.
  - Sistem prikazuje kreirani profil korisnika.

Prikaz stranice za kreiranje računa na sistem prikazan je na slici 6.

Kreiraj profil

Ime: Nejra

Prezime: Mahmutović

Zainteresovan/a za: Muško

Min godine: 21 Max godine: 26

Datum rođenja: 08/15/2004

Spol: Žensko

Grad: Bosanska Krupa

Opis: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Profilna slika: Choose file bilbie.jpg

Nazad SPASI PROFIL

Slika 6: Stranica za kreiranje računa

- Sporedni scenarij SK:

- S esija je istekla i korisnik se vodi ponovno na stranicu za prijavu.

#### **Slučaj korištenja 4: Dashboard**

- Naziv SK: Pregled Dashboard stranice

- Akteri SK: Korisnik

- Učesnici SK: Korisnik i sistem

- Preduslov: Korisnik je prijavljen na sistem

- Osnovni scenarij SK:

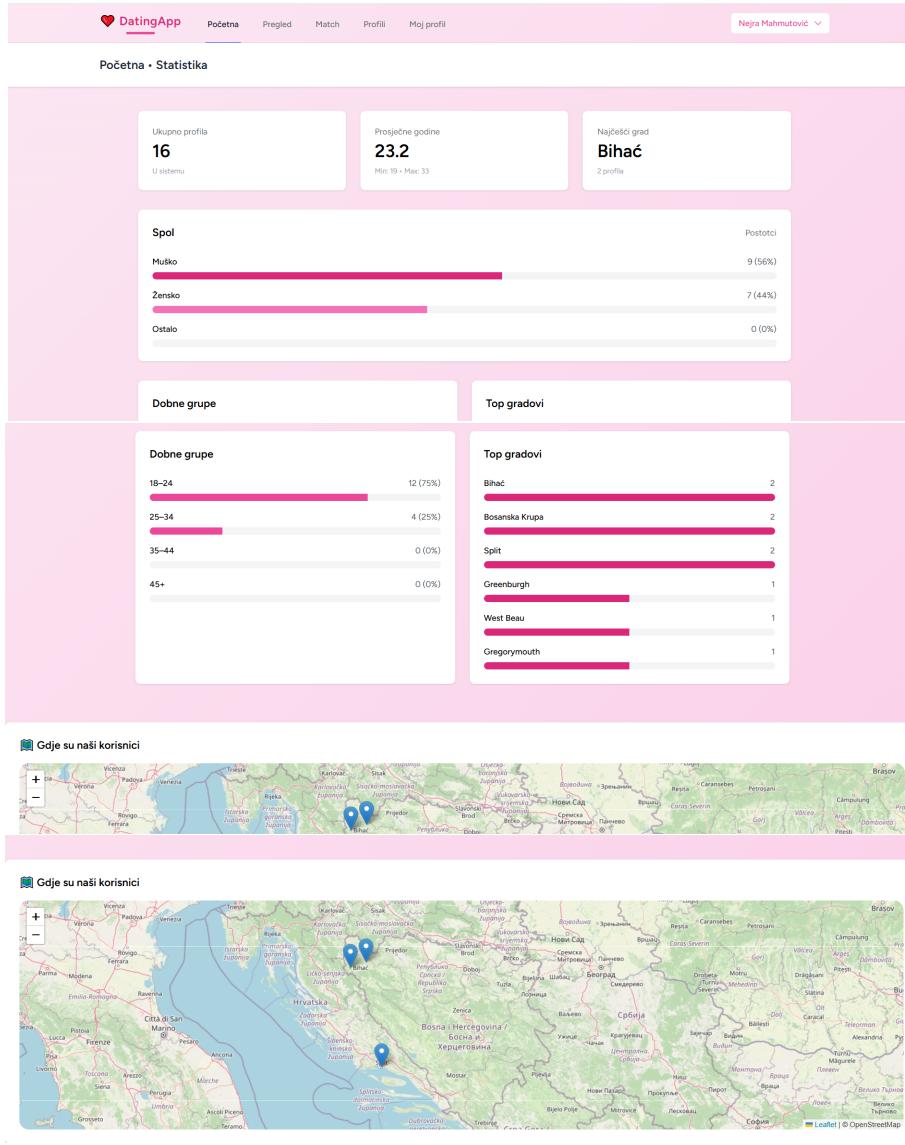
- K orisnik se prijavljuje na sistem.

- S istem prikazuje Dashboard stranicu.

- S istem prikazuje statističke podatke o korisnicima.

- S istem prikazuje mapu korisnika.

Prikaz početna stranica sistema prikazan je na slici 7.



Slika 7: Početna stranica

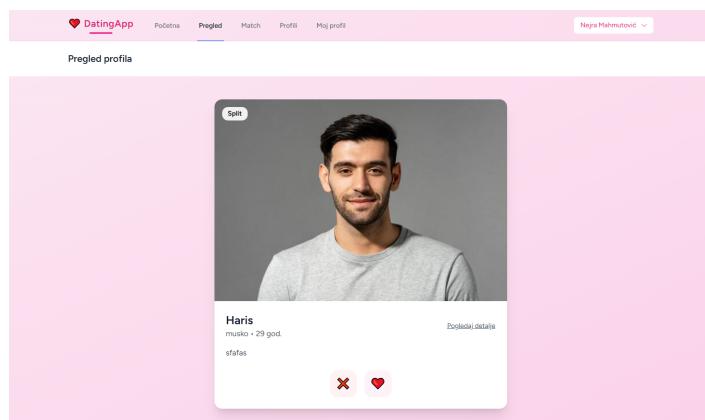
- Sporedni scenarij SK:

- Sesija je istekla i korisnik se vodi ponovno na stranicu za prijavu.

### Slučaj korištenja 5: Pregled profila za like i dislike

- Naziv SK: Like i Dislike profila
- Akteri SK: Korisnik
- Učesnici SK: Korisnik i sistem
- Preduslov: Korisnik je prijavljen i ima kreiran profil
- Osnovni scenarij SK:
  - Korisnik otvara stranicu za pregled profila.
  - Sistem prikazuje profile drugih korisnika.
  - Korisnik odabire opciju Like ili Dislike.
  - Sistem sprema odluku korisnika u bazu podataka.

Prikaz stranice za like i dislike prikazan je na slici 8.



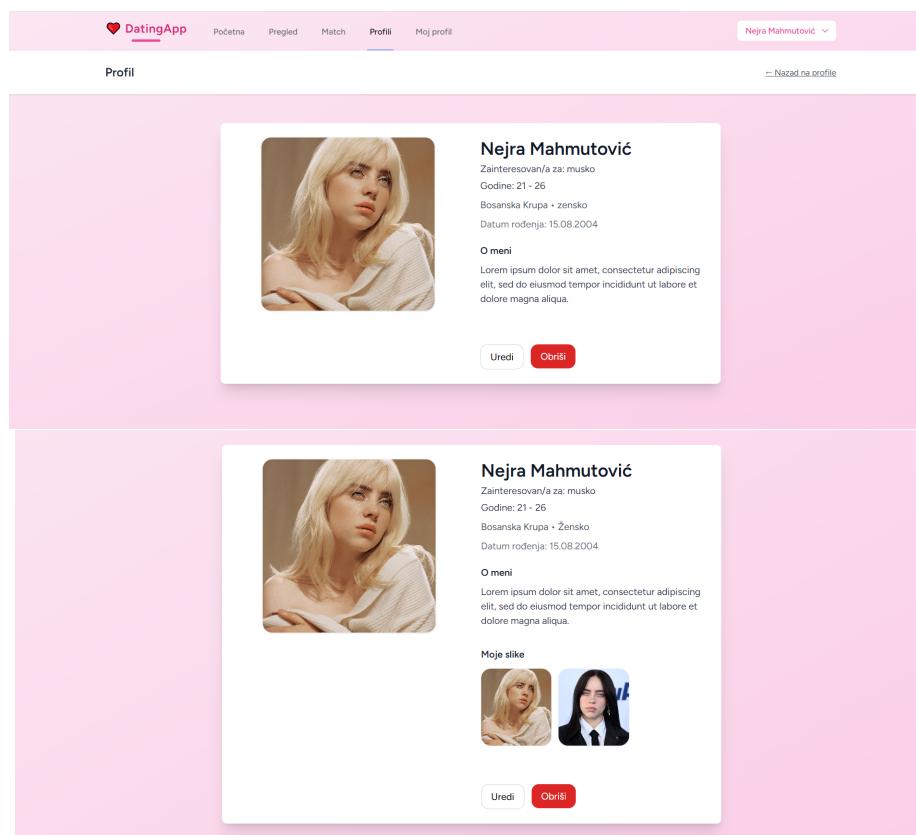
Slika 8: Stranica za like i dislike

- Sporedni scenarij SK:
  - Sesija je istekla i korisnik se vodi ponovno na stranicu za prijavu.

### Slučaj korištenja 6: Moj profil

- Naziv SK: Pregled vlastitog profila
- Akteri SK: Korisnik
- Učesnici SK: Korisnik i sistem
- Preduslov: Korisnik je prijavljen i ima kreiran profil
- Osnovni scenarij SK:
  - Korisnik otvara stranicu Moj profil.
  - Sistem učitava podatke o korisniku iz baze.
  - Sistem prikazuje profilnu sliku i osnovne podatke.
  - Sistem prikazuje galeriju slika ukoliko postoji.

Prikaz stranice za like i dislike prikazan je na slici 9.



Slika 9: Stranica vlastitog profila

- Sporedni scenarij SK:

- Korisnik nema dodanih slika u galeriji te sistem prikazuje samo profilnu sliku i osnovne podatke.

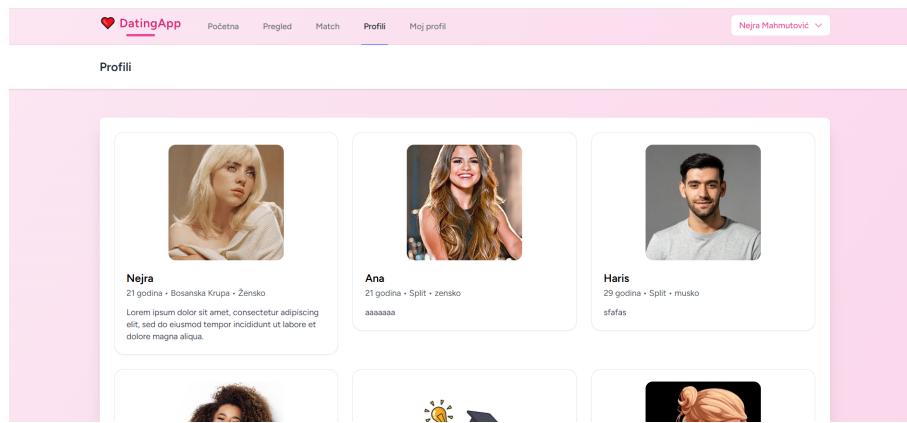
#### **Slučaj korištenja 7: Pregled svih profila**

- Naziv SK: Pregled svih profila
- Akteri SK: Korisnik
- Učesnici SK: Korisnik i sistem
- Preduslov: Korisnik je prijavljen na sistem
- Osnovni scenarij SK:

- Korisnik na navigacionom meniju odabire opciju za pregled svih profila.

- Sistem provjerava da li je korisnik prijavljen.
- Sistem učitava profile iz baze podataka.
- Sistem prikazuje listu svih profila korisnika.

Prikaz stranice za sve profile prikazan je na slici 10.

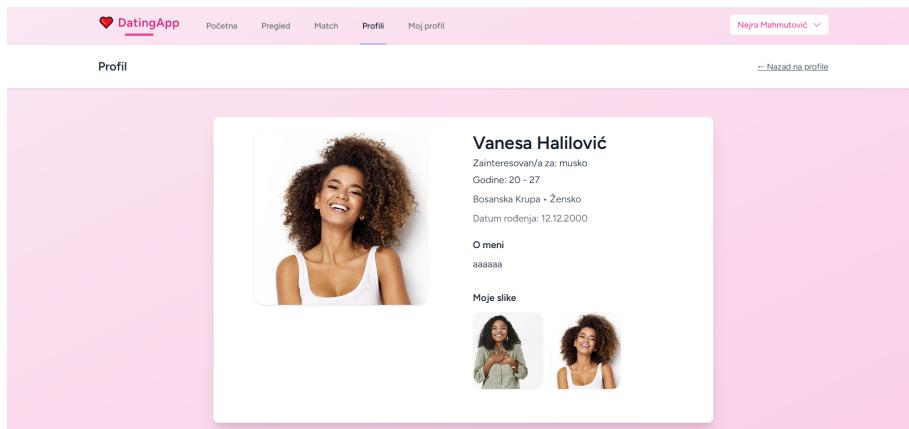


Slika 10: Stranica za sve profile

- Sporedni scenarij SK:

- Sesija je istekla i korisnik se vodi ponovno na stranicu za prijavu.

Ulazak na drugi profil je prikazan na slici 11.

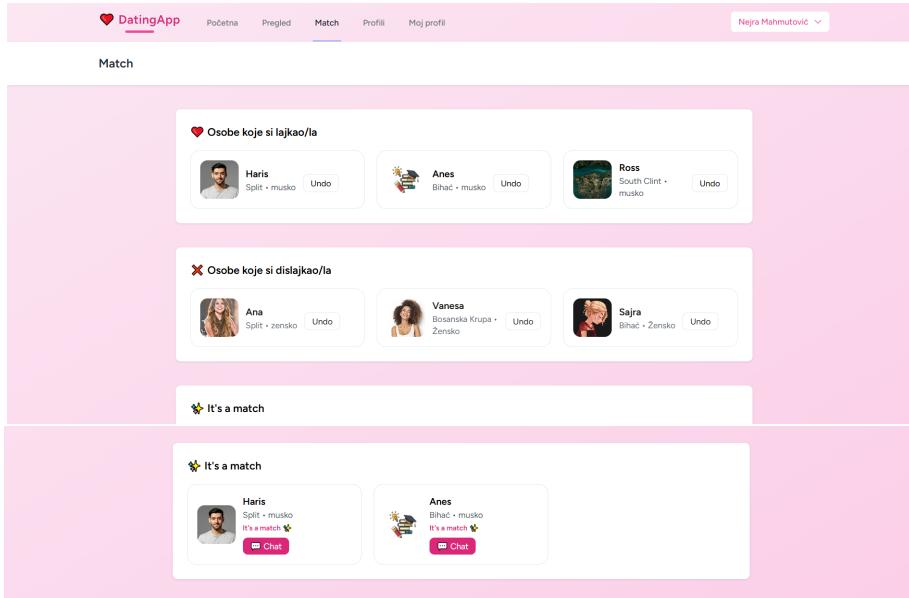


Slika 11: Stranica jednog profila

#### Slučaj korištenja 8: Match sistem

- Naziv SK: Pregled match-eva
- Akteri SK: Korisnik
- Učesnici SK: Korisnik i sistem
- Preduslov: Korisnik je prijavljen i postoje ostvareni match-evi
- Osnovni scenarij SK:
  - Korisnik otvara stranicu Match.
  - Sistem pronalazi obostrane like-ove u bazi podataka.
  - Sistem prikazuje listu match korisnika.
  - Korisnik može otvoriti chat (demo verzija).

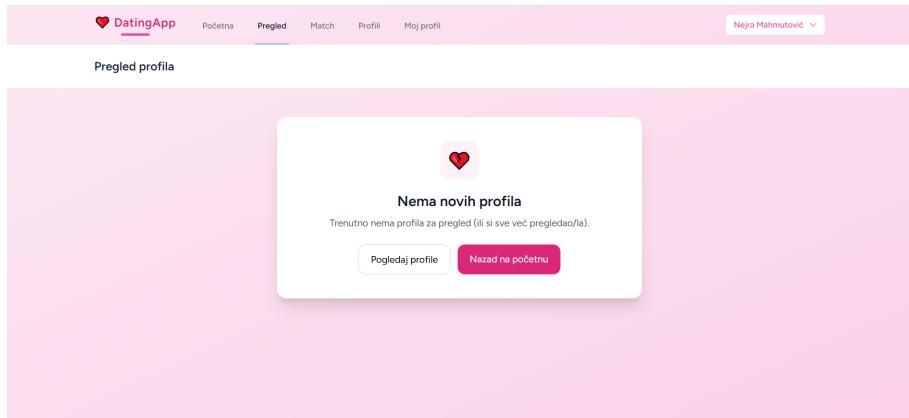
Prikaz stranice za match prikazana je na slici 12.



Slika 12: Stranica za match sistem

- Sporedni scenarij SK:

- K orisnik nema nijedan match te sistem prikazuje odgovarajuću poruku.



Slika 13: Prikaz kada korisnik nema nijedan match

## 5 Zaključak

Razvoj web aplikacije *DatingApp*, namijenjene upoznavanju i povezivanju korisnika putem interneta, predstavlja uspješnu primjenu savremenih web tehnologija u cilju izrade funkcionalnog i preglednog sistema za online upoznavanje. Aplikacija omogućava korisnicima registraciju i prijavu na sistem, kreiranje i uređivanje vlastitog profila, pregled profila drugih korisnika, kao i ostvarivanje međusobnih podudaranja (match-eva) putem sistema lajkova i dislajkova. Na ovaj način omogućeno je jednostavno i intuitivno korištenje aplikacije, uz jasno organizovane funkcionalnosti dostupne prijavljenim korisnicima. Aplikacija je razvijena korištenjem Laravel PHP frameworka, koji omogućava organizovan i strukturiran razvoj web sistema kroz MVC (Model–View–Controller) arhitekturu. Ovakav pristup omogućava jasnu podjelu odgovornosti između poslovne logike, prikaza i upravljanja podacima, čime se olakšava održavanje i daljnje proširenje aplikacije. Za čuvanje podataka korištena je relacijska baza podataka, koja omogućava pouzdano i efikasno upravljanje informacijama o korisnicima, profilima i interakcijama između korisnika. Korištenjem Eloquent ORM sistema omogućen je jednostavan rad sa bazom podataka kroz objektno-orientisani pristup. U okviru razvoja sistema korišteni su UML dijagrami za modeliranje aplikacije, uključujući dijagram slučajeva korištenja, klasni dijagram i ER dijagram baze podataka. Ovi dijagrami omogućili su jasnu vizualizaciju strukture sistema, odnosa između entiteta i načina interakcije korisnika sa aplikacijom. Implementirane funkcionalnosti aplikacije uključuju upravljanje korisničkim profilima, galeriju slika, pregled profila drugih korisnika, like i dislike sistem, automatsko prepoznavanje match-eva, kao i prikaz statističkih podataka i lokacija korisnika putem interaktivne mape. Testiranjem aplikacije potvrđena je ispravnost rada osnovnih funkcionalnosti kao što su registracija korisnika, kreiranje profila, pregled profila i ostvarivanje match-eva. Na osnovu realizovanog projekta može se zaključiti da Laravel predstavlja stabilno i efikasno rješenje za razvoj web aplikacija srednje složenosti, kao što je *DatingApp*. Implementirani sistem pokazuje kako se savremene web tehnologije mogu koristiti za izgradnju funkcionalne i pregledne web aplikacije za upoznavanje korisnika. Projekat predstavlja dobru osnovu za buduća unapređenja, kao što su uvođenje real-time chata između korisnika, naprednije filtriranje profila, sistem notifikacija ili razvoj mobilne verzije aplikacije, čime bi se dodatno unaprijedila funkcionalnost i korisničko iskustvo.

## **Literatura**

- [1] Zagreb, Element.hr, Robert Manger, Softversko inžinjerstvo, 2016.
- [2] Zagreb, Element.hr, Robert Manger, Baze podataka, 2014.
- [3] Indiana University Bloomington, Alan Dennis, Systems Analysis and Design: An Object-Oriented Approach with UML, 2020.
- [4] Rufus Stewart, Laravel: The Ultimate Beginner's Guide to Learn Laravel Step by Step, 2020.
- [5] Grady Booch, The Unified Modeling Language User Guide, 2005.