Programsko inženjerstvo

Ak. god. 2025./2026.

BreakoutSystems

Dokumentacija, Rev. *2*

Grupa: *BreakoutSystems*

Voditelj: *Iva Sović*

Datum predaje: *23. 1. 2026.*

Nastavnik: *Vlado Sruk*

Sadržaj

1 Dnevnik promjena dokumentacije 4

2 Opis projektnog zadatka 6

2.1 Cilj i opis projekta . . . 6

2.2 Problematika zadatka i postojeća rješenja . . . 8

2.2.1 Postojeća slična rješenja i analiza . . . 8

2.3 Potencijalna korist i ciljana skupina korisnika . . . 11

2.3.1 Ciljana skupina korisnika . . . 11

2.3.2 Potencijalna korist . . . 11

2.4 Opseg projekta i ključne funkcionalnosti . . . 12

2.4.1 MVP (Faza 1) . . . 12

2.5 Mogućnost prilagodbe i nadogradnje . . . 13

2.5.1 Mogućnost prilagodbe rješenja . . . 13

2.5.2 Moguće nadogradnje projektnog zadatka . . . 13

3 Analiza zahtjeva 14

3.1 Funkcionalni zahtjevi . . . 14

4 Specifikacija programske potpore 16

4.1 Obrasci uporabe . . . 16

4.1.1 Opis obrazaca uporabe . . . 16

4.2 Dijagrami obrazaca uporabe . . . 27

4.2.1 Visokorazinski dijagram obrazaca uporabe cijelog sustava . . . 27

4.2.2 Dijagram obrazaca uporabe za ključne značajke . . . 28

4.2.3 Dijagram obrazaca uporabe za korisničke uloge . . . 29

4.2.4 Dijagram obrazaca uporabe za osnovne poslovne procese . . . 33

4.2.5 Dijagram obrazaca uporabe za kritične sustave i integracije . . . 34

4.3 Sekvencijski dijagrami . . . 35

4.3.1 Ostali zahtjevi . . . 41

5 Arhitektura i dizajn sustava 42

5.1 Organizacija sustava . . . 42
5.1.1 Podsustavi i konektori . . . 42
5.1.2 Principi oblikovanja . . . 43
5.2 Baza podataka . . . 44
5.2.1 Opis tablica . . . 44
5.2.2 Dijagram baze podataka . . . 49
5.2.3 Dijagram razreda . . . 50
5.3 Dinamičko ponašanje aplikacije . . . 51
5.3.1 Dijagram stanja . . . 51
5.3.2 Dijagram aktivnosti . . . 52

6 Arhitektura komponenata i razmještaja 54
6.1 Dijagram komponenata . . . 54
6.1.1 Frontend podsustav . . . 54
6.1.2 Backend podsustav . . . 55
6.1.3 Vanjski servisi . . . 56
6.1.4 Web preglednik . . . 56
6.2 Dijagram razmještaja . . . 57
6.2.1 Microsoft Azure cloud platforma . . . 57
6.2.2 Komunikacija s vanjskim servisima . . . 58

7 Ispitivanje programskog rješenja 59
7.1 Ispitivanje komponenti . . . 59
7.1.1 Test klase . . . 59
7.2 Rezultati ispitivanja komponenti . . . 60
7.3 Ispitivanje sustava . . . 61
7.3.1 Ispitni slučajevi . . . 61
7.4 Rezultati ispitivanja sustava . . . 65
7.5 Zaključak . . . 66

8 Implementacija i korisničko sučelje 67
8.1 Tehnologije za implementaciju aplikacije . . . 67
8.1.1 Korištene tehnologije i alati . . . 67
8.1.2 Backend tehnologije . . . 68
8.1.3 Frontend tehnologije . . . 68
8.1.4 Baza podataka . . . 68
8.1.5 Ispitivanje sustava . . . 68

8.1.6 CI/CD i deployment 69
8.1.7 Licence 69
8.1.8 Ispitivanje programskog rješenja 69
8.2 Upute za puštanje u pogon 71
8.2.1 Instalacija potrebne programske potpore 71
8.2.2 Izgradnja frontenda 71
8.2.3 Pokretanje backenda 72
8.2.4 Konfiguracija environment varijabli 74
8.2.5 Pokretanje aplikacije 74

9 Zaključak i budući rad 76
9.1 Prva faza projekta 76
9.2 Druga faza projekta 76
9.3 Tehnički izazovi 77
9.4 Stečena znanja 77
9.5 Budući rad 77
9.6 Zaključne misli 78

10 Popis literature 79

Indeks slika i dijagrama 81

11 Dodatak: Prikaz aktivnosti grupe 82
11.1 Plan rada 82
11.1.1 Dijagram pregleda promjena 85

1. Dnevnik promjena dokumentacije

Tablica 1.1: Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak dokumentacije, inicijalna Home stranica	Antonio Mamić	23.10.2025.
0.2	Dodan opis projektnog zadatka	Hana Dolovčak	23.10.2025.
0.3	Proširena analiza zahtjeva	Antonio Mamić	24.10.2025.
0.4	Dopune analize zahtjeva	Antonio Mamić	27.–28.10.2025.
0.5	Dodan opis projektnog zadatka – proširenje	Hana Dolovčak	04.–07.11.2025.
0.6	Dodana relacijska baza podataka	Antonio Mamić	08.–09.11.2025.
0.7	Dodana specifikacija zahtjeva sustava – obrasci uporabe	Hana Dolovčak	10.–11.11.2025.
0.8	Ispravak opisa projektnog zadatka i analize zahtjeva	Antonio Mamić, Aurora Necko, Hana Dolovčak	12.11.2025.
0.9	Dodani dijagrami arhitekture i dizajna sustava	Hana Dolovčak	12.11.2025.
0.9.1	Dodan dijagram razreda	Lucija Stipetić	13.11.2025.
0.9.2	Dopune specifikacije zahtjeva sustava	Bruno Pećnik, Antonio Mamić, Hana Dolovčak	13.–14.11.2025.
1.0	Završena dokumentacija za prvu predaju	Hana Dolovčak, Antonio Mamić	14.11.2025.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.1	Ažuriranje arhitekture i dizajna sustava	Antonio Mamić	05.01.2026.
1.2	Dodana arhitektura komponenata i razmještaja	Bruno Pećnik	14.–15.01.2026.
1.3	Ažuriran opis projektnog zadatka, specifikacija zahtjeva, popis literature	Hana Dolovčak	15.01.2026.
1.4	Dodano ispitivanje programskog rješenja i tehnologije za implementaciju	Hana Dolovčak	16.01.2026.
1.5	Dodani dijagrami stanja i aktivnosti	Bruno Pećnik	16.01.2026.
1.6	Prošireno ispitivanje programskog rješenja, specifikacija zahtjeva	Hana Dolovčak	21.01.2026.
1.7	Ažurirana arhitektura i dizajn sustava – dijagrami	Bruno Pećnik	21.01.2026.
1.8	Dodane upute za puštanje u pogon, zaključak i budući rad	Antonio Mamić, Hana Dolovčak	22.01.2026.
1.9	Ažuriran dnevnik promjena dokumentacije, prikaz aktivnosti grupe	Aurora Necko, Hana Dolovčak, Lucija Stipetić	22.01.2026.
2.0	Konačna verzija dokumentacije	23.01.2026.	

2. Opis projektnog zadatka

2.1 Cilj i opis projekta

Escape Room industrija u Hrvatskoj i svijetu bilježi sve veći rast, ali još uvijek nema cjeloviti sustav koji bi objedinio sve aspekte poslovanja. Trenutačno se rezervacije obavljaju putem telefona, e-maila ili raznih booking platformi koje nisu prilagođene specifičnim potrebama Escape Roomova. Rezultati timova i rang-liste često se vode ručno, u Excel tablicama ili na papiru. Neautomatizirano praćenje rezultata nije efikasno i stvara nepotrebno opterećenje.

Cilj projekta je razvoj web platforme koja služi kao cjeloviti sustav za Escape Roomove. Igračima platforma omogućuje pretraživanje Escape Roomova, rezervaciju termina te praćenje rezultata i rangiranje timova, a vlasnicima oglašavanje njihovih soba te upravljanje terminima i rezultatima.

Neprijavljenom korisniku prikazuje se interaktivna karta dostupnih Escape Roomova s nazivima, lokacijama, opisima, cijenama i dostupnim terminima. Neprijavljeni korisnik može pregledavati Escape Roomove i rang-ljestvice timova, no za rezervaciju termina ili sudjelovanje u timovima treba se registirati.

Autentifikacija korisnika provodi se putem OAuth 2.0 protokola. Sustav razlikuje četiri vrste korisnika: neprijavljenog korisnika, korisnika polaznika Escape Rooma, korisnika vlasnika Escape Rooma i administratora sustava.

Polaznik Escape Rooma može osnivati timove ili se pridruživati postojećim timovima. U ime tima, korisnik koji je osnovao tim rezervira termine za Escape Room igre. Sustav osigurava da tim može rezervirati isključivo one Escape Roomove u kojima nijedan član tima nije prethodno sudjelovao. Prilikom rezervacije, korisnik plaća termin putem platforme po cijeni koju je definirao vlasnik Escape Rooma. Ako je rezervirani termin udaljen više od 24 sata, sustav automatski šalje podsjetnik svim članovima tima putem elektroničke pošte.

Vlasnik Escape Rooma putem platforme dodaje Escape Roomove, unosi njihove podatke (naziv, opis, slike, geografska lokacija, cijena i inicijalna težina) te definira dostupne termine. Nakon završetka igre, unosi rezultat za tim. Vlasnik plaća mjesečnu ili godišnju naknadu putem integriranog sustava plaćanja.

Platforma uključuje javno dostupan sustav rangiranja timova. Svaki Escape Room ima vlastitu ljestvicu rezultata temeljenu na vremenu završetka, dok globalna ljestvica rangira timove prema bodovima koji se izračunavaju na temelju brzine rješavanja i dinamičke težine Escape Rooma. Težina Escape Rooma kontinuirano se prilagođava na temelju ocjena korisnika nakon odigrane igre.

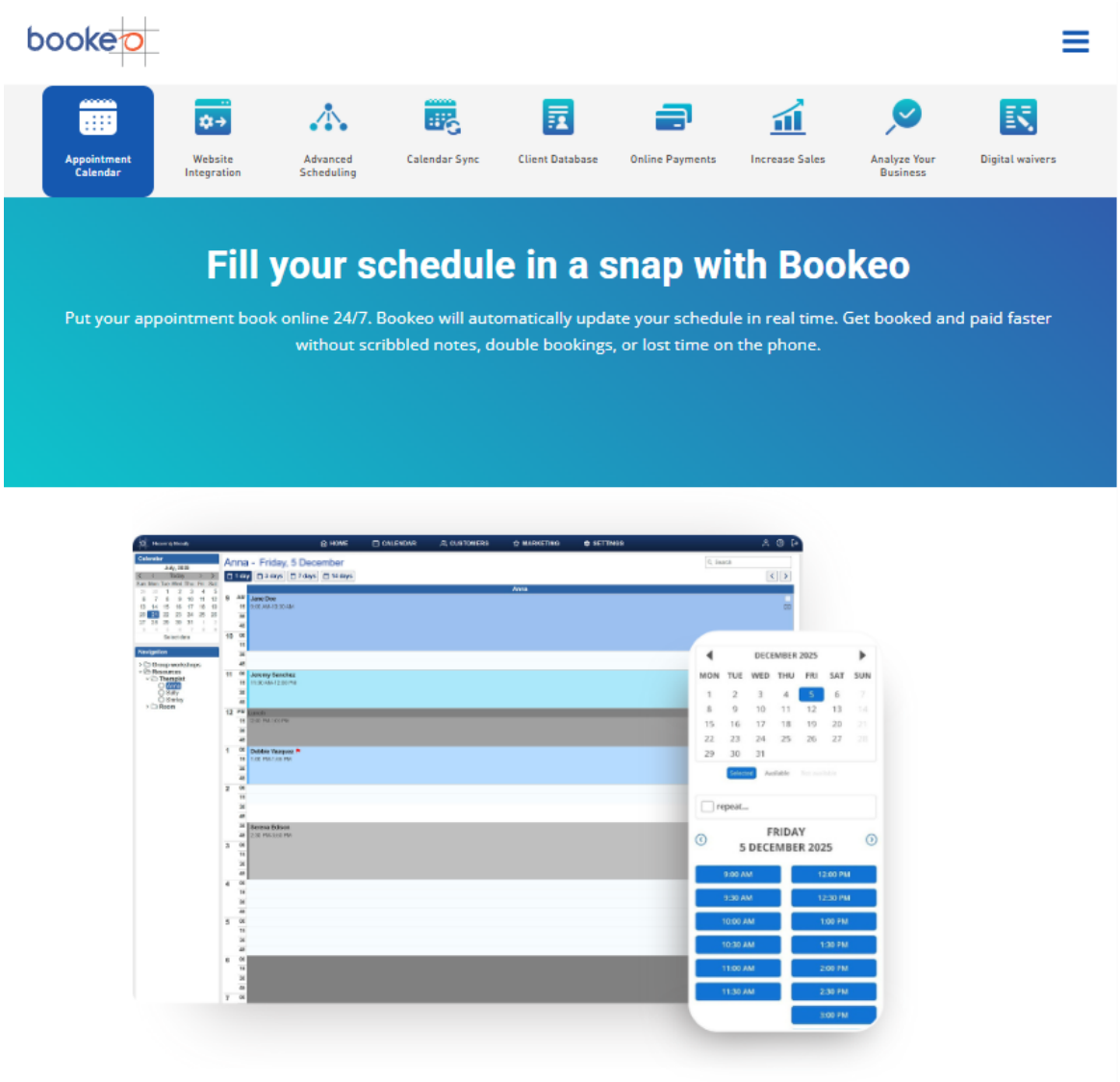
Plaćanja i prikaz geografskih lokacija ostvaruju se integracijom s vanjskim servisima.

2.2 Problematika zadatka i postojeća rješenja

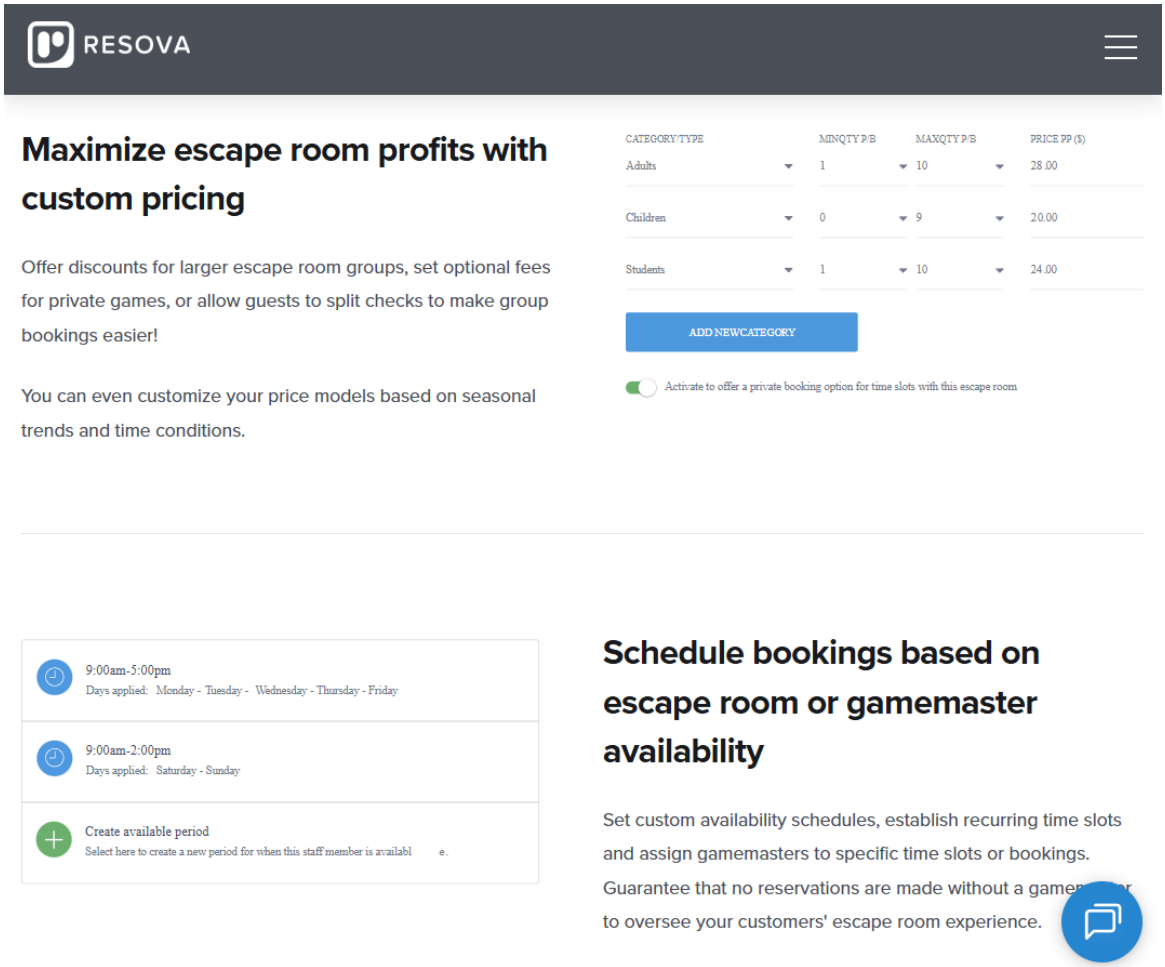
2.2.1 Postojeća slična rješenja i analiza

Na tržištu postoje platforme koje rješavaju dio opisanih problema, no nijedna ne pokriva sve potrebe. Najpoznatija rješenja su Bookeo i Resova, no one su specijalizirane za upravljanje rezervacijama u uslužnim djelatnostima.

Bookeo i Resova fokusirani su isključivo na upravljanje i optimizaciju rasporeda, upravljanje resursima, slanje automatskih potvrda i obradu online plaćanja. Te platforme fokusirane su samo na operativni dio. Ne sadrže sustav za rangiranje timova, niti poslovnu logiku specifičnu za Escape Room industriju.



Slika 2.1: Primjer sučelja Bookeo.



Slika 2.2: Primjer sučelja Resova.

Postojeći alati odlično obavljaju svoju primarnu funkciju upravljanja rezervacijama i mogu se koristiti kao dio rješenja, međutim, Escape Room industrija zahtijeva specifične funkcionalnosti koje trenutno nisu dostupne ni u jednom pojedinačnom sustavu. BreakoutSystems proširuje svojstva booking sustava i dodaje cjelovito rješenje potrebno za karakteristike Escape Roomova.

2.3 Potencijalna korist i ciljana skupina korisnika

2.3.1 Ciljana skupina korisnika

Realizacija projekta jamči ekonomsku i operativnu vrijednost za sve dionike. Projekt je usmjeren na dvije primarne skupine:

- **Polaznike Escape Roomova:** Korisnici koji žele brzo i jednostavno pronaći sobu koja odgovara njihovoj razini, rezervirati termin i natjecati se u timovima.
- **Vlasnike Escape Roomova:** Osobe ili tvrtke kojima treba jednostavan alat za upravljanje sobama, terminima i rezultatima na jednom mjestu.

2.3.2 Potencijalna korist

Vrijednost za polaznike Escape Roomova

Igrači mogu vidjeti kako stoje u odnosu na druge timove, a sustav rangiranja motivira ih da isprobavaju različite sobe. Pravilo da se ista soba ne može ponovo igrati (čak ni u drugom timu) potiče ih na istraživanje i igranje novih soba. Ocjene težine soba pomažu igračima pri odabiru sljedeće igre i vlasnicima daju točniju povratnu informaciju.

Vrijednost za vlasnike Escape Roomova

Umjesto plaćanja i održavanja više zasebnih programskih rješenja, vlasnicima soba platforma donosi objedinjenje poslovanja i smanjenje operativnih troškova.

2.4 Opseg projekta i ključne funkcionalnosti

Projekt se razvija kroz tri razvojne faze pri čemu prva faza predstavlja temeljnu implementaciju i minimalno održivi proizvod (MVP). Fokus je na postavljanju osnovnih funkcionalnosti koje platforma mora imati kako bi bila korisna i spremna za nadogradnju.

2.4.1 MVP (Faza 1)

Sustav rezervacija:

- pregled termina u kalendaru
- odabir željenog termina
- plaćanje putem integriranog vanjskog servisa

Sigurnost i autentifikacija:

- prijava korisnika pomoću OAuth 2.0 standarda

Sustav rangiranja:

- dinamički algoritam koji računa bodove timova na temelju vremena završetka i težine sobe

Geografska lokacija:

- interaktivna karta koja prikazuje lokacije svih dostupnih Escape Roomova

2.5 Mogućnost prilagodbe i nadogradnje

2.5.1 Mogućnost prilagodbe rješenja

Sustav je prilagodljiv i može se jednostavno primijeniti na druge srodne djelatnosti:

- Laser Tag
- Paintball
- Axe Throwing arene
- Karting staze
- Rezervaciju sportskih terena s evidencijom rezultata

2.5.2 Moguće nadogradnje projektnog zadatka

Faza 2

Platforma ima potencijal za razvoj dodatnih funkcionalnosti kao što su:

- Chat unutar aplikacije za komunikaciju i dogovore članova tima
- Sustav bodovanja i nagrađivanja korisnika za poticanje ponovnih rezervacija (popusti, prioriteti u rezervaciji)
- Napredni izvještaji za vlasnike: grafički prikazi ključnih metrika, predviđanje popunjenosti i ROI
- Sustav achievementa i badgeva za timove i pojedince
- Automatska sinkronizacija rezervacija s osobnim kalendarima (Google, Outlook)

Faza 3

- AI modul koji predlaže Escape Roomove koje timovi mogu igrati sljedeće
- Marketplace za fizičke suvenire i digitalne poklon-bonove

3. Analiza zahtjeva

3.1 Funkcionalni zahtjevi

Dionici

1. Vlasnici Escape Roomova
2. Polaznici Escape Roomova
   (a) Voditelj tima
   (b) Član tima
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. <u>Anonimni korisnik može:</u>

(a) pregledavati dostupne Escape Roomove

(b) koristiti interaktivnu kartu za pregled Escape Roomova

(c) pregledavati rang-ljestvice timova

(d) registrirati korisnički račun

2. <u>Član tima može:</u>

(a) sve što i anonimni korisnik

(b) pregledavati vlastite rezultate i povijest posjećenih Escape Roomova

(c) organizirati se u timove i slati pozivnice

(d) primati e-mail podsjetnike o nadolazećim terminima

(e) ocjenjivati posjećene Escape Roomove

3. Vođa tima može:

(a) sve što i član tima

(b) rezervirati dostupne termine Escape Roomova za svoj tim

(c) spriječiti rezervaciju već posjećenih Escape Roomova od strane članova tima

(d) platiti rezervaciju prilikom same rezervacije

4. Vlasnik Escape Rooma može:

(a) sve što i vođa tima

(b) dodavati nove Escape Roomove i uređivati njihove podatke

(c) otvarati termine za posjete Escape Roomovima

(d) plaćati naknadu za dodane Escape Roomove

5. Administrator može:

(a) sve što i vlasnik Escape Rooma

(b) mijenjati podatke sustava

(c) brisati korisničke profile i Escape Roomove

(d) pristupati statistikama i nadgledati aktivnost korisnika

(e) dodavati nove administratore i upravljati pravima korisnika

6. Vanjski servisi mogu:

(a) omogućiti autentifikaciju korisnika

(b) omogućiti online plaćanje

(c) integrirati interaktivnu kartu

7. Baza podataka može:

(a) pohranjivati sve podatke o korisnicima, timovima i njihovim ovlastima

(b) pohranjivati sve podatke o Escape Roomovima i terminima

(c) pohranjivati informacije o rezultatima, rezervacijama i rang-listama timova

(d) podržavati pristup vanjskih servisa i administrativne funkcionalnosti

4. Specifikacija programske potpore

4.1 Obrasci uporabe

4.1.1 Opis obrazaca uporabe

UC1 – Registracija korisnika

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti novi korisnički račun na platformi
- **Sudionici:** Neregistrirani korisnik, OAuth 2.0 servis (GitHub), Baza podataka
- **Preduvjet:** Neregistrirani korisnik ima GitHub račun
- **Opis osnovnog tijeka:**
  1. Korisnik otvara stranicu za registraciju.
  2. Korisnik popunjava podatke (vlasnik Escape Rooma ili igrač).
  3. Korisnik klikne "Registriraj se".
  4. Korisnik se preusmjerava na GitHub OAuth servis.
  5. Kreira se novi zapis u bazi podataka s podatcima o novom korisniku.
  6. Učitava se stranica profila.
- **Opis mogućih odstupanja:**

3. a) GitHub servis nedostupan
   1. Prikazuje se poruka da servis za prijavu trenutačno nije dostupan.
5. a) Korisnik već postoji u bazi
   1. Korisnik se preusmjerava na stranicu za prijavu uz prikaz popup poruke o grešci.

UC2 – Prijava korisnika

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Prijaviti se na platformu
- **Sudionici:** OAuth 2.0 servis (GitHub), Baza podataka
- **Preduvjet:** Registrirani korisnik ima GitHub račun
- **Opis osnovnog tijeka:**
  1. Korisnik otvara stranicu za prijavu.

2. Korisnik klikne "Prijava".
3. Korisnik se preusmjerava na GitHub OAuth servis.
4. Pretražuje se baza podataka, odnosno podatci o korisniku.
5. Korisnik se preusmjerava na stranicu profila.

- **Opis mogućih odstupanja:**

3. a) GitHub servis nedostupan
    1. Prikazuje se poruka da servis za prijavu trenutačno nije dostupan.
4. a) Korisnik ne postoji u bazi
    1. Korisnik dobiva poruku da nije registriran.
    2. Korisnik se preusmjerava na stranicu za registraciju.

UC3 – Pregled dostupnih Escape Roomova

- **Glavni sudionik:** Korisnik (anonimni ili prijavljeni)
- **Cilj:** Pregledati sve dostupne Escape Roomove na platformi
- **Sudionici:** Baza podataka, Kartografski servis
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
    1. Korisnik otvara stranicu "Escape Rooms".
    2. Dohvaćaju se svi podaci o Escape Roomovima iz baze podataka.
    3. Prikazuje se lista Escape Roomova.
    4. Integrira se kartografski servis i prikazuje interaktivna karta s markerima.
    5. Korisnik može kliknuti na pojedini Escape Room za više detalja.
- **Opis mogućih odstupanja:**

2. a) Nema Escape Roomova u bazi
    1. Prikazuje se poruka "Trenutačno nema dostupnih Escape Roomova".
4. a) Kartografski servis nedostupan
    1. Prikazuje se karta bez pozadine.

UC4 – Filtriranje Escape Roomova

- **Glavni sudionik:** Korisnik
- **Cilj:** Pronaći Escape Roomove prema specifičnim kriterijima
- **Sudionici:** Baza podataka
- **Preduvjet:** UC3 (Pregled dostupnih Escape Roomova)
- **Opis osnovnog tijeka:**

1. Korisnik odabire kriterije filtriranja (lokacija, tim, žanr, težina).
2. Izvršava se upit u bazi podataka.
3. Dohvaćaju se Escape Roomovi koji zadovoljavaju sve kriterije.
4. Ažurira se prikaz liste i karte s filtriranim rezultatima.

- **Opis mogućih odstupanja:**

3. a) Nijedan Escape Room ne zadovoljava sve kriterije
    1. Prikazuje se poruka "Nema rezultata za odabrane kriterije".

UC5 – Pregled detalja Escape Rooma

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati detaljne informacije o Escape Roomu
- **Sudionici:** Baza podataka
- **Preduvjet:** UC3 (Pregled dostupnih Escape Roomova)
- **Opis osnovnog tijeka:**
    1. Korisnik odabire Escape Room.
    2. Dohvaćaju se detaljni podaci iz baze podataka (naziv, opis, slike, težina, cijena, geografska lokacija).
    3. Dohvaćaju se dostupni termini iz baze podataka.
    4. Dohvaća se podaci o najboljim rezultatima timova za Escape Room iz baze podataka.
    5. Prikazuju se sve informacije na stranici.
- **Opis mogućih odstupanja:**

3. a) Nema dostupnih termina
    1. Prikazuje se poruka "Trenutačno nema dostupnih termina".

UC6 – Kreiranje tima

- **Glavni sudionik:** Prijavljeni korisnik (polaznik)
- **Cilj:** Osnovati novi tim
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava korisnika)
- **Opis osnovnog tijeka:**
    1. Korisnik odabire "Kreiraj novi tim".
    2. Prikazuje se forma za unos podataka tima.
    3. Korisnik unosi podatke i klikne "Kreiraj".

4. Provjerava se postoji li već tim s istim nazivom u tablici timovi.
5. Kreira se novi zapis u bazi podataka.
6. Korisnik se preusmjerava na stranicu profila.

- **Opis mogućih odstupanja:**

4. a) Tim s istim nazivom već postoji
    1. Prikazuje se poruka "Tim s ovim nazivom već postoji".
    2. Korisnik se vraća na korak 3.

UC7 – Pozivanje članova u tim

- **Glavni sudionik:** Voditelj tima
- **Cilj:** Pozvati druge korisnike da se pridruže timu
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Kreiranje tima)
- **Opis osnovnog tijeka:**
    1. Voditelj otvara novi tim.
    2. Voditelj odabire "Pozovi članove".
    3. Voditelj unosi korisnička imena korisnika koje želi pozvati.
    4. Provjerava se u bazi podataka postoje li korisnici s tim korisničkim imenima.
    5. Šalje se poziv preko platforme.
    6. Prikazuje se poruka "Poslano".
- **Opis mogućih odstupanja:**

4. a) Korisnik s tim korisničkim imenom ne postoji u sustavu
    1. Prikazuje se poruka "Korisnik s korisničkim imenom '{korisničko ime}' ne postoji".
4. b) Korisnik je već član tima
    1. Provjerava se baza podataka.
    2. Prikazuje se poruka "Korisnik je već član tima".

UC8 – Prihvaćanje poziva u tim

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Prihvatiti ili odbiti pozivnicu za tim
- **Sudionici:** Baza podataka
- **Preduvjet:** UC7 (Pozivanje članova u tim)
- **Opis osnovnog tijeka:**

1. Poziv u tim korisniku se prikazuje na profilu.
2. Prikazuju se detalji tima (naziv, broj članova, voditelj).
3. Korisnik odabire "Prihvati" ili "Odbij".
4. Ažurira se status članstva u bazi podataka.
5. Prikazuje se odgovarajuća poruka potvrde.

- **Opis mogućih odstupanja:**

4. a) Korisnik odbije pozivnicu
    1. Zahtjev za članstvo briše se iz baze podataka.

UC9 – Napuštanje tima

- **Glavni sudionik:** Član tima
- **Cilj:** Napustiti tim
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Kreiranje tima) i UC8 (Prihvaćanje poziva u tim)
- **Opis osnovnog tijeka:**
    1. Korisnik otvara stranicu profila.
    2. Korisnik klikne "Napusti tim".
    3. Prikazuje se poruka za potvrdu akcije.
    4. Briše se korisnik iz tablice članovi tima u bazi podataka.
    5. Prikazuje se poruka "Uspješno ste napustili tim".
- **Opis mogućih odstupanja:** -

UC10 – Pregled tima

- **Glavni sudionik:** Član tima
- **Cilj:** Pregledati informacije o timu
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Kreiranje tima) i UC8 (Prihvaćanje poziva u tim)
- **Opis osnovnog tijeka:**
    1. Korisnik odabire tim iz liste svojih timova.
    2. Dohvaćaju se osnovni podaci tima iz baze.
    3. Prikazuju se informacije: naziv i članovi.
- **Opis mogućih odstupanja:** -

UC11 – Rezervacija termina

- **Glavni sudionik:** Voditelj tima
- **Cilj:** Rezervirati termin za tim u Escape Roomu
- **Sudionici:** Baza podataka, servis za plaćanje
- **Preduvjet:** UC6 (Kreiranje tima), UC5 (Pregled detalja Escape Rooma)
- **Opis osnovnog tijeka:**
  1. Voditelj otvara stranicu Escape Rooma.
  2. Dohvaćaju se svi dostupni termini za odabrani Escape Room.
  3. Prikazuju se dostupni termini (datum, vrijeme, cijena).
  4. Voditelj odabire željeni termin.
  5. Dohvaćaju se svi članovi tima iz baze podataka.
  6. Provjerava se za svakog člana u tablici rezultati je li već bio u tom Escape Roomu.
  7. Korisnik se preusmjerava na servis za plaćanje.
  8. Servis za plaćanje obrađuje transakciju i vraća potvrdu plaćanja.
  9. Status plaćanja sprema se u bazu podataka.
  10. Ako je termin udaljen više od 24 sata, zakazuje se slanje e-mail podsjetnika članovima tima.
- **Opis mogućih odstupanja:**

3. a) Nema dostupnih termina
   1. Prikazuje se poruka "Trenutačno nema dostupnih termina za ovaj Escape Room".
6. a) Jedan ili više članova već je igralo ovaj Escape Room
   1. Prikazuje se upozorenje "Netko od članova tima već je igrao ovu sobu.".
   2. Nudi se opcija "Nastavi s rezervacijom" ili "Odustani".
   3. Ako nastavi nastavlja se s korakom 7.
9. a) Plaćanje odbijeno
   1. Servis za plaćanje vraća grešku.
   2. Prikazuje se poruka "Plaćanje neuspješno. Molimo pokušajte ponovno".

UC12 – Slanje e-mail podsjetnika o terminu

- **Glavni sudionik:** Član tima
- **Cilj:** Primiti podsjetnik o nadolazećem terminu
- **Sudionici:** Baza podataka, e-mail servis, pozadinski proces
- **Preduvjet:** UC11 (Rezervacija termina)
- **Opis osnovnog tijeka:**
  1. Pozadinski proces pokreće se svakih 5 minuta.

2. Dohvaćaju se sve rezervacije iz baze podataka čije je vrijeme slanja dostiglo trenutačno vrijeme (24 sata prije termina).
3. Za svaku rezervaciju dohvaćaju se svi članovi tima.
4. Dohvaćaju se e-mail adrese članova.
5. Šalje se e-mail podsjetnik svim članovima putem e-mail servisa.

- **Opis mogućih odstupanja:**

5. a) E-mail servis nedostupan
    1. Pokušava se ponovno poslati e-mail nakon 10 minuta.

UC13 – Plaćanje pretplate vlasnika

- **Glavni sudionik:** Vlasnik Escape Rooma
- **Cilj:** Platiti mjesečnu ili godišnju pretplatu za oglašavanje Escape Rooma
- **Sudionici:** Baza podataka, Servis za plaćanje
- **Preduvjet:** UC2 (Prijava korisnika)
- **Opis osnovnog tijeka:**
    1. Vlasnik otvara stranicu profila.
    2. Prikazuju se dostupni planovi pretplate (mjesečni, godišnji) s cijenama izračunatim s obzirom na broj Escape Roomova.
    3. Vlasnik odabire plan pretplate.
    4. Vlasnik se preusmjerava na servis za plaćanje.
    5. Servis za plaćanje obrađuje transakciju i vraća potvrdu plaćanja.
    6. Kreira se ili ažurira zapis u bazi podataka.
    7. Prikazuje se poruka "Plaćanje izvršeno".
- **Opis mogućih odstupanja:**

5. a) Plaćanje odbijeno
    1. Servis za plaćanje vraća grešku.
    2. Prikazuje se poruka "Plaćanje neuspješno. Molimo pokušajte ponovno".

UC14 – Dodavanje Escape Rooma

- **Glavni sudionik:** Vlasnik Escape Rooma
- **Cilj:** Dodati novi Escape Room na platformu
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava korisnika), korisnik je tipa vlasnik
- **Opis osnovnog tijeka:**

1. Vlasnik odabire "Dodaj novu sobu".
2. Prikazuje se forma za unos podataka (naziv, opis, slike, geografska lokacija, inicijalna težina, cijena, adresa).
3. Vlasnik popunjava formu i klikne "Objavi".
4. Kreira se novi zapis u bazi podataka s podacima o sobi.
5. Prikazuje se poruka "Escape Room uspješno dodan".

- **Opis mogućih odstupanja:** -

UC15 – Upravljanje terminima Escape Rooma

- **Glavni sudionik:** Vlasnik Escape Rooma
- **Cilj:** Dodati, urediti ili ukloniti termine za rezervacije
- **Sudionici:** Baza podataka
- **Preduvjet:** UC14 (Dodavanje Escape Rooma)
- **Opis osnovnog tijeka:**
  1. Vlasnik otvara svoj profil i odabire Escape Room koji želi uređivati.
  2. Sustav dohvaća sve postojeće termine iz tablice termin za odabrani Escape Room.
  3. Sustav prikazuje termine u obliku kalendara ili tablice.
  4. Vlasnik odabire jednu od akcija (dodaj novi termin, uredi postojeći termin, obriši termin).
  5. Sustav provjerava je li novi ili izmijenjeni termin u konfliktu s postojećim terminima.
  6. Ako nema preklapanja, sustav ažurira zapise u bazi podataka.
  7. Sustav prikazuje poruku „Termini su uspješno ažurirani."
- **Opis mogućih odstupanja:** -

UC16 – Unos rezultata tima

- **Glavni sudionik:** Vlasnik Escape Rooma
- **Cilj:** Unijeti rezultat tima nakon što su završili Escape Room
- **Sudionici:** Baza podataka
- **Preduvjet:** UC11 (Rezervacija termina)
- **Opis osnovnog tijeka:**
  1. Vlasnik otvara stranicu profila.
  2. Prikazuje se forma za unos naziva tima, termina i vremena koje je bilo potrebno timu za rješavanje ili statusa neriješenosti.
  3. Vlasnik unosi podatke i klikne "Unesi".

4. Kreira se novi zapis u bazi podataka s podacima.
5. Ažurira se rang ljestvica i ljestvica za pojedini Escape Room.
6. Prikazuje se poruka "Rezultat uspješno spremljen".

- **Opis mogućih odstupanja:** -

UC17 – Ocjenjivanje težine Escape Rooma

- **Glavni sudionik:** Član tima
- **Cilj:** Ocijeniti težinu odigranog Escape Rooma
- **Sudionici:** Baza podataka
- **Preduvjet:** UC16 (Unos rezultata tima)
- **Opis osnovnog tijeka:**
  1. Korisnik otvara stranicu profila.
  2. Prikazuje se povijest odigranih Escape Roomova.
  3. Korisnik odabire Escape Room i klikne "Ocijeni težinu".
  4. Korisnik odabire ocjenu i klikne "Ocijeni".
  5. Kreira se ili ažurira zapis u bazi podataka.
  6. Izračunava se nova prosječna težina Escape Rooma.
  7. Ažurira se težina u bazi podataka.
- **Opis mogućih odstupanja:**

2. a) Korisnik nije odigrao nijedan Escape Room
   1. Prikazuje se poruka "Nema dostupnih soba za ocijeniti".

UC18 – Pregled rang ljestvica

- **Glavni sudionik:** Korisnik (anonimni ili prijavljeni)
- **Cilj:** Pregledati ljestvice timova
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire stranicu "Leaderboard".
  2. Prikazuje se opcija izbora: globalna ljestvica ili ljestvica po Escape Roomu.
  3. Korisnik odabire opciju.
  4. Dohvaćaju se timovi i njihovi bodovi iz baze podataka.
  5. Sortiraju se timovi prema bodovima.
- **Opis mogućih odstupanja:**

4. a) Nema rezultata (još nema odigranih Escape Roomova)
    1. Prikazuje se poruka "Još nema timova na ljestvici".

UC19 – Upravljanje sadržajem

- **Glavni sudionik:** Administrator
- **Cilj:** Uređivati i nadzirati sadržaj na platformi
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava korisnika)
- **Opis osnovnog tijeka:**
    1. Administrator odabire Escape Room koji želi urediti.
    2. Dohvaćaju se podaci iz odgovarajuće tablice u bazi.
    3. Administrator uređuje ili briše sadržaj.
    4. Ažurira se ili briše zapis u bazi podataka.
- **Opis mogućih odstupanja:** -

UC20 – Pregled profila korisnika

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledati svoj profil
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava korisnika)
- **Opis osnovnog tijeka:**
    1. Korisnik odabire "Moj profil".
    2. Dohvaćaju se korisnički podaci.
    3. Dohvaćaju se svi timovi korisnika iz baze podataka.
    4. Dohvaćaju se prethodno odigrane sobe.
- **Opis mogućih odstupanja:** -

UC21 – Pregled raspoloživosti po karti

- **Glavni sudionik:** Korisnik
- **Cilj:** Pronaći slobodne termine prema lokaciji
- **Sudionici:** Baza podataka, Kartografski servis
- **Preduvjet:** UC3 (Pregled dostupnih Escape Roomova)

- **Opis osnovnog tijeka:**
  1. Korisnik otvara interaktivnu kartu na stranici s Escape Roomovima.
  2. Dohvaćaju se svi Escape Roomovi iz baze podataka.
  3. Korisnik klikne na pin i otvara se stranica Escape Rooma (UC5).
- **Opis mogućih odstupanja:**

2. a) Nema dostupnih Escape Roomova
   1. Prikazuje se poruka "Nema Escape Roomova u odabranom području".

UC22 – Integracija s kartografskim servisom

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazati geografske lokacije Escape Roomova na karti
- **Sudionici:** Baza podataka, Kartografski servis
- **Preduvjet:** UC3 (Pregled dostupnih Escape Roomova)
- **Opis osnovnog tijeka:**
  1. Dohvaćaju se geografske koordinate svih Escape Roomova iz baze podataka.
  2. Šalje se zahtjev kartografskom servisu s koordinatama.
  3. Kartografski servis vraća kartu s markerima.
  4. Prikazuje se interaktivna karta korisniku.
- **Opis mogućih odstupanja:**

2. a) Kartografski servis nedostupan
   1. Prikazuje se karta bez pozadine.

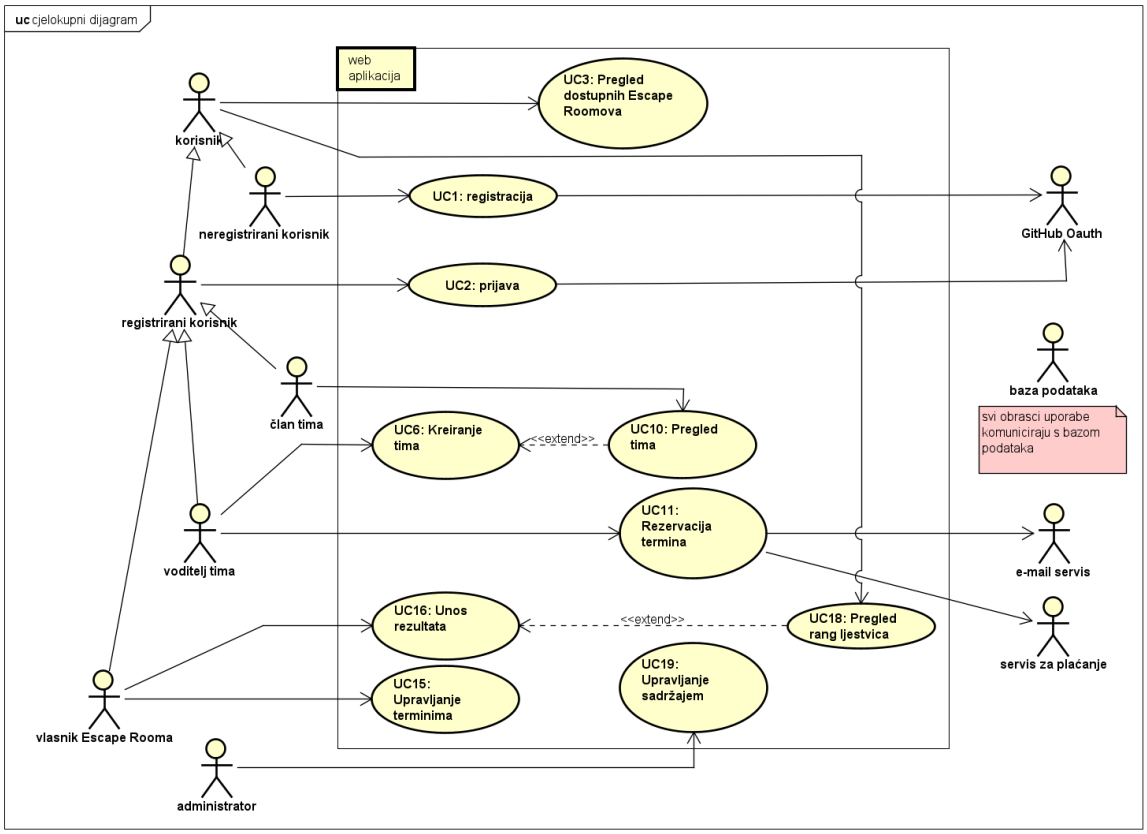
UC23 – Integracija sa servisom za plaćanje

- **Glavni sudionik:** Korisnik
- **Cilj:** Obraditi plaćanja rezervacija i pretplata
- **Sudionici:** Baza podataka, servis za plaćanje
- **Preduvjet:** UC11 (Rezervacija termina) ili UC13 (Plaćanje pretplate)
- **Opis osnovnog tijeka:**
  1. Korisnik se preusmjerava na servis za plaćanje.
  2. Servis za plaćanje vraća potvrdu plaćanja.
  3. Spremaju se podaci o transakciji u bazu podataka.
  4. Ažurira se status rezervacije ili pretplate.
- **Opis mogućih odstupanja:**

4. a) Plaćanje odbijeno
   1. Prikazuje se poruka "Plaćanje nije uspjelo. Molimo pokušajte ponovno".

4.2 Dijagrami obrazaca uporabe

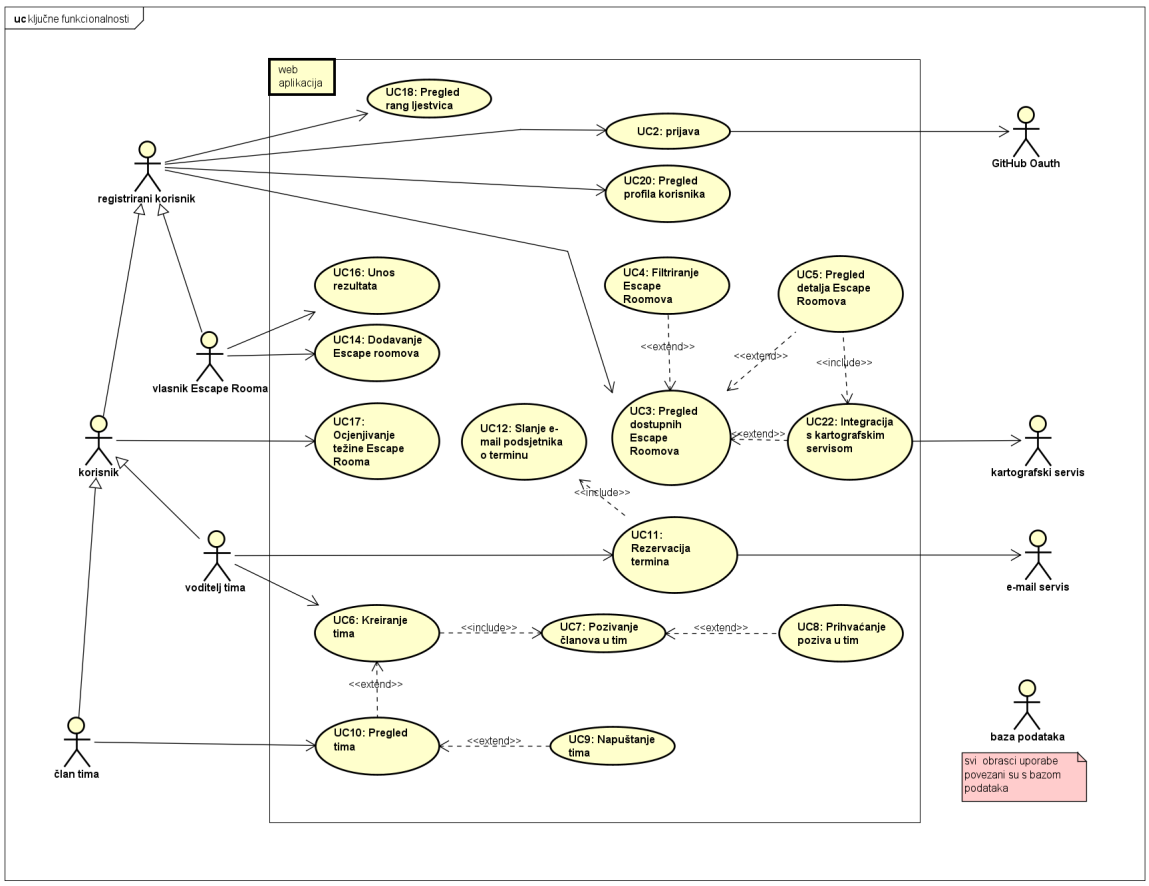
4.2.1 Visokorazinski dijagram obrazaca uporabe cijelog sustava



Slika 4.1: Cjelokupni dijagram obrazaca uporabe

Dijagram prikazuje sve komponente sustava i njihove međusobne veze. Prikazani su glavni aktori uključujući neprijavljene korisnike, članove tima, voditelja tima i vlasnike soba. Dijagram obuhvaća funkcionalnosti pregledavanja dostupnih soba, rezervacije termina, dodavanja timova, dodavanja Escape Roomova, unosa rezultata te globalnog sustava rangiranja.

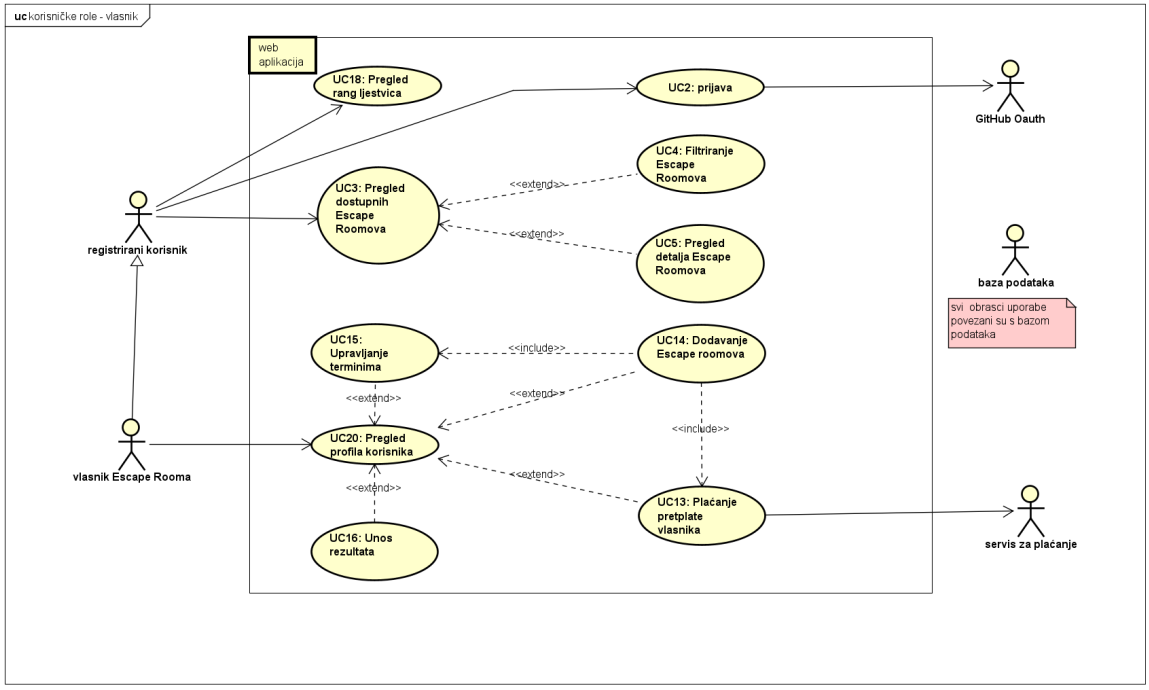
4.2.2 Dijagram obrazaca uporabe za ključne značajke



Slika 4.2: Dijagram obrazaca uporabe ključnih funkcionalnosti sustava

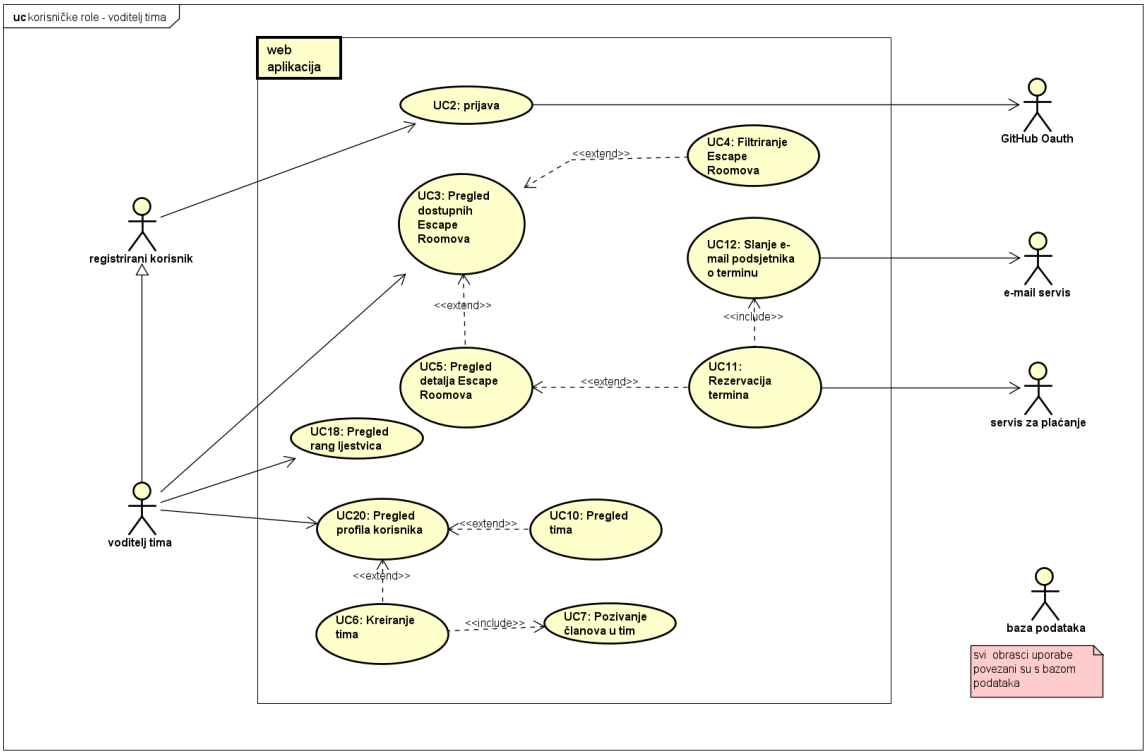
Dijagram ključnih funkcionalnosti fokusira se na najvažnije funkcije sustava. Prikazuje registraciju i prijavu korisnika, pretraživanje i dodavanje Escape Roomova, kreiranje tima, rezervaciju termina te praćenje i unos rezultata.

4.2.3 Dijagram obrazaca uporabe za korisničke uloge



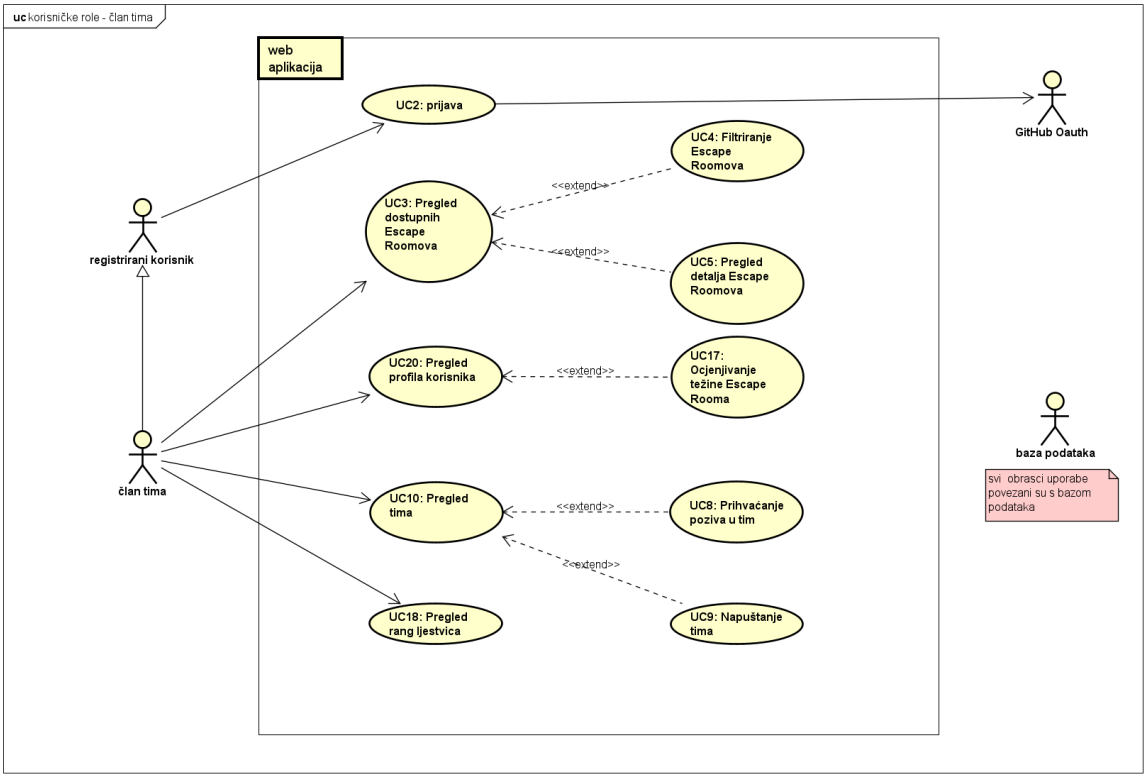
Slika 4.3: Dijagram obrazaca uporabe korisničkih uloga – vlasnik Escape Rooma

Dijagram prikazuje aktivnosti vlasnika Escape Rooma: dodavanje soba, upravljanje terminima, unos rezultata timova, pregled rang-ljestvica te interakcije s plaćanjima i integracijama sustava.



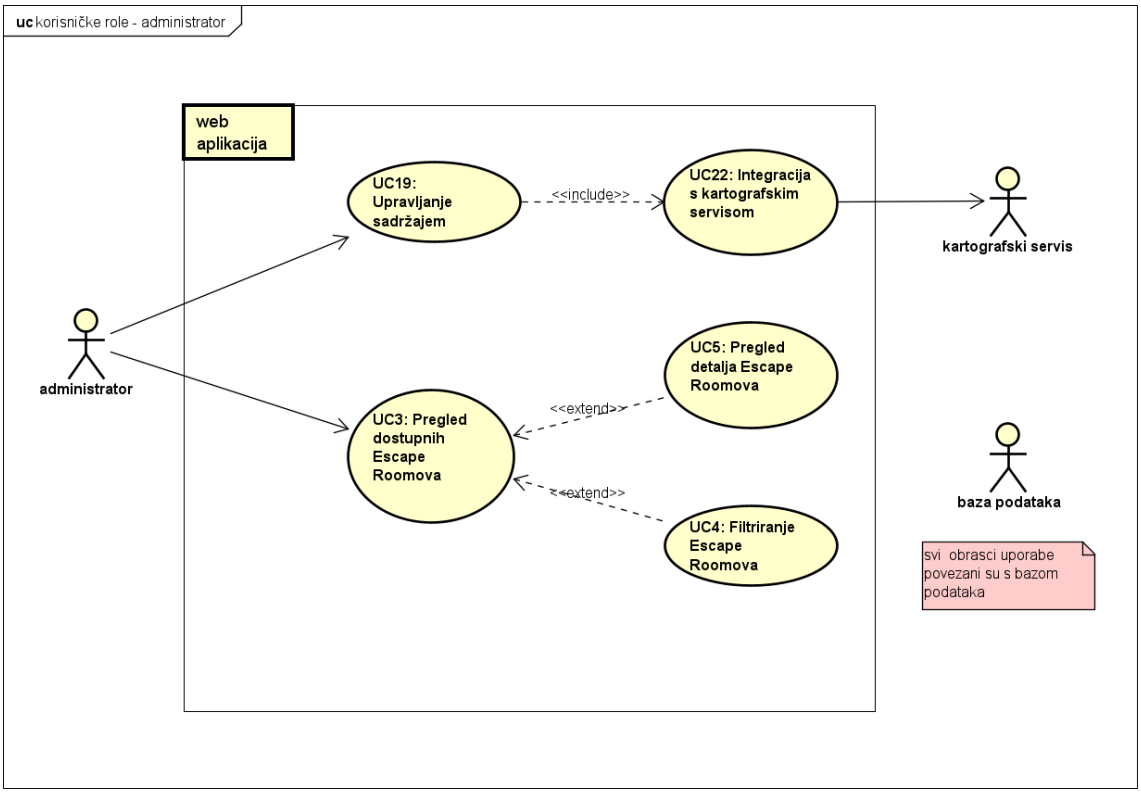
Slika 4.4: Dijagram obrazaca uporabe korisničkih uloga – voditelj tima

Dijagram prikazuje funkcionalnosti dostupne voditelju tima: kreiranje i upravljanje timom, pregled rang-ljestvica, pregled dostupnih soba te rezervaciju termina u ime tima.



Slika 4.5: Dijagram obrazaca uporabe korisničkih uloga – član tima

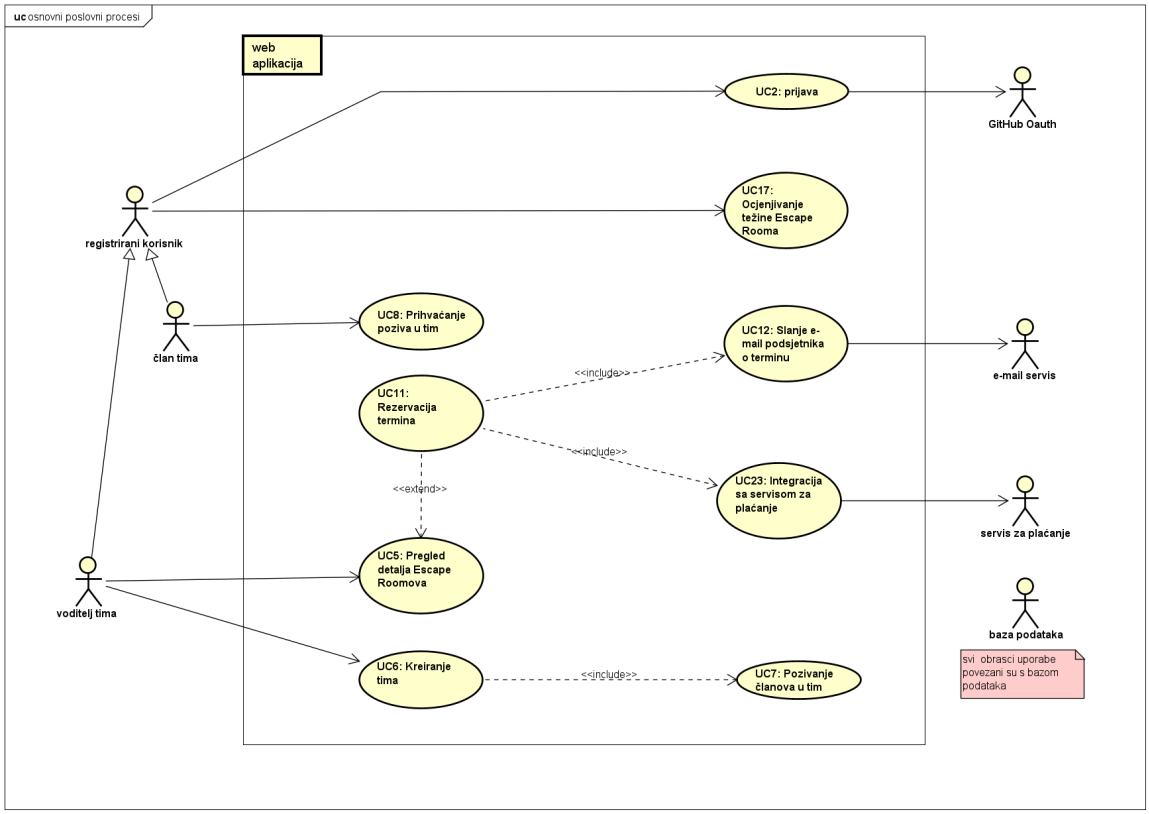
Dijagram prikazuje funkcionalnosti dostupne članovima tima: pregled tima, pregled soba te ocjenjivanje sobe nakon odigrane partije.



Slika 4.6: Dijagram obrazaca uporabe korisničkih uloga – administrator

Dijagram prikazuje aktivnosti administratora: upravljanje sadržajem, praćenje sustava i kontrolu ispravnosti rada platforme.

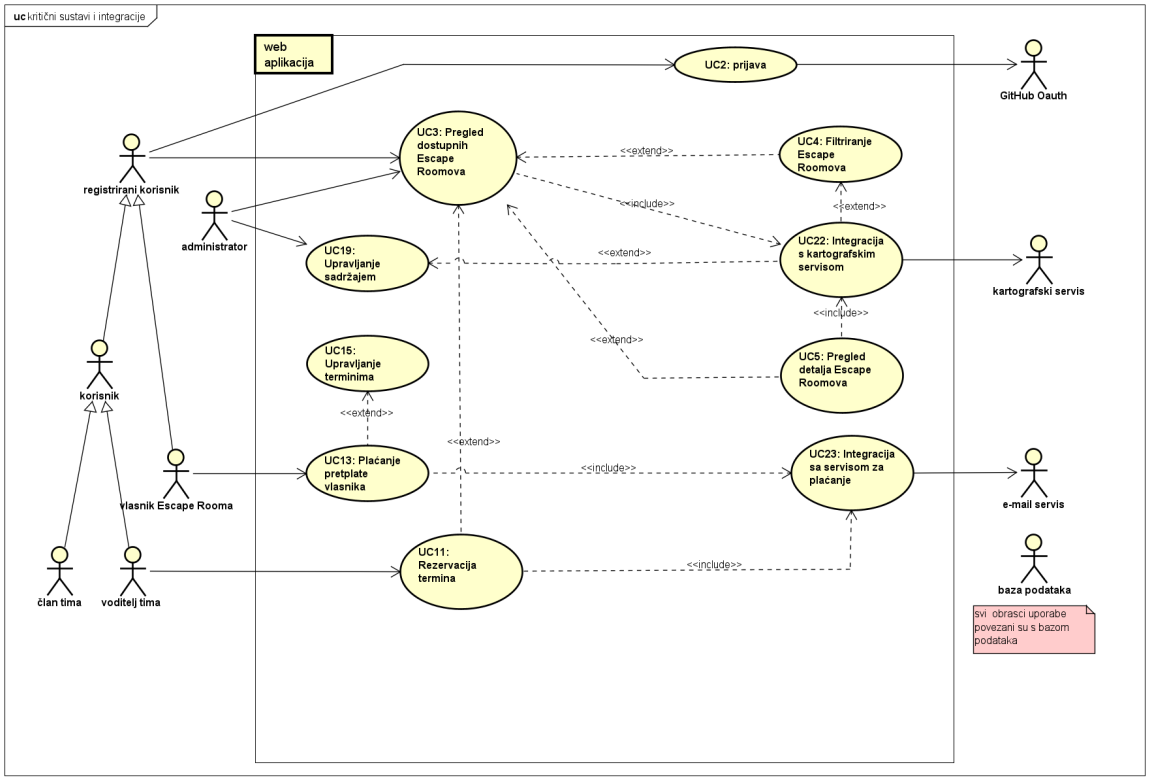
4.2.4 Dijagram obrazaca uporabe za osnovne poslovne procese



Slika 4.7: Dijagram obrazaca uporabe osnovnih poslovnih procesa

Dijagram ilustrira tijek osnovnih poslovnih procesa: od registracije i kreiranja tima, preko rezervacije termina i plaćanja, do ocjenjivanja i rangiranja timova.

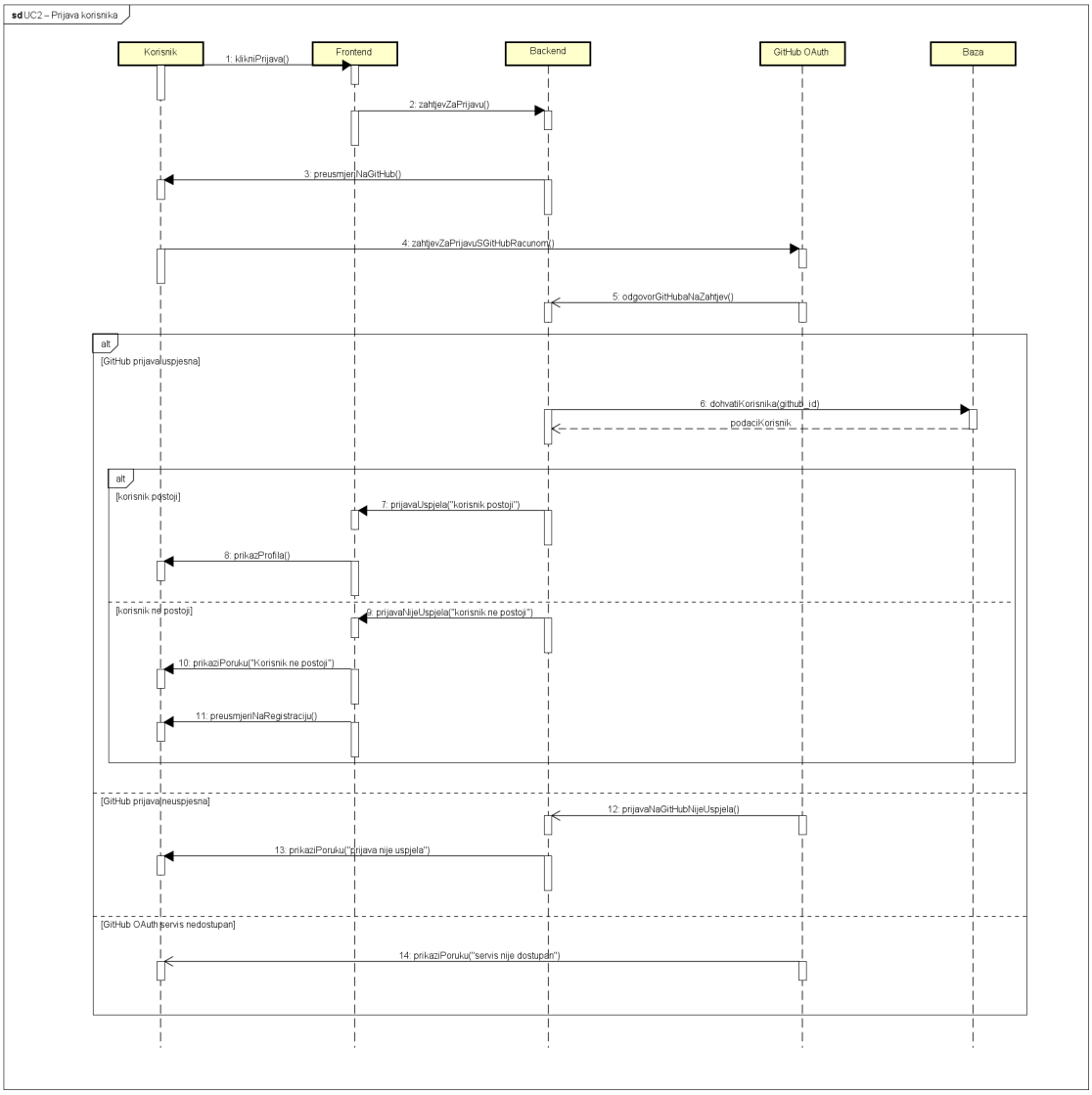
4.2.5 Dijagram obrazaca uporabe za kritične sustave i integracije



Slika 4.8: Dijagram obrazaca uporabe Kritičnih sustava i integracije

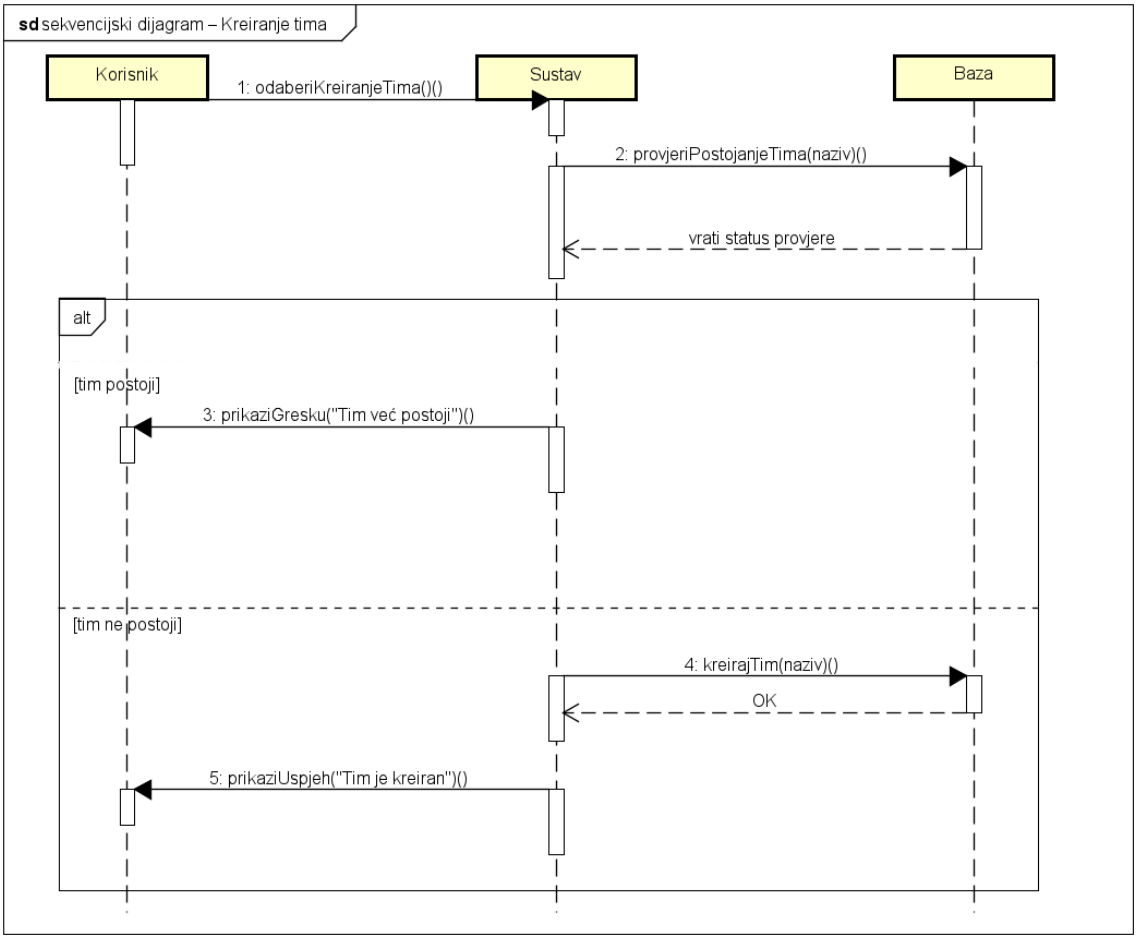
Dijagram prikazuje kritične dijelove sustava uključujući integracije s vanjskim servisima za plaćanje, interaktivnu kartu, e-mail podsjetnike i sigurnosne mehanizme poput OAuth 2.0.

4.3 Sekvencijski dijagrami



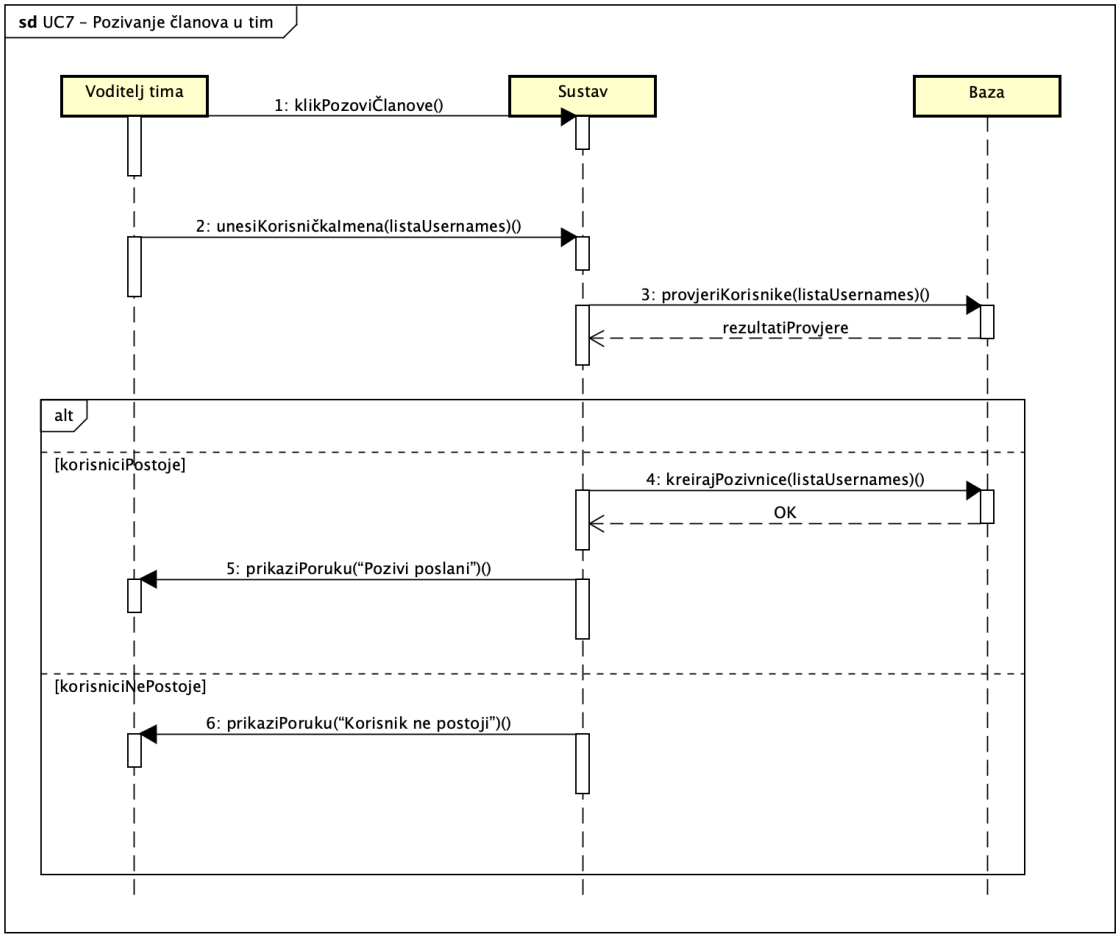
Slika 4.9: Sekvencijski dijagram - Prijava korisnika

Proces započinje kada korisnik odabere opciju prijave, nakon čega frontend šalje zahtjev za prijavu backendu. Backend preusmjerava korisnika na GitHub. Korisnik unosi svoje podatke za prijavu na GitHub servis. Ako GitHub servis nije dostupan prikazuje se poruka, inače GitHub validira prijavu i vraća backendu je li prijava uspjela ili nije i github_id korisnika. Ako prijava nije uspjela, prikazuje se poruka "prijava nije uspjela". Ako prijava uspije, provjerava se postoje li podaci o korisniku u bazi i rezultat provjere određuje daljnji tijek. Ako korisnik postoji u bazi, backend vraća podatke o korisniku frontendu koji ih prikazuje. Ako korisnik ne postoji u bazi, prikazuje se poruka "korisnik ne postoji" te se korisnik preusmjerava na stranicu za registraciju.



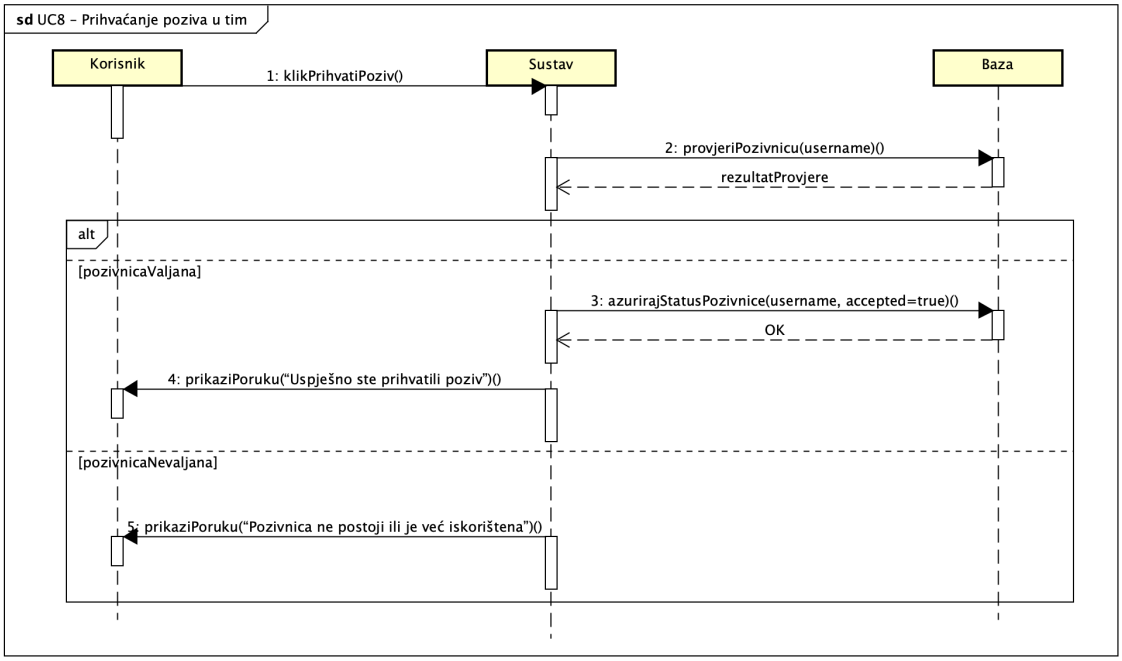
Slika 4.10: Sekvencijski dijagram – Kreiranje tima

Proces započinje kada korisnik odabere opciju kreiranja tima, nakon čega sustav provjerava postoji li već tim s istim nazivom u bazi podataka. Dijagram sadrži alt fragment koji prikazuje dva moguća ishoda provjere: (1) Ako tim već postoji, sustav korisniku prikazuje obavijest da je naziv zauzet te se postupak prekida. (2) Ako tim ne postoji, sustav upisuje novi tim u bazu te nakon uspješnog spremanja korisniku vraća potvrdu.



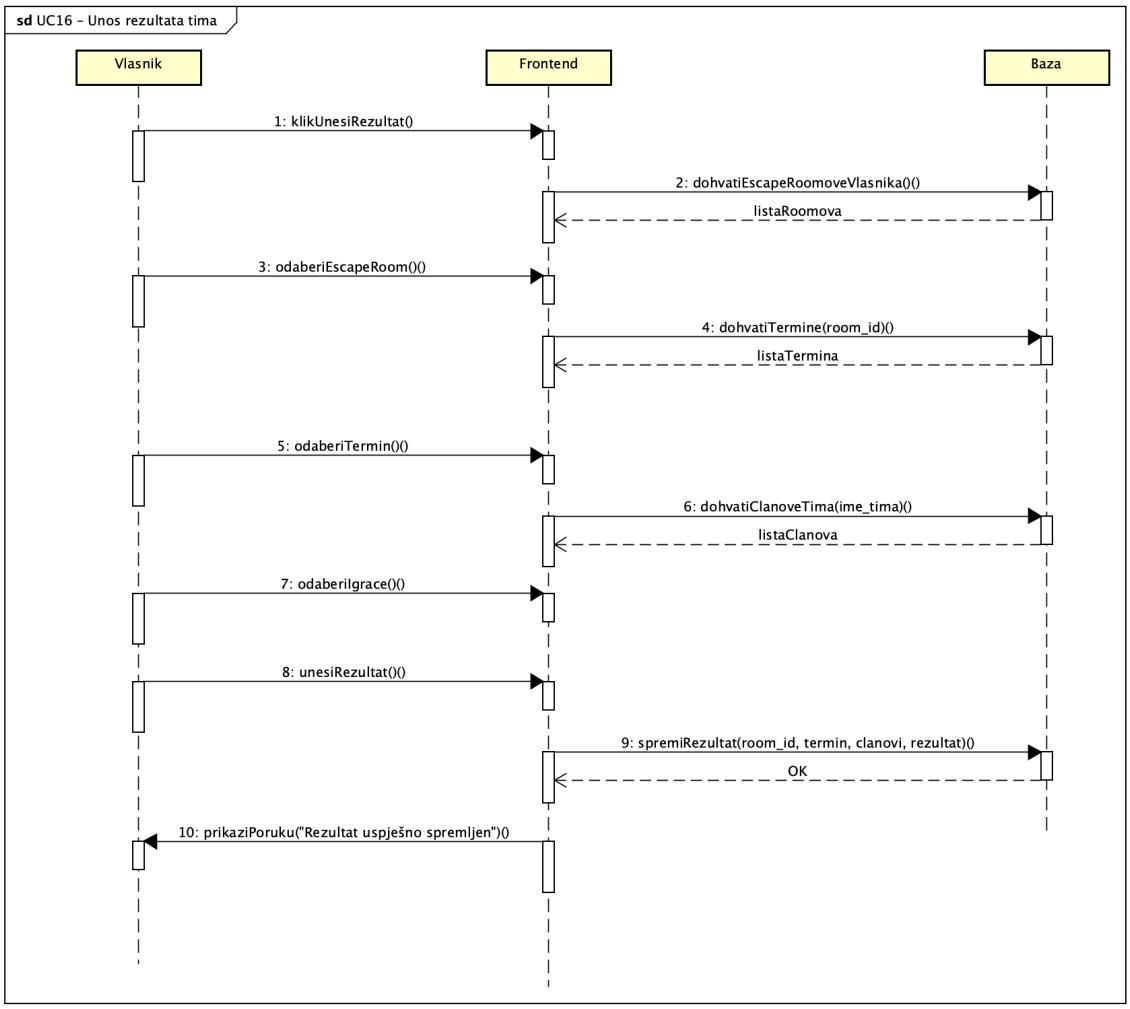
Slika 4.11: Sekvencijski dijagram – Pozivanje članova u tim

Tijek započinje kada voditelj tima odabere opciju za pozivanje članova i unese korisnička imena osoba koje želi pozvati. Sustav zatim dohvaća iz baze informacije o tome postoje li ti korisnici u sustavu. Dijagram sadrži alt fragment s dva scenarija: (1) Ako svi navedeni korisnici postoje, sustav kreira pozivnice i spremi ih u bazu, nakon čega voditelju prikazuje poruku o uspješno poslanim pozivima. (2) Ako jedan ili više korisnika ne postoji, sustav prekida postupak i prikazuje voditelju obavijest.



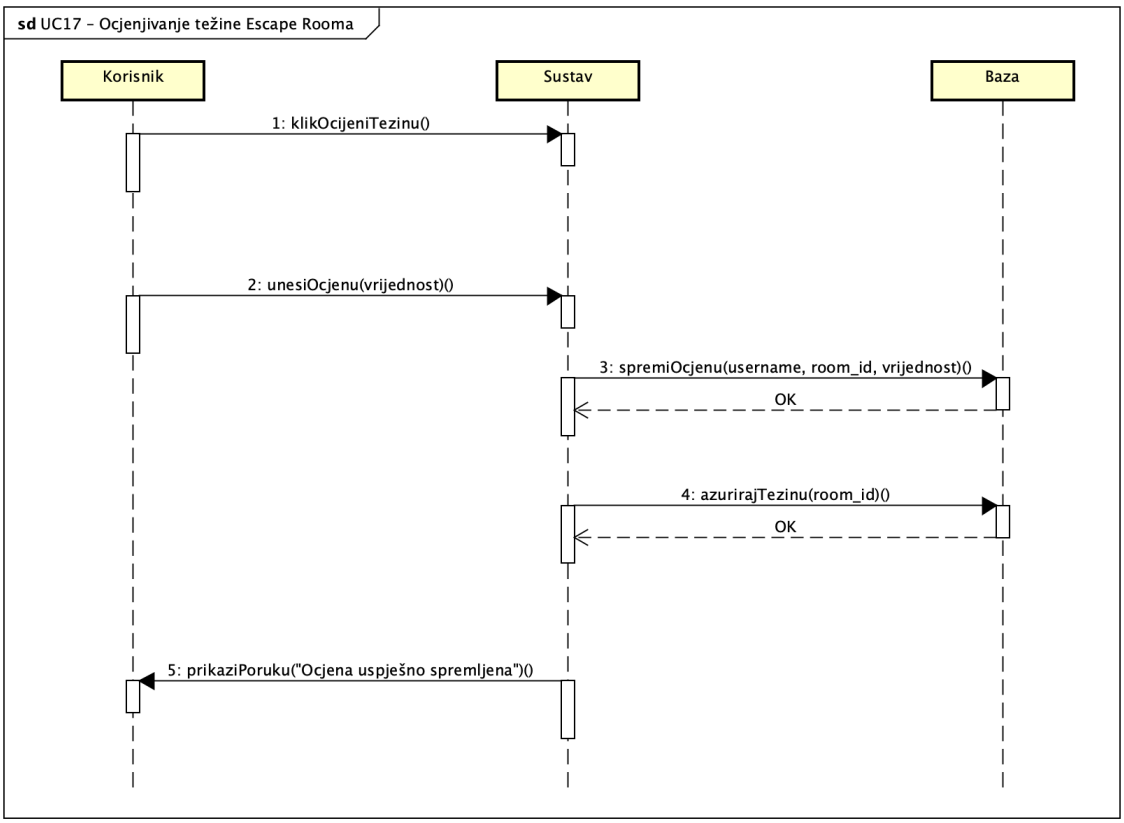
Slika 4.12: Sekvencijski dijagram – Prihvaćanje poziva u tim

Postupak započinje kada korisnik odabere opciju za prihvaćanje poziva. Sustav u bazi provjerava postoji li pozivnica koja odgovara korisniku i je li još uvijek važeća. Dijagram sadrži alt fragment s dva scenarija: (1) Ako je pozivnica valjana, sustav ažurira status u bazi tako da označi da je prihvaćena, nakon čega korisniku prikazuje poruku o uspješnom prihvaćanju poziva. (2) Ako pozivnica nije valjana, npr. ako ne postoji ili je već iskorištena, sustav korisniku prikazuje obavijest da prihvaćanje nije moguće.



Slika 4.13: Sekvencijski dijagram – Unos rezultata tima

Proces započinje kada vlasnik odabere opciju za unos rezultata. Sustav najprije dohvaća popis escape roomova kojima upravlja, a vlasnik zatim odabire željeni prostor. Nakon odabira sustav dohvaća sve termine povezane s tim escape roomom. Vlasnik potom bira termin za koji želi upisati rezultat. Na temelju odabranog termina sustav dohvaća članove pripadajućeg tima te ih prikazuje vlasniku, koji zatim označava igrače i unosi završni rezultat. Kada su svi podaci uneseni, sustav sprema rezultat u bazu podataka i na kraju prikazuje poruku o uspješnom spremanju.



Slika 4.14: Sekvencijski dijagram – Ocjenjivanje težine Escape Rooma

Postupak započinje kada korisnik odabere opciju za ocjenjivanje. Nakon toga korisnik unosi željenu vrijednost ocjene. Sustav sprema ocjenu u bazu podataka. Nakon spremanja pojedinačne ocjene sustav ažurira ukupnu prosječnu težinu Escape Rooma, uzimajući u obzir sve postojeće ocjene. Kada su sve radnje uspješno izvršene, sustav obavještava korisnika prikazom poruke o uspješnom spremanju ocjene.

4.3.1 Ostali zahtjevi

- Sustav mora biti implementiran kao web-aplikacija korištenjem objektno-orijentiranih programskih jezika.
- Pristup aplikaciji mora biti omogućen putem internetskog preglednika iz javne mreže koristeći HTTPS protokol.
- Sustav mora omogućiti istovremeni rad više korisnika u stvarnom vremenu.
- Korisničko sučelje mora biti intuitivno, jednostavno za korištenje i ne smije zahtijevati opširne upute.
- Dizajn korisničkog sučelja mora biti responzivan kako bi aplikacija bila dostupna i na mobilnim uređajima.
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost niti stabilnost sustava.
- Postupak prijave korisnika mora biti siguran.
- Sustav mora podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja.
- Trajanje pristupa bazi podataka ne smije trajati dulje od nekoliko sekundi.
- Baza podataka mora biti odvojena od ostatka sustava te joj se ne smije pristupati izravno, već isključivo putem back-end dijela aplikacije.
- Veza s bazom podataka mora biti sigurna, brza i otporna na vanjske greške.
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava.
- Sustav kao valutu koristi EUR.

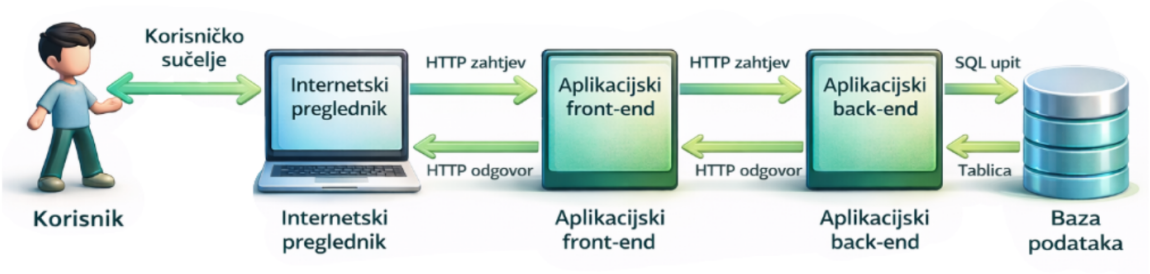
5. Arhitektura i dizajn sustava

5.1 Organizacija sustava

Sustav BreakoutSystems sastoji se od tri glavna dijela:

- baza podataka (SQLite),
- frontend aplikacija (React)
- backend aplikacija (Python).

Frontend omogućuje korisniku interakciju putem web preglednika na računalu ili mobilnom uređaju. Korisnik šalje zahtjeve backendu preko HTTP protokola, backend obrađuje zahtjev, pristupa podacima i vraća odgovor frontendu koji ga prikazuje korisniku.



Slika 5.1: Prikaz arhitekture sustava

5.1.1 Podsustavi i konektori

Arhitektura sustava je klijent-poslužitelj, pri čemu frontend i backend komuniciraju preko REST API-ja. Backend obrađuje poslovnu logiku, validira podatke, pristupa SQLite bazi te povezuje sustav s vanjskim servisima. Azure hosta backend i omogućuje automatski deployment, dok frontend može biti hostan kao statička stranica.

Glavne komponente sustava uključuju:

- Frontend (React) – korisničko sučelje, prikaz podataka, upravljanje sesijama i tokenima
- Backend (Python) – poslovna logika, moduli za autentifikaciju, rezervacije, rangiranje i ocjenjivanje
- Baza podataka (SQLite) – pohrana podataka o sobama, rezervacijama, korisnicima i rezultatima
- Vanjski servisi – GitHub OAuth, Stripe, OpenStreetMap API
- Konektori – REST API, HTTPS, sqlite3, SMTP

5.1.2 Principi oblikovanja

Sustav je dizajniran prema nekoliko ključnih principa:

- Podjela odgovornosti – frontend se bavi prikazom i interakcijom, backend obradom i pohranom podataka
- Modularnost – backend podijeljen u neovisne module olakšava testiranje i održavanje
- Niska povezanost – komponente se mogu nadograditi ili zamijeniti bez utjecaja na ostatak sustava
- Sigurnost – JWT tokeni i HTTPS protokol osiguravaju sigurnu komunikaciju
- Jednostavna infrastruktura – SQLite baza i Azure hosting pojednostavljuju postavljanje i održavanje

5.2 Baza podataka

Koristi se SQLite implementacija SQL baze podataka. SQLite je odabran zbog jednostavnosti implementacije jer se u potpunosti sprema u jednu datoteku kojoj se iz programa pristupa putem odgovarajuće SQLite biblioteke. Ne moramo brinuti o pokretanju servera DBMS-a (npr. SQLite) i upravljanju dozvolama pristupa bazi. Budući da je arhitektura baze SQL, komponente naše baze su klasične tablice/relacije kao u svim SQL sustavima. Svi atributi tipa `TIMESTAMP` označeni su kao `VARCHAR(255)` jer SQLite nema poseban tip podataka `TIMESTAMP`.

5.2.1 Opis tablica

Korisnik

Tablica `Korisnik` pohranjuje osnovne podatke svih korisnika sustava. Primarni ključ je `username`, koji jedinstveno identificira korisnika i povezuje ga s ostalim tablicama. Atribut `oauth_id` omogućuje autentifikaciju preko vanjskih servisa (OAuth), dok `uloga` definira tip korisnika i prava pristupa unutar sustava (POLAZNIK, VLASNIK, ADMIN).

Atribut	Tip podatka	Opis varijable
username (PK)	VARCHAR(255)	Korisničko ime vezano uz račun korisnika
oauth_id	VARCHAR(255)	ID korisnika dobiven od vanjskog servisa za prijavu OAuth protokolom
uloga	VARCHAR(8)	Vrsta korisnika. Može biti POLAZNIK, VLASNIK ili ADMIN

Tablica 5.1: Korisnik

Polaznik

Tablica `Polaznik` sadrži dodatne informacije specifične za korisnike s ulogom POLAZNIK. Povezuje se s tablicom `Korisnik` preko atributa `username`. Pohranjuje e-mail adresu i profilnu sliku korisnika, što omogućuje personalizaciju profila i kontakt s korisnikom.

Atribut	Tip podatka	Opis varijable
username (PK)(FK)	VARCHAR(255)	Korisničko ime vezano uz račun korisnika
email	VARCHAR(255)	E-mail adresa korisnika
profImgUrl	VARCHAR(255)	Ime datoteke profilne slike korisnika

Tablica 5.2: Polaznik

Vlasnik

Tablica `Vlasnik` pohranjuje podatke o korisnicima s ulogom VLASNIK. Sadrži podatke o tvrtki (naziv, adresa, grad, telefon) i logotipu, te datum do kada je vlasnik aktivan (`clanarinaDoDatVr`). Povezuje se s tablicom `Korisnik` preko `username`.

Atribut	Tip podatka	Opis varijable
username (PK)(FK)	VARCHAR(255)	Korisničko ime vezano uz račun korisnika
naziv_tvrtke	VARCHAR(255)	Naziv tvrtke
adresa	VARCHAR(255)	Adresa tvrtke
grad	VARCHAR(255)	Grad ili mjesto tvrtke
telefon	VARCHAR(255)	Telefon tvrtke
logoImgUrl	VARCHAR(255)	Ime datoteke logotipa tvrtke
clanarinaDoDatVr	VARCHAR(255)	Datum i vrijeme do kada korisnik ima plaćenu članarinu

Tablica 5.3: Vlasnik

Tim

Tablica `Tim` definira timove u sustavu. Svaki tim ima jedinstveno ime i opcionalno logotip. Atribut `voditelj_username` povezuje tim s korisnikom koji je voditelj, a primarni ključ je `ime`. Timovi služe za rezervacije Escape Rooma i praćenje članova.

Atribut	Tip podatka	Opis varijable
ime (PK)	VARCHAR(255)	Ime tima, pomoću njega se tim identificira
image_url	VARCHAR(255)	Ime datoteke logotipa tima
voditelj_username (FK)	VARCHAR(255)	Korisničko ime vezano uz račun osnivača/voditelja tima

Tablica 5.4: Tim

Član tima

Tablica `ClanTima` bilježi članstvo korisnika u timovima. Primarni ključ je kombinacija `ime_tima` i `username`. Atribut `accepted` označava je li korisnik prihvatio poziv u tim. Tablica povezuje korisnike i timove te omogućuje praćenje statusa članstva.

Atribut	**Tip podatka**	**Opis varijable**
ime_tima (FK)	VARCHAR(255)	Ime tima
username (FK)	VARCHAR(255)	Korisničko ime vezano uz račun člana tima
accepted	BOOLEAN	Oznaka prihvaćanja poziva u tim
PK = ime_tima, username		

Tablica 5.5: ClanTima

Escape Room

Tablica `EscapeRoom` sadrži osnovne informacije o Escape Roomovima, uključujući naziv, opis, lokaciju, cijenu, veličinu tima i kategoriju. Primarni ključ je `room_id`, a `vlasnik_username` povezuje Escape Room s njegovim vlasnikom. Tablica omogućuje upravljanje ponudom i rezervacijama.

Atribut	**Tip podatka**	**Opis varijable**
room_id (PK)	INT	ID sobe
vlasnik_username (FK)	VARCHAR(255)	Korisničko ime vlasnika
naziv	VARCHAR(255)	Naziv sobe
opis	VARCHAR(255)	Kratki opis sobe
geo_lat	DOUBLE	Zem. širina
geo_long	DOUBLE	Zem. dužina
adresa	VARCHAR(255)	Adresa sobe
grad	VARCHAR(255)	Grad
inicijalna_tezina	DOUBLE	Početna težina
cijena	DOUBLE	Cijena termina
minBrClanTima	INT	Min. broj članova
maxBrClanTima	INT	Maks. broj članova
kategorija	VARCHAR(255)	Tema sobe

Tablica 5.6: EscapeRoom

Escape Room image

Tablica `EscapeRoomImage` sadrži slike povezanih Escape Roomova. Atribut `cover` označava glavnu (naslovnu) sliku, dok `room_id` povezuje sliku s odgovarajućim Escape Roomom. Primarni ključ je `image_url`.

Atribut	Tip podatka	Opis varijable
image_url (PK)	VARCHAR(255)	Ime datoteke slike Escape Rooma
image_index	INT	Indeks slike trenutne sobe
room_id (FK)	INT	Identifikator Escape Rooma na koji se slika odnosi

Tablica 5.7: EscapeRoomImage

Termin

Tablica `Termin` definira termine rezervacija Escape Roomova. Svaki termin je jedinstven kombinacijom `room_id` i `datVrPoc`. Atribut `ime_tima` označava tim koji je rezervirao termin (može biti NULL ako termin nije rezerviran), a `rezultatSekunde` bilježi vrijeme koje je tim proveo u Escape Roomu.

Atribut	Tip podatka	Opis varijable
room_id (FK)	INT	Identifikator Escape Rooma
datVrPoc	VARCHAR(255)	Datum i vrijeme termina
ime_tima (FK)	VARCHAR(255)	Ime tima koji je rezervirao termin ili NULL
rezultatSekunde	INT	Vrijeme koje je bilo potrebno timu da završi Escape Room u sekundama
notified	BOOLEAN	Jesu li poslani e-mail podsjetnici za dani termin
PK = room_id, datVrPoc		

Tablica 5.8: Termin

Ocjena težine

Tablica `OcjenaTezine` pohranjuje ocjene težine Escape Roomova od strane korisnika. Primarni ključ je kombinacija `room_id` i `username`. Omogućuje izračun prosječne težine Escape Rooma i pruža preporuke budućim posjetiteljima.

Atribut	Tip podatka	Opis varijable
room_id (FK)	INT	Identifikator Escape Rooma
username (FK)	VARCHAR(255)	Korisničko ime vezano uz račun korisnika
vrijednost_ocjene	DOUBLE	Vrijednost ocjene težine Escape Rooma
PK = room_id, username		

Tablica 5.9: OcjenaTezine

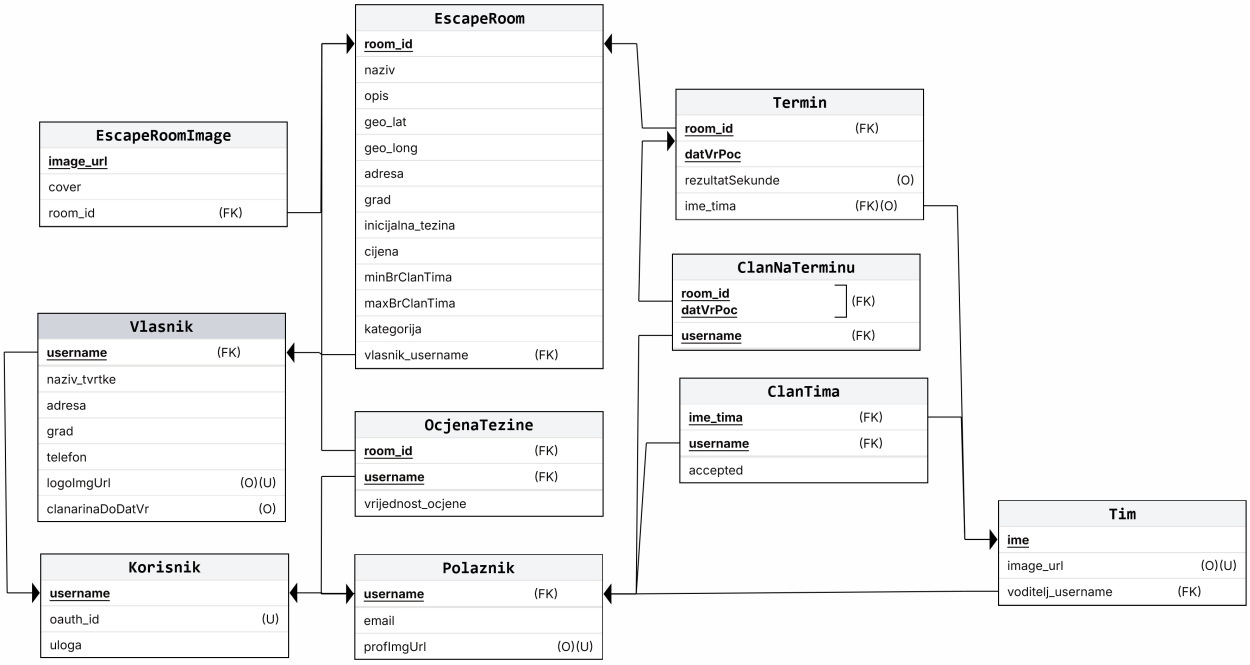
Član na terminu

Tablica `ClanNaTerminu` bilježi članove timova koji su sudjelovali u određenom terminu Escape Rooma. Primarni ključ kombinira `room_id`, `datVrPoc` i `username`. Tablica povezuje korisnike, timove i termine te služi za statistiku i praćenje prisutnosti.

Atribut	Tip podatka	Opis varijable
room_id	INT	Identifikator Escape Rooma
datVrPoc	VARCHAR(255)	Datum i vrijeme termina
username (FK)	VARCHAR(255)	Korisničko ime vezano uz račun korisnika
FK = room_id, datVrPoc		
PK = room_id, datVrPoc, username		

Tablica 5.10: ClanNaTerminu

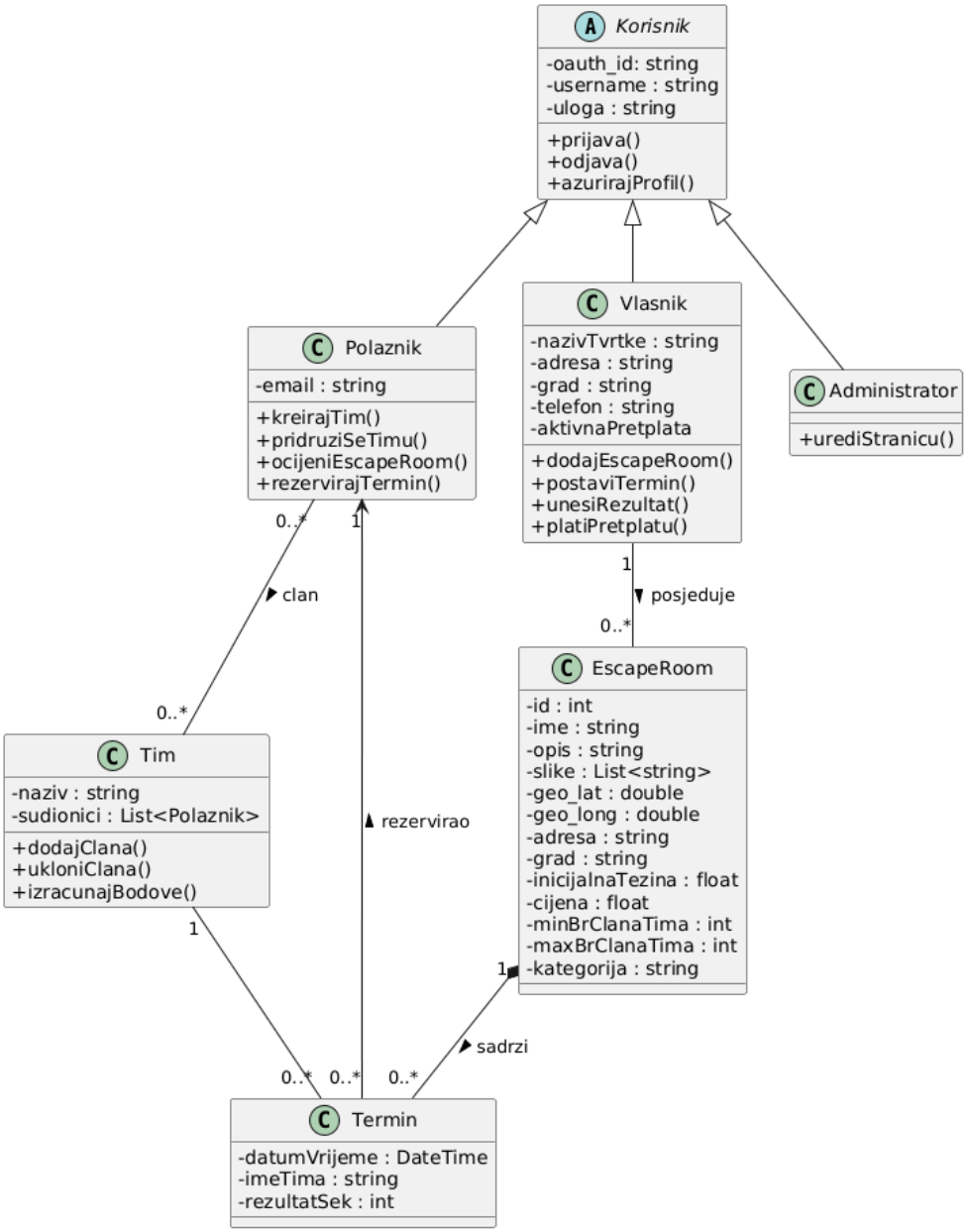
5.2.2 Dijagram baze podataka



Slika 5.2: Relacijski dijagram baze podataka

5.2.3 Dijagram razreda

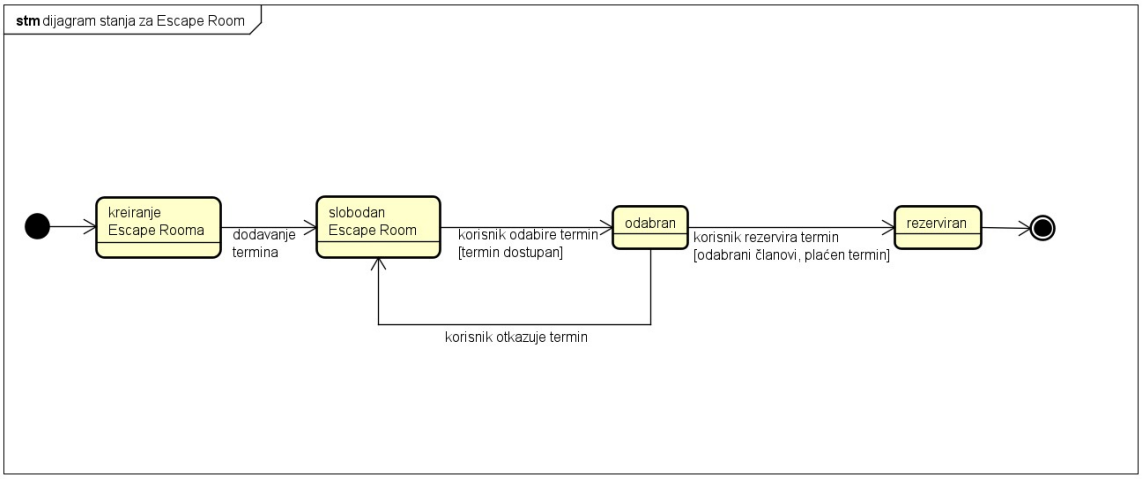
Prikazani dijagram razreda preslikava strukturu baze podataka. Apstraktni razred Korisnik implementiraju razredi Administrator, Polaznik i Vlasnik, koji se međusobno razlikuju po ovlastima. Klasa EscapeRoom sadrži atribute relevantne za Escape Roomove. Klasa Tim omogućuje dodavanje i uklanjanje članova iz tima te pregled trenutačnih članova. Klasa Termin sadrži ključne podatke o terminu.



Slika 5.3: Dijagram razreda

5.3 Dinamičko ponašanje aplikacije

5.3.1 Dijagram stanja



Slika 5.4: Dijagram stanja za Escape Room

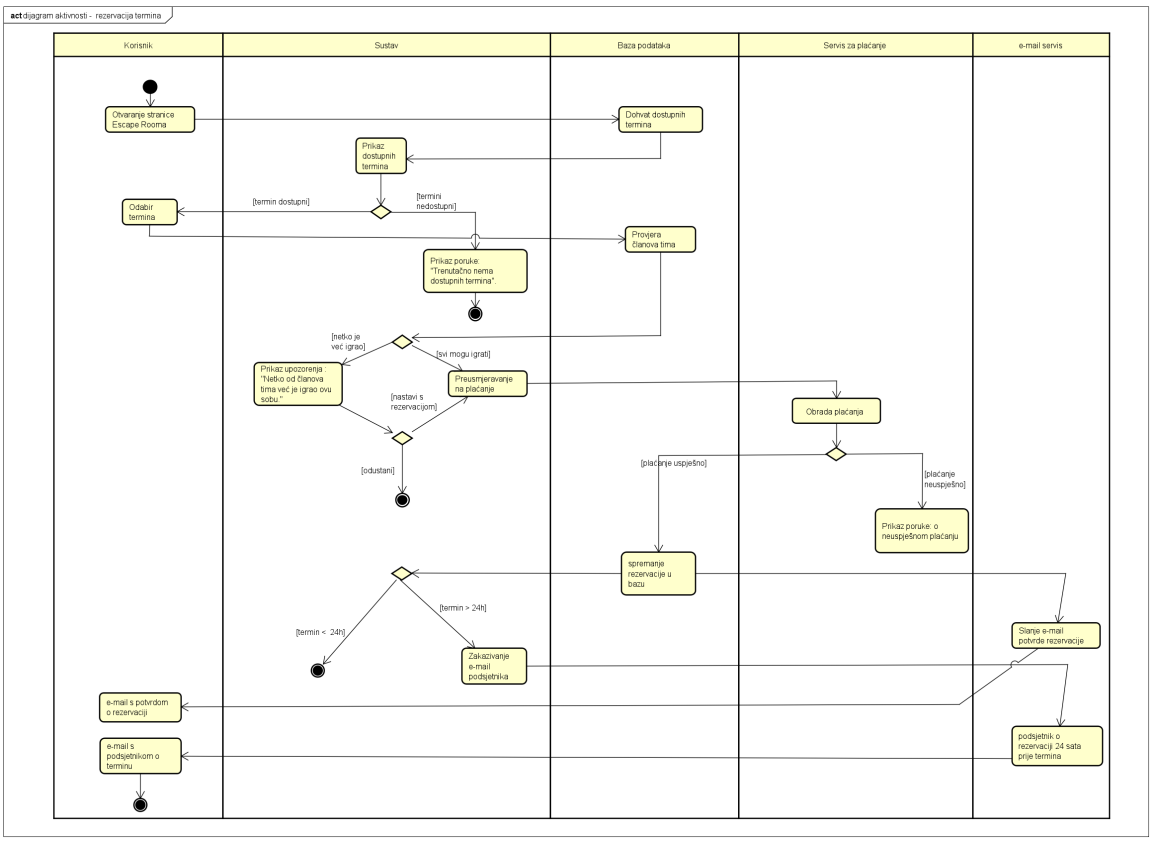
Životni ciklus termina escape rooma prikazan je kroz tri glavna stanja: slobodan, odabran i rezerviran.

Nakon kreiranja termina od strane vlasnika, termin je u početnom stanju slobodan i dostupan za rezervaciju. Prijelaz u stanje odabran aktivira se kada korisnik odabere termin.

Iz stanja odabran moguće su dvije tranzicije:

- Uspješno plaćanje i rezervacija vode u konačno stanje rezerviran (uz uvjete: termin dostupan i plaćen)
- Neuspješno plaćanje vraća termin u slobodno stanje

5.3.2 Dijagram aktivnosti



Slika 5.5: Dijagram aktivnosti za rezervaciju Escape Rooma

Dijagram aktivnosti prikazuje proces rezervacije termina u Escape Roomu, od otvaranja stranice do potvrde rezervacije.

Proces započinje kada korisnik otvori stranicu Escape Rooma. Sustav zatim pokreće dohvat dostupnih termina iz baze podataka te ih prikazuje korisniku. Na temelju dostupnosti termina proces se grana u dva smjera.

Ako nema dostupnih termina, sustav prikazuje poruku "Trenutačno nema dostupnih termina". Ako termini postoje, korisnik može odabrati željeni termin. Tok se nastavlja prema provjeri članova tima u bazi podataka kako bi se utvrdilo je li netko od članova već igrao tu sobu.

Nakon provjere članova tima, proces se ponovno grana. U slučaju da je netko od članova tima već igrao odabranu sobu, sustav prikazuje upozorenje "Netko od članova tima već je igrao ovu sobu." Korisnik tada može odlučiti nastaviti s rezervacijom ili odustati - ako odustane, proces završava. Ako svi članovi mogu igrati (nitko nije igrao tu sobu) ili korisnik odluči nastaviti unatoč upozorenju, sustav preusmjerava

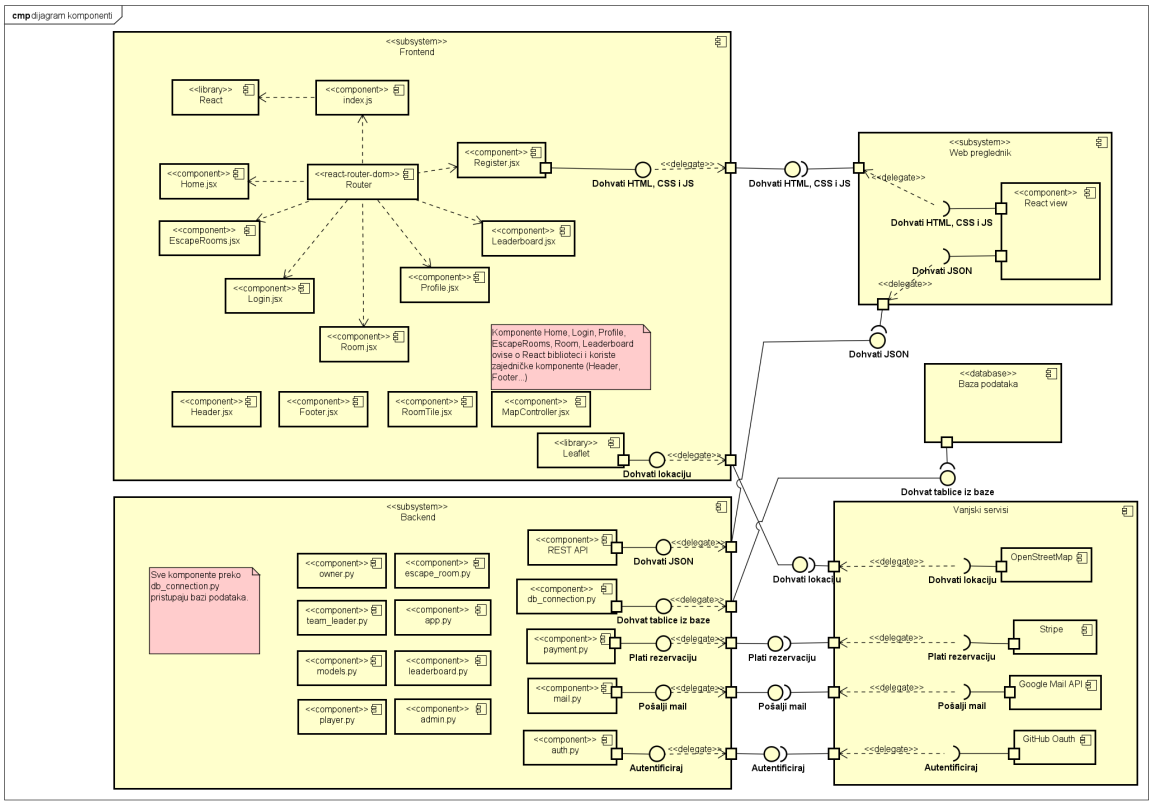
korisnika na plaćanje.

Servis za plaćanje obrađuje transakciju. Ako je plaćanje neuspješno, prikazuje se poruka o neuspješnom plaćanju. Ako je plaćanje uspješno, rezervacija se sprema u bazu podataka, a e-mail servis šalje potvrdu rezervacije korisniku.

Nakon uspješne rezervacije, sustav provjerava vrijeme do termina. Ako je termin udaljen više od 24 sata, zakazuje se slanje e-mail podsjetnika koji će korisnik primiti 24 sata prije termina. Ako je termin unutar 24 sata, podsjetnik se ne zakazuje. U oba slučaja korisnik prima e-mail s potvrdom rezervacije i proces završava.

6. Arhitektura komponenata i razmještaja

6.1 Dijagram komponenata



Slika 6.1: Dijagram komponenata

6.1.1 Frontend podsustav

Frontend podsustav izgrađen je korištenjem React biblioteke. Ulazna točka aplikacije je `index.js` koji pokreće renderiranje React komponenti. Router komponenta iz paketa `react-router-dom` upravlja navigacijom i na temelju URL-a određuje koja će se stranica prikazati korisniku.

Frontend se sastoji od sedam glavnih stranica. `Home.jsx` predstavlja početnu

stranicu s prikazom popularnih Escape Roomova. `Login.jsx` i `Register.jsx` omogućuju prijavu postojećih korisnika odnosno registraciju novih korisnika u sustav. `Profile.jsx` prikazuje korisnički profil s osobnim podacima i poviješću rezervacija. `EscapeRooms.jsx` nudi pregled svih dostupnih Escape Roomova s mogućnošću filtriranja, dok `Room.jsx` prikazuje detalje pojedine sobe s mogućnošću rezervacije termina. `Leaderboard.jsx` prikazuje globalne i lokalne rang liste timova.

Sve navedene komponente ovise o React biblioteci i koriste zajedničke komponente koje osiguravaju konzistentan izgled kroz aplikaciju. `Header.jsx` i `Footer.jsx` koriste se za navigaciju i podnožje stranice, `RoomTile.jsx` služi za prikaz pojedine sobe u listi, a `MapController.jsx` upravlja interaktivnom kartom. Za prikaz lokacija Escape Roomova koristi se Leaflet biblioteka koja dohvaća kartografske podatke putem sučelja od OpenStreetMap servisa.

6.1.2 Backend podsustav

Backend podsustav implementiran je u programskom jeziku Python korištenjem Flask frameworka. Glavna komponenta REST API prima HTTP zahtjeve od frontenda i prosljeđuje ih odgovarajućim modulima za obradu.

Backend se sastoji od komponenti organiziranih prema funkcionalnostima. `app.py` predstavlja glavnu Flask aplikaciju i konfiguraciju sustava. `auth.py` implementira autentifikaciju i autorizaciju korisnika putem GitHub OAuth protokola. `admin.py` sadrži administratorske funkcionalnosti za upravljanje sustavom, dok `owner.py` implementira funkcionalnosti specifične za vlasnike Escape Roomova poput dodavanja soba i upravljanja terminima.

`team_leader.py` sadrži funkcionalnosti za voditelje timova uključujući kreiranje timova i rezervaciju termina, a `player.py` implementira funkcionalnosti za igrače poput pregledavanja povijesti igara i ocjenjivanja soba. `escape_room.py` upravlja podacima o Escape Roomovima, dok `leaderboard.py` implementira izračun i prikaz rang lista timova.

Za integraciju s vanjskim servisima koriste se `payment.py` koji upravlja komunikacijom sa Stripe platnim servisom te `mail.py` koji šalje e-mail obavijesti putem Google Mail API-ja. `models.py` definira strukturu podataka i modele koji se koriste kroz aplikaciju, a `db_connection.py` uspostavlja vezu s bazom podataka. Sve komponente koje trebaju pristup podacima koriste `db_connection.py` koji komunicira s SQLite bazom podataka.

6.1.3 Vanjski servisi

Sustav komunicira s četiri vanjska servisa koji proširuju funkcionalnosti aplikacije.

OpenStreetMap servis koristi se za prikaz interaktivnih karata. Frontend aplikacija putem Leaflet biblioteke dohvaća kartografske podatke koji omogućuju vizualizaciju lokacija Escape Roomova na karti.

Stripe platni servis omogućuje sigurno procesiranje online plaćanja. Backend aplikacija putem `payment.py` modula komunicira sa Stripe API-jem za obradu plaćanja članarina vlasnika i rezervacija termina od strane igrača.

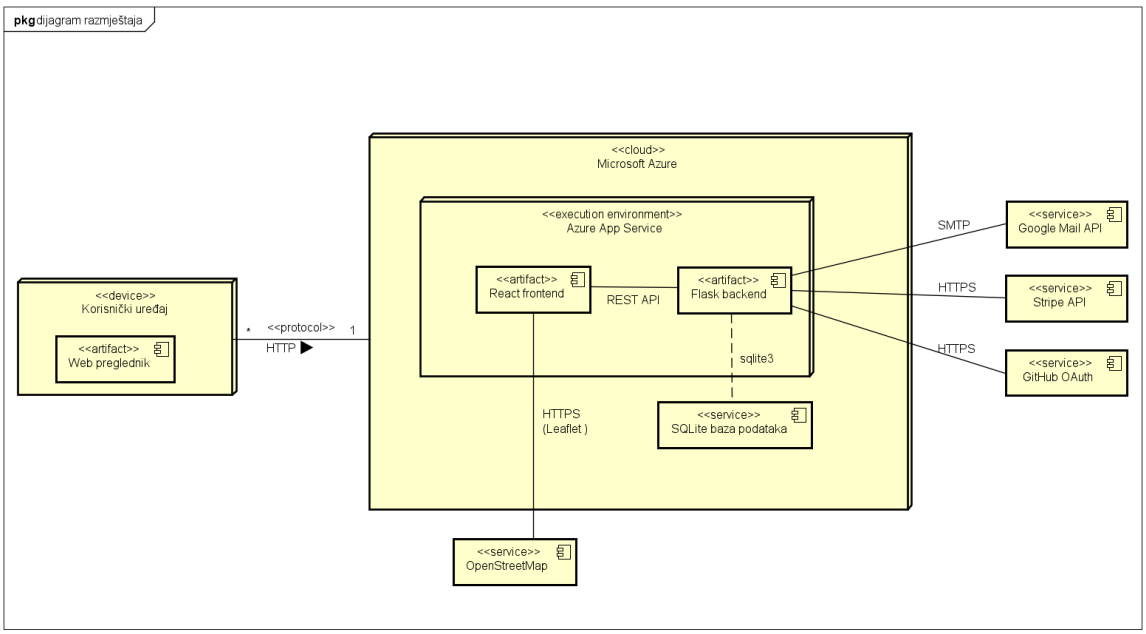
Google Mail API koristi se za slanje e-mail obavijesti korisnicima. Backend putem `mail.py` modula šalje potvrde rezervacija, podsjetnike 24 sata prije termina te druge transakcijske e-mailove.

GitHub OAuth servis omogućuje autentifikaciju korisnika. Umjesto implementacije vlastitog sustava za upravljanje lozinkama, korisnici se prijavljuju putem svojih GitHub računa što pojednostavljuje proces registracije i povećava sigurnost sustava.

6.1.4 Web preglednik

Komponenta React view unutar web preglednika predstavlja klijentsku stranu aplikacije. Preglednik putem HTTP protokola prima statičke datoteke (HTML, CSS i JavaScript) od frontend poslužitelja, a putem REST API sučelja komunicira s backendom za dohvat i slanje dinamičkih podataka u JSON formatu.

6.2 Dijagram razmještaja



Slika 6.2: Dijagram razmještaja

Korisnici sustavu pristupaju putem web preglednika na svojim uređajima, bilo da se radi o osobnom računalu, mobilnom telefonu ili tabletu. Web preglednik izvršava React aplikaciju koja je preuzeta s poslužitelja i prikazuje korisničko sučelje kroz koje korisnik interagira sa sustavom.

6.2.1 Microsoft Azure cloud platforma

Cjelokupna poslužiteljska infrastruktura smještena je na Microsoft Azure cloud platformi unutar Azure App Service execution environmenta. Ovaj pristup omogućuje jednostavno skaliranje, visoku dostupnost i automatski deployment aplikacije.

React Frontend poslužuje statičke datoteke korisnicima. Kada korisnik pristupi aplikaciji, preglednik preuzima HTML, CSS i JavaScript datoteke koje zatim lokalno izvršava i prikazuje korisničko sučelje.

Flask Backend predstavlja Python aplikaciju koja obrađuje poslovnu logiku i API zahtjeve. Backend prima zahtjeve od frontend aplikacije, procesira ih, pristupa bazi podataka i vraća odgovore u JSON formatu.

SQLite baza podataka pohranjena je kao datoteka na istom poslužitelju gdje se izvršava backend aplikacija. Backend pristupa bazi korištenjem sqlite3 biblioteke

koja je ugrađena u Python standardnu biblioteku.

Frontend i backend komuniciraju putem internog REST API sučelja. Svi zahtjevi s frontenda šalju se na definirane API endpointe koje backend obrađuje i vraća odgovarajuće odgovore.

6.2.2 Komunikacija s vanjskim servisima

Backend aplikacija komunicira s vanjskim servisima putem HTTPS protokola koji osigurava enkriptiranu i sigurnu komunikaciju.

GitHub OAuth servis koristi se za autentifikaciju korisnika. Kada korisnik želi se prijaviti, backend preusmjerava korisnika na GitHub stranicu za autorizaciju, nakon čega GitHub vraća autorizacijski token koji backend koristi za dohvat korisničkih podataka.

Stripe API koristi se za sigurno procesiranje online plaćanja. Backend šalje zahtjeve za kreiranje plaćanja, a Stripe obrađuje transakciju i vraća status plaćanja.

Google Mail API koristi se za slanje transakcijskih e-mailova putem SMTP protokola. Backend formira e-mail poruke i šalje ih kroz Google infrastrukturu do krajnjih korisnika.

Frontend aplikacija putem Leaflet biblioteke komunicira s OpenStreetMap API servisom za dohvat kartografskih podataka. Komunikacija se odvija izravno iz korisnikovog preglednika korištenjem HTTPS protokola, bez posredovanja backend aplikacije.

7. Ispitivanje programskog rješenja

7.1 Ispitivanje komponenti

Cilj ispitivanja komponenti je provjera osnovnih funkcionalnosti implementiranih u backend API-ju sustava. Testovi su napisani u **pytest** frameworku za Python i testiraju izolirane komponente sustava. Komponentni testovi obuhvaćaju provjeru autentikacije, pristupa podacima, CRUD operacija, validacije ulaza te pravilnog rukovanja pogreškama.

Korišteni alati: pytest (Python testing framework), Flask test client (simulacija HTTP zahtjeva), SQLite in-memory baza (izolacija testova).

Pokretanje testova: Testovi se pokreću iz direktorija `backend` naredbom:

```
pytest tests_komponente.py -v --tb=short
```

Parametar `-v` omogućuje detaljan ispis rezultata za svaki test, dok `-tb=short` skraćuje prikaz traceback poruka u slučaju pogreške.

7.1.1 Test klase

TestAuth Autentikacija – pristup podacima prijavljenog korisnika, neautorizirani pristup (401), provjera obaveznih polja pri registraciji.

TestRooms Pregled soba – dohvat svih escape roomova i pojedinačno po ID-u, informacije o vlasniku, struktura podataka i izračun težine.

TestRoomFiltering Pretraga i filtriranje – filtriranje po gradu, kategoriji ili kombinaciji, rubni slučajevi.

TestLeaderboard Rang ljestvice – globalna i lokalna rang lista, provjera strukture podataka.

TestMetadata Pomoćni podaci – dohvat gradova i kategorija bez duplikata.

TestAppointments Termini – dohvat slobodnih i zauzetih termina, završeni termini s vremenom.

TestPlayer Korisnik (Polaznik) – pregled timova, povijest igranja, pozivnice, ocjenjivanje soba.

TestTeamLeader Voditelj tima – kreiranje tima, upravljanje članovima, provjera dozvola (403).

TestOwner Vlasnik objekta – pregled i uređivanje soba, unos rezultata, restrikcije.

TestPayment Plaćanja – pretplate (mjesečna/godišnja), rezervacije (polog).

TestStaticAndErrors Sustav i greške – učitavanje React aplikacije, JSON odgovori za 404.

7.2 Rezultati ispitivanja komponenti

```
C:\Users\hanad\Radna površina\faks\5. sem\progi\BreakoutSystems\aj\BreakoutSystems\backend>pytest tests_komponente.py -v --tb=short
============================================================================ test session starts ============================================================================
platform win32 -- Python 3.12.3, pytest-8.3.2, pluggy-1.5.0 -- C:\Users\hanad\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\hanad\Radna površina\faks\5. sem\progi\BreakoutSystems\aj\BreakoutSystems\backend
plugins: anyio-4.8.0
collected 67 items

tests_komponente.py::TestAuth::test_me_unauthorized PASSED                                                                                                                [  1%]
tests_komponente.py::TestAuth::test_me_authorized_polaznik PASSED                                                                                                         [  2%]
tests_komponente.py::TestAuth::test_me_authorized_vlasnik PASSED                                                                                                          [  4%]
tests_komponente.py::TestAuth::test_logout_unauthorized PASSED                                                                                                            [  5%]
tests_komponente.py::TestAuth::test_logout_authorized PASSED                                                                                                              [  7%]
tests_komponente.py::TestAuth::test_edit_unauthorized PASSED                                                                                                              [  8%]
tests_komponente.py::TestAuth::test_register_missing_username PASSED                                                                                                      [ 10%]
tests_komponente.py::TestRooms::test_get_all_rooms_via_filter PASSED                                                                                                      [ 11%]
tests_komponente.py::TestRooms::test_get_room_by_id PASSED                                                                                                                [ 13%]
tests_komponente.py::TestRooms::test_get_room_not_found PASSED                                                                                                            [ 14%]
tests_komponente.py::TestRooms::test_get_room_owner PASSED                                                                                                                [ 16%]
tests_komponente.py::TestRooms::test_get_room_owner_not_found PASSED                                                                                                      [ 17%]
tests_komponente.py::TestRooms::test_room_has_all_fields PASSED                                                                                                           [ 19%]
tests_komponente.py::TestRooms::test_room_tezina_calculation PASSED                                                                                                       [ 20%]
tests_komponente.py::TestRoomFiltering::test_filter_by_city PASSED                                                                                                        [ 22%]
tests_komponente.py::TestRoomFiltering::test_filter_by_category PASSED                                                                                                    [ 23%]
tests_komponente.py::TestRoomFiltering::test_filter_combined PASSED                                                                                                       [ 25%]
tests_komponente.py::TestRoomFiltering::test_filter_no_results PASSED                                                                                                     [ 26%]
tests_komponente.py::TestRoomFiltering::test_filter_no_body_returns_error PASSED                                                                                          [ 28%]
tests_komponente.py::TestLeaderboard::test_global_leaderboard PASSED                                                                                                      [ 29%]
tests_komponente.py::TestLeaderboard::test_room_leaderboard PASSED                                                                                                        [ 31%]
tests_komponente.py::TestLeaderboard::test_leaderboard_structure PASSED                                                                                                   [ 32%]
tests_komponente.py::TestLeaderboard::test_nonexistent_room_leaderboard PASSED                                                                                            [ 34%]
tests_komponente.py::TestMetadata::test_get_categories PASSED                                                                                                             [ 35%]
tests_komponente.py::TestMetadata::test_get_cities PASSED                                                                                                                 [ 37%]
tests_komponente.py::TestMetadata::test_categories_unique PASSED                                                                                                          [ 38%]
tests_komponente.py::TestMetadata::test_cities_unique PASSED                                                                                                              [ 40%]
tests_komponente.py::TestAppointments::test_get_appointments PASSED                                                                                                       [ 41%]
tests_komponente.py::TestAppointments::test_get_appointments_missing_room_id PASSED                                                                                       [ 43%]
tests_komponente.py::TestAppointments::test_appointments_structure PASSED                                                                                                 [ 44%]
tests_komponente.py::TestAppointments::test_appointments_for_room_with_bookings PASSED                                                                                    [ 46%]
tests_komponente.py::TestAppointments::test_appointments_for_empty_room PASSED                                                                                            [ 47%]
tests_komponente.py::TestAppointments::test_appointments_with_results PASSED                                                                                              [ 49%]
tests_komponente.py::TestPlayer::test_my_teams_unauthorized PASSED                                                                                                        [ 50%]
```

Slika 7.1: Prikaz pokretanja pytest testova

```
tests_komponente.py::TestStaticAndErrors::test_serve_react_index PASSED                                                                                                   [ 94%]
tests_komponente.py::TestStaticAndErrors::test_api_not_found PASSED                                                                                                       [ 95%]
tests_komponente.py::TestStaticAndErrors::test_api_returns_json_error PASSED                                                                                              [ 97%]
tests_komponente.py::TestStaticAndErrors::test_instance_images_route PASSED                                                                                               [ 98%]
tests_komponente.py::TestStaticAndErrors::test_invalid_json_body PASSED                                                                                                   [100%]

============================================================================= 67 passed in 14.65s ==============================================================================

C:\Users\hanad\Radna površina\faks\5. sem\progi\BreakoutSystems\aj\BreakoutSystems\backend>
```

Slika 7.2: Prikaz rezultata pytest testova

7.3 Ispitivanje sustava

Cilj ispitivanja sustava je testiranje ponašanja cijelog sustava u uvjetima stvarnog korištenja. Ispitivanje je provedeno korištenjem **Selenium WebDriver** frameworka koji simulira korisničke akcije u web pregledniku. Provedeno je 28 ispitnih slučajeva organiziranih u 6 testnih skupina: testovi navigacije (5), testovi header komponente (5), testovi prijave (3), testovi registracije (5), testovi Escape Rooms stranice (6) i testovi Leaderboard stranice (4).

Korišteni alati: Selenium WebDriver, Chrome WebDriver, Python unittest.

Pokretanje testova: Testovi se pokreću iz direktorija `backend` naredbom:

```
python selenium_tests.py
```

Prije pokretanja potrebno je osigurati da je aplikacija aktivna na `http://localhost:5000` te da je Chrome WebDriver instaliran i dostupan u sistemskoj putanji.

7.3.1 Ispitni slučajevi

IS1 – Učitavanje početne stranice
Ulaz: Otvaranje `http://localhost:5000`
Očekivani rezultat: Stranica se uspješno učitava i prikazuje se header komponenta.
Rezultat: Prolazi ako se header element pronađe na stranici.

IS2 – Učitavanje stranice Escape Rooms
Ulaz: Otvaranje `http://localhost:5000/escape-rooms`
Očekivani rezultat: Stranica se učitava s `escape-rooms-page` wrapperom i sekcijom s naslovom „Istraži".
Rezultat: Prolazi ako se pronađe sekcija s odgovarajućim naslovom.

IS3 – Učitavanje stranice Leaderboard
Ulaz: Otvaranje `http://localhost:5000/leaderboard`
Očekivani rezultat: Stranica se učitava s `leaderboard-page` wrapperom i sekcijom s naslovom „Leaderboard".
Rezultat: Prolazi ako se pronađe sekcija.

IS4 – Učitavanje stranice za prijavu
Ulaz: Otvaranje `http://localhost:5000/login`
Očekivani rezultat: Stranica se učitava s `login-container` elementom.
Rezultat: Prolazi ako se pronađe login container.

IS5 – Učitavanje stranice za registraciju
Ulaz: Otvaranje `http://localhost:5000/register`

Očekivani rezultat: Prikazuje se register forma.
Rezultat: Prolazi ako se pronađe register forma.

IS6 – Provjera elemenata headera
Ulaz: Otvaranje početne stranice
Očekivani rezultat: Header sadrži logo i najmanje 3 navigacijska linka.
Rezultat: Prolazi ako je uvjet ispunjen.

IS7 – Navigacija putem loga
Ulaz: Otvaranje `/escape-rooms`, zatim klik na logo
Očekivani rezultat: Preusmjeravanje na početnu stranicu.
Rezultat: Prolazi ako URL odgovara početnoj stranici.

IS8 – Navigacija na Escape Rooms preko headera
Ulaz: Klik na link „Escape Rooms" u headeru
Očekivani rezultat: Preusmjeravanje na `/escape-rooms`.
Rezultat: Prolazi ako se stranica učita.

IS9 – Navigacija na Leaderboard preko headera
Ulaz: Klik na link „Leaderboard" u headeru
Očekivani rezultat: Preusmjeravanje na `/leaderboard`.
Rezultat: Prolazi ako se stranica učita.

IS10 – Provjera ikone profila
Ulaz: Otvaranje početne stranice
Očekivani rezultat: Postoji element `header-profile`.
Rezultat: Prolazi ako se pronađe element.

IS11 – Provjera elemenata login stranice
Ulaz: Otvaranje `/login`
Očekivani rezultat: Postoji naslov „Prijavi se" i GitHub link „PRIJAVA".
Rezultat: Prolazi ako su elementi prisutni.

IS12 – Link na registraciju iz login stranice
Ulaz: Klik na link za registraciju na login stranici
Očekivani rezultat: Preusmjeravanje na `/register`.
Rezultat: Prolazi ako URL sadrži `/register`.

IS13 – Zaštita profila bez prijave
Ulaz: Pokušaj pristupa `/profile` bez prijave
Očekivani rezultat: Preusmjeravanje na login stranicu ili prikaz loadera.
Rezultat: Prolazi ako pristup nije dopušten.

IS14 – Provjera register forme
Ulaz: Otvaranje `/register`
Očekivani rezultat: Postoji polje `username`.
Rezultat: Prolazi ako se pronađe polje.

IS15 – Provjera radio gumba za uloge
Ulaz: Otvaranje `/register`
Očekivani rezultat: Postoje radio gumbi „POLAZNIK" i „VLASNIK".
Rezultat: Prolazi ako su oba pronađena.

IS16 – Odabir uloge Polaznik
Ulaz: Klik na radio gumb „POLAZNIK"
Očekivani rezultat: Prikazuje se email polje.
Rezultat: Prolazi ako je polje vidljivo.

IS17 – Odabir uloge Vlasnik
Ulaz: Klik na radio gumb „VLASNIK"
Očekivani rezultat: Prikazuju se dodatna polja (tvrtka, adresa, grad, telefon).
Rezultat: Prolazi ako su polja vidljiva.

IS18 – Validacija prazne register forme
Ulaz: Submit forme bez popunjavanja polja
Očekivani rezultat: Korisnik ostaje na `/register`.
Rezultat: Prolazi ako forma ne prolazi validaciju.

IS19 – Sekcija Escape Rooms
Ulaz: Otvaranje `/escape-rooms`
Očekivani rezultat: Sekcija s naslovom „Istraži".
Rezultat: Prolazi ako postoji.

IS20 – Forma za filtriranje
Ulaz: Otvaranje `/escape-rooms`
Očekivani rezultat: Postoje select elementi za grad i kategoriju.
Rezultat: Prolazi ako su pronađeni.

IS21 – Gumb PRETRAŽI
Ulaz: Otvaranje `/escape-rooms`
Očekivani rezultat: Postoji gumb „PRETRAŽI".
Rezultat: Prolazi ako se pronađe.

IS22 – Prikaz mape
Ulaz: Otvaranje `/escape-rooms`
Očekivani rezultat: Prikazana je mapa escape roomova.

Rezultat: Prolazi ako se pronađu map elementi.

IS23 – Sekcija kartica soba
Ulaz: Otvaranje `/escape-rooms`
Očekivani rezultat: Postoji sekcija kartica soba ili prazno stanje.
Rezultat: Prolazi ako se pronađe barem jedan element.

IS24 – Prikaz kartica soba
Ulaz: Otvaranje `/escape-rooms`
Očekivani rezultat: Kartice imaju naslov i gumb „DETALJI" (ako sobe postoje).
Rezultat: Prolazi ako je struktura ispravna.

IS25 – Sekcija Leaderboard
Ulaz: Otvaranje `/leaderboard`
Očekivani rezultat: Sekcija s naslovom „Leaderboard".
Rezultat: Prolazi ako je naslov točan.

IS26 – Tablica rang ljestvice
Ulaz: Otvaranje `/leaderboard`
Očekivani rezultat: Tablica s najmanje 3 zaglavlja.
Rezultat: Prolazi ako tablica postoji.

IS27 – Filter za odabir sobe
Ulaz: Otvaranje `/leaderboard`
Očekivani rezultat: Postoji select za odabir sobe.
Rezultat: Prolazi ako je pronađen.

IS28 – Forma za filtriranje karte
Ulaz: Otvaranje `/escape-rooms`
Očekivani rezultat: Postoji forma za filtriranje karte.
Rezultat: Prolazi ako se pronađe barem jednu formu.

7.4 Rezultati ispitivanja sustava

```
C:\Users\hanad\Radna površina\faks\5. sem\progi\BreakoutSystems\aj\BreakoutSystems\backend>py selenium_tests.py
BreakoutSystems - Selenium E2E Testovi
------------------------------------------------------------

test_01_pocetna_stranica (__main__.TestNavigacija.test_01_pocetna_stranica)
Početna stranica se uspješno učitava ... Početna stranica učitana
ok
test_02_escape_rooms_stranica (__main__.TestNavigacija.test_02_escape_rooms_stranica)
Stranica Escape Rooms se učitava s hero sekcijom ... Escape Rooms stranica učitana
ok
test_03_leaderboard_stranica (__main__.TestNavigacija.test_03_leaderboard_stranica)
Stranica Leaderboard se učitava s tablicom ... Leaderboard stranica učitana
ok
test_04_login_stranica (__main__.TestNavigacija.test_04_login_stranica)
Stranica za prijavu se učitava ... Login stranica učitana
ok
test_05_register_stranica (__main__.TestNavigacija.test_05_register_stranica)
Stranica za registraciju se učitava ... Register stranica učitana
ok
test_01_header_elementi (__main__.TestHeader.test_01_header_elementi)
Header ima logo i navigacijske linkove ... Header ima logo i 4 nav linkova
ok
test_02_logo_vodi_na_pocetnu (__main__.TestHeader.test_02_logo_vodi_na_pocetnu)
Klik na logo vodi na početnu stranicu ... Logo vodi na početnu
ok
test_03_nav_escape_rooms (__main__.TestHeader.test_03_nav_escape_rooms)
Navigacija na Escape Rooms preko headera ... Nav link Escape Rooms radi
ok
test_04_nav_leaderboard (__main__.TestHeader.test_04_nav_leaderboard)
Navigacija na Leaderboard preko headera ... Nav link Leaderboard radi
ok
test_05_profil_ikona (__main__.TestHeader.test_05_profil_ikona)
Profil ikona postoji u headeru ... Profilna ikona postoji
ok
test_01_login_elementi (__main__.TestLogin.test_01_login_elementi)
Login stranica ima sve potrebne elemente ... Login elementi prisutni
ok
test_02_register_link (__main__.TestLogin.test_02_register_link)
Link na registraciju iz login stranice ... Link na registraciju radi
ok
test_03_profil_redirect_bez_prijave (__main__.TestLogin.test_03_profil_redirect_bez_prijave)
Pristup profilu bez prijave redirecta na login ... |
```

Slika 7.3: Prikaz pokretanja Selenium testova

```
Ran 28 tests in 62.497s

OK

------------------------------------------------------------
UKUPNO: 28 testova
USPJEŠNO: 28
NEUSPJEŠNO: 0
GREŠKE: 0
```

Slika 7.4: Prikaz rezultata Selenium testova

7.5 Zaključak

Provedeno je sveobuhvatno ispitivanje sustava BreakoutSystems na razini komponenti i sustava.

Ispitivanje komponenti (pytest): Razvijeno je 67 unit/integration testova koji pokrivaju sve glavne API endpointe. Testirani su redovni slučajevi, rubni uvjeti i pogreške. Svi testovi prolaze (67/67).

Ispitivanje sustava (Selenium): Razvijeno je 28 end-to-end testova koji testiraju stvarnu interakciju korisnika kroz web preglednik. Pokrivene funkcionalnosti: navigacija, registracija, pregled soba, leaderboard, autentikacija. Svi testovi prolaze (28/28).

Ukupni rezultati: 95 testova, 95 uspješnih, 0 neuspješnih, 0 otkrivenih grešaka.

Sustav je uspješno prošao sve definirane ispitne slučajeve. Testovi su dostupni u repozitoriju: `tests_komponente.py` i `selenium_tests.py`.

8. Implementacija i korisničko sučelje

8.1 Tehnologije za implementaciju aplikacije

8.1.1 Korištene tehnologije i alati

Razvoj sustava BreakoutSystems zahtijevao je koordiniran rad tima uz primjenu modernih tehnologija i alata. U nastavku su opisane tehnologije korištene za komunikaciju, dokumentaciju, razvoj i deployment aplikacije.

Komunikacija i suradnja

Komunikacija unutar tima odvijala se putem aplikacije WhatsApp koja je omogućila brzu razmjenu informacija i koordinaciju dnevnih aktivnosti. Za video-sastanke i dijeljenje ekrana korištene su funkcionalnosti aplikacije Microsoft Teams, što je bilo posebno korisno tijekom rada na kompleksnijim dijelovima sustava koji su zahtijevali detaljniju diskusiju.

Za upravljanje izvornim kodom i dokumentacijom korišten je sustav za verzioniranje Git s udaljenim repozitorijem na platformi GitHub. GitHub je omogućio praćenje promjena u kodu, pregled i odobravanje pull requestova te automatizaciju CI/CD procesa.

Dokumentacija i dijagrami

Tehnička dokumentacija pisana je u LaTeX sustavu za pripremu dokumenata koji omogućuje profesionalno formatiranje i jednostavno upravljanje referencama, tablicama i slikama. Za izradu UML dijagrama korišten je alat Astah UML koji omogućuje brzu izradu dijagrama i izvoz u različite formate pogodne za umetanje u LaTeX dokumentaciju. ER dijagrami baze podataka izrađeni su pomoću online alata ERDPlus dostupnog na adresi `https://erdplus.com/`.

Razvojna okruženja

Za pisanje backend koda primarno je korišteno razvojno okruženje Visual Studio Code, uz povremeno korištenje PyCharm Community Edition za naprednije funk-

cionalnosti debugiranja Python koda. Frontend dio aplikacije razvijan je uz pomoć WebStorm okruženja koje nudi naprednu podršku za JavaScript i React ekosustav. Sva tri razvojna okruženja podržana su na Windows, Linux i macOS operacijskim sustavima, što je omogućilo fleksibilnost u radu članova tima.

8.1.2 Backend tehnologije

Backend aplikacija razvijena je u programskom jeziku Python verzije 3.11 koristeći Flask 2.3 mikro-framework. Flask je odabran jer nudi dovoljno funkcionalnosti za izradu REST API-ja bez dodatnog opterećenja koje dolazi s većim frameworkima poput Djanga. Minimalistički pristup Flaska omogućio je brži razvoj i lakše razumijevanje strukture aplikacije, dok su dodatne funkcionalnosti po potrebi dodavane kroz Flask ekstenzije.

8.1.3 Frontend tehnologije

Frontend dio aplikacije izgrađen je korištenjem React biblioteke verzije 19.2 uz JavaScript ES2022 standard. React omogućuje izgradnju modularnih komponenti koje se mogu ponovno koristiti kroz aplikaciju. Virtualni DOM koji React koristi osigurava da se ažuriraju samo one komponente čije se stanje promijenilo, čime se poboljšavaju performanse aplikacije i olakšava održavanje koda. Za prikaz interaktivnih karata korištena je Leaflet biblioteka koja komunicira s OpenStreetMap servisom.

8.1.4 Baza podataka

Za pohranu podataka korištena je SQLite 3.41 relacijska baza podataka. SQLite se čuva u jednoj datoteci i ne zahtijeva instalaciju i konfiguraciju zasebnog database servera, što je značajno pojednostavilo postavljanje razvojnog okruženja i omogućilo brzo pokretanje aplikacije. Ovaj izbor omogućuje jednostavno testiranje bez potrebe za postavljanjem PostgreSQL ili MySQL servera, a u slučaju potrebe za skaliranjem, migracija na robusniji DBMS bila bi relativno jednostavna zahvaljujući korištenju standardnog SQL-a.

8.1.5 Ispitivanje sustava

Za end-to-end testiranje korišten je Selenium WebDriver verzije 4.15 koji pokreće stvarni Chrome preglednik i simulira korisnikove akcije poput klikova, unosa teksta i navigacije kroz aplikaciju. Ovakav pristup omogućuje testiranje aplikacije iz

perspektive krajnjeg korisnika i otkrivanje problema koji se možda ne bi primijetili u izoliranim testovima.

Unit i integracijski testovi pisani su u pytest 7.4 frameworku koji omogućuje jednostavnu sintaksu pisanja testova i nudi brojne dodatke poput pytest-flask za testiranje Flask aplikacija i pytest-cov za mjerenje pokrivenosti koda testovima.

8.1.6 CI/CD i deployment

Kontinuirana integracija i deployment implementirani su pomoću GitHub Actions platforme. Svaki push na main granu repozitorija automatski pokreće niz koraka: najprije se izvršavaju pytest testovi kako bi se provjerila ispravnost backend koda, zatim se pokreće `npm run build` naredba za izgradnju produkcijske verzije frontenda, i na kraju se izvršava deployment na Azure App Service.

Microsoft Azure App Service odabran je kao hosting platforma jer omogućuje automatsko skaliranje aplikacije ovisno o opterećenju, podršku za Python i Node.js runtime okruženja te direktan deployment iz GitHub repozitorija. Integracija s GitHubom značajno je pojednostavila proces deploymenta i omogućila brzo iteriranje kroz verzije aplikacije.

8.1.7 Licence

Sve navedene tehnologije besplatne su za studentsku i nekomercijalnu uporabu. Većina korištenih alata otvorenog je koda, dok cloud servisi poput Microsoft Azurea nude besplatne tierove za manje projekte. Ovo je omogućilo razvoj i hosting aplikacije bez dodatnih troškova tijekom cijelog trajanja projekta.

8.1.8 Ispitivanje programskog rješenja

Ovo poglavlje opisuje provedena ispitivanja implementiranih funkcionalnosti na razini komponenti i sustava. Fokus je na odabiru i izvedbi ispitnih slučajeva koji obuhvaćaju redovne, rubne uvjete i testiranje grešaka.

Ispitivanje komponenti

Cilj ispitivanja komponenti je provjera osnovnih funkcionalnosti implementiranih u backend API-ju sustava. Testovi su napisani u **pytest** frameworku za Python i testiraju izolirane komponente sustava.

Korišteni alati

- **pytest** - Python testing framework
- **Flask test client** - simulacija HTTP zahtjeva
- **SQLite in-memory baza** - izolacija testova

8.2 Upute za puštanje u pogon

8.2.1 Instalacija potrebne programske potpore

Kako bi se aplikacija ispravno pokrenula, potrebno je preuzeti i instalirati sljedeće programske pakete:

- `NodeJS 24` i `npm` za izgradnju frontend aplikacije
- `Python 3.11` za pokretanje backend aplikacije

Nakon instalacije potrebnih programa, treba klonirati repozitorij aplikacije naredbom `git clone` ili ga preuzeti kao ZIP arhivu s GitHub platforme i raspakirati na željenu lokaciju.

```
PS C:\Users\Antonio\Desktop> git clone https://github.com/FER-PROGI2025-LA15-4/BreakoutSystems.git
Cloning into 'BreakoutSystems'...
remote: Enumerating objects: 2379, done.
remote: Counting objects: 100% (204/204), done.
remote: Compressing objects: 100% (74/74), done.
remote: Total 2379 (delta 144), reused 140 (delta 129), pack-reused 2175 (from 2)
Receiving objects: 100% (2379/2379), 85.77 MiB | 3.38 MiB/s, done.
Resolving deltas: 100% (1505/1505), done.
PS C:\Users\Antonio\Desktop>
```

Slika 8.1: Kloniranje repozitorija

8.2.2 Izgradnja frontenda

Za izgradnju frontenda potrebno je u terminalu otići u direktorij `izvorni_kod/frontend`. Tamo je potrebno pokrenuti naredbu `npm install` kako bi se instalirale sve potrebne JavaScript biblioteke definirane u `package.json` datoteci.

```
C:\Users\Antonio\Desktop\BreakoutSystems\frontend>npm install

added 589 packages, and audited 590 packages in 16s

107 packages are looking for funding
  run `npm fund` for details

2 vulnerabilities (1 moderate, 1 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
```

Slika 8.2: Pokretanje naredbe `npm install`

Nakon uspješne instalacije potrebnih biblioteka, frontend se može izgraditi pokretanjem naredbe `npm run build`. Ta naredba pokreće React build proces koji optimizira i kompilira izvorni kod u produkcijsku verziju.

```
C:\Users\Antonio\Desktop\BreakoutSystems\frontend>npm run build

> frontend@0.1.0 build
> webpack --mode production

assets by status 18.5 MiB [big]
  assets by path *.png 15.8 MiB
    asset f4cf911157ad9f942a19.png 2.88 MiB [emitted] [immutable] [from: src/assets/images/fantasy-background.png] [big]
 (auxiliary name: main)
    asset eabcff5c899ffa86df83.png 2.65 MiB [emitted] [immutable] [from: src/assets/images/other-background.png] [big] (
auxiliary name: main)
    asset 7cffbace8f65c930561f.png 2.57 MiB [emitted] [immutable] [from: src/assets/images/family-background.png] [big]
(auxiliary name: main)
    asset 32eb58efc5bd86321ee9.png 2.38 MiB [emitted] [immutable] [from: src/assets/images/pocetna-background2.png] [big
] (auxiliary name: main)
    + 5 assets
  asset bundle.js 1.46 MiB [emitted] [minimized] [big] (name: main) 1 related asset
  asset 0ba081b76521874cf46c.ttf 672 KiB [emitted] [immutable] [from: src/assets/fonts/Montserrat/Montserrat-VariableFon
t_wght.ttf] [big] (auxiliary name: main)
  asset 442ec1f3d1c43dc9a113.jpg 627 KiB [emitted] [immutable] [from: src/assets/images/sf-background.jpg] [big] (auxili
ary name: main)
asset c3d3e30c8d61fca6a9d3.png 196 KiB [emitted] [immutable] [from: src/assets/images/escape_rooms-background.png] (auxi
liary name: main)
asset 167be10bc5e3dd5f0560.png 189 KiB [emitted] [immutable] [from: src/assets/images/logo.png] (auxiliary name: main)
asset favicon.ico 130 KiB [emitted]
```

Slika 8.3: Pokretanje naredbe `npm run build`

Nakon izvođenja build naredbe, statička verzija izgrađenog frontenda nalazi se u direktoriju `izvorni_kod/frontend/dist`. U tom direktoriju nalaze se HTML, CSS i JavaScript datoteke koje backend poslužuje korisnicima.

8.2.3 Pokretanje backenda

Za pokretanje backenda potrebno je najprije napraviti i aktivirati novi Python Virtual Environment kako bi se izolirala okolina i ovisnosti projekta od sistemske

Python instalacije.

Stvaranje virtualnog okruženja izvodi se odlaskom u korijenski direktorij `izvorni_kod` i pokretanjem naredbe:

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

Nakon stvaranja virtualnog okruženja, potrebno ga je aktivirati. Na Windows operacijskom sustavu to se izvodi odlaskom u direktorij `izvorni_kod/.venv/Scripts` i pokretanjem odgovarajuće skripte ovisno o korištenom terminalu: `activate` za Git Bash, `activate.bat` za CMD ili `activate.ps1` za PowerShell. Na Linux i macOS sustavima koristi se naredba `source .venv/bin/activate`.

```
C:\Users\Antonio\Desktop\BreakoutSystems>python -m venv .venv

C:\Users\Antonio\Desktop\BreakoutSystems>cd .venv\Scripts

C:\Users\Antonio\Desktop\BreakoutSystems\.venv\Scripts>activate.bat

(.venv) C:\Users\Antonio\Desktop\BreakoutSystems\.venv\Scripts>
```

Slika 8.4: Aktivacija virtualnog okruženja: ulazak u `.venv/Scripts` i pokretanje `activate` (cmd) ili `activate.ps1` (PowerShell).

Nakon aktiviranja virtualnog okruženja, potrebno je otići u direktorij `izvorni_kod/backend` i instalirati potrebne Python biblioteke pokretanjem naredbe:

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

```
(.venv) C:\Users\Antonio\Desktop\BreakoutSystems\backend>python -m pip install -r requirements.txt
Collecting Flask==3.0.0 (from -r requirements.txt (line 1))
  Using cached flask-3.0.0-py3-none-any.whl.metadata (3.6 kB)
Collecting Flask-SQLAlchemy==3.1.1 (from -r requirements.txt (line 2))
  Using cached flask_sqlalchemy-3.1.1-py3-none-any.whl.metadata (3.4 kB)
Collecting Flask-Login==0.6.3 (from -r requirements.txt (line 3))
  Using cached Flask_Login-0.6.3-py3-none-any.whl.metadata (5.8 kB)
Collecting authlib==1.3.0 (from -r requirements.txt (line 4))
  Using cached Authlib-1.3.0-py2.py3-none-any.whl.metadata (3.8 kB)
Collecting python-dotenv==1.0.0 (from -r requirements.txt (line 5))
  Using cached python_dotenv-1.0.0-py3-none-any.whl.metadata (21 kB)
Collecting requests==2.31.0 (from -r requirements.txt (line 6))
  Using cached requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)
Collecting stripe==14.1.0 (from -r requirements.txt (line 7))
  Using cached stripe-14.1.0-py3-none-any.whl.metadata (18 kB)
Collecting APScheduler==3.11.2 (from -r requirements.txt (line 8))
  Using cached apscheduler-3.11.2-py3-none-any.whl.metadata (6.4 kB)
Collecting icalendar==6.3.2 (from -r requirements.txt (line 9))
  Using cached icalendar-6.3.2-py3-none-any.whl.metadata (9.0 kB)
```

Slika 8.5: Pokretanje naredbe `python -m pip install -r requirements.txt`

Naredba čita popis potrebnih biblioteka iz datoteke `requirements.txt` i instalira ih u aktivirano virtualno okruženje.

8.2.4 Konfiguracija environment varijabli

Prije pokretanja backenda potrebno je definirati environment varijable koje sadrže osjetljive podatke poput API ključeva i tajnih vrijednosti. Potrebne varijable su:

- `SECRET_KEY` – tajni ključ za potpisivanje sesija
- `GITHUB_CLIENT_ID` – ID aplikacije za GitHub OAuth
- `GITHUB_CLIENT_SECRET` – tajni ključ za GitHub OAuth
- `GITHUB_REDIRECT_URI` – URL za preusmjeravanje nakon GitHub autentifikacije
- `STRIPE_SECRET_KEY` – tajni ključ za Stripe API
- `STRIPE_PUBLIC_KEY` – javni ključ za Stripe API
- `GMAIL_PASSWORD` – lozinka za slanje e-mailova putem Gmail servisa

Konfiguracija se izvodi stvaranjem `.env` datoteke u direktoriju `izvorni_kod/backend` i upisivanjem navedenih varijabli u formatu `NAZIV_VARIJABLE=vrijednost`.

8.2.5 Pokretanje aplikacije

Nakon konfiguracije environment varijabli, aplikacija se pokreće iz direktorija `izvorni_kod/backend` naredbom:

```
python app.py
```

```
(.venv) C:\Users\Antonio\Desktop\BreakoutSystems\backend>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.129.6.37:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 446-030-959
```

Slika 8.6: Pokretanje backenda odnosno cijele aplikaciju naredbom `python app.py`

Backend aplikacija pri prvom pokretanju automatski stvara SQLite bazu podataka u direktoriju `izvorni_kod/instance`. Također, backend poslužuje ranije izgrađeni frontend iz `dist` direktorija, tako da je cijela aplikacija dostupna na jednoj adresi.

Nakon uspješnog pokretanja, aplikacija je dostupna u web pregledniku na adresi `http://localhost:5000` ili na adresi koja je ispisana u terminalu.

9. Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije za pretraživanje i rezervaciju Escape Roomova uz mogućnost upravljanja timovima, praćenja rezultata putem ELO ranking sustava te ostavljanja recenzija. Nakon dvanaest tjedana rada u timu, ostvarili smo zadani cilj. Sama provedba projekta bila je kroz dvije faze.

9.1 Prva faza projekta

Prva faza projekta uključivala je okupljanje tima za razvoj aplikacije, dodjelu projektnog zadatka i intenzivan rad na dokumentiranju zahtjeva. Kvalitetna provedba prve faze uvelike je olakšala daljnji rad pri realizaciji osmišljenog sustava.

Izrađeni obrasci i dijagrami, uključujući obrasce uporabe, sekvencijske dijagrame, model baze podataka i dijagram razreda, bili su od velike pomoći podtimovima zaduženima za razvoj backenda i frontenda. Izrada vizualnih prikaza idejnih rješenja uštedjela je značajnu količinu vremena u drugoj fazi kada su članovi tima nailazili na nedoumice oko implementacije pojedinih funkcionalnosti.

9.2 Druga faza projekta

Druga faza projekta, iako nešto kraća od prve, bila je puno intenzivnija po pitanju samostalnog rada članova. Manjak iskustva članova u izradi sličnih implementacijskih rješenja primorao je članove na samostalno učenje odabranih alata i programskih jezika, ponajprije Flaska i Reacta, kako bi ispunili dogovorene ciljeve.

Osim realizacije programskog rješenja, u drugoj fazi bilo je potrebno dokumentirati ostale UML dijagrame, uključujući dijagram stanja, dijagram aktivnosti, dijagram komponenti i dijagram razmještaja. Također je bilo potrebno izraditi popratnu dokumentaciju kako bi budući korisnici mogli lakše koristiti sustav ili vršiti preinake na njemu, što je zahtijevalo značajnu količinu vremena.

9.3 Tehnički izazovi

Tijekom razvoja prepoznali smo nekoliko tehničkih izazova koje smo morali prevladati.

Implementacija OAuth autentifikacije zahtijevala je dodatno proučavanje dokumentacije GitHub OAuth protokola i razumijevanje tijeka razmjene tokena između klijenta, poslužitelja i GitHub servisa.

Sinkronizacija frontend i backend razvoja pokazala se izazovnom jer je paralelni rad dovodio do situacija u kojima je frontend tim čekao na implementaciju određenih API endpointa kako bi mogao nastaviti s razvojem korisničkog sučelja.

Deployment na produkcijsku okolinu također je predstavljao izazov. Konfiguracija Azure platforme i prilagodba SQLite baze zahtijevala je pažljivo postavljanje environment varijabli i razumijevanje specifičnosti cloud okoline.

Svi navedeni izazovi uspješno su riješeni kroz timsku suradnju, konzultacije s dostupnom dokumentacijom i međusobnu pomoć članova tima.

9.4 Stečena znanja

Pri izradi projekta članovi tima stekli su brojna praktična znanja. Naučili smo razvijati REST API-je korištenjem Flask frameworka te izrađivati jednostranične aplikacije u React biblioteci. Stekli smo iskustvo u radu s relacijskim bazama podataka, konkretno SQLite sustavom, te u pisanju automatiziranih testova korištenjem pytest i Selenium alata. Također smo naučili osnove deploymenta aplikacija na cloud platforme i produbili znanje o timskom radu i verzioniranju koda putem Git sustava.

Znanja koja bi bila posebno korisna za brže i kvalitetnije ostvarenje projekta uključuju dublje poznavanje OAuth protokola, prethodno iskustvo s CI/CD pipelineima te poznavanje naprednih React obrazaca poput hooks mehanizma i context API-ja.

9.5 Budući rad

Aplikacija ima prostor za daljnje unapređenje i razvoj novih funkcionalnosti. Implementacija sustava preporuka temeljenog na prethodnim rezervacijama i preferencijama korisnika značajno bi poboljšala korisničko iskustvo. Sustav bi mogao analizirati koje su Escape Roomove korisnik već posjetio, koje težine preferira i koje tematike ga zanimaju te na temelju toga predlagati nove sobe koje bi mu se mogle svidjeti.

9.6 Zaključne misli

Komunikacija među članovima tima odvijala se putem WhatsAppa čime smo postigli informiranost svih članova grupe o napretku projekta i trenutnim zadacima. Sudjelovanje na ovakvom projektu bilo je vrijedno iskustvo svim članovima tima jer smo kroz intenzivnih nekoliko tjedana rada iskusili zajednički rad na istom projektu.

Osjetili smo važnost dobre vremenske organiziranosti i koordiniranosti između članova tima. Zadovoljni smo postignutim rezultatom bez obzira na prostor za usavršavanje izrađene aplikacije, što je dijelom posljedica neiskustva na takvim i sličnim projektima.

10. Popis literature

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. Laganière, *Object-Oriented Software Engineering*, 2nd ed., McGraw-Hill, 2005.
4. I. Maršić, *Software Engineering*, 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
7. Selenium WebDriver, https://www.selenium.dev/
8. LLM modeli kao pomoć pri razvoju softvera i otklanjanju pogrešaka (GitHub Copilot, ChatGPT, Gemini, Claude)

Indeks slika i dijagrama

2.1 Primjer sučelja Bookeo. 9
2.2 Primjer sučelja Resova. 10

4.1 Cjelokupni dijagram obrazaca uporabe 27
4.2 Dijagram obrazaca uporabe ključnih funkcionalnosti sustava 28
4.3 Dijagram obrazaca uporabe korisničkih uloga – vlasnik Escape Rooma . 29
4.4 Dijagram obrazaca uporabe korisničkih uloga – voditelj tima 30
4.5 Dijagram obrazaca uporabe korisničkih uloga – član tima 31
4.6 Dijagram obrazaca uporabe korisničkih uloga – administrator 32
4.7 Dijagram obrazaca uporabe osnovnih poslovnih procesa 33
4.8 Dijagram obrazaca uporabe Kritičnih sustava i integracije 34
4.9 Sekvencijski dijagram - Prijava korisnika 35
4.10 Sekvencijski dijagram – Kreiranje tima 36
4.11 Sekvencijski dijagram – Pozivanje članova u tim 37
4.12 Sekvencijski dijagram – Prihvaćanje poziva u tim 38
4.13 Sekvencijski dijagram – Unos rezultata tima 39
4.14 Sekvencijski dijagram – Ocjenjivanje težine Escape Rooma 40

5.1 Prikaz arhitekture sustava 42
5.2 Relacijski dijagram baze podataka 49
5.3 Dijagram razreda 50
5.4 Dijagram stanja za Escape Room 51
5.5 Dijagram aktivnosti za rezervaciju Escape Rooma 52

6.1 Dijagram komponenata 54
6.2 Dijagram razmještaja 57

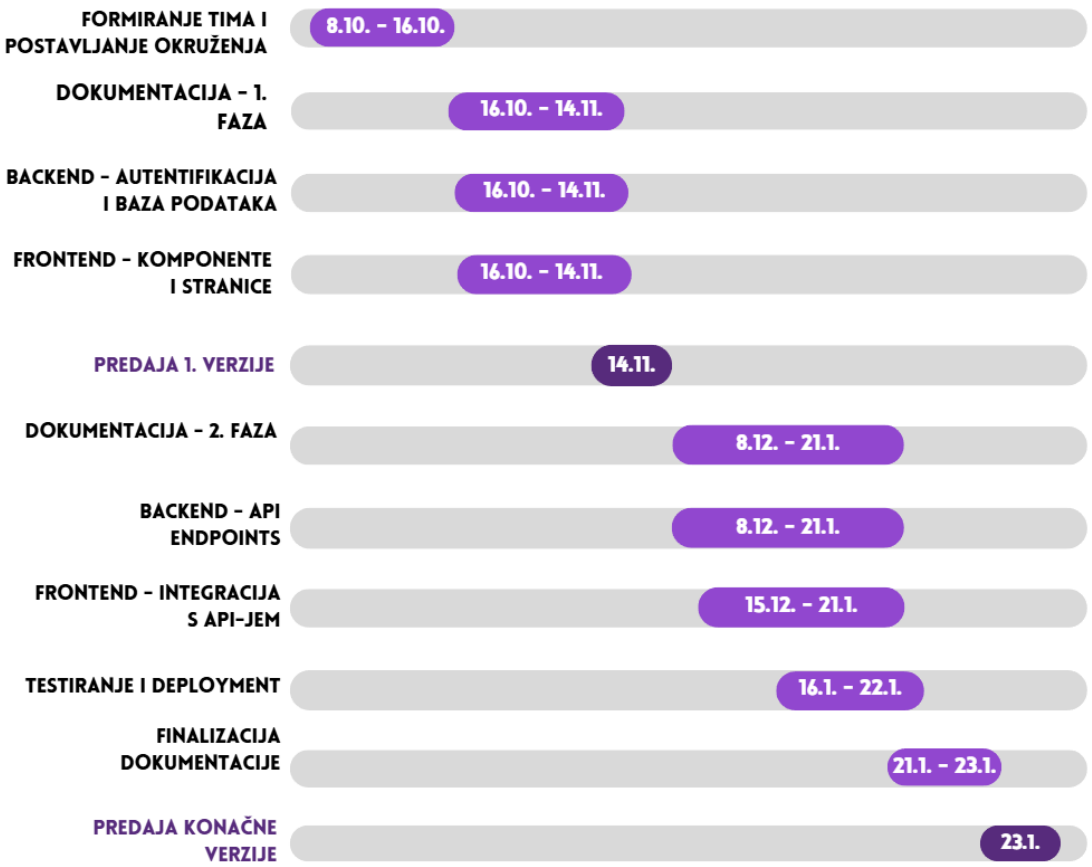
7.1 Prikaz pokretanja pytest testova 60
7.2 Prikaz rezultata pytest testova 60
7.3 Prikaz pokretanja Selenium testova 65
7.4 Prikaz rezultata Selenium testova 65

8.1 Kloniranje repozitorija 71
8.2 Pokretanje naredbe `npm install` 72
8.3 Pokretanje naredbe `npm run build` 72
8.4 Aktivacija virtualnog okruženja: ulazak u `.venv/Scripts` i pokretanje `activate` (cmd) ili `activate.ps1` (PowerShell). 73
8.5 Pokretanje naredbe `python -m pip install -r requirements.txt` . 73
8.6 Pokretanje backenda odnosno cijele aplikaciju naredbom `python app.py` 74

11.1 Gantt prikaz plana rada 82
11.2 Prikaz dijagrama commitova 85
11.3 Prikaz dijagrama commitova po članu tima 85

11. Dodatak: Prikaz aktivnosti grupe

11.1 Plan rada



Slika 11.1: Gantt prikaz plana rada

Dnevnik sastajanja

1. sastanak
   - Datum: 16. listopada 2025.
   - Prisustvovali: H. Dolovčak, A. Mamić, A. Necko, B. Pećnik, L. Stipetić, F. Šušak
   - Teme sastanka:
     - dogovorene okvirne uloge članova tima
     - određene tehnologije koje ćemo koristiti (React, Python/Flask, SQLite)
     - podjela zadataka za prvu fazu dokumentacije
2. sastanak
   - Datum: 28. listopada 2025.
   - Prisustvovali: H. Dolovčak, A. Mamić, A. Necko, B. Pećnik, L. Stipetić, F. Šušak
   - Teme sastanka:
     - pregled napisanog opisa projektnog zadatka i analize zahtjeva
     - dogovor oko obrazaca uporabe i dionika sustava
     - dogovor oko dizajna baze podataka
3. sastanak
   - Datum: 8. studenoga 2025.
   - Prisustvovali: H. Dolovčak, A. Mamić, A. Necko, B. Pećnik, L. Stipetić
   - Teme sastanka:
     - rasprava o arhitekturi sustava i dijagramima
     - dogovor oko ER dijagrama i relacijske sheme
4. sastanak
   - Datum: 13. studenoga 2025.
   - Prisustvovali: H. Dolovčak, A. Mamić, A. Necko, B. Pećnik, L. Stipetić, F. Šušak
   - Teme sastanka:
     - finalizacija dokumentacije za prvu predaju

    - pregled svih dijagrama i ispravak nedostataka
    - provjera konzistentnosti obrazaca uporabe

5. sastanak

    - Datum: 17. prosinca 2025.
    - Prisustvovali: H. Dolovčak, A. Mamić, A. Necko, B. Pećnik, L. Stipetić, F. Šušak
    - Teme sastanka:
        - planiranje implementacije za drugu fazu
        - podjela zadataka za backend i frontend
        - dogovor oko deployanja aplikacije

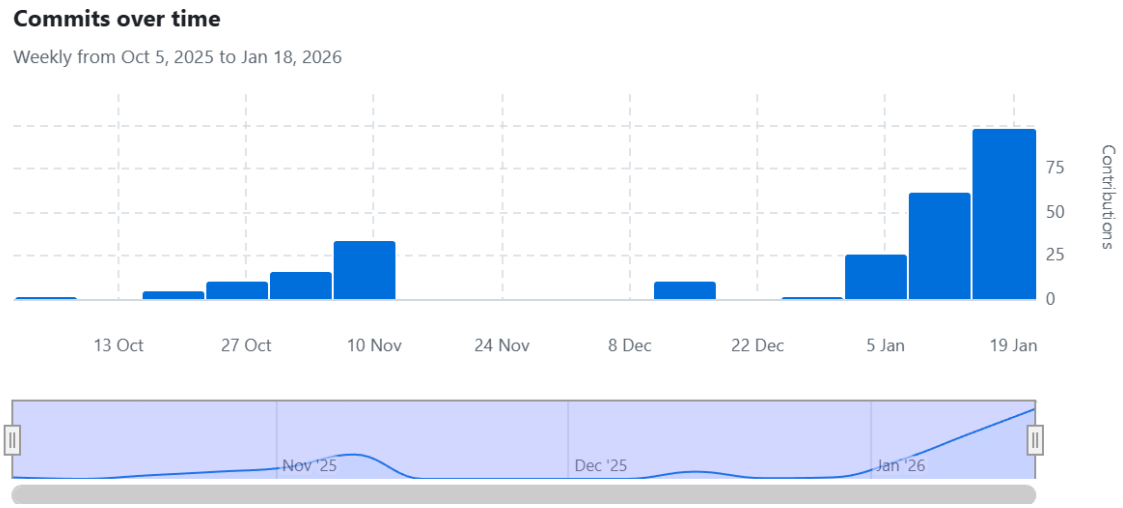
6. sastanak

    - Datum: 15. siječnja 2026.
    - Prisustvovali: H. Dolovčak, A. Mamić, A. Necko, L. Stipetić, F. Šušak
    - Teme sastanka:
        - pregled implementiranog koda i testiranje funkcionalnosti
        - pisanje dokumentacije za ispitivanje programskog rješenja
        - dogovor o izradi dijagrama komponenata i razmještaja

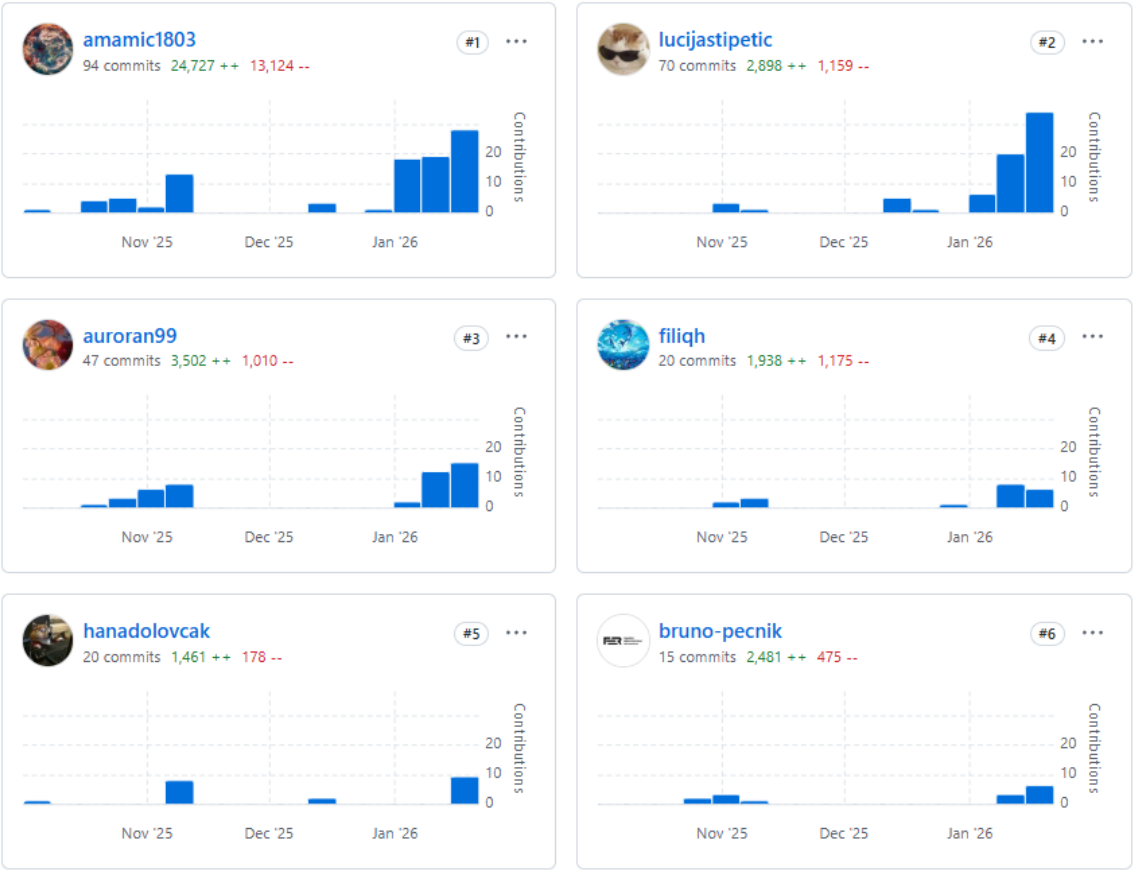
7. sastanak

    - Datum: 21. siječnja 2026.
    - Prisustvovali: H. Dolovčak, A. Mamić, A. Necko, B. Pećnik, L. Stipetić, F. Šušak
    - Teme sastanka:
        - finalizacija dokumentacije za drugu predaju
        - provjera svih sekcija i ispravak grešaka
        - priprema uputa za puštanje u pogon i zaključka

11.1.1 Dijagram pregleda promjena



Slika 11.2: Prikaz dijagrama commitova



Slika 11.3: Prikaz dijagrama commitova po članu tima

Tablica aktivnosti

Tablica 11.1: Pregled aktivnosti po članovima tima

	Hana Dolovčak	**Antonio Mamić**	**Aurora Necko**	**Bruno Pećnik**	**Lucija Stipetić**	**Filip Šušak**
Upravljanje projektom		5				
Opis projektnog zadatka	4					
Funkcionalni zahtjevi		2				
Opis obrazaca uporabe	26					
Dijagram obrazaca uporabe	4					
Sekvencijski dijagrami				9		
Opis ostalih zahtjeva		3				
Arhitektura sustava	8					
Dizajn sustava		3				
Baza podataka (ER, relacijska shema)		1				
Baza podataka - testna baza	3			3		
Dijagram razreda					5	
Dijagram stanja	1			3		
Dijagram aktivnosti	1			4		
Dijagram komponenti	5			5		
Dijagram razmještaja	1			3		
Korištene tehnologije i alati	2					
Ispitivanje programskog rješenja	8					
Upute za puštanje u pogon		2				
Dnevnik sastajanja	1					
Zaključak i budući rad	2					
Popis literature	1					

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Hana Dolovčak	**Antonio Mamić**	**Aurora Necko**	**Bruno Pećnik**	**Lucija Stipetić**	**Filip Šušak**
Izrada PDF dokumenta	25					
Postavljanje Flask projekta						4
Izrada modela i baze podataka		3				
Spajanje s bazom podataka (SQLite)					2	3
Autentifikacija (OAuth)					11	20
Slanje e-mailova (Gmail)					7	
Plaćanje (Stripe)					8	10
API endpointi					34	17
ELO ranking					5	
Organizacija koda					4	
Izrada početne stranice		9	9	9		
Izrada stranice za prijavu/registraciju		23	18			
Izrada stranice za pregled escape roomova		15	13	15		
Izrada leaderboard stranice		1	1	4		
Izrada stranice za rezervacije		20	20			
Izrada korisničkog profila		35	25			
Izrada admin panela		6	4			
Vizualni identitet platforme			21			
Deployment		3				
Izrada prezentacije	2	2	2	2	2	2