

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Zadatak: PlayPadel

Tim: T18.1

Ime tima: BADELPADEL

Nastavnik: Vlado Sruk

Cilj

Cilj projekta je razviti web-aplikaciju *PlayPadel* koja služi kao platforma koja će olakšati rezervaciju terena, prijavu i organizaciju turnira za padel. Osim praktičnosti, korisnicima će omogućiti stvaranje sportske zajednice, ali i educirati sve zainteresirane o sve popularnijem sportu – padelu.

Korisnici

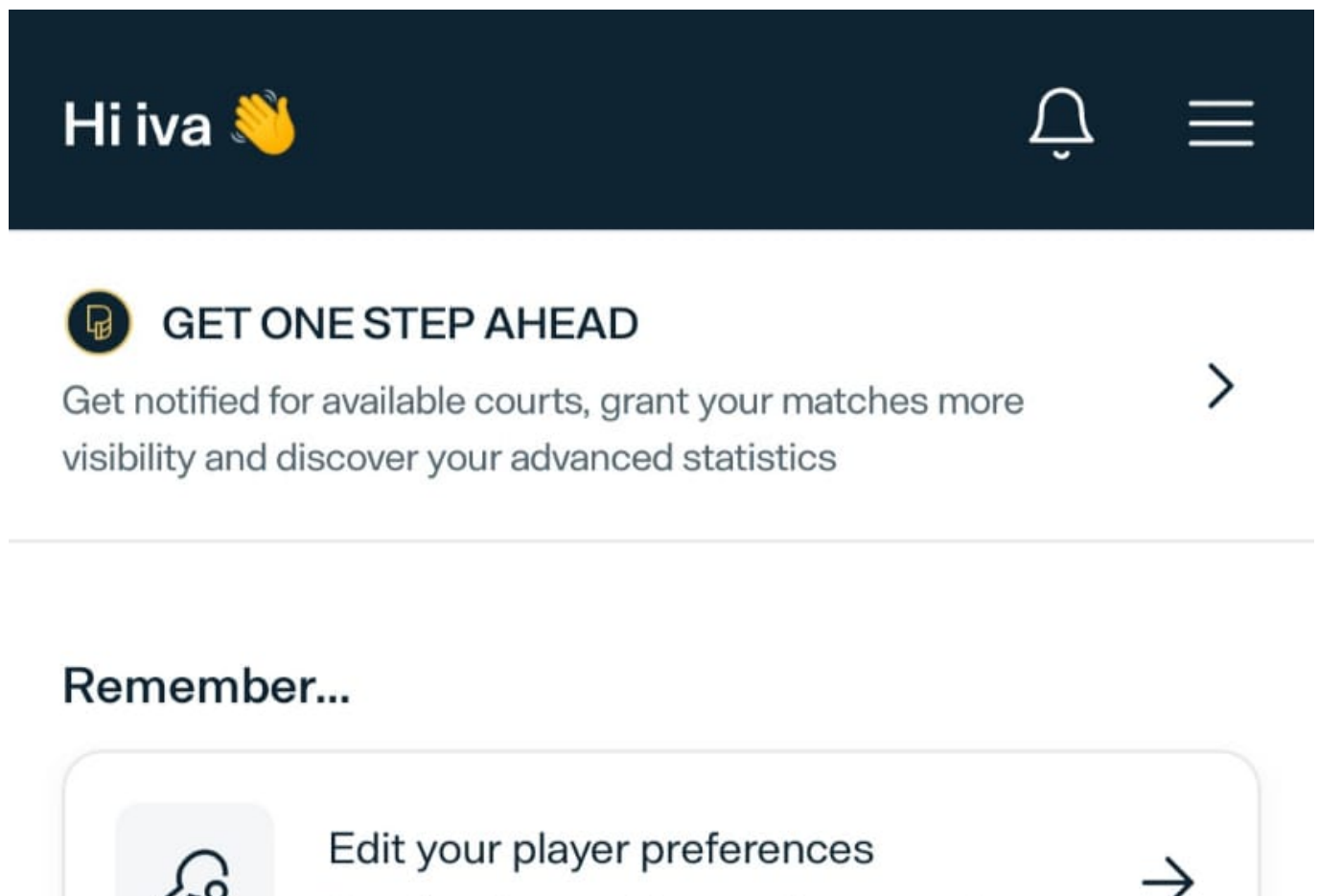
Ovakva web-aplikacija privukla bi primarno igrače padela. Osobe koje traže jednostavan način za rezervaciju termina, sudjelovanje na turnirima i praćenje rezultata. Osim igrača, važna skupina korisnika su i vlasnici dvorana i terena. Oni plaćaju godišnju članarinu kako bi se oglašavali i upravljali rezervacijama i turnirima. Najmanje aktivna skupina ljudi koju bi *PlayPadel* mogao privući su zaljubljenici u sport. Zanimaju ih novosti i događaji u lokalnoj zajednici, u potrazi su za novim hobiem i slično.

Potencijalna korist

Postojanje ove web-aplikacije olakšalo bi rezervaciju i upravljanje terminima, ali i pronalaska potencijalnih partnera za igru. U isto vrijeme, vlasnici terena dobivaju prostor za reklamaciju svojih objekata i mogućnost optimiziranja korištenja svojih kapaciteta. Vlasnici terena bi također profitirali od rezervacija, dok bi *PlayPadel* zarađivao kroz naplatu članstva i potencijalno od postotka zarade dobivene rezervacijama terena i turnirima.

Slična rješenja

Pronašli smo nekoliko sličnih rješenja, ali nijedna aplikacija ne nudi sve mogućnosti koje nudi *PlayPadel*. Najpoznatija slična aplikacija bila bi *Playtomic* koja omogućuje rezervaciju sportskih terena, s funkcijama pretrage po lokaciji i mogućnosti rezervacije termina.



Best hand, court side, match type, preferre...

Play your perfect match



Book a court

If you already know who you are playing with



Play an open match

If you are looking for players at your level



Classes



Competitions



Home



Community



Profile



Vaše ime

Standard account



[Go Premium](#)[Share profile](#)

Your account



Edit profile

Name, email, phone, location, gender, date of bir...



Your activity

Matches, classes, competitions, groups



Your payments

Payment methods, transactions, club member...



Settings

Configure privacy, notifications, security, delete ...



Support



Help



How Playtomic works



Legal information

Osim nje, pronašli smo i *SportyPlus* čiji su primarni korisnici sportski klubovi. Za razliku od PlayPadel, prilagođena je različitim sportovima, ali također omogućuje rezervaciju terena, organizaciju turnira i sl.

The screenshot displays the SportyPlus website, which is a platform for sports clubs. The header includes the logo "SPORTY.PLUS" and navigation links: MODULES, PRICING, DEMO, FAQ, CASE STUDY, CONTACT, and LOGIN. A blue button in the top right corner asks, "Do you want a live presentation?". The main content area features the text "SPORT CLUB MANAGEMENT APPLICATION" and a sub-headline "Tailored just for sports clubs". Below this, it states: "Boost the profitability, efficiency, and simplicity of your club's operations, and enhance the satisfaction and loyalty of your members". There are icons for Apple, Android, Windows, and a mobile phone. The central image shows a smartphone displaying the app's interface, which includes a calendar, a list of events, and a section for "Natjecanja" (Tournaments). The app also lists supported sports: Tennis, Padel, Pickleball, and Badminton, each with a checkmark. A blue button at the bottom of the app interface says "And all other racket sports!". The background of the website features a tennis player in action.

MATCHi je sveobuhvatna platforma za rezervaciju terena i organizaciju aktivnosti u sportovima s reketom, uključujući padel, tenis, badminton, stolni tenis, pickleball i squash.



MATCHi

PLAY MORE

Login to my account

Create a new account

Latest version

Play



Fri

Sat

Sun

Mon

Tue

Wed

Thu

Fri

15
NOV

16
NOV

17
NOV


18
NOV

19
NOV

20
NOV

21
NOV

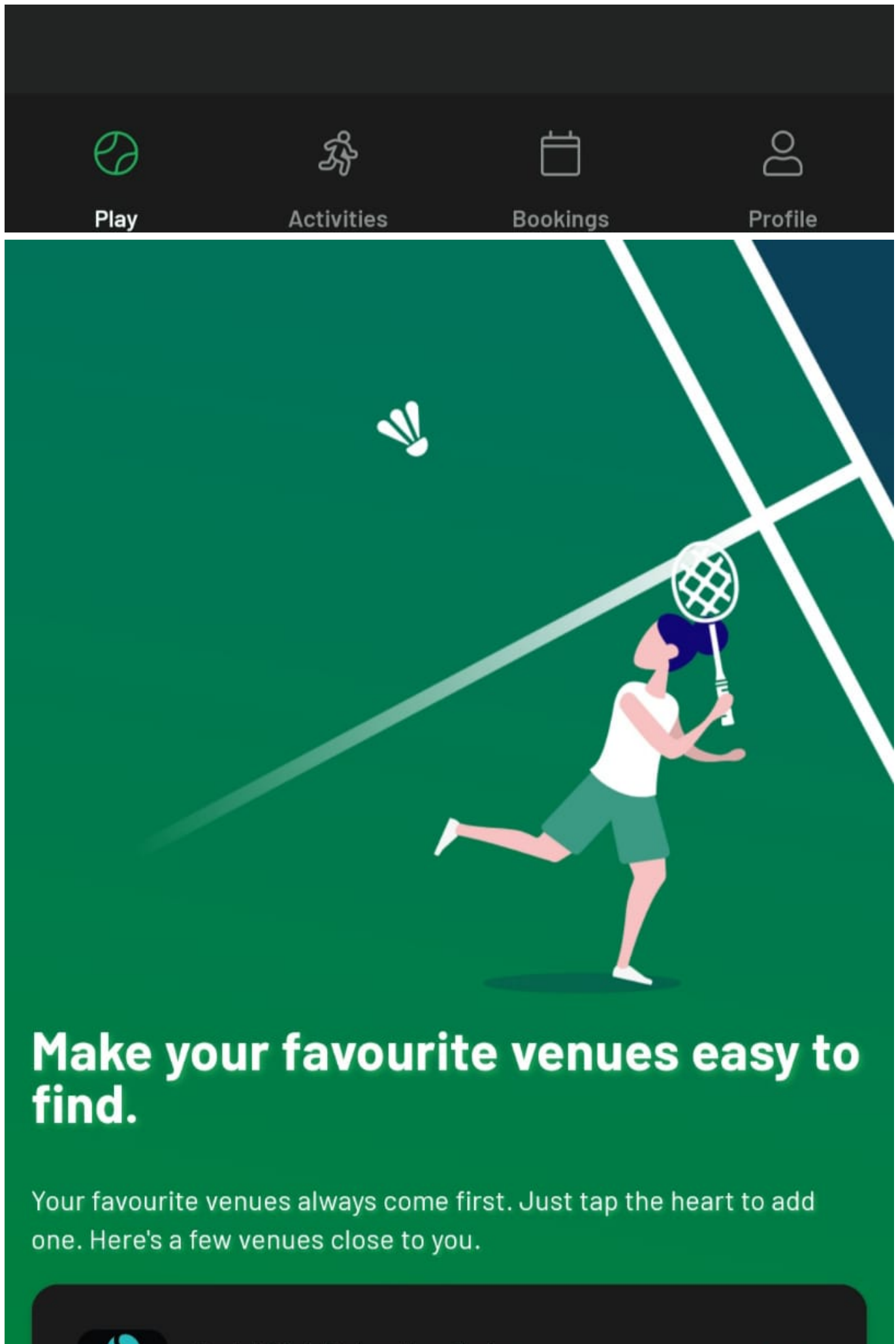
22
NOV

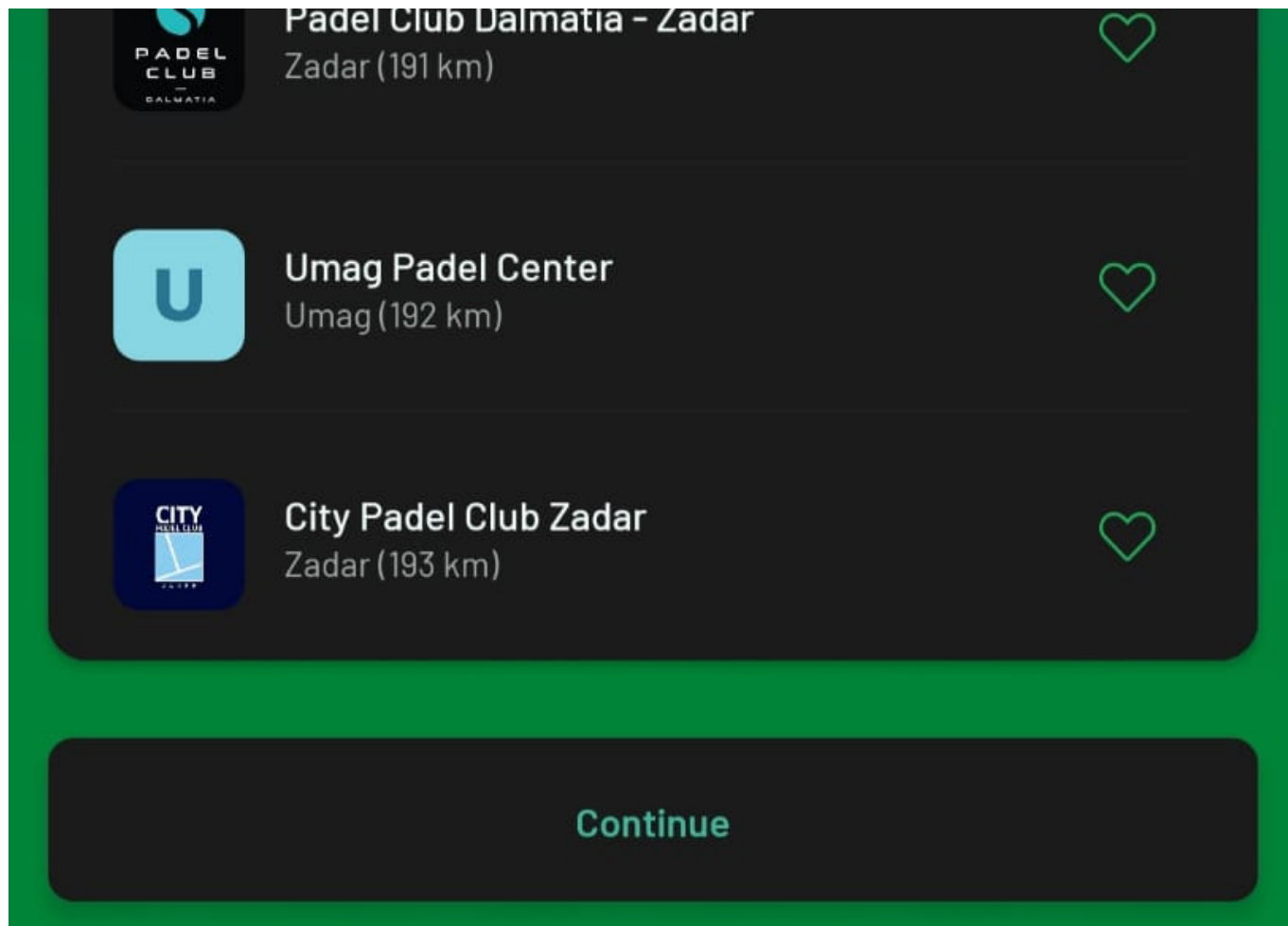


No venues found

We could not find any venues at this location that match your filters.

Change Filters





Mogućnost prilagodbe rješenja

Ovu web-aplikaciju bi bilo moguće prilagoditi i drugim sportovima, jezicima i valutama.

Moguće nadogradnje projektnog zadatka

U budućnosti bismo mogli dodati sustav za ocjenjivanje terena, omogućiti obavjesti, povezati aplikaciju s drugim aplikacijama kao što su MyFitnessPal, ali i fokusirati se više na vlasnike terena te im omogućiti pregled popunjenosti termina, popularnosti turnira i demografskih podataka o igračima. Također bismo im mogli ponuditi pristup statističkim podacima, savjete za organizaciju turnira, mogućnost bolje promocije i slično.

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-001	Sustav omogućuje korisnicima kreiranje profila pomoću e-mail adrese.	Visok	Zahtjev dionika	Korisnik se može registrirati e-mailom, primiti e-mail za potvrdu i uspješno se prijaviti.
F-002	Sustav omogućuje korisnicima odabir uloge prilikom registracije (igrač/vlasnik terena)	Visok	Zahtjev dionika	Korisnik može odabrati ulogu i time dobiti odgovarajuće funkcionalnosti
F-003	Sustav omogućuje vlasnicima terena plaćanje godišnje članarine putem PayPal-a, kreditnih kartica ili gotovine	Visok	Zahtjev dionika	Plaćanje more biti uspješno zabilježeno
F-004	Sustav omogućuje vlasnicima unos osnovnih informacija na javni profil (naziv, adresa, kontakt).	Visok	Zahtjev dionika	Vlasnik može popuniti osnovne podatke koji će biti vidljivi igračima.
F-005	Sustav omogućuje vlasnicima postavljanje detalja o terenu (lokacija, slika, tip terena, termini, cijena).	Visok	Dokument zahtjeva	Vlasnik može dodati potrebne podatke o terenu koji će biti vidljivi igračima.
F-006	Sustav omogućuje igračima pregledavanje dostupnih termina za rezervaciju.	Visok	Povratne informacije korisnika	Korisnik može pregledati listu slobodnih termina prema kriterijima (npr. datum).
F-007	Sustav omogućuje igračima rezervaciju terena putem platforme	Visok	Zahtjev dionika	Korisnik može odabrati termin i rezervirati teren.
F-008	Administratori sustava mogu upravljati korisničkim računima (dodavanje, brisanje, izmjena) te postavljanje cijena članarine.	Visok	Zahtjev dionika	Administrator može upravljati svim korisničkim podacima.
F-009	Sustav omogućuje povezivanje s vanjskim uslugama kalendara (Google Kalendar)	Srednji	Dokument zahtjeva	Termini rezervacija sinkronizirani su s Google Kalendarom

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-010	Sustav omogućuje igračima otkazivanje rezervacije najkasnije 24 sata prije termina.	Srednji	Dokument zahtjeva	Korisnik može otkazati termin i dobiti potvrdu o tome
F-011	Sustav omogućuje vlasnicima unos rezultata završenih turnira	Srednji	Dokument zahtjeva	Vlasnik može unijeti rezultate turnira (koji su kasnije vidljivi igračima)
F-012	Sustav omogućuje igračima pregled rezultata završenih turnira	Srednji	Povratne informacije korisnika	Korisnik može pregledati rezultate turnira u organiziranoj formi
F-013	Sustav omogućuje igračima dodavanje komentara i slika na turnire.	Srednji	Povratne informacije korisnika	Korisnik može objaviti komentar ili sliku na turnirima na kojima je sudjelovao.

Registracija korisnika

Korisnici platforme mogu se registrirati kao igrači ili vlasnici terena:

Igrači: Pristup platformi za igrače je besplatan, omogućujući im korištenje svih funkcija vezanih uz rezervacije, prijave na turnire i praćenje rezultata. **Vlasnici terena:** Vlasnici terena plaćaju godišnju članarinu kako bi mogli oglašavati svoje terene i turnire, upravljati rezervacijama i komunicirati s igračima. Podjela korisnika i javni profili

Platforma razlikuje korisnike prema ulogama:

Vlasnici terena imaju javni profil koji uključuje osnovne informacije kao što su naziv, adresa i kontakt broj. Na svom profilu vlasnici mogu oglašavati dostupne terene i nadolazeće turnire, pružajući pregled mogućnosti za igru. Igrači mogu ostavljati komentare i objavljivati slike s turnira na kojima su sudjelovali, čime dodatno pridonose sportskoj zajednici na platformi. Upravljanje terenima i rezervacijama

Platforma omogućava jednostavno upravljanje rezervacijama i terminima:

Vlasnici terena mogu postavljati detaljne informacije za svaki teren, uključujući lokaciju, fotografije, tip terena (unutarjni ili vanjski), dostupne termine i cijene. Kalendar rezervacija: Integriran s vanjskim kalendarskim uslugama poput Google kalendara, sustav omogućava pregled i rezervaciju termina. Rezervacija termina: Igrači mogu pretraživati dostupne termine i izvršiti rezervaciju kroz platformu, a plaćanje je moguće izvršiti gotovinom, PayPal-om ili kreditnom karticom. Otkazivanje rezervacije: Igrači imaju mogućnost otkazati rezervaciju do 24 sata prije zakazanog termina. Praćenje rezultata i objava s turnira

###PlayPadel podržava praćenje rezultata i interakciju nakon turnira:

Vlasnici terena mogu unositi rezultate završenih turnira i objavljivati fotografije s tih događaja kako bi omogućili igračima uvid u postignuća i atmosferu turnira. Igrači mogu objavljivati komentare i slike vezane uz odigrane mečeve, što dodatno doprinosi dinamici sportske zajednice i motivaciji novih igrača.

Nefunkcionalni zahtjevi

Sigurnost i privatnost

ID zahtjeva	Opis	Prioritet
NF-1.1	Svi podaci o korisnicima (igrači i vlasnici terena) moraju biti zaštićeni enkripcijom prema AES-256 standardu.	Visok
NF-1.2	Korisnički podaci ne smiju se dijeliti s trećim stranama bez njihovog izričitog pristanka, u skladu s GDPR propisima.	Visok
NF-1.3	Sustav mora koristiti sigurnosne protokole (SSL/TLS) za svu komunikaciju s korisnicima.	Visok
NF-1.4	Plaćanja moraju biti procesuirana putem sigurnih vanjskih servisa kao što su PayPal i sustavi kreditnih kartica.	Visok

Kompatibilnost

ID zahtjeva	Opis	Prioritet
NF-2.1	Aplikacija mora biti dostupna i optimizirana za mobilne telefone, tablete i desktop uređaje.	Visok
NF-2.2	Kalendar rezervacija mora biti kompatibilan s popularnim vanjskim servisima (Google Kalendar, Calendar.online)	Visok

Jednostavnost korištenja

ID zahtjeva	Opis	Prioritet
NF-3.1	Korisničko sučelje mora biti intuitivno i jednostavno za upotrebu za sve tipove korisnika, uključujući igrače, vlasnike terena i administratore.	Visok
NF-3.2	Navigacija kroz aplikaciju treba biti jasna i zahtijevati minimalno učenje za nove korisnike.	Srednji

Skalabilnost

ID zahtjeva	Opis	Prioritet
-------------	------	-----------

ID zahtjeva	Opis	Prioritet
NF-4.1	Sustav mora podržavati dodavanje novih funkcionalnosti bez narušavanja postojećih performansi.	Visok
NF-4.2	Sustav mora biti sposoban podržati povećanje broja korisnika i rezervacija s minimalnim utjecajem na performanse.	Visok

Integracija s vanjskim servisima

ID zahtjeva	Opis	Prioritet
NF-5.1	Sustav mora omogućiti integraciju s Google Kalendarom radi sinkronizacije termina rezervacija.	Visok
NF-5.2	Sustav mora podržavati integraciju s PayPal servisom za sigurno procesuiranje plaćanja.	Visok
NF-5.3	Sustav mora podržavati integraciju sa sustavom za kartično plaćanje.	Visok

Dionici sustava *PlayPadel*

Korisnici: Primarni korisnici aplikacije, uključujući igrače i vlasnike terena.

Administratori sustava: Osobe zadužene za održavanje i upravljanje aplikacijom, postavljanje pravila, moderiranje korisnika i određivanje članarina.

Naručitelji: Stranke koje financiraju ili podržavaju razvoj aplikacije, obično tvrtka ili organizacija koja želi podržati padel zajednicu i sportsku infrastrukturu.

Razvojni tim: Programeri, dizajneri, analitičari i testeri zaduženi za razvoj, testiranje i održavanje aplikacije.

Aktori i funkcionalni zahtjevi

Igrači Igrači koriste aplikaciju kako bi rezervirali termine, sudjelovali na turnirima i pratili rezultate. Oni imaju inicijatorsku ulogu jer započinju procese poput rezervacije ili prijave na turnire.

Funkcionalnosti:

Pregledavanje slobodnih termina i rezervacija terena
Prijava na turnire i pregled raspoloživih turnira
Otkazivanje rezervacije do 24 sata prije termina
Komentiranje i postavljanje slika s turnira

Pregledavanje i filtriranje turnira (koji filteri postoje??) Praćenje rezultata završenih turnira

Vlasnik terena Vlasnici terena upravljaju svojim terenima i terminima, organiziraju turnire te komuniciraju s igračima. Oni započinju procese poput postavljanja termina, kreiranja turnira i unosa rezultata.

Funkcionalnosti:

Postavljanje termina za rezervaciju terena, uključujući lokaciju, tip terena, cijene i vrijeme Organizacija turnira s detaljima kao što su datum, lokacija, cijena kotizacije i

Unos rezultata završenih turnira i postavljanje fotografija s događaja

Pregled i odobravanje prijava igrača za turnire Upravljanje profilom, uključujući kontakt podatke i osnovne informacije o terenu

Administrator Administratori su odgovorni za upravljanje pravilima platforme, postavljanje cijena članarina i moderiranje korisnika. Oni ne iniciraju procese izravno, već podržavaju rad sustava.

Fukcionalnosti Postavljanje cijena članarine za vlasnike terena Upravljanje korisničkim računima i moderiranje sadržaja Nadzor nad sigurnosnim postavkama

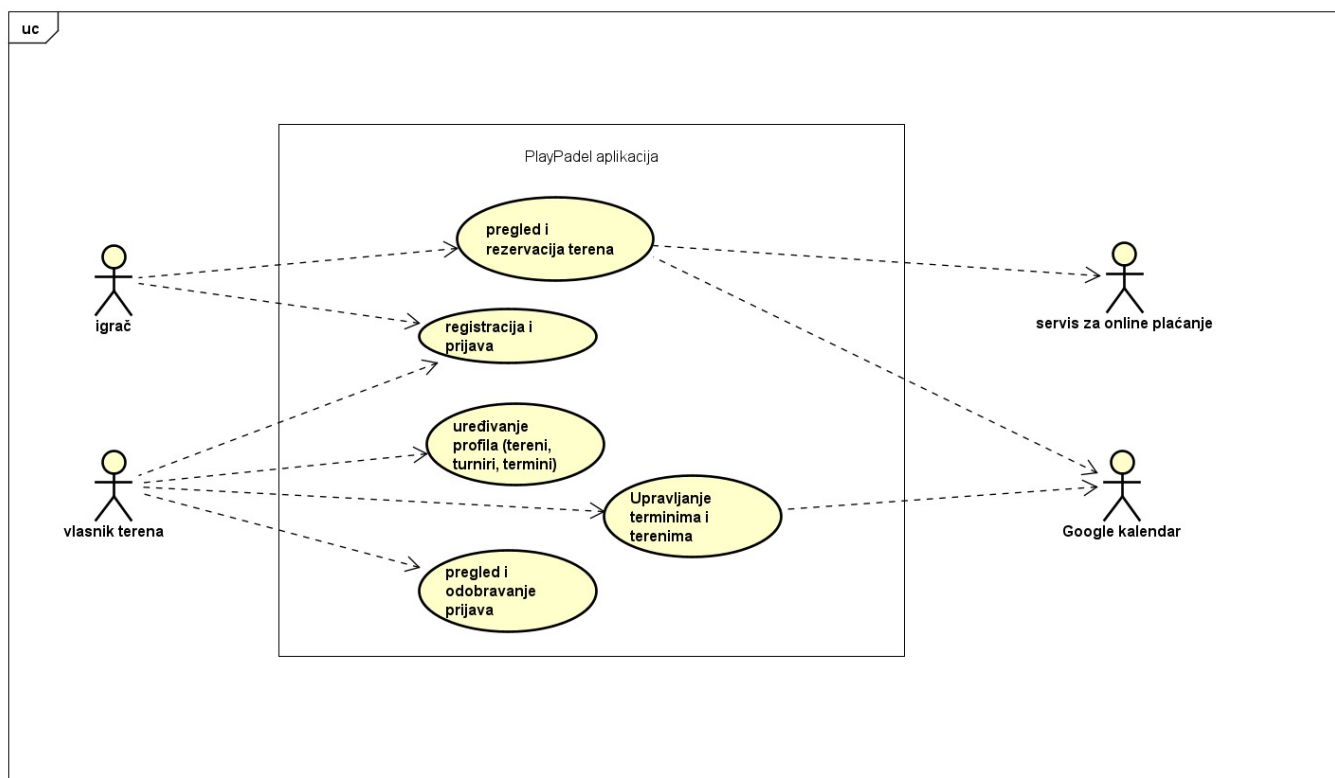
Razvojni tim Razvojni tim je odgovoran za tehnički razvoj i održavanje aplikacije, dodavanje novih funkcionalnosti te praćenje korisničkih povratnih informacija kako bi se unaprijeđivala platforma.

Obrasci uporabe

Opis obrazaca uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

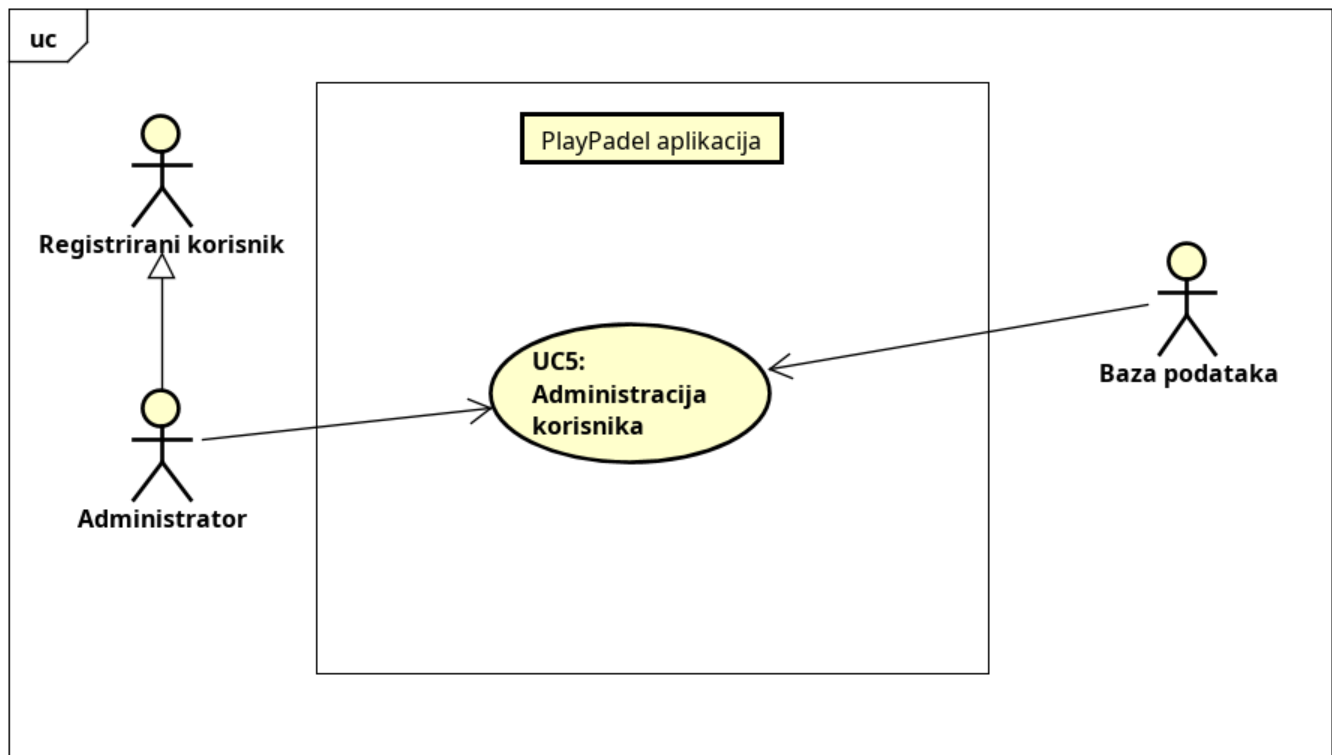
| Prikazuje najvažnije funkcionalnosti sustava iz perspektive krajnjih korisnika. | Slika sustava - osnova za razumijevanje glavnih ciljeva projekta.



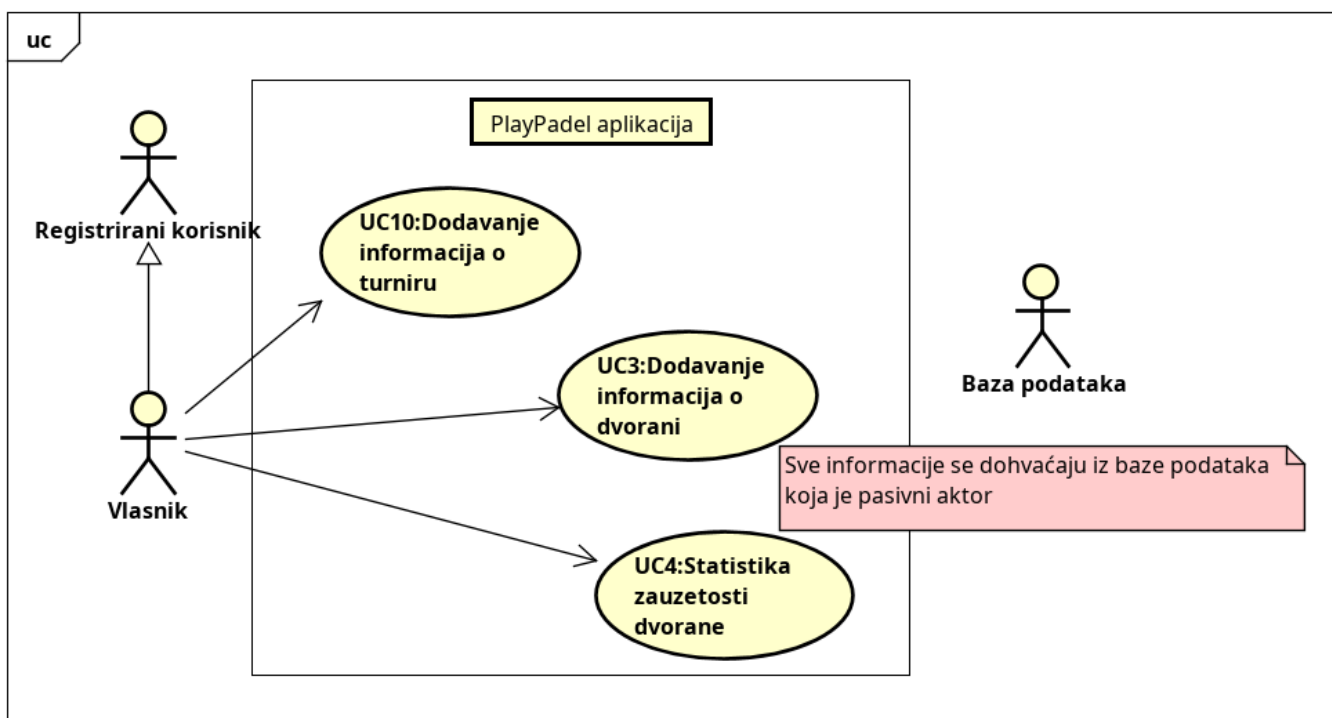
2. dijagram obrazaca uporabe za ključne funkcionalnosti

| detaljnije dijagrame koji se odnose na specifične funkcionalnosti, poput procesa prijave korisnika, upravljanja podacima, ... obrade transakcija. | U ranoj fazi projekta, važno je jasno definirati koje su osnovne funkcionalnosti sustava, omogućava preciznije planiranje razvoj i podijele funkcionalnosti na manje zadatke.

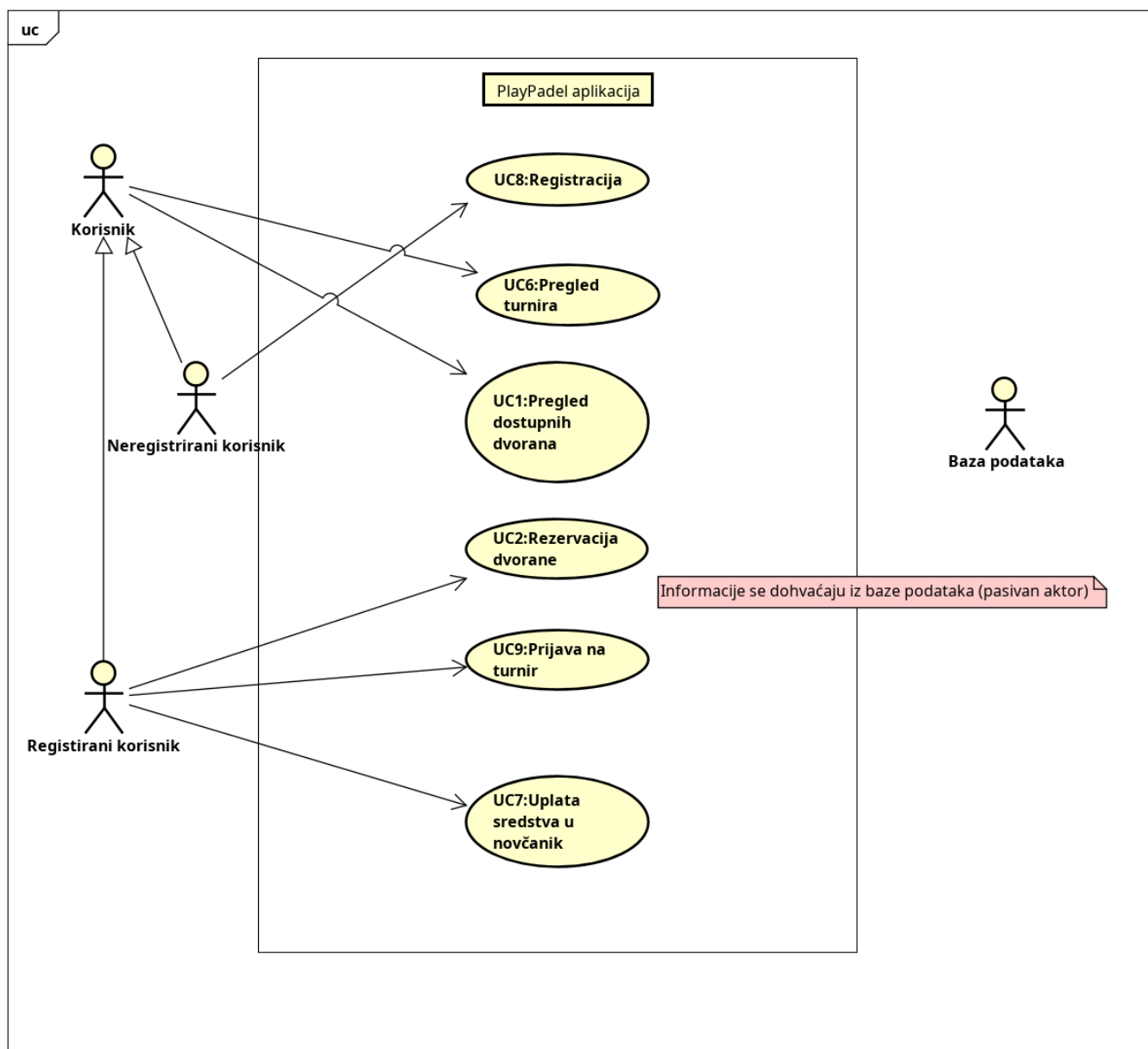
3. dijagram obrazaca uporabe za korisničke role



slika 3.1: Dijagram obrasca uporabe između admina i baze podataka



slika 3.2: Dijagram obrasca uporabe između vlasnika i baze podataka



slika 3.2: Dijagram obrasca uporabe između korisnika i baze podataka

| Identificira glavne korisničke aktore i njihove interakcije sa sustavom. | definira različitih vrsta korisnika i funkcionalnosti koje su im potrebne | Osigurava bolje razumjevanje prioritete i specifične potrebe različitih grupa korisnika.

4. dijagram obrazaca uporabe za osnovne poslovne procese

| prikazuje kako sustav podržava osnovne poslovne procese organizacije.

5. dijagram obrazaca uporabe za kritične sustave i integracije

| interakcije sustava s vanjskim sustavima ili integracijama

Svaki obrazac je potrebno razraditi prema donjem predlošku. Ukoliko u nekom koraku može doći do odstupanja, potrebno je to odstupanje opisati i po mogućnosti ponuditi rješenje kojim bi se tijekom obrasca vratio na osnovni tijek.

UC1 - pregled dostupnih dvorana

- Glavni sudionik: Klijent
- Cilj: Pregledati dostupne dvorane i njihovu zauzetost
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav
- Opis osnovnog tijeka:

1. Klijent ulazi u aplikaciju i pregledava popis sa svim dostupnim dvoranama
2. Klijent bira dvoranu koja ga zanima
3. Aplikacija prikazuje informacije o dvorani i njezinoj zauzetosti

- Opis mogućih odstupanja:

2.a Klijent odabere dvoranu koje trenutno nema dostupnih termina

- Aplikacija obavještava klijenta da nema slobodnih mjesta na odabranoj dvorani

2.b Klijent pokušava pregledati dvoranu koja ne postoji u sustavu.

- Aplikacija obavještava klijenta o nepostojećoj dvorani

UC2 - Rezervacija dvorane

- Glavni sudionik: Klijent
- Cilj: Rezervirati dvoranu za svoj događaj
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav
- Opis osnovnog tijeka:

1. Klijent odabire dvoranu s popisa te željeni datum i vrijeme rezervacije.
2. Aplikacija prikazuje slobodne termine za odabranu dvoranu za navedeni datum i vrijeme.
3. Klijent odabire slobodan termin i potvrđuje rezervaciju.
4. Aplikacija omogućuje plaćanje rezervacije.
5. Nakon uspješne rezervacije, klijent prima potvrdu rezervacije putem e-maila.

- Opis mogućih odstupanja:

1.a Klijent odabire dvoranu koja trenutno nema slobodnih termina.

- Aplikacija obavještava klijenta da nema slobodnih termina u odabranoj dvorani.

3.a Klijent odabire slobodan termin koji u međuvremenu postane zauzet.

- Aplikacija obavještava klijenta o promjeni dostupnosti termina i predlaže novi slobodan termin.

UC3 - Dodavanje informacija o dvorani (za voditelje dvorana)

- Glavni sudionik: Voditelj dvorane
- Cilj: Dodati informacije o svojoj dvorani
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav kao voditelj dvorane
- Opis osnovnog tijeka:

1. Voditelj dvorane unosi informacije o svojoj dvorani, uključujući naziv, opis, fotografiju, cjenik i slično.
2. Voditelj dvorane može definirati kapacitet dvorane i raspoložive resurse (npr. projektor, ozvučenje).
3. Voditelj dvorane postavlja mogućnost rezervacije i unosi informacije o terminima dostupnim za rezervaciju.

- Opis mogućih odstupanja:

1.a Voditelj dvorane pokušava dodati informacije o dvorani koja već postoji u sustavu.

- Aplikacija obavještava voditelja o već postojećim informacijama i omogućava izmjenu postojećih podataka.

3.a Voditelj dvorane pokušava dodati termine ili resurse za nepostojeću dvoranu.

- Aplikacija obavještava voditelja o nepostojećoj dvorani i sugerira unos postojećih podataka o dvorani.

UC4 - Statistika zauzetosti dvorane

- Glavni sudionik: Voditelj dvorane
- Cilj: Pregledati statistiku zauzetosti dvorane i termina kroz vrijeme
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav kao voditelj dvorane
- Opis osnovnog tijeka:

1. Voditelj dvorane bira dvoranu za koju želi pregledati statistiku.
2. Aplikacija prikazuje grafički prikaz statistike zauzetosti dvorane i rezerviranih termina tijekom vremena.

UC5 - Administracija korisnika

- Glavni sudionik: Administrator
- Cilj: Upravljanje korisnicima i njihovim osobnim podacima
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav kao administrator
- Opis osnovnog tijeka:

1. Administrator pregledava popis svih registriranih korisnika.
2. Administrator može mijenjati osobne podatke korisnika.

- Opis mogućih odstupanja:

2.a Administrator pokušava izmijeniti podatke korisnika na nedozvoljen način.

- Sustav obavještava administratora o neispravnoj izmjeni podataka i onemogućava ju.

UC6 - Prikaz dostupnih turnira

- Glavni sudionik: Klijent

- Cilj: Pregledati dostupne turnire
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav
- Opis osnovnog tijeka:

1. Klijent pregledava dostupne dvorane za sastanke na popisu ili karti.
2. Aplikacija prikazuje informacije o dvoranama za sastanke i ukupnom broju slobodnih termina.

- Opis mogućih odstupanja:

1.a Klijent pregledava dvoranu za sastanke koja trenutno nema slobodnih termina.

- Aplikacija obavještava klijenta da nema slobodnih termina za odabranu dvoranu.

UC7 - Uplata sredstava u novčanik

- Glavni sudionik: Klijent
- Cilj: Nadopuniti novčanik sredstvima za plaćanje rezervacija dvorane
- Sudionici: Bankovni sustav
- Preduvjet: Prijava u sustav
- Opis osnovnog tijeka:

1. Klijent odabire opciju za uplatu sredstava.
2. Klijent unosi iznos koji želi uplatiti.
3. Aplikacija preusmjerava korisnika na sigurnu stranicu za plaćanje gdje unosi bankovne podatke.
4. Nakon uspješne uplate, sredstva se dodaju u novčanik korisnika.

- Opis mogućih odstupanja:

4.a Klijent pokušava izvršiti uplatu, ali bankovni sustav ne uspijeva obraditi transakciju.

- Klijent prima obavijest o neuspjeloj uplati i dobiva priliku ponovno pokušati uplatu.

UC8 - Registracija

- Glavni sudionik: Neregistrirani korisnik
- Cilj: Stvoriti korisnički račun za pristup aplikaciji
- Sudionici: Baza podataka
- Preduvjet: -
- Opis osnovnog tijeka:

1. Korisnik odabire opciju za registraciju.
2. Korisnik unosi potrebne korisničke podatke.
3. Korisnik prima obavijest o uspješnoj registraciji.

- Opis mogućih odstupanja:

2.a Korisnik odabire već zauzeto korisničko ime i/ili e-mail, unosi podatak u nedozvoljenom formatu ili pruža neispravan e-mail.

- Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju.
- Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije.

UC9 - Prijava na turnir

- Glavni sudionik: Klijent
- Cilj: Prijaviti se na otvoreni turnir
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav
- Opis osnovnog tijeka:

1. Klijent odabire odabire prijavu na turnire
2. Aplikacija prikazuje otvorene turnire
3. Klijent odabire turnir i šalje prijavu
4. Nakon uspješne prijave, klijent prima potvrdu prijave putem e-maila.

- Opis mogućih odstupanja:

4.a Prijava na turnir je neuspješna

- Aplikacija obavještava klijenta da prijava nije uspjela

UC10 - Dodavanje informacija o turniru (za voditelje dvorana)

- Glavni sudionik: Voditelj dvorane
- Cilj: Dodati informacije o turniru
- Sudionici: Baza podataka
- Preduvjet: Prijava u sustav kao voditelj dvorane
- Opis osnovnog tijeka:

1. Voditelj dvorane unosi informacije o svojem turniru, uključujući naziv, lokacija, datum, cijena kotizacije, nagrada, opis
2. Voditelj dvorane postavlja mogućnost prijave na turnir

- Opis mogućih odstupanja:

1.a Voditelj dvorane pokušava dodati informacije o turniru koji već postoji u sustavu

- Aplikacija obavještava voditelja o već postojećim informacijama i omogućava izmjenu postojećih podataka.

1.b Voditelj dvorane pokušava dodati turnir za nepostojeću dvoranu

- Aplikacija obavještava voditelja o nepostojećoj dvorani i sugerira unos postojećih podataka o dvorani.

UC11 - Prijava u sustav

- Glavni sudionik: Korisnik
- Cilj: Prijaviti se u sustav
- Sudionici: Baza podataka
- Preduvjet: Registracija
- Opis osnovnog tijeka:

1. Korisnik upisuje korisničko ime i lozinku

2. Sustav provjerava ispravnost unesenih podataka

3. Uspješna prijava

- Opis mogućih odstupanja:

1.a Korisnik nije registriran

- Aplikacija obavještava korisnika o nepostojećem korisničkom računu

2.a Podatci za prijavu nisu ispravni

- Aplikacija obavještava korisnika o neispravnom pokušaju

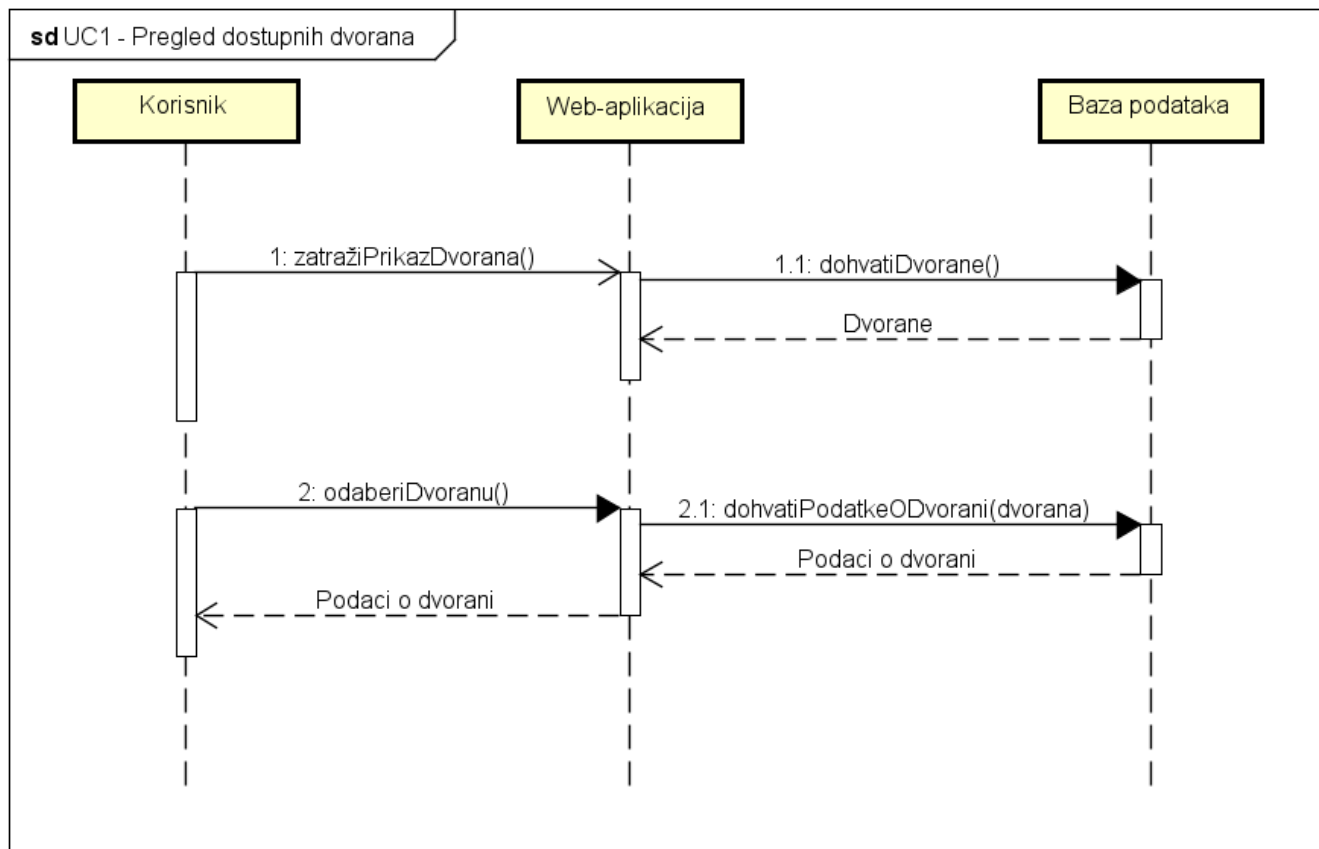
Dijagrami obrazaca uporabe

Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.

Sekvencijski dijagrami

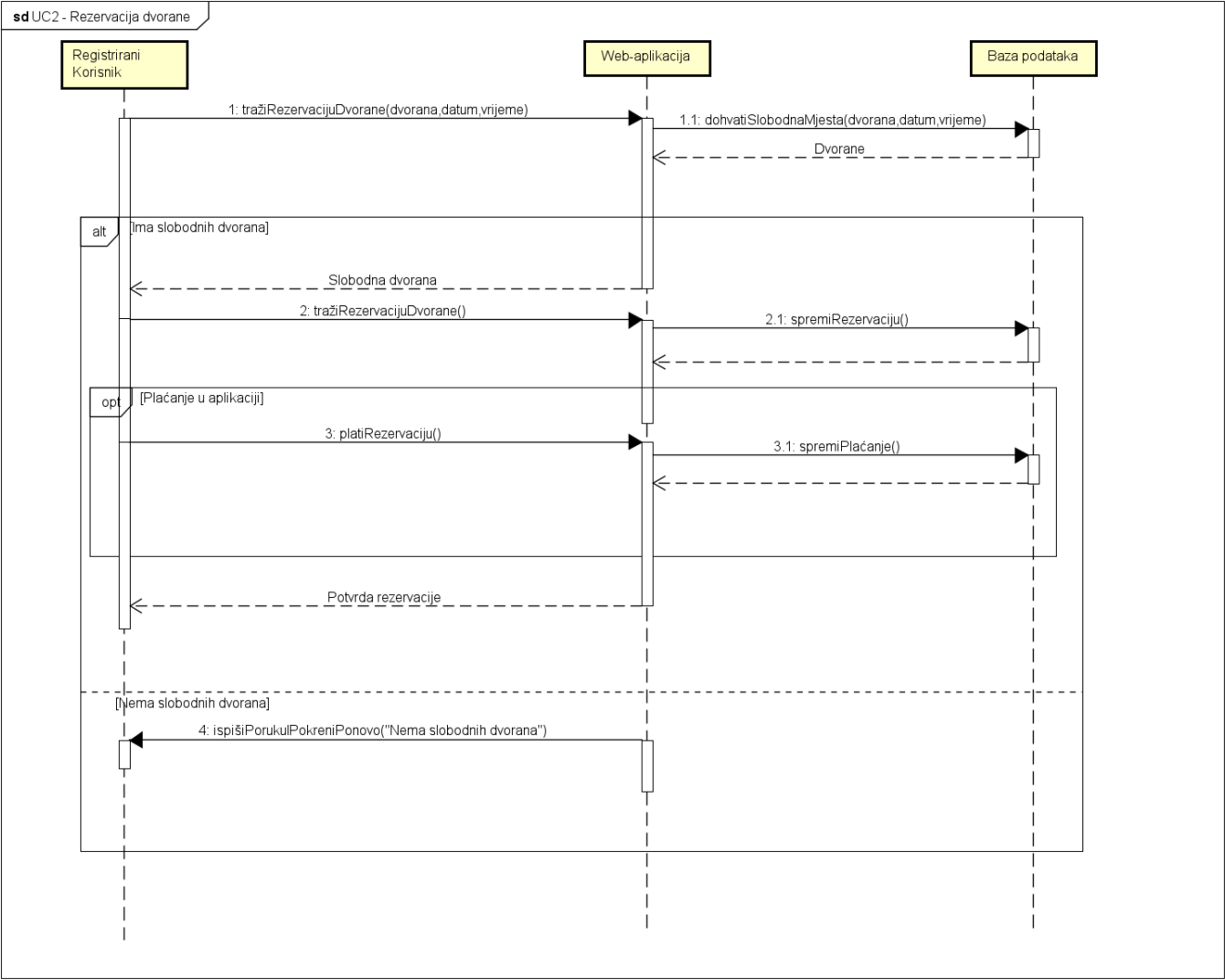
Obrazac uporabe UC1 - pregled dostupnih dvorana

Klijent šalje zahtjev za kartografskim prikazom s dostupnim dvoranama kako bi odabrao dvoranu. Aplikacija dohvaća trenutne podatke o svim dvoranama iz baze podataka i prikazuje ih korisniku. Nakon što klijent odabere dvoranu na karti, aplikacija šalje upit bazi podataka kako bi dohvatila osnovne informacije o odabranoj dvorani. Baza podataka odgovara na upit i šalje informacije o dvorani i zauzetosti dvorane, što aplikacija prikazuje korisniku.



Obrazac uporabe UC2 - Rezervacija dvorane

Klijent šalje zahtjev za rezervaciju parkirališta odabirom parkirališta na karti i specificiranjem datuma i vremena rezervacije. Aplikacija šalje upit bazi podataka kako bi provjerila dostupnost slobodnih parkirališnih mjesta na odabranoj lokaciji za navedeni datum i vrijeme. Baza podataka provjerava dostupna mjesta i šalje informacije o slobodnim parkirališnim mjestima aplikaciji. Ako nema slobodnih mjesta na odabranom parkiralištu aplikacija obavještava klijenta o tome uz poruku. Aplikacija prikazuje klijentu dostupna slobodna mjesta i omogućuje odabir. Klijent bira slobodno parkirališno mjesto i potvrđuje rezervaciju. Aplikacija šalje upit bazi podataka za rezervaciju parkirališta. Baza podataka rezervira parkiralište i šalje potvrdu rezervacije aplikaciji. Aplikacija omogućuje klijentu plaćanje rezervacije. Aplikacija šalje potvrdu plaćanja bazi podataka. Baza podataka ažurira status rezervacije i potvrđuje plaćanje. Aplikacija šalje potvrdu rezervacije klijentu. Klijent prima potvrdu rezervacije.



Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava. Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.

| izvorne kodove dijagrama pohranjujte u svom GIT-u

Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

| za prikazane obrasce uporabe tablično navedite koje funkcionalne zahtjeve obuhvaćaju

Aplikacija je podjeljena na više slojeva koji su međusobno povezani. Imamo:

1. Klijenstka strana (Frontend)

- Uglavnom se sastoji od HTML, CSS-a i JavaScripta. Za izradu korisničkog sučelja koristimo React. Za upravljanje navigacijom između stranica koristi se React Router.

2. Poslužiteljska strana (Backend)

- Izvršava se na web poslužitelju i na njemu se nalazi većina logike: upravljanje bazom podataka, obrada zahtjeva korisnika, autentifikacija, itd. Mi smo izabrali Django, framework baziran na Python-u. Koristimo Django Email Framework za slanje obavijesti o rezervacijama i turnirima, a za upravljanje prijavama i registracijama Django Allauth.

3. Komunikacija između klijenta i poslužitelja

- Komunikacija se odvija putem HTTP protokola. Pri svakoj radnji korisnika koja zahtjeva odgovor od poslužitelja (slanje obrazaca) web poslužitelj obrađuje taj zahtjev i šalje odgovor natrag korisniku. Koristimo REST API koji omogućava razmjenu podataka između klijenta i poslužitelja.

4. Povratna informacija korisniku

- Nakon što poslužitelj obradi korisnikov zahtjev, klijentska strana dobiva odgovor koji se potom prikazuje korisniku. Ovaj proces uključuje dohvaćanje i prikaz podataka u korisničkom sučelju koristeći HTML, CSS i JavaScript i slično.

Arhitektura aplikacije

- temeljena na Django MVT konceptu, primarno zbog njegove sposobnosti zasebnog razvoja pojedinih dijelova aplikacije.

1. Model

- organizacija podataka unutar baza i definiranje njihovih međusobnih odnosa i međudjelovanja.

2. Views

- posrednik između Model-a i Template-a. Sadržava poslovnu logiku pri zahtjevu za prikaz stranice od korisnika. Taj zahtjev se prosljeđuje njemu te on prema isprogramiranoj logici odlučuje koje podatke će zatražiti od baze podataka i koje od njih treba prosljediti dalje.

3. Template

- podrazumijeva sve što se vizualno prikazuje korisniku. Sadrži informacije koje se prosljeđuju web klijentu, uglavnom HTML dokumente.

Baza podataka

Za potrebe razvoja aplikacije BadelPadel koristi se relacijska baza podataka. Osnovne zadaće baze podataka su pohrana i organizacija podataka te brzo pretraživanje i dohvaćanje podataka kako bi ih se moglo dalje

obraditi. Svaki je entitet korištene relacijske baze naveden i opisan u daljnjem tekstu, a ispod opisa nalazi se tablični prikaz opisanog entiteta i njegovih atributa.

Baza podataka aplikacije BadelPadel sastoji se od sljedećih entiteta:

- Note (Tablica: note)
- Vlasnik (Tablica: vlasnik)
- Igrac (Tablica: igrac)
- Teren (Tablica: teren)
- Turnir (Tablica: turnir)
- Post (Tablica: post)
- Komentar (Tablica: komentar)
- TurnirPrijava (Tablica: turnirprijava)
- Termin (Tablica: termin)
- ZauzetiTermin (Tablica: zauzetitermin)

Opis tablica

tablica Note

Atribut	Tip podatka	Opis varijable
title	VARCHAR	Naslov bilješke, maksimalna duljina je 100 znakova.
content	TEXT	Sadržaj bilješke.
created_at	DATETIME	Datum i vrijeme kada je bilješka kreirana
author	INT	ID korisnika koji je autor bilješke (referenca na User)

Entitet note sadrži attribute title, content i created at i author. Veže se na tablicu user putem atributa user, tj author je zapravo ID korisnika.

Vlasnik

Atribut	Tip podatka	Opis varijable
user	INT	ID korisnika (referenca na tablicu User).
adresa	VARCHAR	Adresa vlasnika, maksimalna duljina je 100 znakova
slika	VARCHAR	URL slike vlasnika (opcionalno)
telefon	VARCHAR	Broj telefona vlasnika, maksimalna duljina je 20 znakova

Entitet Vlasnik sadrži attribute User (foreign key), Adresa, Slika, Telefon. Služi za upravljanje informacijama o vlasnicima terena. Veže se na tablicu User.

Igrač

Atribut	Tip podatka	Opis varijable
adresa	VARCHAR	Adresa vlasnika, maksimalna duljina je 100 znakova
slika	VARCHAR	URL slike vlasnika (opcionalno)
telefon	VARCHAR	Broj telefona vlasnika, maksimalna duljina je 20 znakova

Entitet Igrač sadrži attribute Adresa, Slika i Telefon, automatski mu se generira ID. Tablica ima istu strukturu kao Vlasnik, ali je specifična za igrače. Povezana je sa tablicom User putem atributa Vlasnik.

Teren

Atribut	Tip podatka	Opis varijable
lokacija_grad	VARCHAR	Grad u kojem se nalazi teren, max duljina je 100 znakova
lokacija_ulica	VARCHAR	Ulica i adresa terena, maksimalna duljina je 200 znakova.
slika	VARCHAR	URL slike terena (opcionalno, pohranjeno u direktoriju teren_images/)
tip	NVARCHAR	Tip terena (unutarnji ili vanjski)
vlasnik	INT	ID korisnika koji je vlasnik terena (referenca na tablicu User).

Entitet Teren sadrži attribute lokacija_grad, lokacija_ulica, Slika, Tip terena i Vlasnik (foreign key). Povezan je sa tablicom User putem atributa Vlasnik.

Turnir

Atribut	Tip podatka	Opis varijable
naziv	VARCHAR	Naziv turnira, maksimalna duljina je 200 znakova
teren	INT	ID terena na kojem se turnir održava (referenca na tablicu Teren).
datum_pocetka	DATETIME	Datum i vrijeme početka turnira.
datum_kraja	DATETIME	Datum i vrijeme kraja turnira.
cijena_kotizacije	DECIMAL	Cijena kotizacije za sudjelovanje na turniru (maksimalno 10 znamenki, 2 decimalne).
nagrade	TEXT	Opis nagrada koje se dodjeljuju na turniru.
opis	TEXT	Dodatni opis ili detalji o turniru.
organizator	INT	ID korisnika koji organizira turnir (referenca na tablicu User).
otvorenost	VARCHAR	Status turnira (otvoren za prijave ili završen).

Entitet Turnir sadrži attribute Naziv, datum_početka, cijena_kotizacije, Nagrade, Organizator, Otvorenost. Povezan je sa tablicom Teren (atribut Teren je foreign key) i tablicom User (User ili ID korisnika je foreign key).

Komentar

Atribut	Tip podatka	Opis varijable
user_id	INT	ID korisnika koji je autor komentara (referenca na tablicu User).
teren_id	INT	ID terena povezanog s komentarom (referenca na tablicu Teren, opcionalno).
turnir_id	INT	ID turnira povezanog s komentarom (referenca na tablicu Turnir, opcionalno).
slika	NVARCHAR	URL slike povezanog s komentarom (pohranjeno u direktoriju post_images/, opcionalno).
naslov	NVARCHAR	Naslov komentara, maksimalna duljina je 100 znakova
opis	NVARCHAR	Sadržaj komentara, maksimalna duljina je 500 znakova.
vrijeme	TIME	Vrijeme objave komentara (zadana vrijednost je trenutno vrijeme).
post_id	INT	ID posta na koji se komentar odnosi (referenca na tablicu Post, opcionalno).

Entitet Komentar sadrži attribute user_id, teren_id, turnir_id, Slika, Naslov, Opis, Vrijeme, post_id. Ova tablica omogućuje korisnicima da ostavljaju komentare. Entitet je povezan sa tablicom User (user_id) i opcionalno može biti povezan sa tablicom Teren ili Turnir.

TurnirPrijava

Atribut	Tip podatka	Opis varijable
user	INT	ID korisnika koji se prijavio na turnir (referenca na tablicu User).
turnir	INT	ID turnira na koji se odnosi prijava (referenca na tablicu Turnir).
status	VARCHAR	Status prijave, maksimalna duljina je 11 znakova (npr. "prihvaćena", "pending").

Entitet TurnirPrijava sadrži attribute User, Turnir i Status. Služi za praćenje statusa prijava igrača na turnire. Povezan je sa tablicom User i Turnir.

Termin

Atribut	Tip podatka	Opis varijable
teren	INT	ID terena za koji je termin rezerviran (referenca na tablicu Teren)
pocetak	DATETIME	Datum i vrijeme početka termina.
kraj	DATETIME	Datum i vrijeme završetka termina.

Atribut	Tip podatka	Opis varijable
cijena	DECIMAL	Cijena termina (maksimalno 10 znamenki, od kojih su 2 decimale).

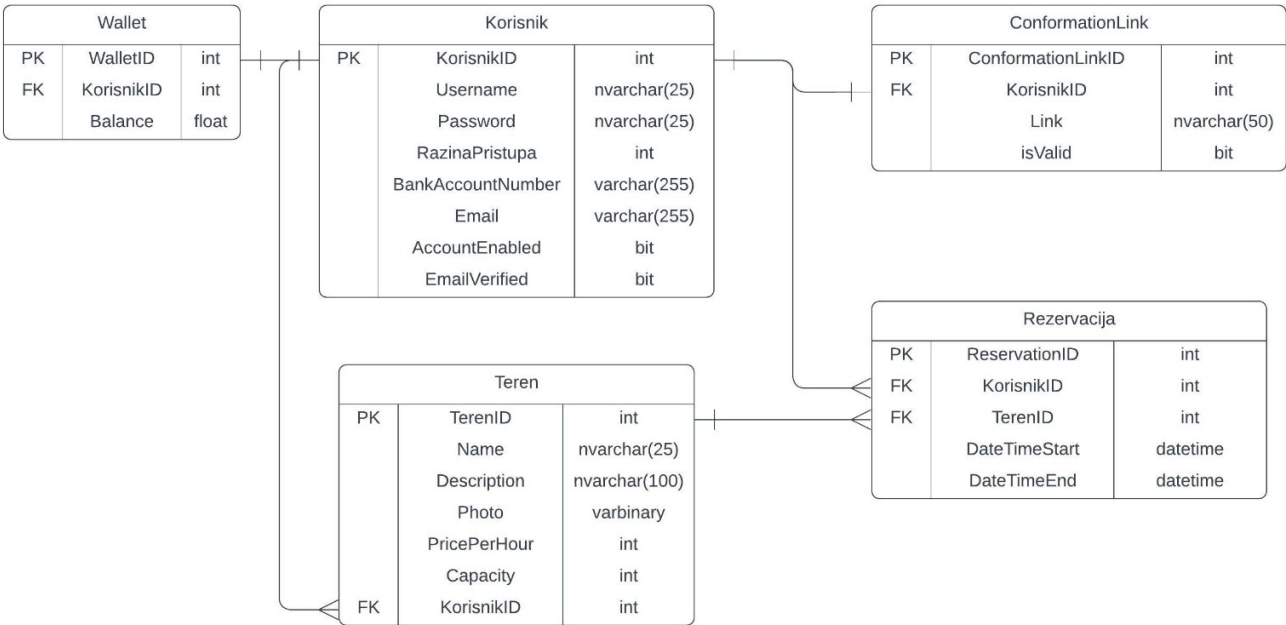
Entitet Termin sadrži attribute Teren, Pocetak, Kraj, Cijena. Povezan je sa tablicom Teren (attribute Teren je Foreign key). Omogućuje korisnicima pregled i rezervaciju slobodnih termina.

ZauzetiTermin

Atribut	Tip podatka	Opis varijable
user	INT	ID korisnika koji je rezervirao termin (referenca na tablicu User).
teren	INT	ID terena za koji je termin rezerviran (referenca na tablicu Teren).
pocetak	DATETIME	Datum i vrijeme početka zauzetog termina.
kraj	DATETIME	Datum i vrijeme završetka zauzetog termina.
cijena	DECIMAL	Cijena zauzetog termina (maksimalno 10 znamenki, od kojih su 2 decimale).

Entitet ZauzetiTermin sadrži attribute User, Teren, Pocetak, Kraj i Cijena. Služi za praćenje informacija o rezerviranim terminima. Povezan je sa tablicama User i Teren.

Dijagram baze podataka




dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

Dijagram stanja

Dijagram stanja za aplikaciju BadelPadel prikazuje moguća stanja korisnika registriranog kao igrač i sustava te prijelaze između tih stanja na temelju različitih događaja. Nakon uspješne prijave korisniku se prikazuje početna stranica koja omogućuje pregled dostupnih padel terena. Odabirom jednog od terena korisniku se prikazuju informacije o terenu i mogućnost odabira dostupnog termina ili turnira. Odabiranjem turnira ili slobodnog termina korisnika se usmjerava na proces plaćanja i obavještava o uspješnosti njegove prijave odnosno rezervacije.

Dijagram stanja


Dijagram aktivnosti

Dijagram aktivnosti prikazuje proces prijave ili registracije korisnika. Korisnik nakon uspješne prijave ili registracije može rezervirati dostupan termin padel terena ili se prijaviti na dostupan turnir. Naposljetku dobiva potvrdu o svojoj rezervaciji.

Dijagram aktivnosti igrača

Dijagram komponenti

Dijagram komponenti prikazuje fizičke i logičke module, te njihova sučelja i kako komponente međusobno surađuju u ostvarenju funkcionalnosti.

Dijagram komponenti

Korištene tehnologije i alati

Komunikacijski alati

Za komunikaciju unutar tima koristili smo aplikaciju [WhatsApp](#) za brze poruke i koordinaciju, te [Google Meet](#) za održavanje virtualnih sastanaka.

Za praćenje promjena u razvoju aplikacije i suradnju koristili smo [GitHub](#) i [GitHub Desktop](#).

Dokumentacija i organizacija

Tijekom pisanja i strukturiranja dokumentacije, bilježaka sa sastanaka te lista zadataka koristili smo [Microsoft Word](#) i [Google Docs](#). Za izradu UML dijagrama koristili smo alat [Astar UML](#).

Tehnologije i alati za izradu aplikacije

Većina koda napisana je u razvojnome okruženju [Visual Studio Code](#), koje je popularno zbog multiplatformske podrške (Windows, Linux, Mac), brzine instalacije i fleksibilnosti u programiranju s raznim jezicima uz pomoć ekstenzija. Primjer korisne ekstenzije za frontend je [Live Server](#), koja omogućuje pregled promjena u kodu u stvarnom vremenu.

Frontend tehnologije

Aplikacija koristi [React](#) za izradu korisničkog sučelja, dok je za brži razvoj primijenjen alat [Vite](#). Pored toga, React koristi virtualni DOM (Document Object Model), što omogućava brzo i efikasno ažuriranje samo onih dijelova stranice koji su se promijenili. Ove tehnologije su proširene dodatnim JavaScript bibliotekama za specifične funkcionalnosti:

[Axios](#) za HTTP zahtjeve. [JWT-decode](#) za dekodiranje JWT tokena. [React-router-dom](#) za upravljanje navigacijom. Date time picker [Google authentication](#) [Rest API](#) [PayPal](#) koji je implementiran po uzoru na kod sa sljedeće [poveznice](#) (frontend) Za odabir palete boja i uređivanje slika koristili smo alate [Canva](#) i [Coolors](#).

Prije kreiranja React aplikacije potrebno je imati instaliran [nodeJS](#) i npm. Npm je upravitelj paketima za programski jezik JavaScript i služi za brzu instalaciju datoteka ili cijelih razvojnih okolina.

Aplikacija je kreirana pomoću sljedećih naredbi

```
npm install
create -react -app
```

Za pokretanje aplikacije potrebno je unutar korijenskog(root) direktorija pokrenuti skriptu start ili run dev

```
cd frontend_root
npm run dev
```

Backend tehnologije

Backend aplikacije izrađen je u radnom okviru [Django](#) programskim jezikom [Python](#). Django je aplikacijski okvir napisan u Python programskom jeziku, besplatan je i otvorenog koda. Django je korišten za implementaciju servera i rad s bazom podataka, a njegova integracija s REST API komponentama omogućila je komunikaciju između frontenda i backenda. Django koristi Model-View-Template (MVT) arhitekturu, tj. dijelovi aplikacije su odvojeni u komponente na temelju njihovih funkcija. Za pohranu podataka koristili smo SQLite, jednostavnu bazu podataka koja ne zahtijeva dodatnu instalaciju i koristi običnu datoteku za spremanje podataka.

Za instalaciju Django potrebno je uspostaviti virtualno okruženje pa instalirati Python i [Pip](#) te naredbom

```
pip install django
```

instalirati Django.

Naredbom `python manage.py runserver` pokreće se Django poslužitelj na localhostu, port 8000

iz {project_root}/src/backend foldera

```
python -m venv ../../.venv
../../.venv/Scripts/activate (na Windows računalima)
source ../../.venv/bin/activate (na Linux računalima)
pip install -r requirements.txt
python manage.py makemigrations api
python manage.py migrate
python manage.py runscript api.create_groups
python manage.py createsuperuser
python manage.py runserver
```

Dodatne Python biblioteke korištene u backendu: [Python virtual environment](#)

- Python virtual environment je alat koji omogućava izolaciju Python okruženja za pojedinačni projekt. To znači da svaka aplikacija ili projekt može imati svoje vlastite biblioteke i zavisnosti, neovisno o drugim projektima ili računalima na kojima se razvija

[Pillow](#)

- Biblioteka Pillow je popularna Python biblioteka za obradu slika i unutar Django aplikacija koristimo za različite zadatke poput učitavanja, obrade i čuvanja slika. U kombinaciji s Django-ovim modelima

[Django REST Framework](#)

- biblioteka za izgradnju REST API-ja unutar Django aplikacija. Olakšava proces razvoja API-ja tako što pruža gotove alate za serijalizaciju podataka, autentifikaciju, upravljanje dozvolama i validaciju. Jedan od ključnih elemenata ove biblioteke su Serializers. Oni pretvaraju Django modele ili druge složene tipove podataka u format JSON koji klijent može koristiti. Također, ova biblioteka omogućava jednostavno kreiranje API endpointa korištenjem klasa kao što su `APIView` i `GenericAPIView` i `ViewSet` što smanjuje potrebu za pisanjem velike količine koda.

[Djangorestframework](#)

[Djangorestframework-simplejwt](#)

[Django-cors-headers](#)

[Psycopg2-binary](#)

[Gunicorn](#)

`django.contrib.admin` - Administracijsko sučelje. `django.contrib.auth` - Upravljanje autentifikacijom. `django.contrib.contenttypes` - Upravljanje tipovima sadržaja. `django.contrib.sessions` - Upravljanje sesijama. `django.contrib.messages` - Obrada poruka.

Docker [Docker](#) je virtualni stroj primjenske razine, a omogućuje pokretanje određene aplikacije unutar slabo izoliranog okruženja koje se naziva spremnik (container) Kontejnerizacija je omogućila jednostavno postavljanje i pokretanje aplikacije na različitim okruženjima.

Hosting Za hosting aplikacije koristili smo besplatnu uslugu dinamičkog DNS-a [DuckDNS](#) koja mapira promjenljivu javnu IP adresu na lako pamtljivu domenu. Server na kojem je aplikacija smještena ustupila nam je računalna sekcija KSET-a, koristeći fizički stroj iznajmljen putem usluge Hetzner.

Komunikacija fronta i enda (klijentski i poslužiteljski dio)

Klijentske web aplikacije komuniciraju s poslužiteljima korištenjem HTTP protokola. Klijentski dio web aplikacije pošalje upit poslužitelju. Za vrijeme čekanja na odgovor, poslužitelj komunicira s bazom podataka. Nekada je potrebno odgovor dodatno pretvarati u JSON objekt, ali budući da smo koristili paket Axios to nije bilo potrebno (pretvaranje u JSON događa se automatski) Axios omogućuje slanje GET, POST, PUT, DELETE i drugih vrsta zahtjeva prema backendu.

Paket Axios instaliran je sljedećom naredbom:

```
npm install axios --save
```

Backend (Django) koristi se za obradu zahtjeva, komunikaciju s bazom podataka i slanje odgovora nazad frontend aplikaciji. U `views.py` datoteci nalaze se modeli koji će se koristiti. tj. na taj način dohvaćamo ili spremamo podatke. U `mapi url.py` definirane su rute. Serijalizatori (`serializers.py`) pretvaraju Python objekte u JSON format koji React aplikacija može čitati.

Django REST Framework obrađuje zahtjev te vraća odgovor u JSON formatu nakon čega React na temelju dobivenih uputa ažurira korisničko sučelje.

Ispitivanje programskog rješenja

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.

Ispitivanje komponenti

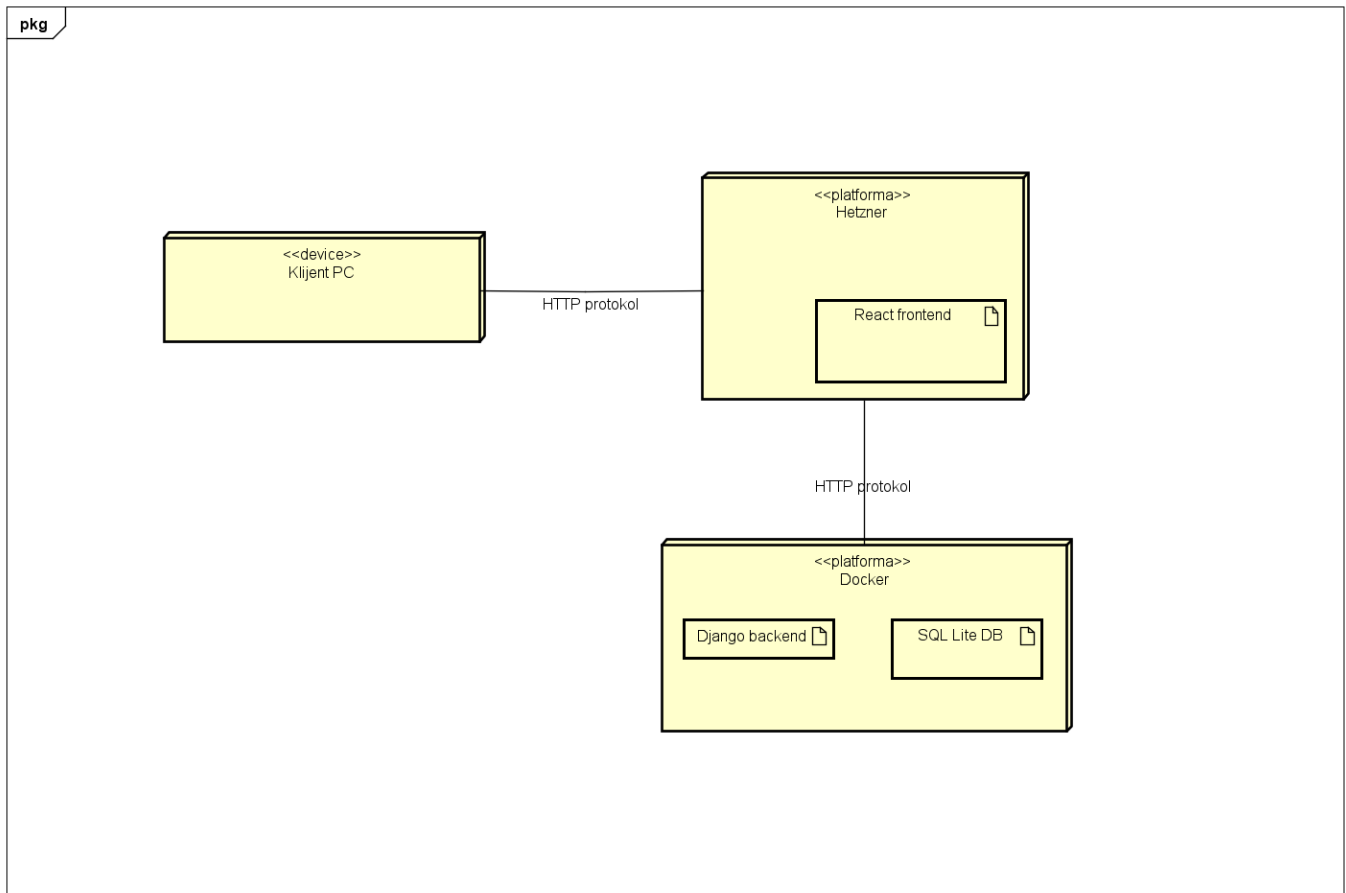
Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi minimalno 6 ispitnih slučajeva u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kod svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).

Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium1. Razraditi minimalno 4 ispitna slučaja u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata. Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

- dodatak za preglednik Selenium IDE - snimanje korisnikovih akcija radi automatskog ponavljanja ispita
- Selenium WebDriver - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje. Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

Dijagram razmještaja



Upute za puštanje u pogon

Lokalno:

Frontend i backend web aplikacije moguće je pokrenuti zasebno (opisano gore). Nakon pokretanja dostupni su na sljedećim adresama <http://localhost:8000> (backend) i <http://localhost:5137>

Backend:

1. Kreirajte virtualno okruženje:

```
python -m venv .venv
```

2. Aktivirajte virtualno okruženje:

Linux/macOS:

```
source ./venv/bin/activate
```

Windows:

```
.\.venv\Scripts\activate
```

3. Smjestite se u direktorij backend:

```
cd ./src/backend
```

4. Instalirajte potrebne pakete:

```
pip install -r requirements.txt
```

5. Provjerite migracije: Ako postoje lokalne migracije koje su slučajno poslane na Git, obrišite sve datoteke u direktoriju BADELPADEL/src/backend/api/migrations, osim **init.py**.

6. Pripremite i izvršite migracije:

```
python manage.py makemigrations api  
python manage.py migrate
```

7. Kreirajte osnovne grupe korisnika:

```
python manage.py runscript api.create_groups
```

8. Kreirajte administratorskog korisnika:

Pokrenite naredbu:

```
python manage.py createsuperuser
```

Slijedite upute na terminalu. Tipične vrijednosti su: Korisničko ime: admin Email: Ostaviti prazno ("") Lozinka: admin Ponovite lozinku: admin

Ako Django upozori na uvjete za lozinku, možete zaobići provjeru unosom y.

9. Pokrenite backend poslužitelj:

```
python manage.py runserver
```

Backend poslužitelj bit će dostupan na: <http://localhost:8000>

Za pristup administratorskoj stranici baze podataka, idite na <http://localhost:8000/admin> i prijavite se s administratorskim korisnikom kreiranim u koraku 8.

Frontend:

1. Smjestite se u direktorij frontend:

```
cd ./src/frontend
```

2. Kreirajte .env datoteku: U direktoriju frontend, napravite datoteku .env i unesite sljedeće podatke:

```
VITE_API_URL = "http://localhost:8000/"  
VITE_GOOGLE_CLIENT_ID = "528593109002-  
9uvvrgh6kkpc4ipapja0ar7fjdjcp05t.apps.googleusercontent.com"  
VITE_PAYPAL_CLIENT_ID =  
"AbSTih5UeQB948SZK07W_zHhQYRDG0wqwixhs21ZMoBdzuSQz8ZDXyBzeqCKwS007oKncWK5Z_j1lCxH"
```

3. Instalirajte potrebne pakete: Smjestite se u direktorij src unutar mape frontend i instalirajte npm:

```
cd ./src  
npm install
```

4. Pokrenite frontend poslužitelj:

```
npm run dev
```

Frontend poslužitelj bit će dostupan na: <http://localhost:5173>

Javno:

Prije pokretanja aplikacije na produkcijskom okruženju, obavezno je dodati sve potrebne API ključeve u .env datoteku frontend i backend komponenti. Ovi ključevi nisu javno dostupni iz sigurnosnih razloga.

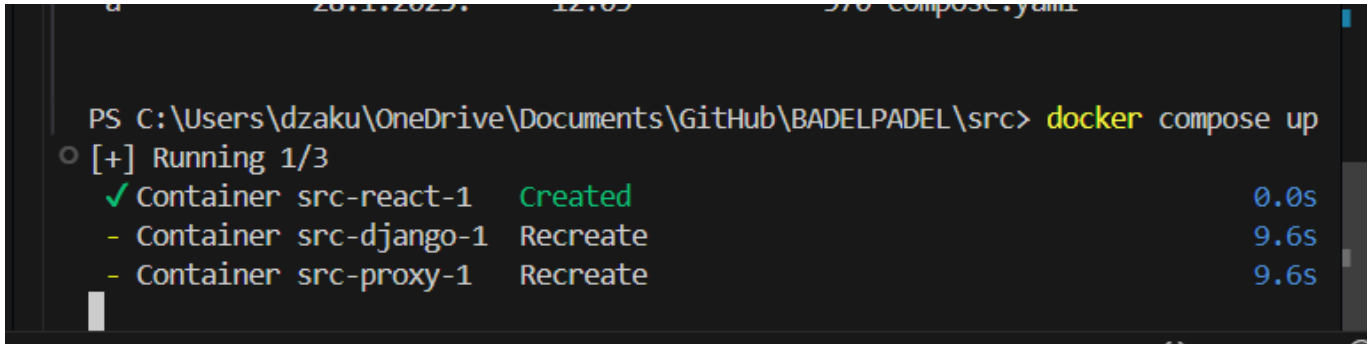
Korak 1: Instalacija Dockera Potrebno je instalirati Docker Desktop prema službenim uputama. Nakon instalacije, pokrećemo terminal i pozicioniramo se u direktorij koji sadrži datoteku compose.yaml.

Korak 2: Pokretanje Docker Kontejnera Pokretanje aplikacije obavlja se naredbom:

```
docker compose up --detach --wait --build --pull always
```

Pomoću docker-compose skripte dižemo tri kontejnera:

Frontend kontejner koji upravlja korisničkim sučeljem. Backend kontejner koji sadrži aplikacijsku logiku. Proxy kontejner za preusmjeravanje zahtjeva između frontend i backend servisa.



```
PS C:\Users\dzaku\OneDrive\Documents\GitHub\BADELPADEL\src> docker compose up
[+] Running 1/3
 ✓ Container src-react-1 Created 0.0s
 - Container src-django-1 Recreate 9.6s
 - Container src-proxy-1 Recreate 9.6s
```

Docker koristi definirane slike iz Dockerfile datoteka za kreiranje kontejnera. Svako novo pokretanje stvara novo (osvježeno) kontejnersko okruženje. Kako bismo zadržali određene podatke (npr. podatke o korisnicima i rezervacijama), koristimo Docker volumene za trajnu pohranu.

Korak 3: Deploy na Server Za pokretanje na serveru slijedimo ove korake:

1. Spojimo se na server pomoću SSH-a

```
ssh korisnik@serverina.kset.org
```

2. Smjestimo se u direktorij s izvornim kodom

```
cd /docker/badelpadel
```

3. Povlačimo najnoviji kod s repozitorija:

```
git pull
```

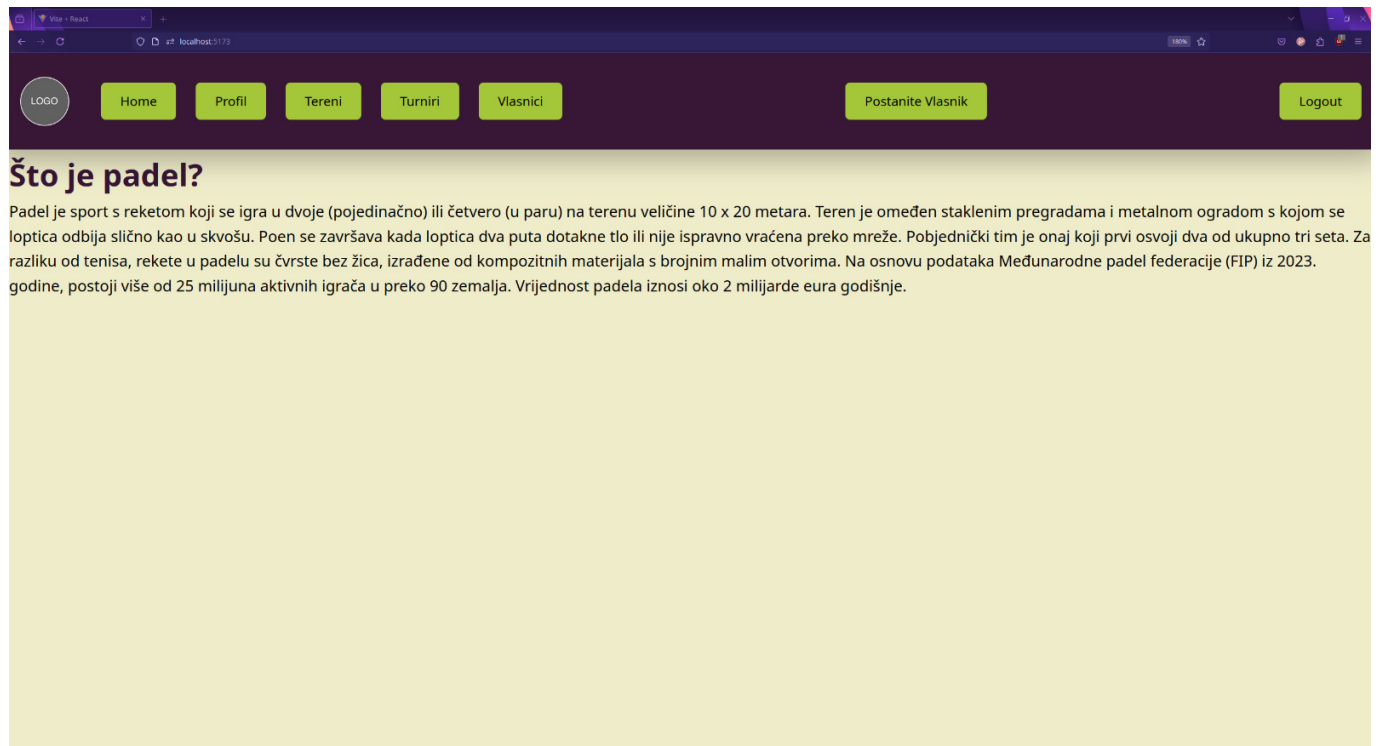
4. Pokrećemo novi deployment:

```
docker compose up --detach --wait --build --pull always // upogonjavanje novog koda
```

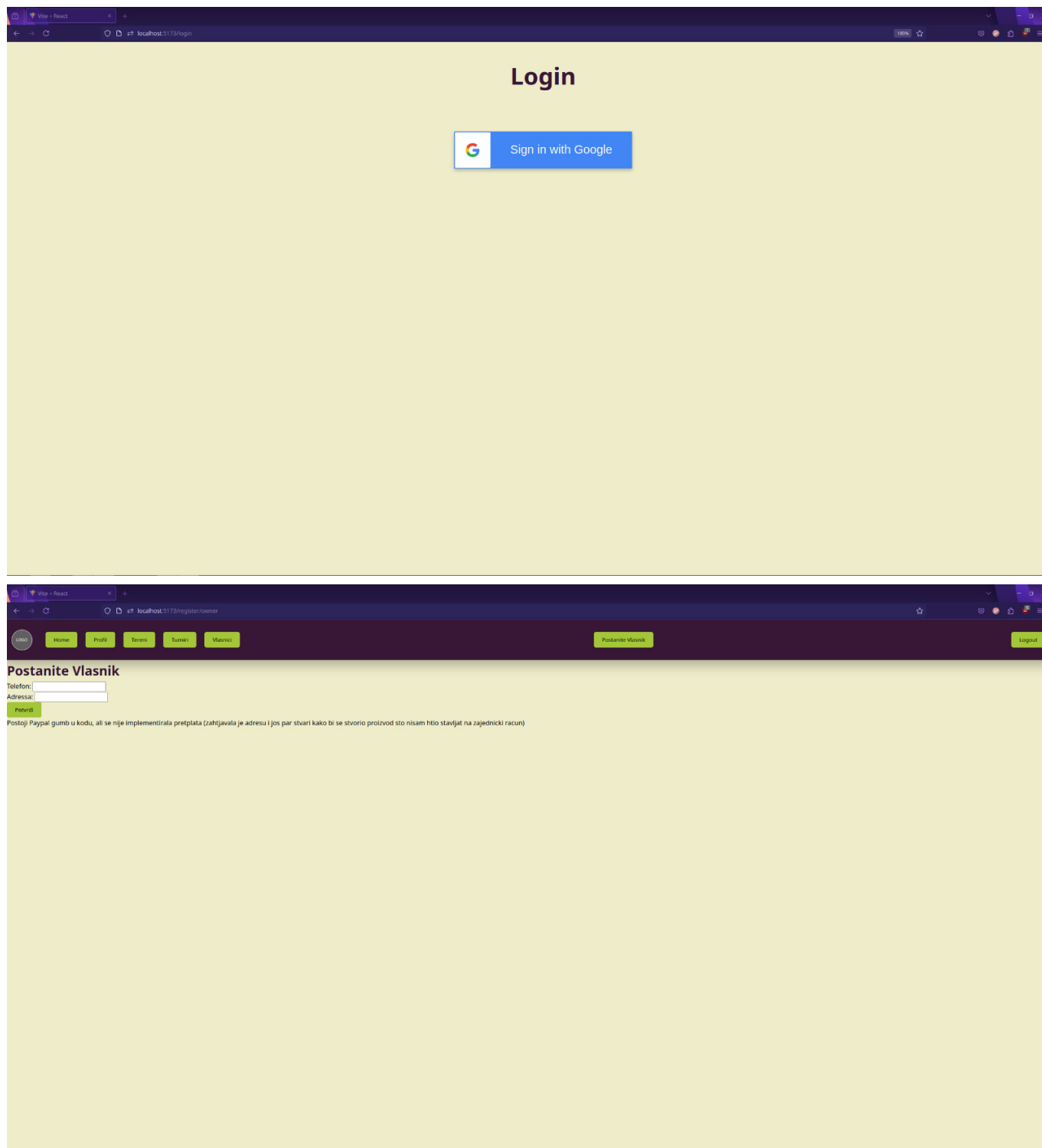
Web aplikacija dostupna je na adresi: <https://badelpadel.duckdns.org>

Slike aplikacije:

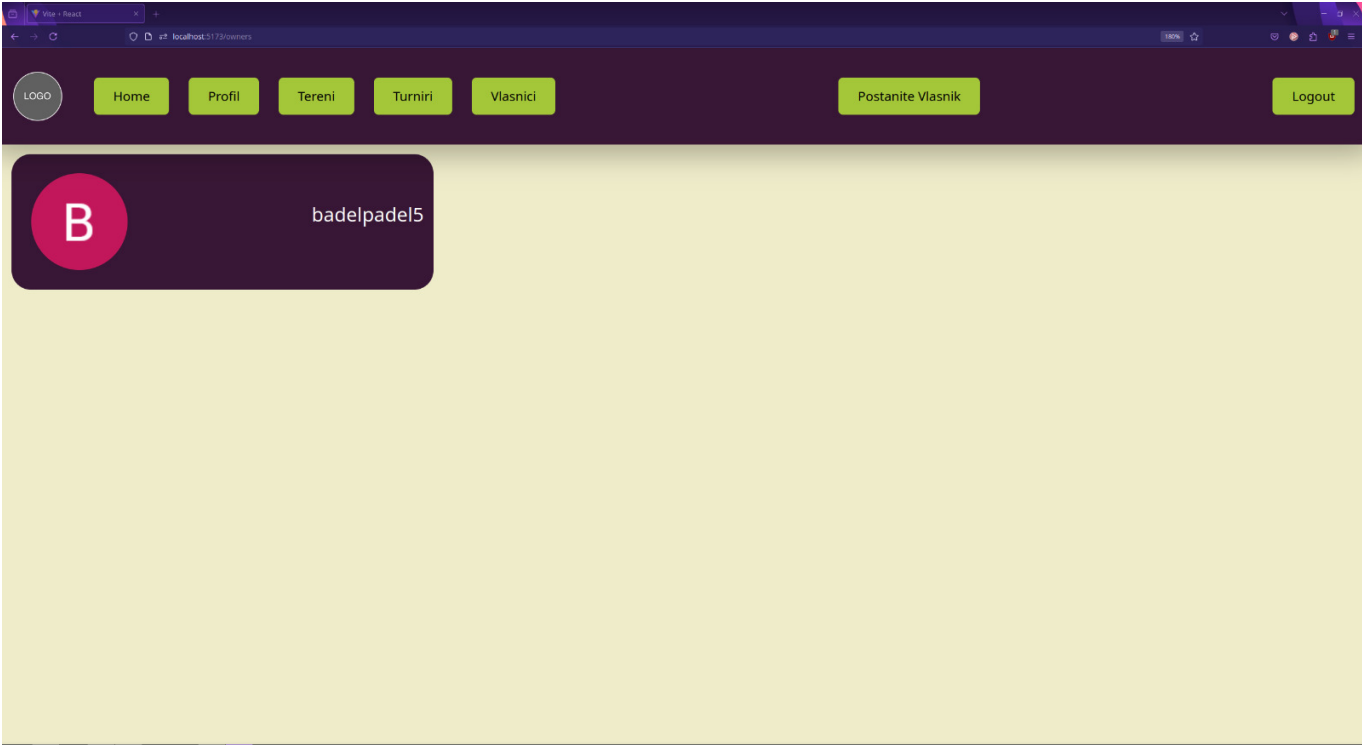
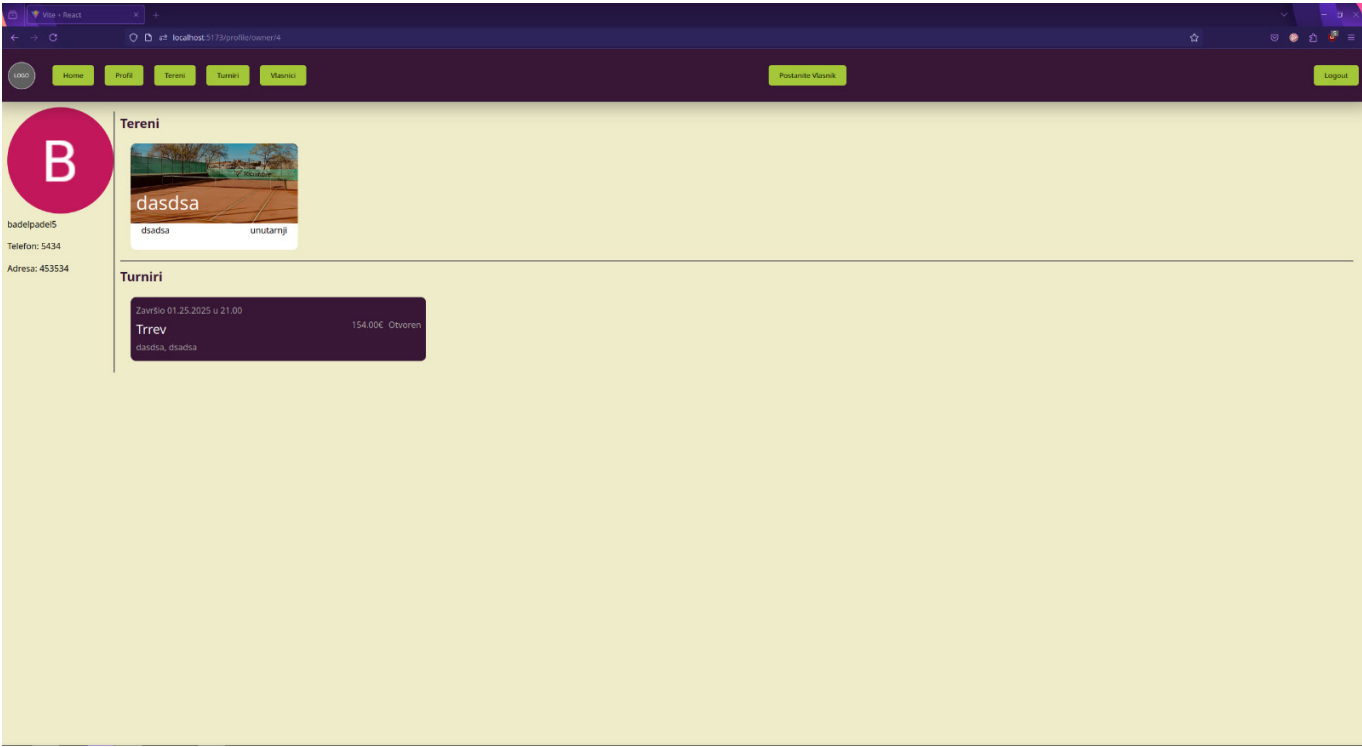
Web aplikacija [PlayPadel](#) se sastoji od naslovne stranice kojoj mogu pristupiti i neregistrirani korisnici.



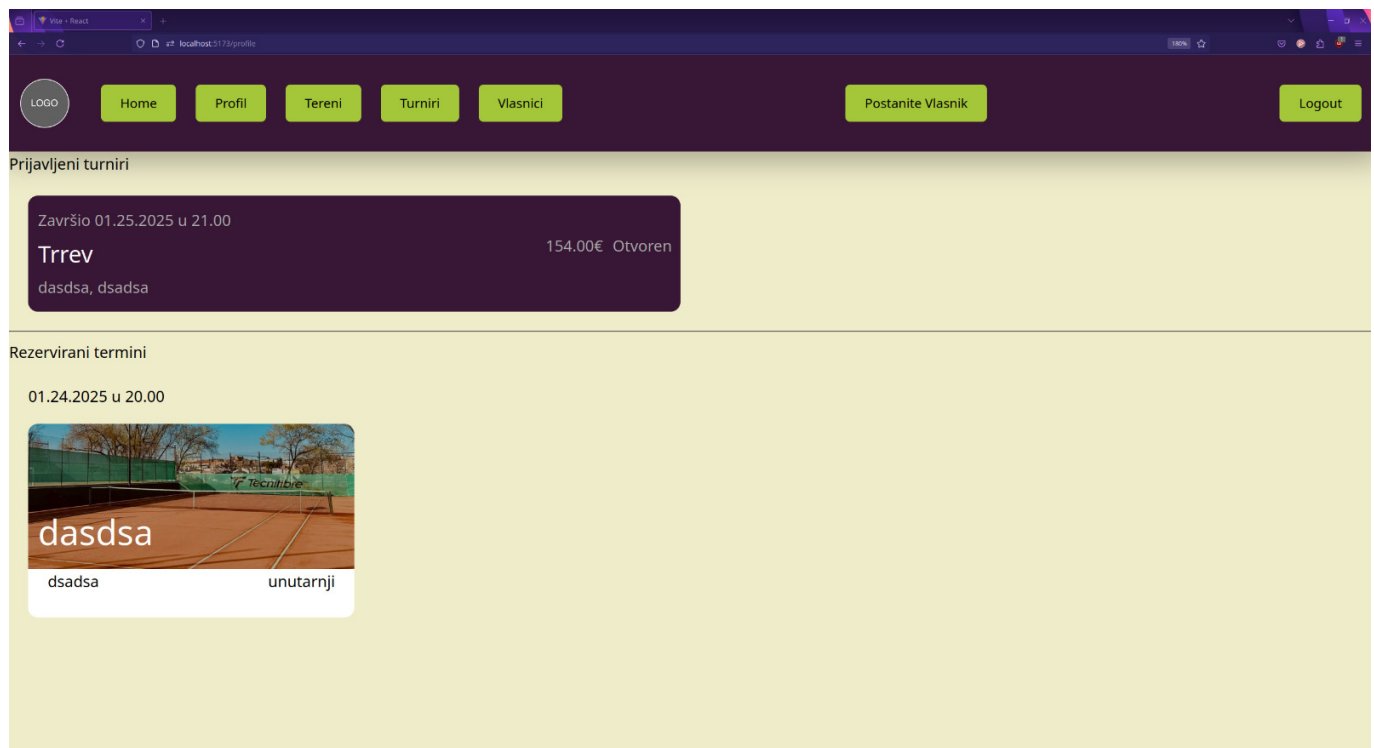
Stranice za Login i prijavu novih vlasnika



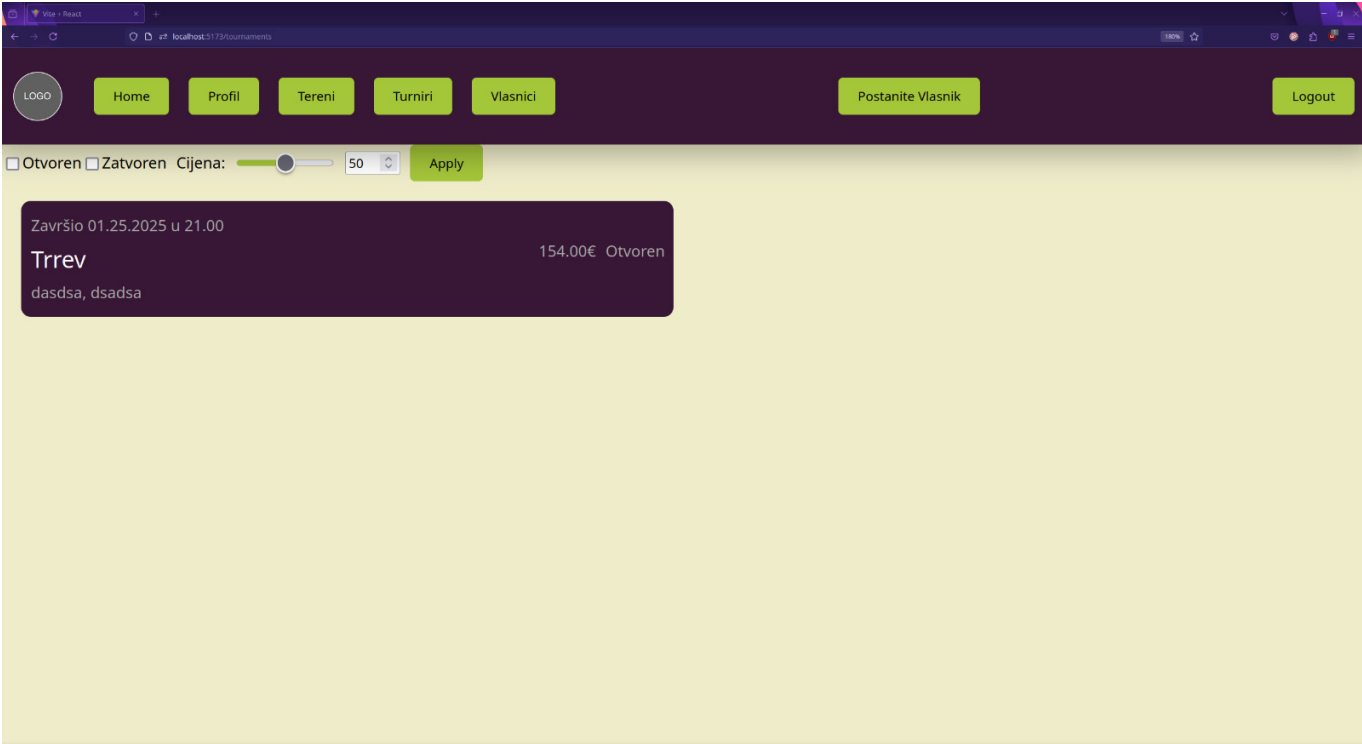
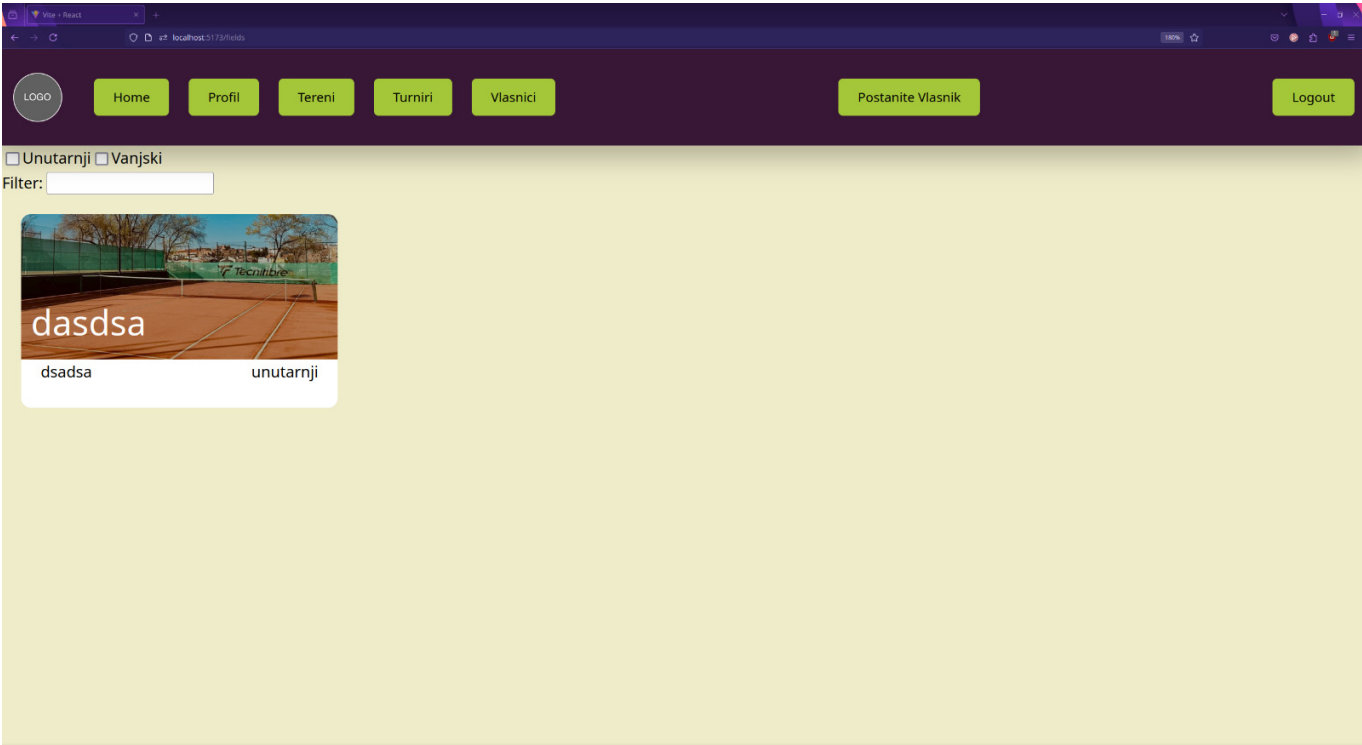
Nakon prijave, igrači mogu pregledavati profile vlasnika



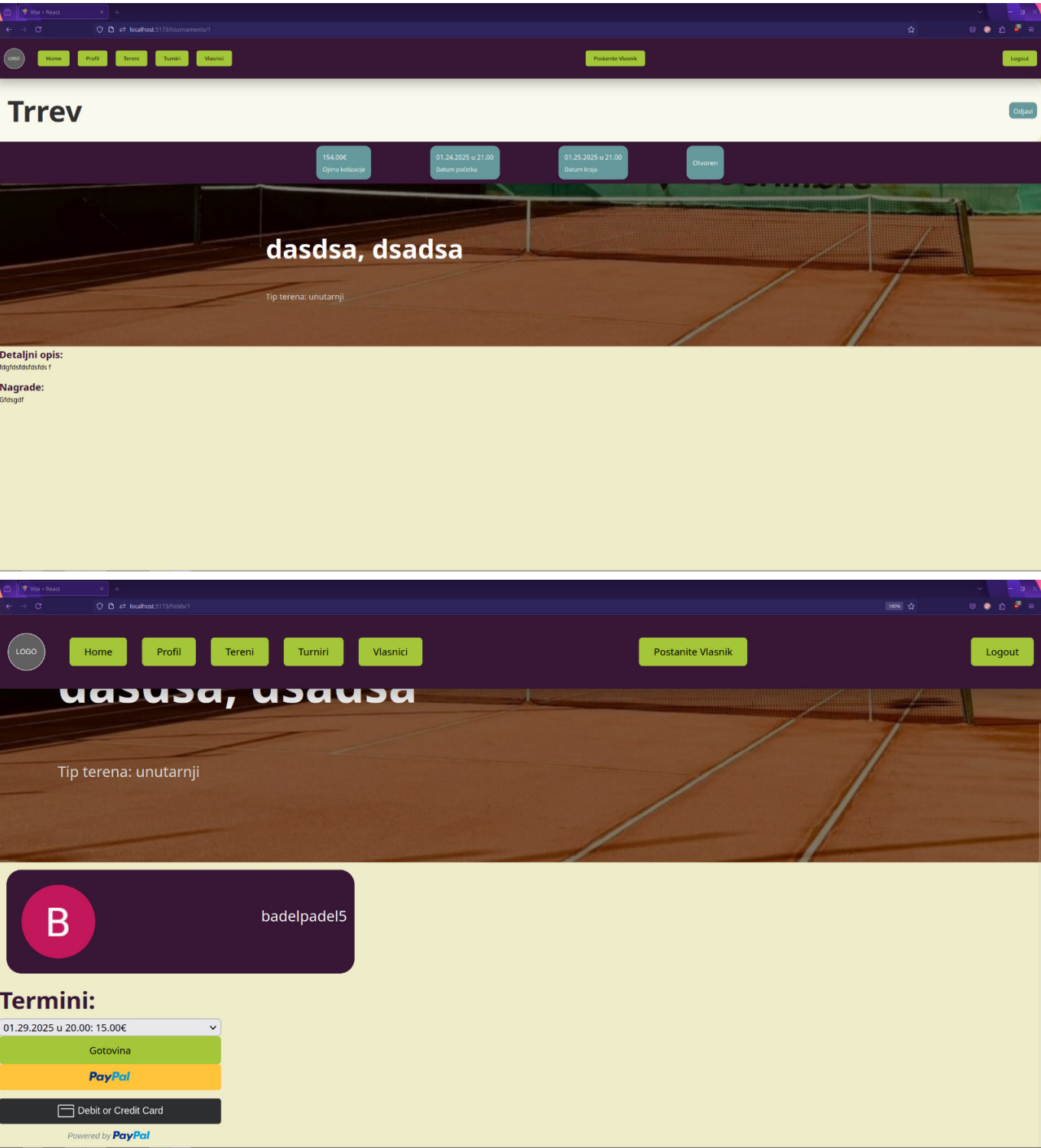
vlastiti profil



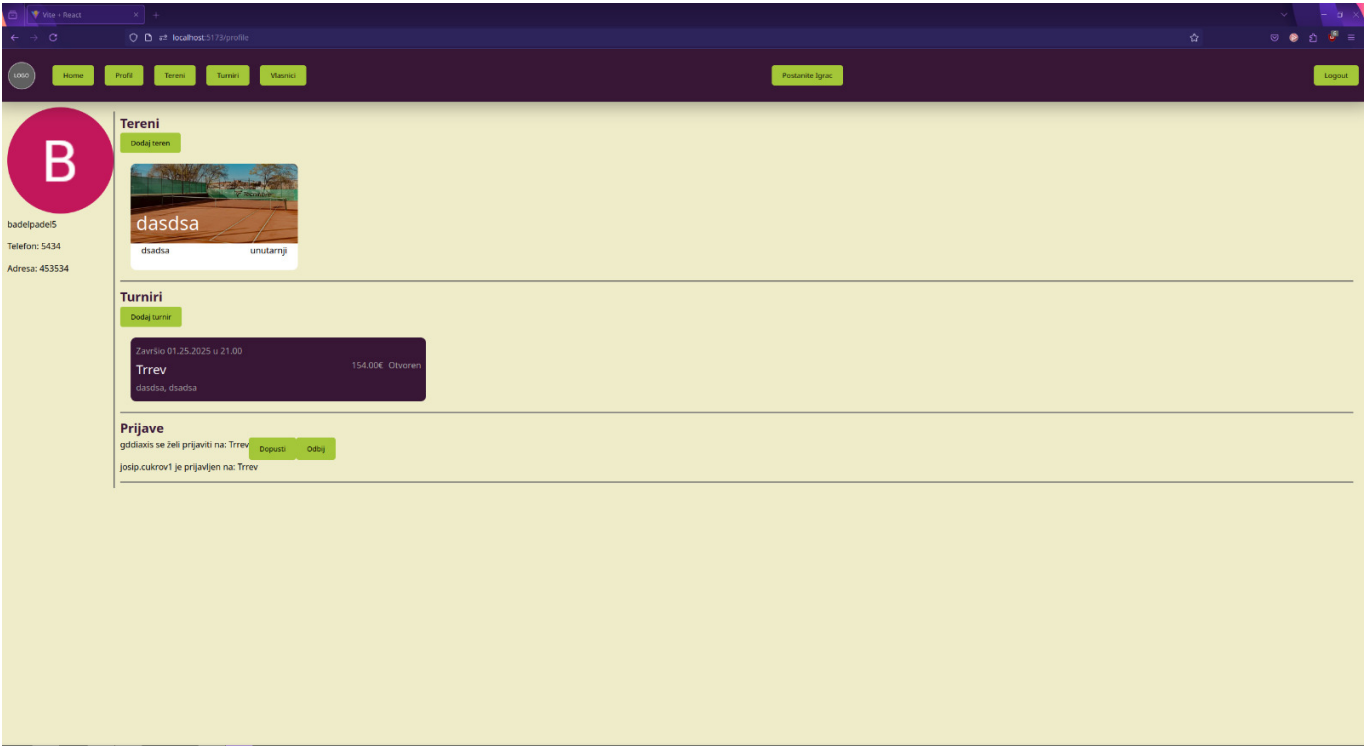
pretraživati terene i turnire



te mogu rezervirati neki teren ili se prijaviti na turnir.



Vlasnici mogu pregledavati vlastiti profil gdje mogu dodavati terene i turnire te prihvatiti ili odbiti prijave



Izrada projektnog zadatka bila je izazovna i s obzirom na naše trenutačno znanje i vještine, vjerojatno bi nam trebalo više vremena kako bismo napravili kvalitetnu web aplikaciju. Glavni razlog tome bila je nedovoljno dobro definirana podjela poslova unutar tima, kao i ograničeno poznavanje programskih jezika i alata koje smo koristili.

Tijekom rada na projektu stekli smo nekoliko važnih znanja i vještina. Naučili smo se služiti Gitom, neki članovi tima su naučili osnove rada s Django frameworkom za backend razvoj, dok su drugi članovi tima savladali osnove Reacta za frontend razvoj ili stekli iskustvo u konfiguriranju i korištenju alata kao što su Docker i Docker Compose. Kroz ovaj proces bolje smo razumjeli složenost kreiranja funkcionalne aplikacije i integraciju različitih tehnologija, ali i da trebamo više vremena uložiti svoje razumijevanje spomenutih tehnologija, pogotovo ako želimo nastaviti raditi ovakve projekte u budućnosti.

Za bržu i kvalitetniju izradu projekta bilo bi ključno unaprijediti naše vještine organizacije i timske komunikacije, tzv. soft skills, ali i tehničke vještine. Konkretno, potrebna su bolja znanja o radu s bazama podataka, korištenju JavaScripta i Pythona te bolje snalaženje u novim i nepoznatim situacijama.

Iako projekt nije u potpunosti dovršen, to nam pruža priliku za daljnji rad i unaprjeđenje. Kao prvi korak, mogli bismo završiti započeto i dodatno razviti funkcionalnosti koje su ostale neimplementirane. Trenutačno aplikacija ne uključuje pretplate za vlasnike, integraciju newslettera ili administrativni pristup putem frontenda. Administrativni dio aplikacije realiziran je korištenjem Django administracije, koja je dostupna putem backend sučelja. Nadalje, rezultati turnira i integracija Google kalendara nisu implementirani, iako smo koristili datetime funkcionalnosti iz React knjižnice.

Aplikaciju bismo mogli napraviti vizualno privlačnijom i poraditi na pristupačnosti (npr. prilagoditi web aplikaciju slijepim i slabovidnim osobama tako da dodamo opise slikama i sličnim značkama za lakše korištenje stranice). Osim toga mogli bismo dodati opciju slanja obavijesti o novim turnirima, objavljenim fotografijama turnira na kojem je korisnik prisustvovao, dodati mogućnost ocjenjivanja terena i traženja suigrača sa sličnim iskustvom i vještinama igranja ili ugraditi opciju dobivanja virtualnog trofeja/bedža i sl. nakon pobjede na turniru ili određenog broja rezerviranih termina.

Kontinuirano osvježavanje Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

#Dnevnik sastajanja

1. sastanak

- Datum: 12. listopada 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Josip Cukrov, Gregor Mihaljević, Tomislav Marenić
- Teme sastanka: napravljena WhatsApp grupa, upoznavanje, razgovor o očekivanjima o projektu i kvalitetama članova koje bi mogle biti korisne za projekt.

2. sastanak

- Datum: 17. listopada 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Josip Cukrov, Gregor Mihaljević, Tomislav Marenić
- Teme sastanka: dogovor oko naziva tima, upoznavanje s gitom za članove koji ga do sada nisu koristili, dogovor oko tehnologija koje planiramo koristiti pri izradi projekta, odabir voditelja

3. sastanak

- Datum: 21. listopada 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Gregor Mihaljević, Tomislav Marenić
- Teme sastanka: definiranje funkcijskih i nefunkcijskih zahtjeva, izrada "kostura" aplikacije

4. sastanak

- Datum: 29. listopada 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Tomislav Marenić, Josip Cukrov
- Teme sastanka: planovi i raspodjela uloga pri izradi dokumentacije i dijagrama

5. sastanak

- Datum: 4. studenoga 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Tomislav Marenić, Josip Cukrov, Gregor Mihaljević
- Teme sastanka: analiza napravljenoga, ispravci grešaka, planovi za dalje (backend i dokumentacija)

6. sastanak

- Datum: 14. studenoga 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Tomislav Marenić, Josip Cukrov
- Teme sastanka: analiza napravljenoga, pronalasci i ispravci grešaka, raspodjela daljnjih uloga, puštanje aplikacije u pogon

7. sastanak

- Datum: 21. studenoga 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Gregor Mihaljević, Tomislav Marenić, Josip Cukrov
- Teme sastanka: dogovor oko API-ja i autentifikacije

8. sastanak

- Datum: 28. studenoga 2024.
- Prisustvovali: Mario Olčar, Iva Džakula, Gregor Mihaljević, Tomislav Marenić, Josip Cukrov

- Teme sastanka: Ragovor o napravljenim greškama i planovima za dalje

9. sastanak

- Datum: 14. prosinca 2024.
- Prisustvovali: Mario Olčar, Tomislav Marenić, Josip Cukrov
- Teme sastanka: backend

10. sastanak

- Datum: 19. prosinca 2024.
- Prisustvovali: Mario Olčar, Tomislav Marenić
- Teme sastanka: backend

11. sastanak

- Datum: 3. siječnja 2025.
- Prisustvovali: Mario Olčar, Tomislav Marenić, Josip Cukrov
- Teme sastanka: spajanje frontenda i backenda

12. sastanak

- Datum: 9. siječnja 2025.
- Prisustvovali: Mario Olčar, Iva Džakula, Gregor Mihaljević, Tomislav Marenić, Josip Cukrov
- Teme sastanka: usporedbe promjena

13. sastanak

- Datum: 16. siječnja 2025.
- Prisustvovali: Iva Džakula, Josip Cukrov
- Teme sastanka: PayPal, daljnji razvoj komponenti

14. sastanak

- Datum: 27. siječnja 2025.
- Prisustvovali: Mario Olčar, Iva Džakula, Gregor Mihaljević, Tomislav Marenić, Josip Cukrov
- Teme sastanka: analiza koda

Plan rada

Aktivnost	Početak	Završetak	Trajanje (dana)
Organizacija tima	10.10.2024	17.10.2024	7
Analiza zadatka	12.10.2024	17.10.2024	5
Izrada osnovne strukture sustava	21.10.2024	23.10.2024	2
Početni dizajn korisničkog sučelja	23.10.2024	24.10.2024	1
Razvoj osnovnih funkcionalnosti sustava	25.10.2024	31.10.2024	6
Izrada dijagrama	29.10.2024	15.11.2024	17

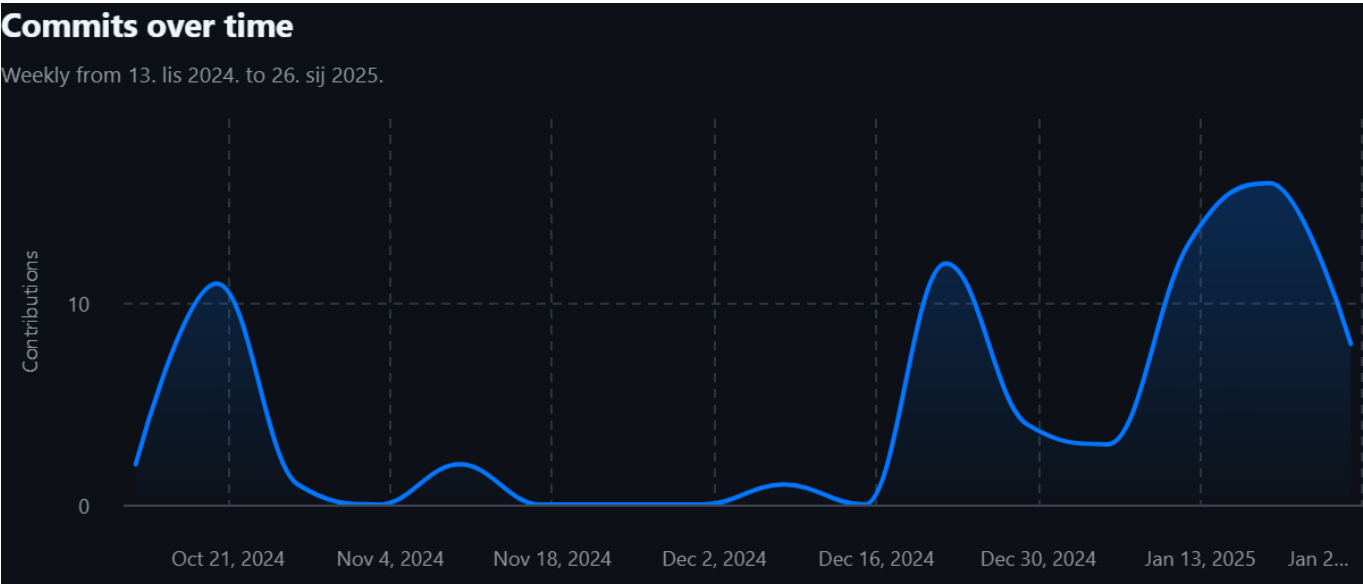
Aktivnost	Početak	Završetak	Trajanje (dana)
Izrada baze podataka	13.11.2024	16.11.2024	3
Pisanje dokumentacije	21.10.2024	16.11.2024	26
Priprema za implementaciju i lansiranje	12.11.2024	15.11.2024	1

Tablica aktivnosti

Aktivnosti	Mario Olčar	Iva Džakula	Gregor Mihaljević	Tomislav Marenić	Josip Cukrov
Postavljanje Git-a	1.5				
Upravljanje projektom	2	1	0.5	1	1
Opis projektnog zadatka		3			
Funkcionalni zahtjevi		2			
Opis pojedinih obrazaca	3	1.5			2
Dijagram obrazaca	3	3.5			4.5
Sekvencijski dijagrami				4	
Opis ostalih zahtjeva		1.5			
Arhitektura i dizajn sustava	5	3	3	3	20
Baza podataka	4.5		3	12	
Dijagram razreda				3.5	
Dijagram stanja				3	
Dijagram aktivnosti					2
Dijagram komponenti			2		
Korištene tehnologije i alati		15			5
Ispitivanje programskog rješenja	3		2	1	
Dijagram razmještaja					
Upute za puštanje u pogon	2	10	1	3	1
Dnevnik sastajanja	2	0.5			
Backend	20		45	40	55
Frontend		40		15	125
Deployment	12.5		30	8	

Aktivnosti	Mario Olčar	Iva Džakula	Gregor Mihaljević	Tomislav Marenić	Josip Cukrov
Dokumentacija	10	30	4	8	7
Sastanci	30	30	30	30	30
Zaključak i budući rad	10				
Popis literature					

Dijagram pregleda promjena



Ključni izazovi i rješenja

- Zaključno
- Opis izazova: Glavni izazovi tijekom projekta (npr. kašnjenje u razvoju, tehnički problemi).
- Rješenja: Način na koji su izazovi riješeni, kao i naučene lekcije koje su doprinijele napretku tima.

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Mario Olčar	30.10.2024
0.2	Opis projektnog zadatka i dodane informacije o dionicima i akterima	Iva Džakula	12.11.2024
0.3	Dodan sekvencijski dijagram rezervacija dvorana	Mario Olčar	
0.4	Dodani Arhitektura i dizajn sustava	Josip Curkov	
	Dodani funkcionalni zahtjevi		
	Dodano poglavlje Specifikacija zahtjeva sustava		
	Dodano poglavlje Implementacija i korisničko sučelje		
	Dodane slike sličnih aplikacija za poglavlje opis projektnog zadatka		
	Dodano poglavlje Analiza zahtjeva		
	Izmjene sekvencijski dijagrami		
	Dodan dijagram obrazaca uporabe		
	Dodan sekvencijski dijagram za pregled dostupnih mjesta		
	Dodan prikaz aktivnosti grupe		

- ovo treba biti vidljivo u samoj dinamici promjena repozitorija

Evidencija promjena sadrži popis promjena izvršenih tijekom cijelo životnog trajanja predmeta evidencije (dokumentacije, projekta i sl.). Osnova svrha je praćenja napretka svake promjene na temelju njezina preispitivanja, odobrenja (ili odbijanja), provedbe, kao i zaključenja. Štoviše, dobar dnevnik promjena sadrži i datum promjene i njegov utjecaj na projekt u smislu rizika, vremena i troškova. Sve te promjene prenose se dionicima. Štoviše, odbačene promjene također su uključene u povijest promjena.

Zašto? Olakšano praćenje ključnim dionicima.