



UNIVERZITET U ZENICI

Politehnički fakultet

Softversko inženjerstvo



## **E - Dom**

Predmet: Metode razvoja softvera

**Studentice:**

Imana Purić

Adna Kargić

Nedžma Imamović

**Profesor:**

Doc. dr. Mujo Hodžić

V. prof. dr. Nermin Goran

v. ass. Edin Tabak

# Sadržaj

|  |           |
|--|-----------|
| <b>1. UVOD.....</b>  | <b>2</b>  |
| <b>2. Opis problema.....</b>                                   | <b>2</b>  |
| <b>3. Uloge i odgovornosti članova tima.....</b>               | <b>3</b>  |
| <b>4. Zainteresovane strane (Stakeholders).....</b>            | <b>4</b>  |
| <b>5. Pozadina i strateško uklapanje projekta.....</b>         | <b>4</b>  |
| <b>6. Ciljevi projekta.....</b>                                | <b>5</b>  |
| <b>7. Opis sistema.....</b>                                    | <b>6</b>  |
| <b>8. Pretpostavke.....</b>                                    | <b>6</b>  |
| <b>9. Ograničenja sistema (Constraints).....</b>               | <b>7</b>  |
| <b>10. Funkcionalnosti sistema.....</b>                        | <b>8</b>  |
| Autentifikacija i upravljanje pristupom.....                   | 8         |
| Dashboard i pregled sistema.....                               | 8         |
| Upravljanje prijavama.....                                     | 8         |
| Upravljanje dokumentima i bodovanjem.....                      | 9         |
| Upravljanje statusima prijava.....                             | 9         |
| Rang lista i izvještaji.....                                   | 9         |
| Profil administratora.....                                     | 9         |
| <b>10. Korisničke priče (User Stories).....</b>                | <b>9</b>  |
| <b>11. Nefunkcionalni zahtjevi.....</b>                        | <b>10</b> |
| <b>12. Šta sistem trenutno ne obuhvata (Out of scope).....</b> | <b>11</b> |
| <b>13. Izgled korisničkog interfejsa.....</b>                  | <b>11</b> |
| Login ekran.....   | 11        |
| Registracija / zaboravljena lozinka.....                       | 12        |
| Dashboard / pregled sistema.....                               | 13        |
| Izgled početnog procesa dodavanja prijave.....                 | 14        |
| <b>14. Modeliranje i dizajn sistema.....</b>                   | <b>14</b> |
| <b>15. Arhitektura sistema / struktura sistema.....</b>        | <b>16</b> |
| Prezentacijski sloj (Presentation Layer).....                  | 16        |
| Poslovna logika (Business Logic Layer).....                    | 17        |
| Sloj pristupa podacima (Data Access Layer – DAO).....          | 17        |
| Model sloj (Model Layer).....                                  | 17        |
| Pomoćni i servisni slojevi.....                                | 17        |
| Napomena o organizaciji i dokumentaciji izvornog koda.....     | 18        |
| <b>16. Korištene tehnologije i alati.....</b>                  | <b>18</b> |
| <b>17. Upravljanje projektom i organizacija rada.....</b>      | <b>19</b> |
| Metodologija i alati (Scrumban).....                           | 19        |
| Faze razvoja (Sprintovi).....                                  | 19        |
| Agilni razvoj i testiranje.....                                | 20        |
| <b>18. Zaključak.....</b>                                      | <b>21</b> |

# **1. UVOD**

Studentski centar Zenica predstavlja ključnu instituciju za smještaj i svakodnevni boravak studenata koji studiraju na Univerzitetu u Zenici. Kao ustanova koja upravlja velikim brojem korisnika, prijava, odluka i prateće dokumentacije, Studentski centar se suočava sa složenim administrativnim procesima koji zahtijevaju visok nivo organizacije i tačnosti.

Trenutni način vođenja evidencije i obrade prijava u Studentskom centru Zenica u velikoj mjeri se oslanja na manuelne procedure, papirnu dokumentaciju ili djelimično digitalizovane sisteme koji nisu u potpunosti povezani. Takav pristup može dovesti do otežane obrade podataka, sporije komunikacije između studenata i administracije, kao i povećane mogućnosti za greške i neefikasnost u radu.

Polazeći od višegodišnjeg ličnog iskustva boravka u Studentskom centru Zenica, kao i neposrednog uvida u način funkcionisanja doma i potrebe njegovih korisnika, javila se ideja za razvoj softverskog rješenja koje bi unaprijedilo postojeći sistem. Ovaj projekat ima za cilj razvoj desktop aplikacije namijenjene isključivo Studentskom centru Zenica, koja bi omogućila centralizovano upravljanje prijavama studenata, evidencijom studenata, statusima prijava i pratećom dokumentacijom (dokumentima prijave).

Predložena aplikacija omogućava pristup administratoru koji je u ovom slučaju sekretar Studentskog centra Zenica.

Razvoj ove aplikacije realizovan je u okviru predmeta Metode razvoja softvera, s ciljem primjene teorijskih znanja kroz praktičan projekat koji odgovara stvarnim potrebama jedne institucije i njenih korisnika.

## **2. Opis problema**

Administrativni procesi u Studentskom centru Zenica, posebno oni koji se odnose na prijave studenata za smještaj, podrazumijevaju obradu velikog broja podataka i dokumenata u ograničenim vremenskim rokovima. Svake akademske godine sekretar Studentskog centra suočava se sa velikim brojem prijava koje je potrebno evidentirati, pregledati, bodovati i svrstati prema unaprijed definisanim kriterijima. Ovaj proces je dodatno otežan kada se koristi manuelni ili djelimično digitalizovani način rada.

Trenutni sistem obrade prijava u velikoj mjeri se oslanja na papirnu dokumentaciju ili nepovezane evidencije, što otežava centralizovano upravljanje podacima. Podaci o studentima, njihovim prijavama, dostavljenim dokumentima i ostvarenim bodovima često nisu objedinjeni u jedinstven sistem, zbog čega je potrebno više vremena za njihovu obradu i provjeru. Ovakav način rada povećava mogućnost grešaka, gubitka dokumentacije, neusklađenosti podataka i ponavljanja administrativnih zadataka.

Poseban problem predstavlja praćenje statusa prijave tokom cijelog procesa obrade. Bez centralizovanog sistema, administracija nema brz i jasan uvid u to koje su prijave zaprimljene, koje su u fazi pregleda, a koje su konačno odobrene ili odbijene. Također, ručno bodovanje prijave i pregled dokumentacije zahtijeva dodatno vrijeme i povećava opterećenje zaposlenika Studentskog centra.

Nedostatak jedinstvene desktop aplikacije prilagođene stvarnim potrebama Studentskog centra Zenica dovodi do smanjene efikasnosti, otežanog donošenja odluka i sporije obrade prijave. U uslovima povećanog broja studenata i prijave, ovakav pristup postaje neodrživ i ne prati savremene zahtjeve digitalizacije administrativnih procesa.

Iz navedenih razloga, javlja se potreba za razvojem pouzdanog i centralizovanog softverskog rješenja koje bi omogućilo efikasno upravljanje prijavama studenata, evidencijom dokumentacije, bodovanjem i statusima prijave. Takvo rješenje treba biti jednostavno za korištenje, prilagođeno administrativnom osoblju i u potpunosti usmjereno na optimizaciju rada Studentskog centra Zenica.

### **3. Uloge i odgovornosti članova tima**

Razvoj aplikacije E-Dom realizovan je kroz timski rad u kojem su sve članice tima aktivno učestvovala u svim fazama projekta. Iako je postojala generalna podjela zaduženja prema afinitetima, struktura tima je bila fleksibilna, te su članice međusobno pomagale i radile na svim dijelovima aplikacije (backend, frontend i baza podataka) kako bi se osigurala kompletna funkcionalnost sistema.

Generalna podjela zaduženja izgledala je ovako:

- Adna Kargić – UI Dizajn i Frontend Bila je primarno zadužena za vizuelni identitet aplikacije. Njen fokus bio je na dizajnu korisničkog interfejsa (User Interface), osiguravanju intuitivnog izgleda ekrana i definisanju stilova koji aplikaciju čine preglednom i jednostavnom za korištenje.
- Nedžma Imamović i Imana Purić – Razvoj funkcionalnosti i Baza podataka Bile su zadužene za implementaciju cjelokupne logike sistema. To je podrazumijevalo rad na backend dijelu, dizajniranje i povezivanje sa bazom podataka, kao i pisanje Controller klase koje povezuju dizajn sa kodom. Njihov zadatak bio je osigurati da sve akcije u aplikaciji (prijave, bodovanje, pretrage) ispravno funkcionišu.

**Napomena o saradnji:** Važno je istaći da podjela posla nije bila striktna. Sve tri članice tima su, po potrebi, radile na svim slojevima aplikacije, zajednički rješavale programske greške (bugove) i testirale sistem kako bi se osigurala stabilnost i kvalitet finalnog rješenja.

## 4. Zainteresovane strane (Stakeholders)

Glavna zainteresovana strana u projektu **E-Dom** je Studentski centar Zenica, odnosno administrativno osoblje koje sistem koristi u svakodnevnom radu i koje ima direktnu korist od digitalizacije procesa obrade prijave. Indirektni korisnici sistema su studenti, čije se prijave, dokumentacija i bodovanje obrađuju putem aplikacije, iako oni nemaju direktan pristup sistemu. Fakultet i nastavno osoblje predstavljaju akademske zainteresovane strane kroz evaluaciju projekta i praćenje primjene metodologije razvoja softvera u okviru nastavnog procesa. Također, tim koji je radio na razvoju aplikacije predstavlja internu zainteresovanu stranu, s obzirom na sticanje praktičnih znanja i iskustva kroz realizaciju projekta.

## 5. Pozadina i strateško uklapanje projekta

Razvoj aplikacije **E-Dom** zasnovan je na realnim potrebama Studentskog centra Zenica, ali i na zakonskim obavezama koje ova institucija ima u pogledu vođenja, čuvanja i obrade službene dokumentacije. Prema važećim propisima i pravilnicima, Studentski centar je dužan posjedovati kompletnu evidenciju o prijavama studenata, pratećim dokumentima, bodovanju i donesenim odlukama, te osigurati njihovu dostupnost i arhiviranje u propisanom obliku.

U praksi, veliki dio ovih procesa i dalje se oslanja na manuelne procedure ili nepovezane digitalne zapise, što otežava rad administracije i povećava rizik od grešaka i gubitka podataka. Iz tog razloga, razvoj centralizovane desktop aplikacije predstavlja optimalno rješenje koje omogućava sigurno, pouzdano i zakonski usklađeno upravljanje podacima unutar same institucije.

Izbor desktop aplikacije strateški je opravdan činjenicom da se radi o internom sistemu koji koristi isključivo administrativno osoblje, te da podaci moraju biti trajno dostupni, kontrolisani i pohranjeni u skladu s internim pravilnicima i zakonskim obavezama. Desktop rješenje omogućava bolju kontrolu pristupa, stabilan rad sa bazom podataka i dugoročno arhiviranje podataka, što ga čini najprikladnijom verzijom za potrebe Studentskog centra Zenica.

Projekat E-Dom se na ovaj način uklapa u širi proces digitalizacije administrativnih sistema, istovremeno poštujući zakonski okvir i stvarne operativne potrebe institucije, te predstavlja čvrstu osnovu za buduće unapređenje i proširenje sistema.

## 6. Ciljevi projekta

Poslovni cilj projekta E-Dom jeste unapređenje i digitalizacija administrativnih procesa u Studentskom centru Zenica kroz centralizovan sistem za obradu prijava.

Timski cilj projekta bio je primjena teorijskih znanja iz predmeta Metode razvoja softvera kroz razvoj funkcionalne desktop aplikacije u timskom i agilnom okruženju. Glavni cilj projekta **E-Dom** jeste razvoj pouzdane i funkcionalne desktop aplikacije koja će unaprijediti administrativne procese u Studentskom centru Zenica, posebno u oblasti obrade prijava studenata za smještaj. Projekat ima za cilj da zamijeni postojeće manuelne i nepovezane procedure jedinstvenim digitalnim sistemom koji omogućava efikasnije upravljanje podacima i dokumentacijom.

Specifični ciljevi projekta su sljedeći:

- Omogućiti centralizovano upravljanje prijavama studenata za smještaj kroz jedinstvenu desktop aplikaciju.
- Omogućiti evidenciju i pregled podataka o studentima, njihovim prijavama.
- Osigurati jednostavno dodavanje, pregled dokumentima koji prate prijavu studenta.
- Omogućiti automatsko bodovanje prijave na osnovu dostavljenih dokumenata i definisanih kriterija.
- Omogućiti praćenje i promjenu statusa prijave tokom procesa obrade (npr. zaključena, odobrena, na pregledu, odbijena).
- Povećati efikasnost i tačnost administrativnog rada smanjenjem manuelnih aktivnosti i mogućnosti za greške.
- Osigurati brzi uvid u stanje prijave i pregled dokumenata prijave radi lakšeg rada.
- Povećati preglednost i transparentnost procesa obrade prijave unutar Studentskog centra Zenica.
- Primijeniti teorijska znanja iz oblasti metoda razvoja softvera kroz praktičan projekat zasnovan na realnim potrebama institucije.

Realizacijom navedenih ciljeva, projekat **E-Dom** doprinosi modernizaciji rada Studentskog centra Zenica, olakšava svakodnevne administrativne aktivnosti i stvara osnovu za dalji razvoj i proširenje sistema u budućnosti.

## 7. Opis sistema

Sistem je koncipiran kao zatvoreno rješenje koje koristi isključivo administrativno osoblje, bez direktnog pristupa studenata.

Aplikacija omogućava upravljanje evidencijom studenata i njihovih prijava po akademskim godinama. Svaka prijava predstavlja cjelinu koja sadrži osnovne podatke o studentu, status prijave, bodove i prateću dokumentaciju. Dokumenti se vezuju direktno za prijavu i koriste se kao osnova za bodovanje i donošenje odluka.

Sistem omogućava dodavanje, pregled i brisanje prijava, pri čemu se prilikom brisanja prijave automatski uklanjaju svi dokumenti povezani s tom prijavom. Time se osigurava dosljednost i integritet podataka unutar sistema.

Za svaku prijavu moguće je pratiti tok obrade kroz različite statuse, čime se administraciji omogućava jasan pregled trenutnog stanja prijava. Na osnovu unesenih bodova po dokumentima, sistem automatski izračunava ukupan broj bodova prijave, što olakšava i ubrzava proces evaluacije.

Korisnički interfejs sistema je prilagođen svakodnevnom administrativnom radu i omogućava pregledno prikazivanje podataka, statistika i detalja pojedinačnih prijava. Poseban akcenat stavljen je na jasno razdvajanje funkcionalnih cjelina sistema, kao što su pregled prijava, promjena statusa prijava.

Sistem **E-Dom** je osmišljen tako da podrži trenutne potrebe Studentskog centra Zenica, ali i da omogući jednostavno proširenje funkcionalnosti u budućnosti, u skladu s razvojem poslovnih procesa i zahtjeva institucije.

## 8. Pretpostavke

Tokom razvoja sistema E-Dom napravljeno je nekoliko tehničkih i organizacionih pretpostavki. Pretpostavlja se da aplikaciju koristi isključivo administrativno osoblje Studentskog centra Zenica. Sistem se koristi na računarima sa stabilnom internet konekcijom, s obzirom na korištenje cloud baze podataka i email servisa. Pretpostavlja se da su podaci koje administrator unosi tačni i potpuni. Također, pretpostavlja se da korisnici imaju osnovno iskustvo rada sa desktop aplikacijama.

## 9. Ograničenja sistema (Constraints)

Sistem E-Dom razvijen je uz jasno definisana tehnička, organizaciona i funkcionalna ograničenja koja proizlaze iz namjene aplikacije, okruženja u kojem se koristi, kao i okvira akademskog projekta. Ova ograničenja ne predstavljaju nedostatke sistema, već su svjesno donesene odluke s ciljem fokusiranja na ključne potrebe Studentskog centra Zenica i uspješne realizacije projekta.

Aplikacija je namijenjena isključivo internom korištenju od strane administrativnog osoblja Studentskog centra Zenica. Sistem ne podržava više različitih korisničkih uloga (npr. studenti, super-administratori ili eksterni korisnici), već je pristup ograničen na autorizovane administratore. Samim tim, funkcionalnosti vezane za upravljanje pravima pristupa ili višekorisničke hijerarhije nisu obuhvaćene trenutnom verzijom sistema.

Sistem je razvijen kao desktop aplikacija i nije prilagođen za rad putem web preglednika niti za korištenje na mobilnim uređajima. Ovakvo ograničenje je u skladu s činjenicom da se aplikacija koristi u kontrolisanom radnom okruženju, na službenim računarima unutar institucije, gdje desktop rješenje predstavlja najstabilniji i najsigurniji oblik implementacije.

Rad aplikacije zavisi od dostupnosti stabilne internet konekcije, s obzirom na korištenje cloud baze podataka i email servisa za autentifikaciju i slanje obavještenja. U slučaju prekida internet konekcije, određene funkcionalnosti sistema, poput pristupa podacima iz baze ili slanja email poruka, mogu biti privremeno onemogućene, dok offline režim rada nije podržan.

Sistem ne uključuje napredne mehanizme za automatizirano testiranje, kao što su unit testovi ili integracioni testovi. Validacija ispravnosti funkcionalnosti vršena je putem manualnog testiranja tokom razvoja aplikacije i završne faze projekta. Ovakav pristup testiranju je prihvatljiv u kontekstu akademskog projekta i obima sistema.

Također, aplikacija ne obuhvata napredne sigurnosne mehanizme na nivou institucije, poput centralizovanog sistema autentifikacije (Single Sign-On), enkripcije kompletne baze podataka ili integracije sa eksternim sigurnosnim servisima. Sigurnost sistema fokusirana je na osnovne i neophodne mjere zaštite, kao što su šifrovanje lozinki i kontrola pristupa aplikaciji.

Navedena ograničenja su u skladu sa ciljevima projekta, vremenskim okvirom i zahtjevima predmeta Metode razvoja softvera. Ona ne umanjuju funkcionalnu vrijednost sistema, već jasno definišu granice trenutne verzije aplikacije i ostavljaju prostor za buduća unapređenja i proširenja u realnom produkcijskom okruženju.



## 10. Funkcionalnosti sistema

Sistem **E-Dom** obuhvata skup funkcionalnosti koje omogućavaju potpunu administrativnu obradu prijava studenata za smještaj u Studentskom centru Zenica. Funkcionalnosti sistema su grupisane prema logičkim cjelinama rada aplikacije.

### Autentifikacija i upravljanje pristupom

Sistem omogućava siguran pristup aplikaciji putem mehanizama autentifikacije administratora, koji uključuju:

- **Prijavu (login)** administratora korištenjem korisničkog imena i lozinke,
- **Registraciju administratora**, uz unos osnovnih podataka,
- **Zaboravljenu lozinku**, pri čemu se proces promjene lozinke vrši putem email servisa. Administratoru se šalje verifikacijski kod na registrovanu email adresu, koji se koristi za postavljanje nove lozinke,
- **Odjavu (logout)** iz sistema, čime se završava aktivna sesija korisnika.

### Dashboard i pregled sistema

Nakon uspješne prijave, administratoru je dostupan **dashboard** koji pruža brzi pregled stanja sistema. Dashboard omogućava:

- Pregled trenutnog broja studenata u bazi,
- Pregled ukupnog broja prijava u bazi,
- Brze akcije za najčešće korištene funkcionalnosti sistema,
- Vizuelni i numerički prikaz ključnih informacija potrebnih za svakodnevni rad administracije.
- Dijagram sa trenutnim statusima prijava u bazi

### Upravljanje prijavama

Sistem omogućava kompletno upravljanje prijavama studenata kroz sljedeće funkcionalnosti:

- **Dodavanje nove prijave**, koje se odvija u više koraka:
  1. Dodavanje studenta,
  2. Unos osnovnih podataka o prijavi (akademska godina, status i ostali relevantni podaci),
  3. Dodavanje i povezivanje potrebnih dokumenata uz prijavu,
- Pregled svih prijava unutar sistema,
- Sortiranje prijava prema različitim kriterijima (status, akademska godina, broj bodova i sl.),
- Pretragu (search) prijava i studenata radi bržeg pronalaska podataka,
- Brisanje prijave, pri čemu se automatski brišu svi dokumenti vezani za tu prijavu.

## Upravljanje dokumentima i bodovanjem

Za svaku prijavu sistem omogućava:

- Dodavanje, pregled i evidenciju dokumenata prijave,
- Označavanje da li je dokument dostavljen,
- Dodjelu bodova po dokumentima,
- Automatski obračun ukupnog broja bodova prijave na osnovu unesenih dokumenata.

## Upravljanje statusima prijava

Sistem omogućava praćenje toka obrade prijave kroz promjenu statusa, kao što su zaključena, u obradi, odobrena ili odbijena. Administrator ima mogućnost promjene statusa prijave u skladu s rezultatima evaluacije, također ima pravo da kada odobri prijavu pošalje email studentu da je prijava odobrena.

## Rang lista i izvještaji

Sistem omogućava generisanje **rang liste prijava** na osnovu ukupnog broja bodova. Rang lista:

- Prikazuje prijave sortirane prema ostvarenim bodovima, odvojeno tabele brucoši i studenti viših godina,
- Omogućava **export rang liste u PDF format**, radi arhiviranja ili službene upotrebe.

## Profil administratora

Administrator ima pristup dijelu **Moj profil**, koji omogućava:

- Pregled osnovnih podataka administratora,
- Upravljanje ličnim podacima,
- Pregled informacija vezanih za nalog.

## 10. Korisničke priče (User Stories)

Korisničke priče predstavljaju opis ključnih funkcionalnosti sistema iz perspektive krajnjeg korisnika, u ovom slučaju administratora Studentskog centra Zenica. One definišu interakciju korisnika sa sistemom i očekivane rezultate pojedinih radnji.

**Kao administrator sistema**, želim da se sigurno prijavim u aplikaciju korištenjem korisničkog imena i lozinke, kako bih dobio autorizovan pristup funkcionalnostima sistema i podacima o studentima i njihovim prijavama.

**Kao administrator sistema,** želim imati mogućnost dodavanja nove prijave studenta kroz strukturiran proces, koji uključuje unos osnovnih podataka o studentu, definisanje prijave i povezivanje potrebne dokumentacije, kako bih osigurao potpunu i tačnu evidenciju zahtjeva za smještaj.

**Kao administrator sistema,** želim pregledati dokumente prijave i automatsko bodovanje na osnovu dostavljenih dokumenata.

**Kao administrator sistema,** želim imati uvid u trenutni status svake prijave i mogućnost njegove izmjene tokom procesa obrade, kako bih mogao pratiti tok prijave od unosa prijave do konačne odluke.

**Kao administrator sistema,** želim generisati rang listu studenata na osnovu ukupnog broja ostvarenih bodova, kako bih imao jasan i pouzdan pregled poretka studenata za potrebe donošenja odluka i izrade službenih izvještaja.

## **11. Nefunkcionalni zahtjevi**

Pored funkcionalnih zahtjeva, sistem E-Dom mora zadovoljiti i nefunkcionalne zahtjeve koji se odnose na sigurnost, pouzdanost, performanse, upotrebljivost i održivost sistema. Sistem osigurava sigurnost podataka administratora i studenata, pri čemu se korisničke lozinke pohranjuju u šifrovanom obliku, a proces resetovanja lozinke realizuje se putem email verifikacije.

Aplikacija omogućava stabilan i pouzdan rad sa cloud bazom podataka, uz očuvanje integriteta i konzistentnosti svih povezanih podataka. Poseban naglasak stavljen je na upotrebljivost sistema kroz intuitivan i pregledan korisnički interfejs prilagođen administrativnom osoblju, bez potrebe za dodatnom obukom.

Sistem omogućava brz odziv prilikom pregleda prijave, pretrage podataka i generisanja rang lista, čak i pri radu sa većim brojem zapisa. Pristup aplikaciji ograničen je isključivo na autorizovane administratore, čime se sprječava neovlašteni pristup osjetljivim podacima. Slojna i modularna arhitektura sistema omogućava jednostavno održavanje, proširenje i prilagođavanje aplikacije budućim potrebama Studentskog centra Zenica.

## 12. Šta sistem trenutno ne obuhvata (Out of scope)

Iako sistem E-Dom pokriva ključne administrativne procese vezane za obradu prijave studenata za smještaj, određene funkcionalnosti nisu obuhvaćene trenutnom verzijom aplikacije. Ove funkcionalnosti su svjesno izostavljene kako bi se fokus zadržao na osnovnim potrebama Studentskog centra Zenica i osigurala stabilnost sistema.

Sistem trenutno ne omogućava direktan pristup studentima, niti online podnošenje prijave od strane studenata. Komunikacija sa studentima ograničena je na slanje email obavještenja o statusu prijave, bez dodatnih interaktivnih funkcionalnosti.

Aplikacija ne obuhvata napredno upravljanje smještajnim kapacitetima (sobe, paviljoni, raspored kreveta), finansijske module (uplate, dugovanja, fakturisanje), niti integraciju sa drugim informacionim sistemima Univerziteta u Zenici. Također, u trenutnoj verziji nisu implementirani napredni analitički izvještaji i statistike, kao ni automatizovani sistemi za dugoročno arhiviranje i backup podataka na nivou institucije.

Navedene funkcionalnosti predstavljaju potencijalne pravce daljeg razvoja sistema, ali su u ovoj fazi projekta ostavljene van obuhvata kako bi se ispunili ciljevi predmeta i osigurala kvalitetna implementacija osnovnih funkcionalnosti.

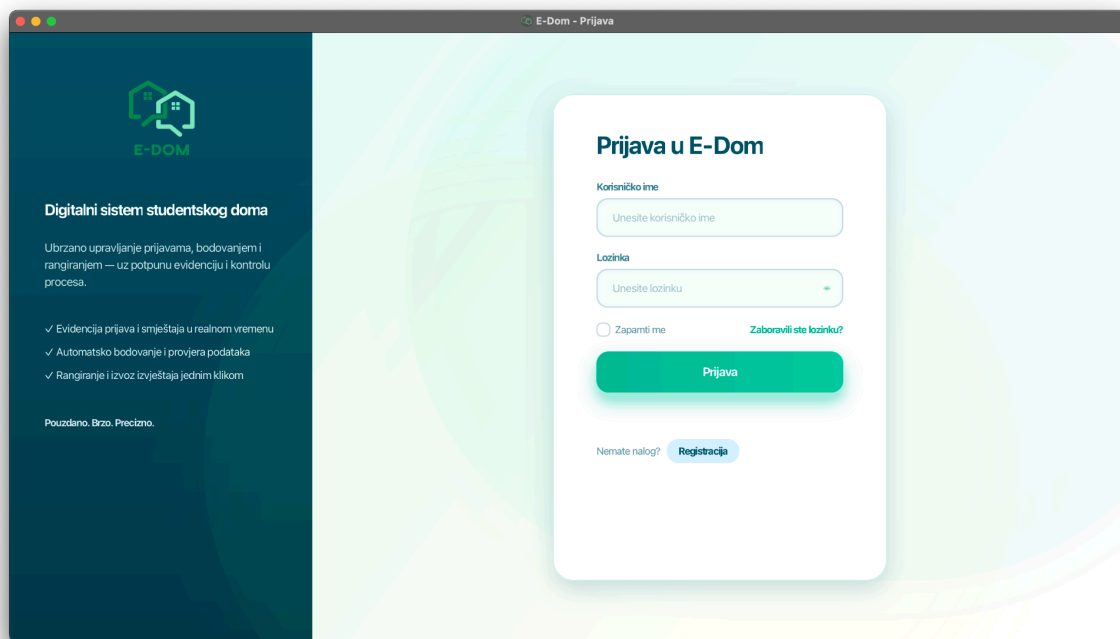
## 13. Izgled korisničkog interfejsa

Korisnički interfejs sistema **E-Dom** dizajniran je s ciljem da omogući jednostavno, pregledno i efikasno korištenje aplikacije. Poseban akcenat stavljen je na intuitivnu navigaciju, jasno razdvajanje funkcionalnih cjelina i brz pristup najčešće korištenim opcijama sistema.

Interfejs je prilagođen svakodnevnom administrativnom radu, pri čemu vizuelni elementi i raspored komponenti omogućavaju lakše praćenje stanja prijave, pregled dokumentacije i donošenje odluka, uz minimalnu potrebu za dodatnom obukom korisnika.

### Login ekran

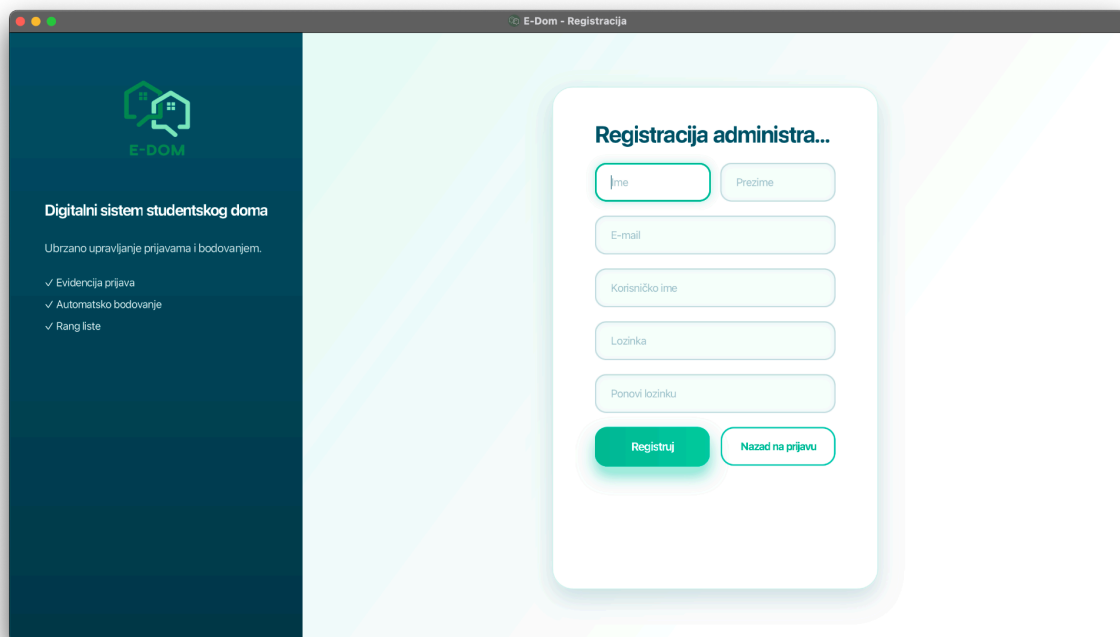
U nastavku slijedi screenshot login ekrana.



**Slika 1. Prijava u E - Dom**

## **Registracija / zaboravljena lozinka**

U nastavku slijedi screenshot ekrana registracije i u slučaju da je administrator zaboravio lozinku.



**Slika 2. Registracija administratora**

**Reset lozinke**

Korisničko ime

Pošalji kod

Verifikacioni kod

Unesite kod

Nova lozinka

Nova lozinka

Potvrda lozinke

Potvrdite lozinku

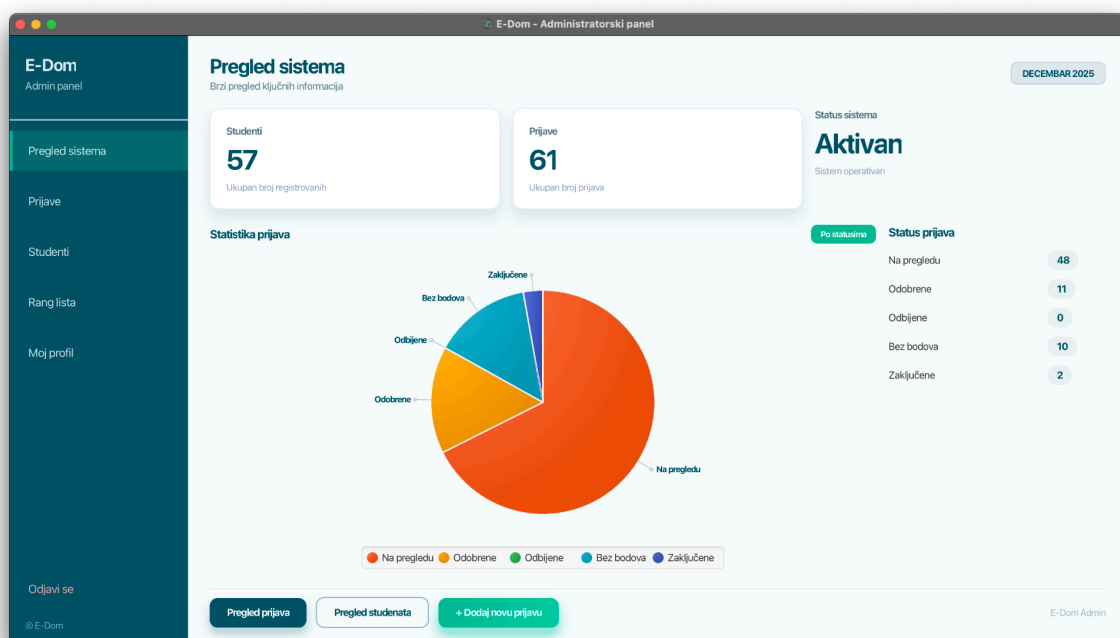
Resetuj lozinku

Nazad na prijavu

**Slika 3. Reset lozinke**

## Dashboard / pregled sistema

U nastavku slijedi screenshot dashboard-a.



**Slika 4. Pregled sistema**

## Izgled početnog procesa dodavanja prijave

U nastavku slijedi screenshot početnog ekrana za dodavanje nove prijave a to je dodavanje studenta nakon čega slijedi ekran sa osnovnim podacima o prijavi, zatim dodavanje dokumenata.

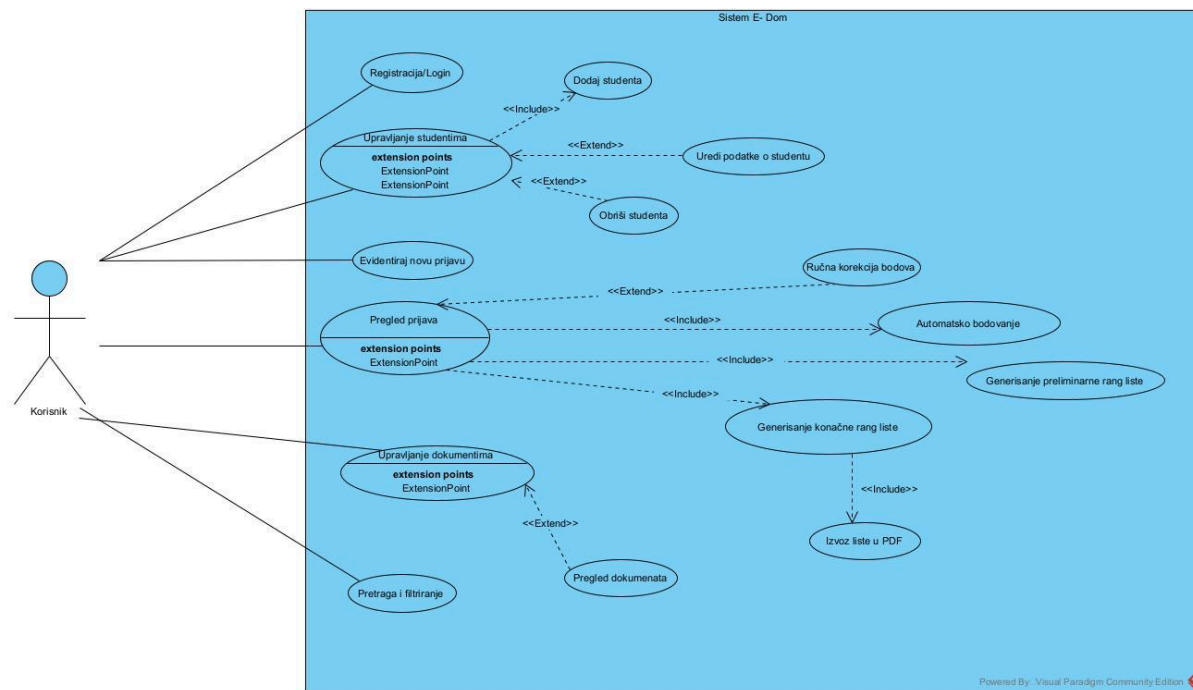
The screenshot shows a web form titled "Dodavanje novog studenta" (Adding a new student). Below the title is a subtitle "Unesite osnovne informacije i kontakt podatke." (Enter basic information and contact data). A progress bar at the top indicates three steps: 1. Osnovno (Basic), 2. Dokumenti (Documents), and 3. Pregled (Review). The first step, "Osnovne informacije", is currently active. The form contains two columns of input fields. The left column includes: "Ime" (Name) with a placeholder "Unesite ime", "Prezime" (Surname) with a placeholder "Unesite prezime", "Ime jednog roditelja" (Name of one parent) with a placeholder "Ime roditelja", "JMBG" (JMBG) with a placeholder "13 cifara", "Broj indeksa" (Index number) with a placeholder "npr. 421", and "Fakultet" (Faculty) with a dropdown menu "Odaberi fakultet". The right column includes: "Godina studija" (Year of study) with a dropdown menu "Odaberi godinu", "Prosjek" (Average) with a placeholder "npr. 8.6", "Email" with a placeholder "ime.prezime@email.com", "Telefon" (Phone) with a placeholder "+387...", "Adresa stanovanja" (Residence address) with a placeholder "Grad, ulica", and "Socijalni status" (Social status) with a dropdown menu "Odaberi status". At the bottom of the form are two buttons: "← Nazad" (Back) and "Dolje →" (Next).

Slika 5. Nova prijava

## 14. Modeliranje i dizajn sistema

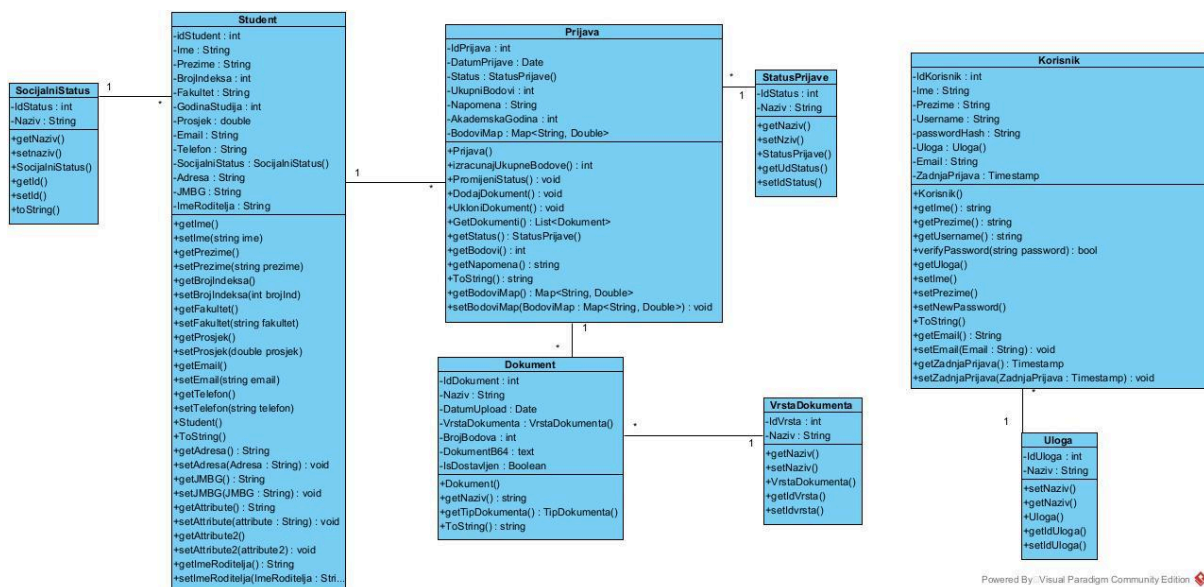
Kako bi se osigurala precizna implementacija funkcionalnosti i ispravna struktura podataka, prije samog kodiranja kreirani su odgovarajući dijagrami. Ovi modeli služe kao plan za razvoj poslovne logike i baze podataka.

**Dijagram slučajeva upotrebe (Use Case Diagram)** Ovaj dijagram prikazuje interakciju korisnika (administratora) sa sistemom. Jasno su definirane ključne funkcionalnosti kao što su Registracija/Login, Upravljanje studentima, Evidentiranje prijave, te Generisanje rang listi. Prikazane su i relacije Include (npr. automatsko bodovanje pri pregledu) i Extend (npr. ručna korekcija bodova), koje opisuju zavisnosti između procesa.



Slika 6. Use Case dijagram

**UML Dijagram klasa:** Statička struktura aplikacije prikazana je kroz dijagram klasa koji definira osnovne entitete u sistemu kao što su Student, Prijava, Dokument i Korisnik. Dijagram prikazuje attribute svake klase, njihove metode (gettere, settere, te poslovnu logiku poput `izracunajUkupneBodove()`), kao i međusobne veze između klasa (asocijacije i kompozicije).



Slika 7. UML dijagram klasa



**ER Dijagram (Entity-Relationship Diagram):** Za potrebe projektovanja baze podataka kreiran je ER dijagram koji definira strukturu tabela i njihove relacije. Centralni entitet je Prijava, koja je povezana sa Studentom, Statusom\_prijave i Dokumentima. Ovakva struktura osigurava integritet podataka i omogućava efikasno uparivanje studenata sa njihovim prijavama i pratećom dokumentacijom unutar baze.



Slika 8. ER dijagram

## 15. Arhitektura sistema / struktura sistema

Sistem **E-Dom** razvijen je kao **desktop aplikacija** korištenjem programskog jezika **Java** i okvira **JavaFX** za izradu grafičkog korisničkog interfejsa. Aplikacija je zasnovana na **slojnoj (layered) arhitekturi**, koja omogućava jasno razdvajanje odgovornosti između pojedinih dijelova sistema, lakše održavanje koda i mogućnost budućeg proširenja funkcionalnosti.

### Prezentacijski sloj (Presentation Layer)

Prezentacijski sloj zadužen je za prikaz korisničkog interfejsa i interakciju administratora sa sistemom. Ovaj sloj realizovan je pomoću **FXML datoteka** koje definišu izgled pojedinih ekrana (login, dashboard, prijave, rang lista i dr.), uz pripadajuće **CSS stilove** za vizuelni dizajn aplikacije.

Logika upravljanja korisničkim interfejsom implementirana je u **controller klasama**, koje obrađuju korisničke akcije, poput klikova na dugmad, unosa podataka i navigacije kroz aplikaciju. Svaki veći ekran aplikacije ima odgovarajući controller, čime je osigurana modularnost i preglednost koda.

## Poslovna logika (Business Logic Layer)

Sloj poslovne logike sadrži pravila i procese koji definišu način rada sistema. U ovom sloju se obrađuju ključne funkcionalnosti aplikacije, kao što su:

- obrada prijava studenata,
- bodovanje na osnovu definisanih kriterija,
- promjena statusa prijava,
- generisanje rang liste,
- slanje email obavještenja studentima.

Poslovna logika je implementirana kroz **service klase**, koje povezuju korisnički interfejs sa podacima iz baze i obezbjeđuju pravilnu obradu zahtjeva sistema.

## Sloj pristupa podacima (Data Access Layer – DAO)

Sloj pristupa podacima zadužen je za komunikaciju sa bazom podataka. U ovom sloju se nalaze **DAO (Data Access Object) klase**, koje omogućavaju čitanje, upis, izmjenu i brisanje podataka vezanih za studente, prijave, dokumente, statuse i korisnike sistema.

Korištenjem DAO sloja postignuto je razdvajanje poslovne logike od direktnog rada sa bazom podataka, čime se povećava sigurnost, fleksibilnost i održivost sistema.

## Model sloj (Model Layer)

Model sloj sadrži klase koje predstavljaju osnovne entitete sistema, kao što su student, prijava, dokument, status prijave i korisnik. Ove klase definišu strukturu podataka i koriste se kroz sve slojeve aplikacije kao nosioci informacija.

## Pomoćni i servisni slojevi

Pored osnovnih slojeva, sistem sadrži i pomoćne komponente, poput:

- util klasa za rad sa lozinkama, tokenima i tekstualnim podacima,
- servisa za generisanje PDF dokumenata i eksport podataka,
- email servisa za slanje obavještenja administratorima i studentima.

## Napomena o organizaciji i dokumentaciji izvornog koda

Izvorni kod aplikacije E-Dom organizovan je u skladu sa principima slojne arhitekture i MVC dizajnerskog uzorka, što olakšava razumijevanje, održavanje i dalji razvoj sistema. Svaki sloj aplikacije ima jasno definisanu odgovornost, pri čemu je prezentacijski sloj odvojen od poslovne logike i pristupa podacima. Komunikacija sa bazom podataka realizovana je putem DAO obrazaca, čime je smanjena zavisnost ostatka sistema od konkretne implementacije baze. Sigurnosne mjere, poput šifrovanja lozinki i kontrole pristupa, implementirane su u okviru poslovne logike. Ovakva struktura koda omogućava lakše snalaženje novim programerima i jasno razdvajanje funkcionalnosti unutar sistema.

## 16. Korištene tehnologije i alati

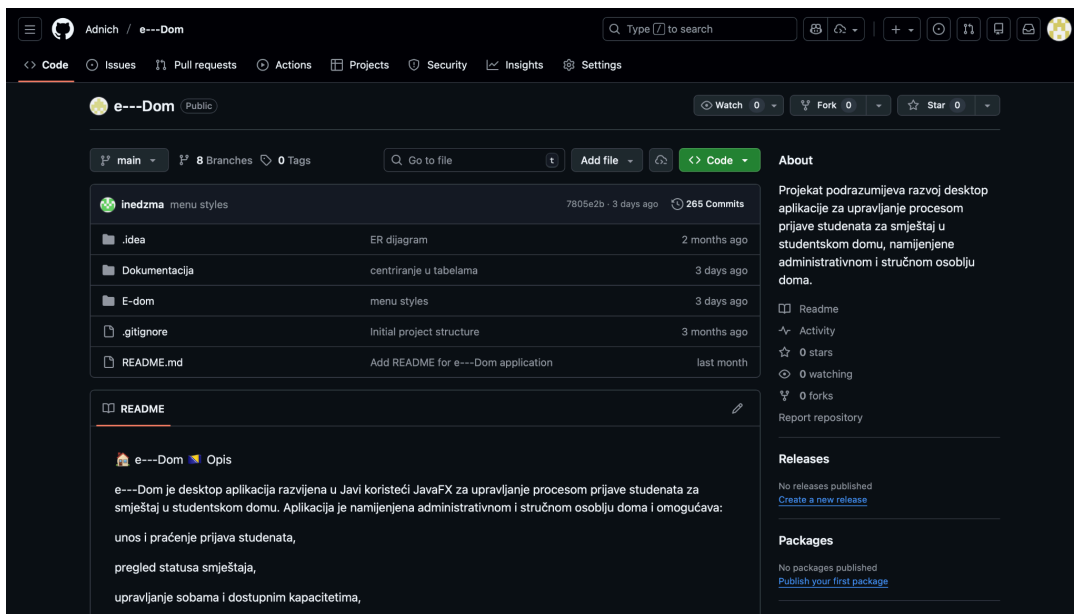
Razvoj desktop aplikacije **E-Dom** realizovan je korištenjem savremenih tehnologija i alata koji omogućavaju stabilan, modularan i timski orijentisan razvoj softverskog rješenja. Za organizaciju i upravljanje projektnim zavisnostima korišten je Apache Maven, koji omogućava jednostavno upravljanje bibliotekama i konzistentno okruženje za razvoj aplikacije.

U razvoju aplikacije korišten je GitHub kao platforma za verzionisanje koda i timsku saradnju. Svaka članica tima koristila je vlastiti GitHub profil, što je omogućilo praćenje individualnog doprinosa, upravljanje izmjenama koda, kao i sigurnu i kontrolisanu saradnju kroz repozitorij projekta.

Za pohranu i upravljanje podacima korištena je cloud baza podataka Aiven, koja omogućava pouzdan i udaljen pristup bazi bez potrebe za lokalnim serverskim okruženjem. Korištenje cloud baze omogućilo je lakše testiranje aplikacije, rad više članova tima nad istim podacima, kao i veću dostupnost i sigurnost podataka. Za pristup bazi podataka korišten je DAO (Data Access Object) pristup, koji omogućava jasnu separaciju između poslovne logike aplikacije i sloja za rad s bazom podataka, čime se povećava održivost i proširivost sistema.

Razvoj i testiranje aplikacije vršeno je u integrisanom razvojnom okruženju (IDE) koje pruža podršku za Java i JavaFX aplikacije, uz korištenje dodatnih alata za testiranje funkcionalnosti, dizajn korisničkog interfejsa i provjeru stabilnosti sistema.

Link za GitHub repository: <https://github.com/Adnich/e---Dom>



Slika 9. GitHub repository

## 17. Upravljanje projektom i organizacija rada

Za upravljanje razvojem aplikacije **E-Dom** odabran je hibridni **Scrumban pristup**, koji kombinuje strukturu Scrum metodologije sa vizuelnom transparentnošću Kanban sistema. Ovakav pristup omogućio je timu da zadrži fokus kroz definisane vremenske okvire (sprintove), dok je istovremeno osigurana fleksibilnost u praćenju svakodnevnih zadataka putem digitalnih ploča.

### Metodologija i alati (Scrumban)

Organizacija zadataka vršila se putem alata **YouTrack**, gdje su svi projektni zadaci bili vizualizirani na **Agile Board** (Kanban) pločama. Svaki zadatak (task) prolazio je kroz jasno definisan životni ciklus, podijeljen u četiri ključne kolone:

- **Open:** Zadaci koji su planirani za trenutni sprint, ali rad na njima još nije započeo.
- **In Progress:** Zadaci na kojima članovi tima trenutno aktivno rade.
- **To Verify:** Faza u kojoj se završena funkcionalnost pregleda i testira prije konačnog odobrenja.
- **Done:** Potpuno završeni i testirani zadaci.

Na osnovu sedmičnih sastanaka tima, **Nedžma Imamović** je bila zadužena za upravljanje ovim pločama, kreiranje novih taskova i ažuriranje njihovog statusa u skladu sa dogovorom.

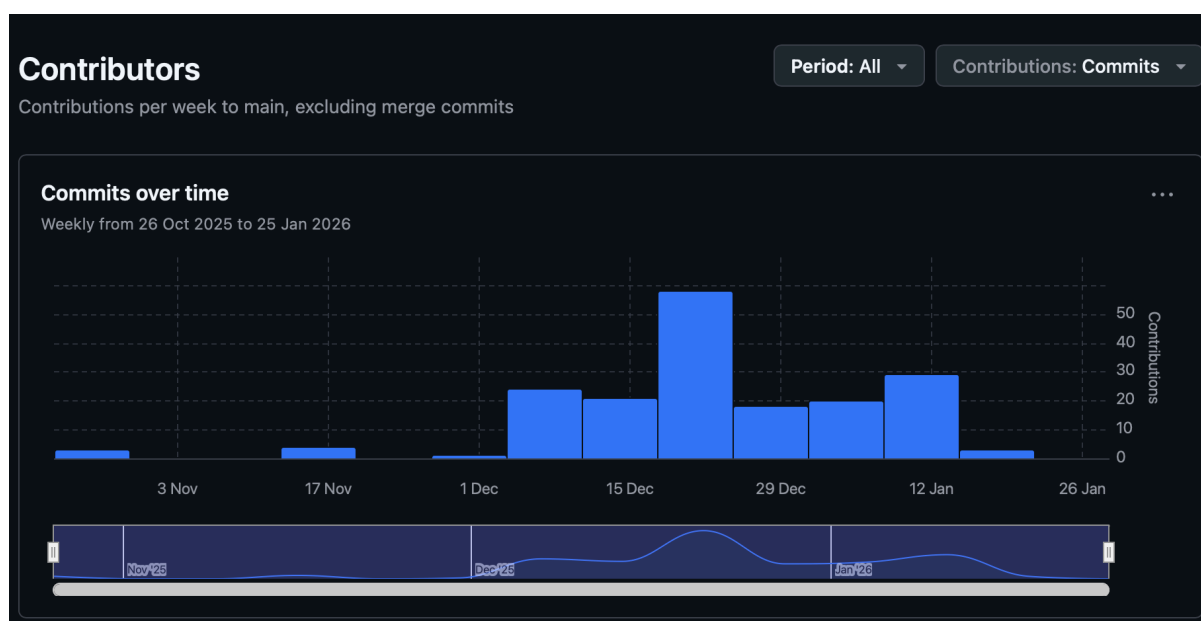
### Faze razvoja (Sprintovi)

Cjelokupan razvojni proces bio je podijeljen u četiri ključna sprinta:

1. **Sprint 1: Analiza i početna dokumentacija** – U ovoj fazi fokus je bio na analizi zahtjeva, izradi UML dijagrama i postavljanju arhitekture sistema.
2. **Sprint 2: Baza podataka i Backend temelji** – Kreiranje relacione baze na *cloud* platformi, razvoj Modela i DAO sloja, te analiza relacija između entiteta.
3. **Sprint 3: Razvoj glavnih funkcionalnosti i Dizajn** – Najintenzivnija faza u kojoj su paralelno razvijane ključne funkcionalnosti (logika aplikacije) i dizajn korisničkog interfejsa.
4. **Sprint 4: Finalizacija i dodatne funkcionalnosti** – Implementacija dodatnih opcija, "poliranje" izgleda aplikacije i završno testiranje sistema.

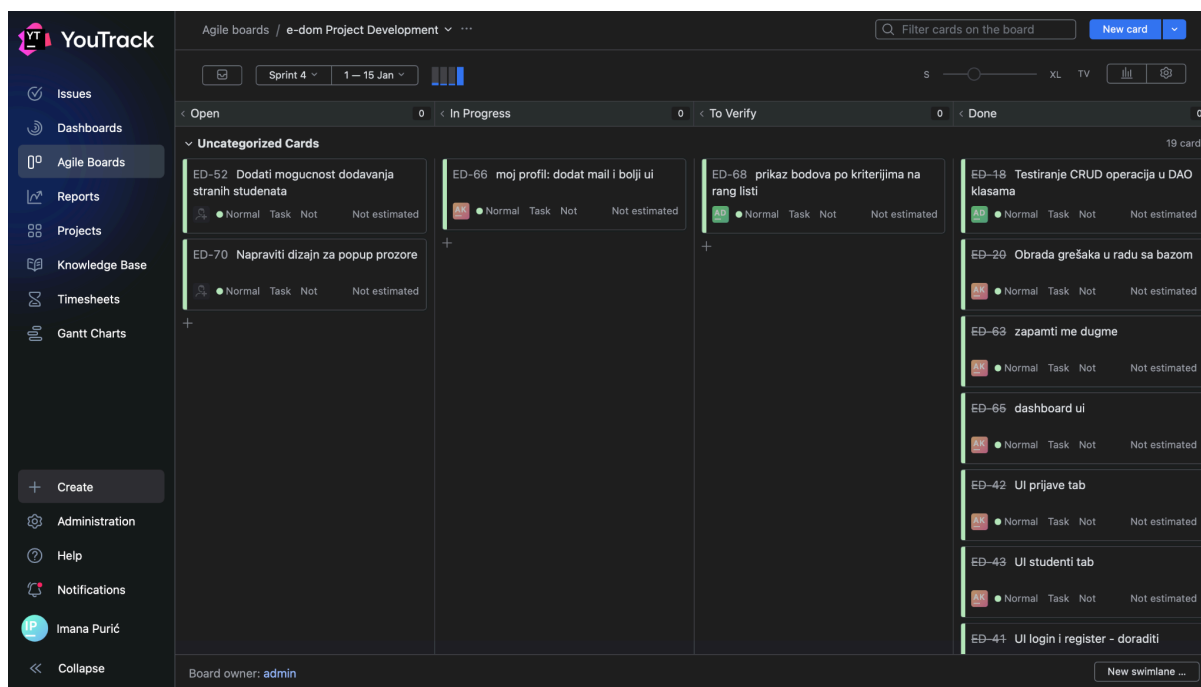
## Agilni razvoj i testiranje

U cilju praćenja kontinuiteta rada i timskog doprinosa, korištena je GitHub platforma koja omogućava transparentan uvid u aktivnosti razvoja kroz commit historiju. Analiza doprinosa po vremenskim periodima pokazuje intenzivnije faze razvoja tokom implementacije ključnih funkcionalnosti sistema. Rad unutar sprintova odvijao se agilno i paralelno. Dok su **Nedžma i Imana** razvijale backend logiku za određeni modul (npr. *Login*), **Adna** je istovremeno radila na dizajnu interfejsa za taj isti dio. Kvalitet softvera osiguran je kroz kontinuirano **manuelno testiranje**. Iako nisu korišteni automatizirani *JUnit* testovi, svaka funkcionalnost je detaljno ručno testirana i verifikovana kroz "To Verify" fazu na Kanban ploči prije nego što bi bila označena kao završena ("Done"), čime su greške otkrivane i ispravljane u ranoj fazi razvoja.



**Slika 10.** Pregled doprinosa članova tima (GitHub commits over time)

Vidljivo je da se najveći broj commit-a odnosi na period razvoja glavnih funkcionalnosti i finalizacije sistema, što odgovara planiranim sprint fazama projekta.



Slika 11. YouTrack agile boards

## 18. Zaključak

U okviru ovog projekta uspješno je razvijena desktop aplikacija **E-Dom**, namijenjena unapređenju administrativnih procesa u Studentskom centru Zenica. Aplikacija omogućava centralizovano upravljanje prijavama studenata, dokumentacijom, bodovanjem i statusima prijava, čime se značajno smanjuje potreba za manuelnim radom. Implementacijom slojne arhitekture postignuta je bolja organizacija koda, lakše održavanje sistema i mogućnost njegovog budućeg proširenja. Korištenjem savremenih tehnologija, kao što su JavaFX, cloud baza podataka i GitHub, osigurana je stabilnost i pouzdanost aplikacije. Sistem je prilagođen stvarnim potrebama administrativnog osoblja i omogućava efikasniji svakodnevni rad. Projekat predstavlja dobru osnovu za dalji razvoj sistema i potencijalno proširenje funkcionalnosti u budućnosti.