

Programsko inženjerstvo

Ak. god. 2020./2021.

Tanky

Dokumentacija, Rev. 2.0

Grupa: ACDC

Voditelj: *Luka Pavlović*

Datum predaje: *14. siječnja. 2021.*

Nastavnik: *Eugen Vušak*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	9
3.1.2 Sekvencijski dijagrami	17
3.2 Ostali zahtjevi	19
4 Arhitektura i dizajn sustava	21
4.1 Baza podataka	22
4.1.1 Opis tablica	22
4.1.2 Dijagram baze podataka	24
4.2 Dijagram razreda	25
4.3 Dijagram stanja	27
4.4 Dijagram aktivnosti	28
4.5 Dijagram komponenti	29
5 Implementacija i korisničko sučelje	31
5.1 Korištene tehnologije i alati	31
5.2 Ispitivanje programskog rješenja	32
5.2.1 Ispitivanje komponenti	32
5.2.2 Ispitivanje sustava	34
5.3 Dijagram razmještaja	35
5.4 Upute za puštanje u pogon	36
6 Zaključak i budući rad	37
Popis literature	38
Indeks slika i dijagrama	39

Dodatak: Prikaz aktivnosti grupe

40

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

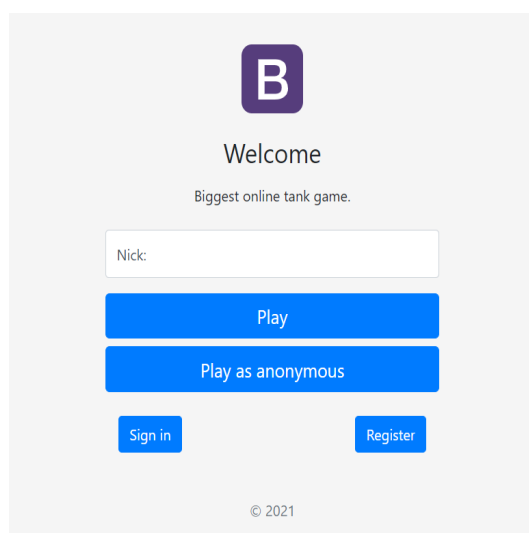
Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Čutura	20.10.2020.
0.11	Dodavanje opisa zadatka	Čutura	20.10.2020.
0.12	Dodavanje poglavlja specifikacija programske potpore	Pavić	20.10.2020.
0.13	Revizija obrazaca uporabe	Pavlović, Pavić	4.11.2020.
0.14	Promjena obrazaca uporabe i izrada dijagrama obrazaca uporabe	Pavić	5.11.2020.
0.15	Razrada opisa zadatka	Mutvar	7.11.2020.
0.16	Promjena obrazaca uporabe i izrada sekvencijskog dijagrama	Pavić	8.11.2020.
0.17	Dodavanje informacija o bazi podataka	Pavlović	8.11.2020.
0.18	Dodavanje ostalih zahtjeva	Kristić	8.11.2020.
0.19	Dodavanje opisa arhitekture	Kristić	8.11.2020.
0.20	Ispravak obrazaca uporabe i dodavanje sekvencijskog dijagrama	Pavić	10.11.2020.
0.21	Izrada dijagrama razreda	Mutvar	12.11.2020.
0.22	Ispravak gramatičkih pogrešaka u dokumentu	Kristić	12.11.2020.
0.23	Ispravak sekvencijskog dijagrama i dopuna opisa zadatka	Pavić	12.11.2020.
0.24	Završna revizija i popravci prije predaje	Pavlović	12.11.2020.
1.0	Verzija dokumentacije spremna za predaju	Pavlović	12.11.2020.
1.1	Izrada dijagrama stanja	Pavić	3.1.2021.
1.2	Ispravak dijagrama stanja i izrada dijagrama aktivnosti	Pavić	4.1.2021.
1.3	Izrada dijagrama razmještaja	Pavić	4.1.2021.
1.4	Ispravak dijagrama stanja	Pavić	13.1.2021.

Rev.	Opis promjene/dodatka	Autori	Datum
1.5	Opis korištenih tehnologija i alata	Pavić, Pavlović, Kristić	13.1.2021.
1.6	Izrada dijagrama komponenti	Pavić, Pavlović	13.1.2021.
2.0	Završetak dokumentacije	Pavić, Pavlović	

2. Opis projektnog zadatka

Cilj projekta je razviti online igricu Tanky. Igra je namijenjena za više igrača te je u potpunosti realizirana u web pregledniku. Svrha igrice je omogućiti zabavu i zanimaciju korisnicima te je namijenjena za svaki uzrast, međutim najviše je prilagođena mlađim te srednje starim korisnicima. Igra simulira rat oklopnih vozila te svaki igrač ima svoje vozilo koje se može kretati po mapi, a ta funkcionalnost izvodi se pomoću tipkovnice. Nadalje, igrači mogu ispaljivati projekte na protivnike kako bi ih eliminirali. Igrač projektil usmjeruje pomoću miša te ga klikom na lijevi gumb ispaljuje. Pogotkom se pogođenom igraču smanjuje život te ukoliko dođe do nule eliminira ga se iz igrice, a igraču koji ga je pogodio se evidentiraju bodovi. Igrači se natječu u rundama ograničenog trajanja i igrač s najviše ostvarenih bodova pobjeđuje.

Ulaskom u igricu korisniku se prikazuje početni izbornik koji je prikazan slikom 2.1. Korisnik se može pridružiti kao registrirani ili kao anonimni korisnik.



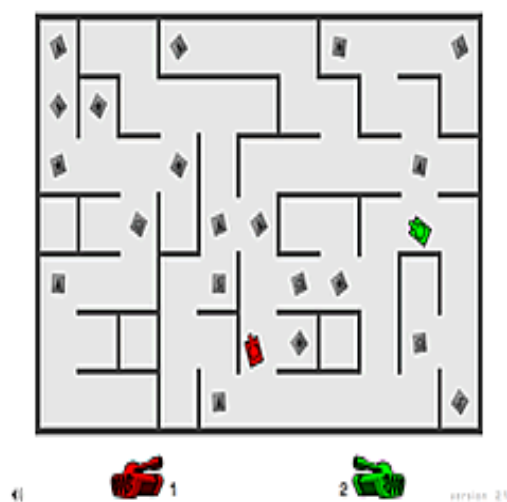
Slika 2.1: Početni izbornik

Prilikom registracije korisnik mora upisati e-mail adresu i lozinku koju će onda koristiti svaki sljedeći put prilikom prijave. Prije pokretanja same igre korisnik upisuje nadimak koji će se koristiti za prikaz rezultata trenutno aktivne runde u gornjem desnom kutu ekrana. Nakon pokretanja korisnik je automatski pozici-

oniran na mapu s ostalim igračima te igra započinje. Na mapama se nalaze razne prepreke i zakloni gdje se igrač može skloniti od protivnika. Također, igrice automatski spaja igrače slične jačine (broja bodova) u iste igre tako da igra bude što zanimljivija i ravnopravnija.

Registriranim korisnicima se prati broj rundi u kojima su sudjelovali te prikazuje statistika pobjeda u izborniku „Moj račun“. Nadalje, registrirani korisnici imaju mogućnost promjene izgleda njihovog oklopnog vozila u tom istom izborniku, klikom na željeni izgled među dostupnima. Izgledi se otključavaju ovisno o postignutim bodovima igrača. Korisnički izbornik nudi i mogućnost promjene zaporke, resetiranja statistike, brisanja računa i pregled statistike drugih registriranih korisnika pretraživanje po nadimku. Unutar same web aplikacije postoji uloga administratora. Administrator ima posebno sučelje unutar kojeg ima niz opcija. Jedna od opcija je određivanje liste mapa koje su trenutno aktivne unutar same igre od ponuđenih. Daljnje opcije su pretraga korisnika po nadimku, prikaz statistike, zabranjivanje korisnicima pristup igri te uklanjanje korisnika. Trenutno aktivna runda u donjem desnom kutu ima prozor za razgovor. Koristeći taj prozor igrači mogu izmjenjivati poruke.

Ideju i motivaciju za igru pronašli smo u sličnim popularnim igrama namijenjenim za više igrača. Jedan od primjera je AZ Tank Game sa slike 2.2, igra s vrlo sličnom tematikom međutim namijenjena samo za igru 1 na 1.

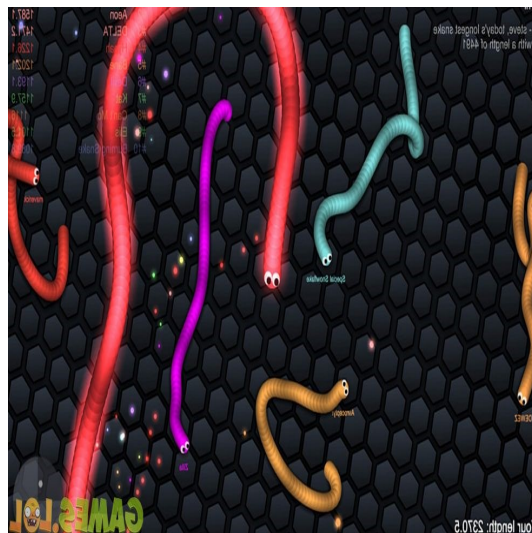


Slika 2.2: AZ Tank Game

Također, igre poput Agar.io i Slither.io koje su tematski različite, ali idejno iste kao Tanky, čija smo rješenja i implementaciju proučavali kako bi dobili ideje za

razvoj naše igrice.

Na slici 2.3 prikazana je igrica Slither.io koja je također namijenjena za više korisnika koji igraju jedan protiv drugog. Međutim za razliku od Tanky-a gdje je cilj pogoditi protivnika projektilom ovdje igrači pokušaju natjerati protivnike da se zabiju u tijelo njihove „zmije“ te ih tako eliminirati iz igre.



Slika 2.3: Slither.io

Problematizacija i opseg zadatka svodi se na razvoj back-end i front-end dijelova aplikacije u skladu s objektno orijentiranom paradigmom te na osmišljavanje i izradu dizajna pojedinih elemenata. Za front-end implementaciju koristili smo HTML, CSS i JavaScript. Za back-end razvoj koristili smo Javascript, SQL te sustav za upravljanje bazama PostgreSQL u kojem smo postavili bazu podataka.

Projekt je sklon brojnim nadogradnjama koje je moguće provesti. Jedna od njih je uvođenje mogućnosti formiranja timova unutar igre. Tim bi se sastojao od minimalno dva igrača. Kreiranje runde bilo bi moguće samo za timove u kojima je isti broj igrača. Također, jedna od nadogradnji je uvođenje lige. U ligu bi mogli ući samo igrači koji su ostvarili potreban kvalifikacijski broj bodova. Liga bi se održavala 3 dana u tjednu. Kvalificirani igrač odigrao bi određen broj rundi i bio plasiran prema svom dostignuću, a isto tako i nagrađen poboljšanjima tenka.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnik
 - (a) Anonimni korisnik
 - (b) Registrirani korisnik
2. Administrator

Aktori i njihovi funkcionalni zahtjevi:

1. anonimni korisnik (inicijator) može:
 - (a) pridružiti se igri kao anoniman korisnik
 - i. sustav mu dodjeljuje nadimak
2. registrirani korisnik (inicijator) može:
 - (a) unijeti nadimak i zaporku
 - (b) pratiti vlastitu statistiku u igri
 - (c) promijeniti izgled vozila u izborniku
 - (d) brisati svoj račun
3. administrator (inicijator) može:
 - (a) uređivati liste mapa aktivnih unutar igre
 - (b) pretraživati korisnika po nadimku i vidjeti prikaz njegove statistike
 - (c) pretraživati najbolje igrače
4. baza podataka (sudionik) može:
 - (a) pohraniti podatke o korisniku

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - registracija

- **Glavni sudionik:** Neregistriran korisnik
- **Cilj:** Stvoriti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb za registraciju
 2. Korisnik unosi potrebne podatke – nadimak i zaporku
 3. Provjerava se ispravnost podataka
 4. Podaci se spremaju u bazu podataka
 5. Korisnik je uspješno registriran
- **Opis mogućih odstupanja:**
 - 2.b Korisnik unosi već korištene podatke
 1. Korisnik prima obavijest da se podaci koriste
 2. Omogućava se ponovni pokušaj korisniku

UC2 - prijava u sustav

- **Glavni sudionik:** Registriran korisnik
- **Cilj:** Korištenje mogućnosti registriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Korisnik odabira gumb "Prijava"
 2. Korisnik upisuje korisničko ime i zaporku
 3. Provjerava se ispravnost podataka
 4. Korisnik je uspješno prijavljen
- **Opis mogućih odstupanja:**
 - 2.b
 1. Korisnik nije registriran
 2. Korisnik je unio krive podatke

UC3 - pregled vlastite statistike

- **Glavni sudionik:** Registrirani korisnik

- **Cilj:** Praćenje broja odigranih rundi i statistika pobjeda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran(UC1)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb „Moj račun“
 2. Korisnik odabire gumb „Prikaži moju statistiku“
 3. Korisniku piše broj odigranih rundi i pobjeda

UC4 - pregled statistike drugih igrača

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Mogućnost pregleda statistike drugih registriranih korisnika pretraživanjem po nadimku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran(UC1)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb „Moj račun“
 2. Korisnik bira polje „Pretraži igrače“
 3. U odabrano polje korisnik upisuje nadimak željenog igrača
 4. Sustav izbacuje statistiku traženog igrača iz baze podataka
- **Opis mogućih odstupanja:**
 - 5.d Traženi korisnik je obrisao profil

UC5 - resetiranje statistike

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Mogućnost resetiranje dosadašnje statistike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran(UC1)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb „Moj Račun“
 2. Korisnik odabire opciju „Resetiraj statistiku“
 3. Sustav nudi korisniku u skočnom prozoru mogućnost da odustane klikom na gumb „Odustani“ ili da resetira statistiku klikom na gumb „Resetiraj“
 4. Korisnikova statistika se u bazi podatak postavi na 0

UC6 - odjava

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen u sustav (UC3)
- **Opis osnovnog tijeka:**
 1. Korisnik zahtjeva odjavu iz sustava odabirom gumba "Odjava" na početnoj stranici
 2. Sustav odjavljuje korisnika
 3. Prikazuje se početna stranica

UC7 - promjena osobnih podataka

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Mogućnost promijene zaporke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb „Moj račun“
 2. Korisnik odabire opcije za promjenu zaporke
 3. Korisnik upisuje novu zaporku
 4. Korisnik sprema promjene
 5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 5.d Korisnik je upisao novu zaporku, ali nije spremio promjene
 1. Sustav obavještava korisnika da nije spremio prije izlaska iz izbornika

UC8 - promjena izgleda vozila

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Mogućnost promjene izgleda vozila među dostupnima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb „Moj Račun“
 2. Korisnik odabire opciju "Odaberi novi izgled"
 3. Korisniku se prikazuje lista otključanih izgleda koje mu dodjeljuje sustav ovisno o broju pobjeda

UC9 - brisanje korisničkog računa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Obrisati korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb „Moj Račun“
 2. Korisnik odabire opciju „Obriši račun“
 3. Korisnički račun se briše iz baze podataka

UC10 - odabir akcije nakon smanjena zdravlja igrača na 0

- **Glavni sudionik:** Korisnik
- **Cilj:** Korisnik može odlučiti hoće li napustiti igru ili će se vratiti nakon što mu je zdravlje smanjeno na 0
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisniku je zdravlje smanjeno na 0 te ga se izbacuje iz trenutno aktivne runde
 2. Korisniku se pojavljuje skočni prozor
 3. Korisnik odabire jednu od ponuđenih opcija
 4. Ako odabere "Napusti" on izlazi iz aktivne igre
 5. Ako odabere "Povratak u igru" on se vraća u aktivnu rundu
- **Opis mogućih odstupanja:**
 - 5.b
 1. Ako je u trenutku povratka u igru igra završila, sustav odbija vratiti korisnika u igru
 2. Ako se korisnik želi vratiti u igru, mora pričekati 30 sekundi od izbacivanja

UC11 - komuniciranje između igrača

- **Glavni sudionik:** Korisnici(Registrirani, anonimni)
- **Cilj:** Mogućnost komunikacije između igrača unutar runde
- **Sudionici:** Korisnici
- **Preduvjet:** -

- **Opis osnovnog tijeka:**

1. Korisnik klikom miša odabire tekstno polje na ekranu
2. Korisnik piše poruku i šalje ostalim igračima

UC12 - ulazak korisnika u rundu

- **Glavni sudionik:** Korisnik (Registrirani, anonimni)

- **Cilj:** Ulazak korisnika u rundu

- **Sudionici:** Baza podataka

- **Preduvjet:** -

- **Opis osnovnog tijeka:**

1. Korisnik odabire gumb "Igraj"
2. Server korisniku dodjeljuje rundu u koju može ući
3. Korisnik ulazi u rundu i igra

- **Opis mogućih odstupanja:**

2.b

1. Korisnik mora čekati da mu server dodjeli rundu za igru jer server raspodjeljuje s obzirom na dosadašnji uspjeh korisnika

UC13 - prijava administratora u sustav

- **Glavni sudionik:** Administrator

- **Cilj:** Korištenje administracijskih ovlasti

- **Sudionici:** Baza podataka

- **Preduvjet:** Postojanje administratora

- **Opis osnovnog tijeka:**

1. Administrator odabira gumb "Prijava administratora"
2. Administrator upisuje ime i zaporku
3. Provjerava se ispravnost podataka
4. Administrator je uspješno prijavljen

- **Opis mogućih odstupanja:**

2.b

1. Ne postoji administrator s upisanim imenom
2. Postoji ime administratora, ali je pogrešna zaporka

UC14 - zabranjivanje pristupa korisniku

- **Glavni sudionik:** Administrator

- **Cilj:** Zabraniti korisnicima pristup igri

- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Administrator pronalazi korisnika kojem želi zabraniti pristup
 2. Administrator bira opciju "Blokiraj korisnika"
 3. Administrator zabrani pristup korisniku trajno ili privremeno

UC15 - dodjela administratorskih ovlasti

- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje novog administratora u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Administrator odabire gumb "Dodaj novog administratora"
 2. Administrator u polje upisuje ime i zaporku
 3. Administrator dodaje novog administratora
- **Opis mogućih odstupanja:**
 - 2.b
 1. U sustavu postoji administrator s istim imenom

UC16 - dodjela liste mape

- **Glavni sudionik:** Administrator
- **Cilj:** Odrediti liste mape koje će biti aktivne unutar igre
- **Sudionici:** Baza podataka
- **Preduvjet:** Nema igara u tijeku
- **Opis osnovnog tijeka:**
 1. Administrator bira opciju "Dodjela liste mapa"
 2. Administrator odabere liste mape koje će biti aktivne unutar same igre
- **Opis mogućih odstupanja:**
 - 2.b
 1. Postoje igre u tijeku te administrator ne može dodijeliti liste mapa
 2. Igrači moraju pričekati da lista mapa bude dodijeljena

UC17 - pretraga najboljih igrača

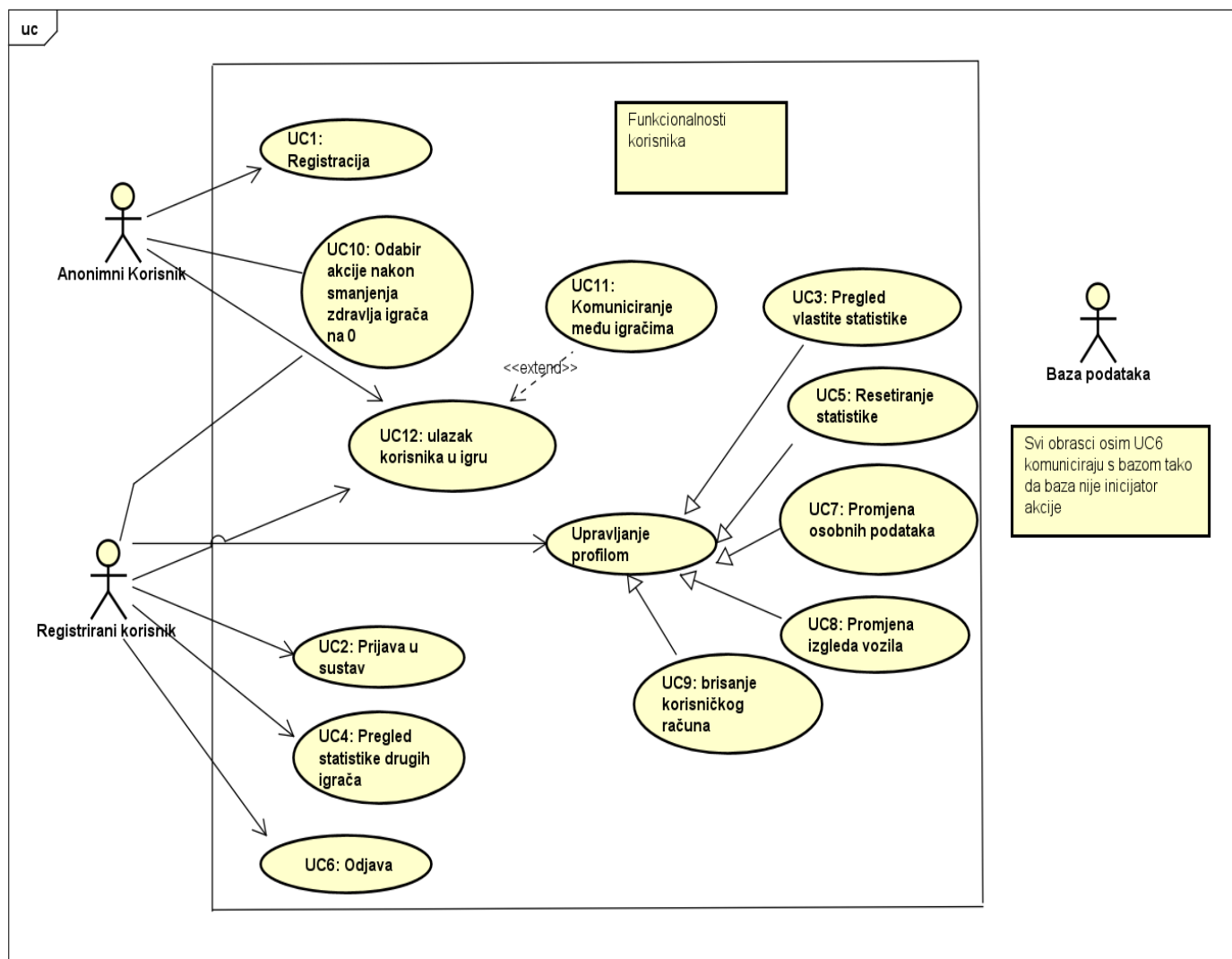
- **Glavni sudionik:** Administrator
- **Cilj:** Pronaći igrače sa najviše ostvarenih bodova

- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Administrator odabire opcije "Pretraga igrača"
 2. Administrator odabire zatim opciju "Pretraži po rezultatu"
 3. Administrator u tekstno polje upisuje broj korisnika koje pretražuje
 4. Administrator odabire korisnika i ima uvid u dosadašnji uspjeh

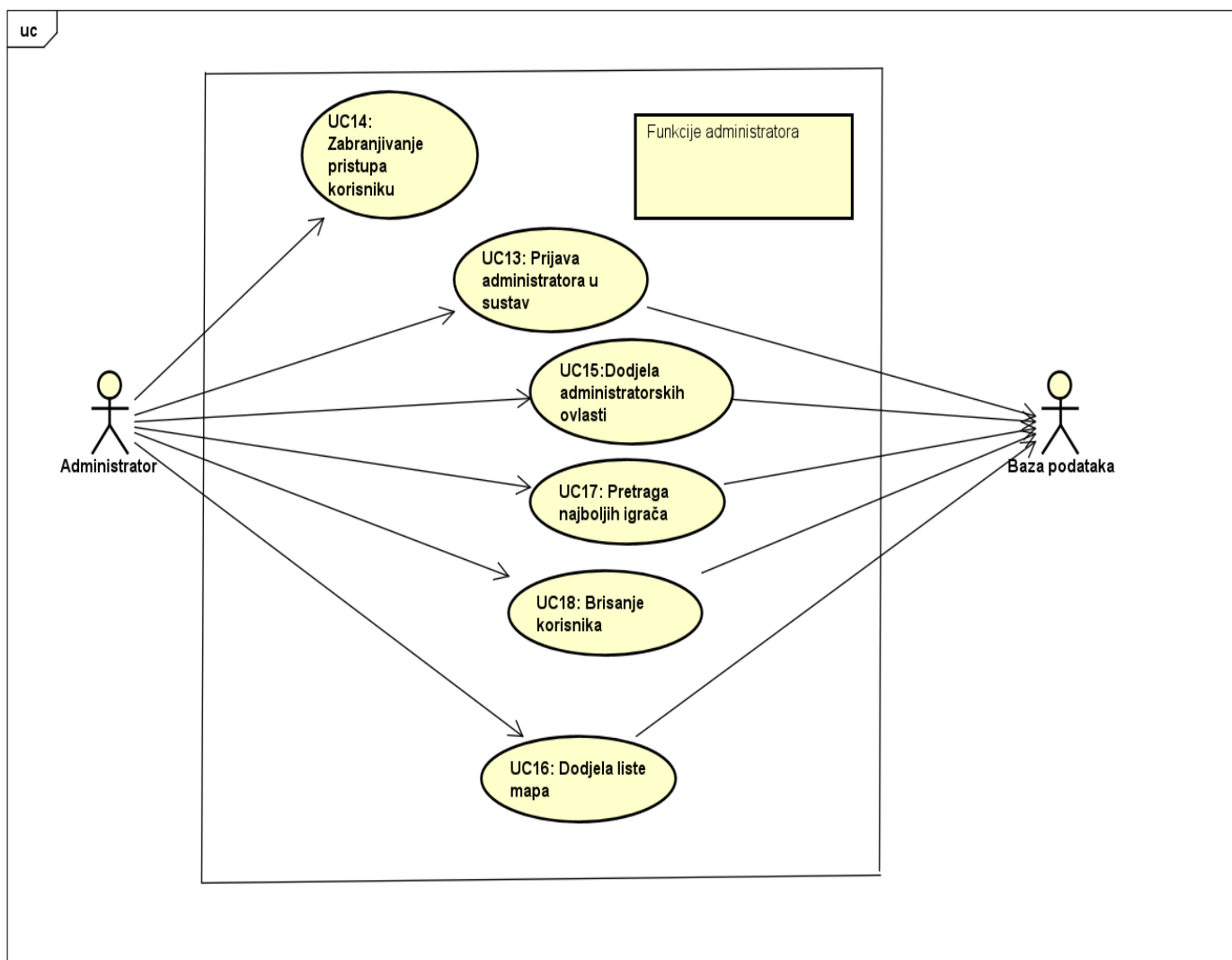
UC18 - brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Administrator odabire opcije "Uklanjanje korisnika"
 2. Administrator u tekstno polje upisuje ime korisnika
 3. Administrator pronalazi korisnika, uklanja ga i njegove podatke iz baze podataka

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe, funkcionalnosti korisnika

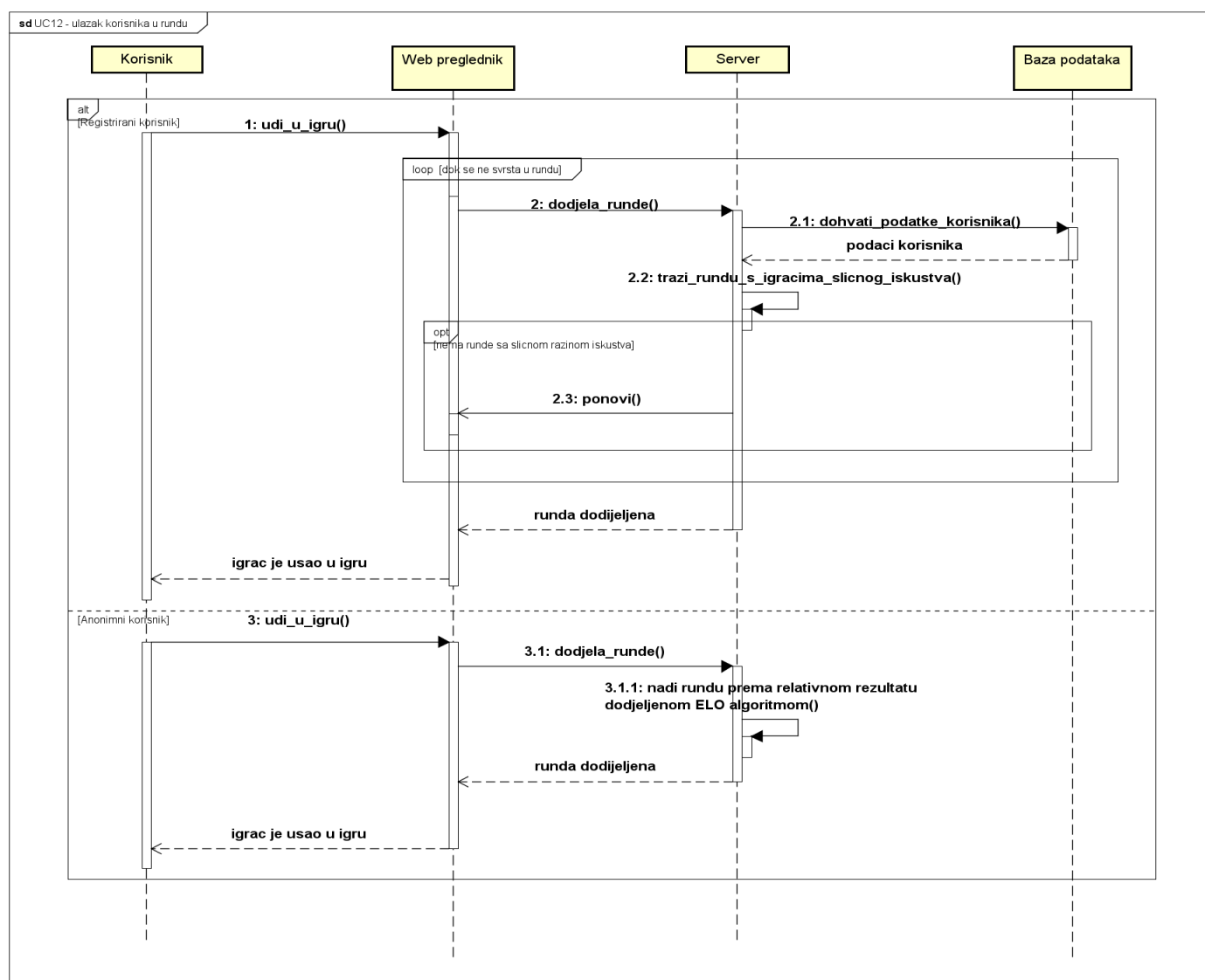


Slika 3.2: Dijagram obrazaca uporabe, funkcionalnosti administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC12 - ulazak korisnika u rundu

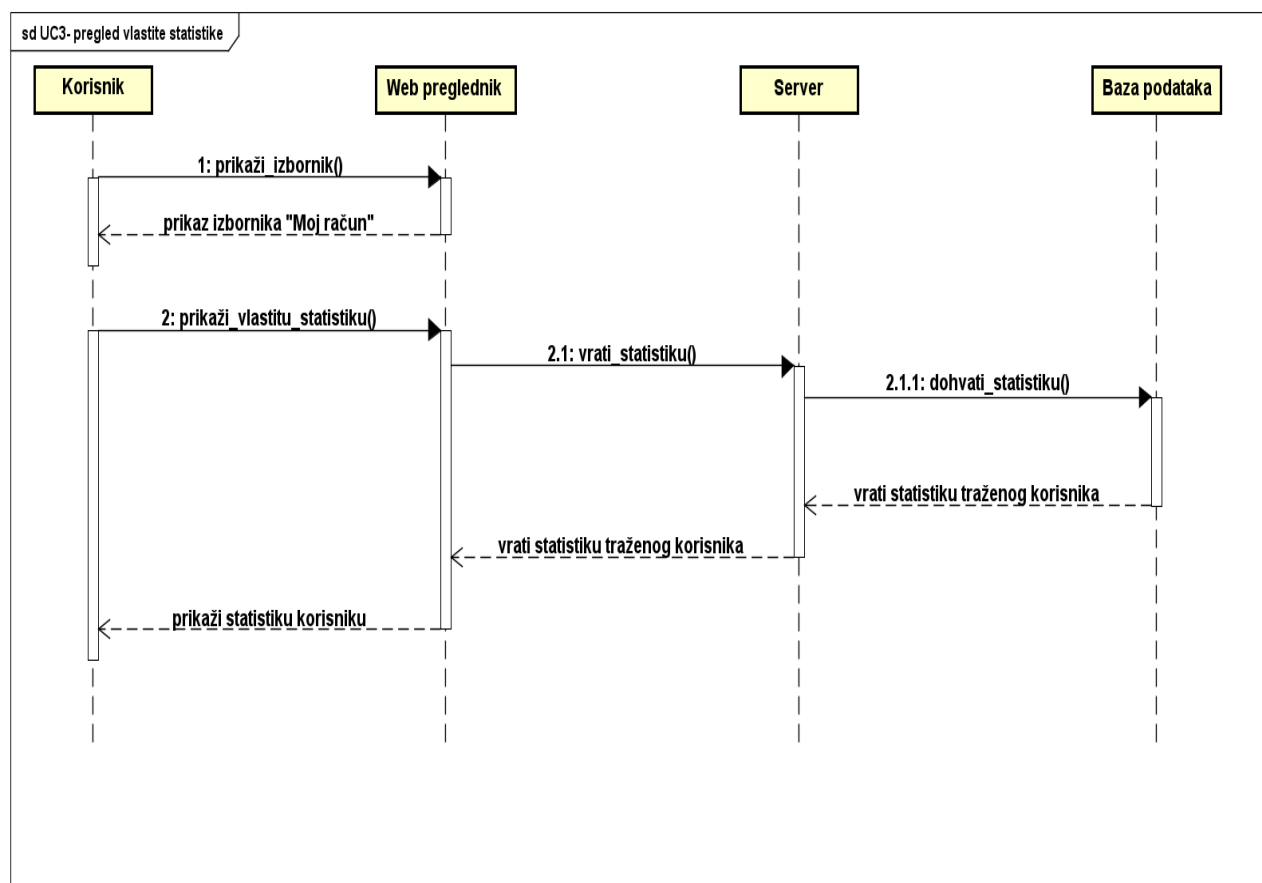
Korisnik želi ući u igru te odabire gumb igraj na web pregledniku. Zahtjev se šalje prema serveru. Korisnik ako je registriran ima svoju dosadašnju statistiku u bazi podataka te se prema njoj traži runda u kojoj su igrači slične razine iskustva. Ukoliko se igrača ne može svrstati u rundu u trenutku zahtjev će se slati opet dok server ne pronade rundu u koju može korisnik ući. Ako u igru pak želi ući anonimni korisnik, on ima fiksni relativni učinak dodijeljen od implementiranog ELO algoritma te će se njega rasporediti u rundu s ostalim anonimnim korisnicima.



Slika 3.3: Sekvencijski dijagram za UC12

Obrazac uporabe UC3 - pregled vlastite statistike

Korisniku se pritiskom na gumb "Moj račun", kojeg vidi na web pregledniku, otvara izbornik gdje ima mogućnost odabrati gumb "Prikaži moju statistiku". Web preglednik će zahtjev za tom akcijom poslati serveru koji taj zahtjev šalje direktno bazi podataka. Baza podataka zatim pronađe statistiku traženog igrača i vraća podatke serveru. Server zatim šalje podatke web pregledniku, a na samom web pregledniku korisnik zatim može vidjeti svoju najnoviju statistiku.



Slika 3.4: Sekvencijski dijagram za UC3

3.2 Ostali zahtjevi

1. Prilikom registracije sustav mora upozoriti korisnika da je korisničko ime već zauzeto.
2. Sustav mora podržati istodobno barem jednog administratora i barem 4 redovna korisnika.
3. Sustav mora biti jednostavan za korištenje, a sučelje mora biti jasno i intuitivno.
4. Sustav mora podržati hrvatske diakritičke znakove za korisnička imena i prilikom komunikacije u prozoru za poruke.
5. Sustav se temelji na HTML-u, CSS-u i JavaScriptu na korisničkoj strani te Node.js-u s express-om na serverskoj strani.

6. Koristi protokol HTTPS za komunikaciju između servera i preglednika weba.
7. Sustav podržava do 50 korisnika.
8. Korisnički podaci moraju biti zaštićeni nekim od algoritama enkripcije.
9. Traženje nove igre ne smije trajati dulje od 5 minuta.
10. Sve naknadne nadogradnje sustava ne smiju narušavati prijašnje funkcionalnosti sustava.
11. Igrici se može pristupiti HTTPS-om iz javne mreže.

4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na tri podsustava:

1. Web poslužitelj
2. Web aplikacija
3. Baza podataka

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Svaki internetski preglednik je prevođitelj. Dakle, stranica je pisana u kodu koji preglednik nakon toga interpretira kao nešto svakome razumljivo. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj osnova je rada web aplikacije. Njegova primarna zadaća je komunikacija klijenta s aplikacijom. Komunikacija s odvija preko HTTPS protokola (engl. *Hyper Text Transfer Protocol Secure*) protokola, koji je protokol u prijenosu informacija na webu. Poslužitelj je onaj koji pokreće web aplikaciju te joj prosljeđuje zahtjev.

Korisnik koristi web aplikaciju za obrađivanje željenih zahtjeva. Web aplikacija obrađuje zahtjev te ovisno o zahtjevu pristupa bazi podataka nakon čega preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku. Programski jezik kojeg smo odabrali za izradu naše web aplikacije je JavaScript te Express radni okvir. Odabrana razvojna okolina je Visual Studio Code. Express podržava MVC koncept i time olakšava razvoj web aplikacije. Arhitektura same aplikacije je podijeljena na klijentski i poslužiteljski dio. Klijentski dio je izgrađen pomoću HTML-a i JavaScripta. Klijentski dio korisnici koriste za interakciju s aplikacijom kroz web preglednik. Klijentski dio šalje zahtjeve poslužitelju ovisno o akcijama korisnika. Poslužiteljski dio izgrađen je pomoću Express razvojnog okvira te primjenjuje MVC koncept. Karakteristika takvog pristupa je razdvajanje domena problema koja rezultira učinkovitijom podjelom članova po timovima. MVC također doprinosi i modularnosti koda i njegovoj čitljivosti.

4.1 Baza podataka

Za potrebu našeg sustava koristiti ćemo relacijsku bazu podataka koja svojom strukturom olakšava modeliranje interakcije u sustavu. Osnovna gradivna jedinica baze je relacija, to jest imenovana tablica s definiranim atributima. Zadaća baze podataka je brza, jednostavna i robusna pohrana, alteracija i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Ima izgled
- Tenk
- Igrao je
- Igra
- Mapa

4.1.1 Opis tablica

Korisnik Ovaj entitet sadrži sve bitne podatke o korisniku. Sadrži attribute: korisničko ime, lozinka, email, razinu ovlasti te datum rođenja. Ovaj entitet je u vezi *One-to-many* s entitetom Igrao je preko korisničkog imena te u vezi *One-to-many* s entitetom Ima izgled preko korisničkog imena.

Korisnik		
korisničko ime	VARCHAR	jedinstveni identifikator korisnika
lozinka	VARCHAR	hash lozinke
email	VARCHAR	e-mail adresa korisnika
razina ovlasti	INT	razina ovlasti korisnika

Ima izgled Ovaj entitet je spojni entitet entiteta Korisnik i Tenk. Sadrži attribute: korisničko ime i tenkId. Ovaj entitet je u vezi *Many-to-one* s entitetom Korisnik preko korisničkog imena te u vezi *Many-to-one* s entitetom Tenk preko tenkId-a.

Ima izgled		
korisničko ime	VARCHAR	jedinstveni identifikator korisnika
tenkId	VARCHAR	jedinstveni identifikator tenka

Tenk Ovaj entitet sadržava informacije o tenku. Sadrži attribute: tenkId, ime tenka, razina otključavanja. Ovaj entitet je u vezi *Many-to-many* s entitetom Ima izgled preko tenkId-a.

Tenk		
tenkId	VARCHAR	jedinstveni identifikator tenka
ime tenka	VARCHAR	naziv tenka
razina otključavanja	INT	razina na kojoj korisnik otključava ovaj tenk

Igrao je Ovaj entitet je spojni entitet entiteta Korisnik i Igra. Sadrži attribute: korisničko ime, broj bodova, puta uništio, puta uništen te igraId. Ovaj entitet je u vezi *Many-to-one* s entitetom Korisnik preko korisničkog imena te u vezi *One-to-many* s entitetom Igra preko igraId.

Igrao je		
korisničko ime	VARCHAR	jedinstveni identifikator korisnika
igraId	INT	jedinstveni idenitifikator igre
puta uništio	INT	broj neprijatelja koje je korisnik uništio za vrijeme igre
puta uništen	INT	broj puta koje je korisnik bio uništen za vrijeme igre

Igra Ovaj entitet sadrži informacije o odigranoj igri. Sadrži attribute: igraId, vrijeme početka, vrijeme kraja, mapaId. Ovaj entitet je u vezi *One-to-many* s entitetom Igrao je preko igraId-a te u vezi *Many-to-one* s entitetom Mapa.

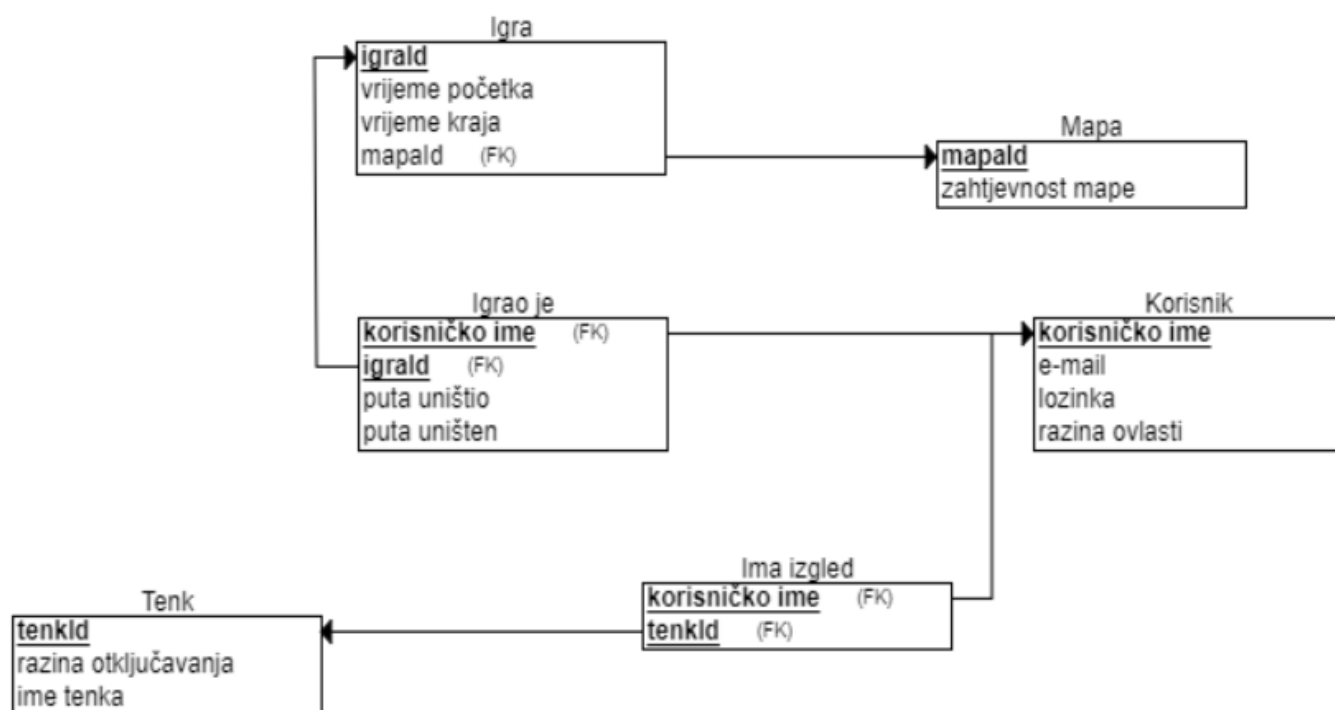
Igra		
igraId	SERIAL INT	jedinstveni identifikator igre
mapaId	INT	jedinstveni identifikator mape

Igra		
vrijeme početka	TIMESTAMP	trenutak početka igre
vrijeme kraja	TIMESTAMP	trenutak kraja igre

Mapa Ovaj entitet sadrži informacije o mapama. Sadrži attribute: mapaId te zahtjevnost mape. Ovaj entitet je u vezi *One-to-many* s entitetom Igra preko mapaId-a.

Mapa		
mapaId	INT	jedinstveni identifikator mape
zahtjevnost mape	INT	zahtjevnost mape za igrati na njoj

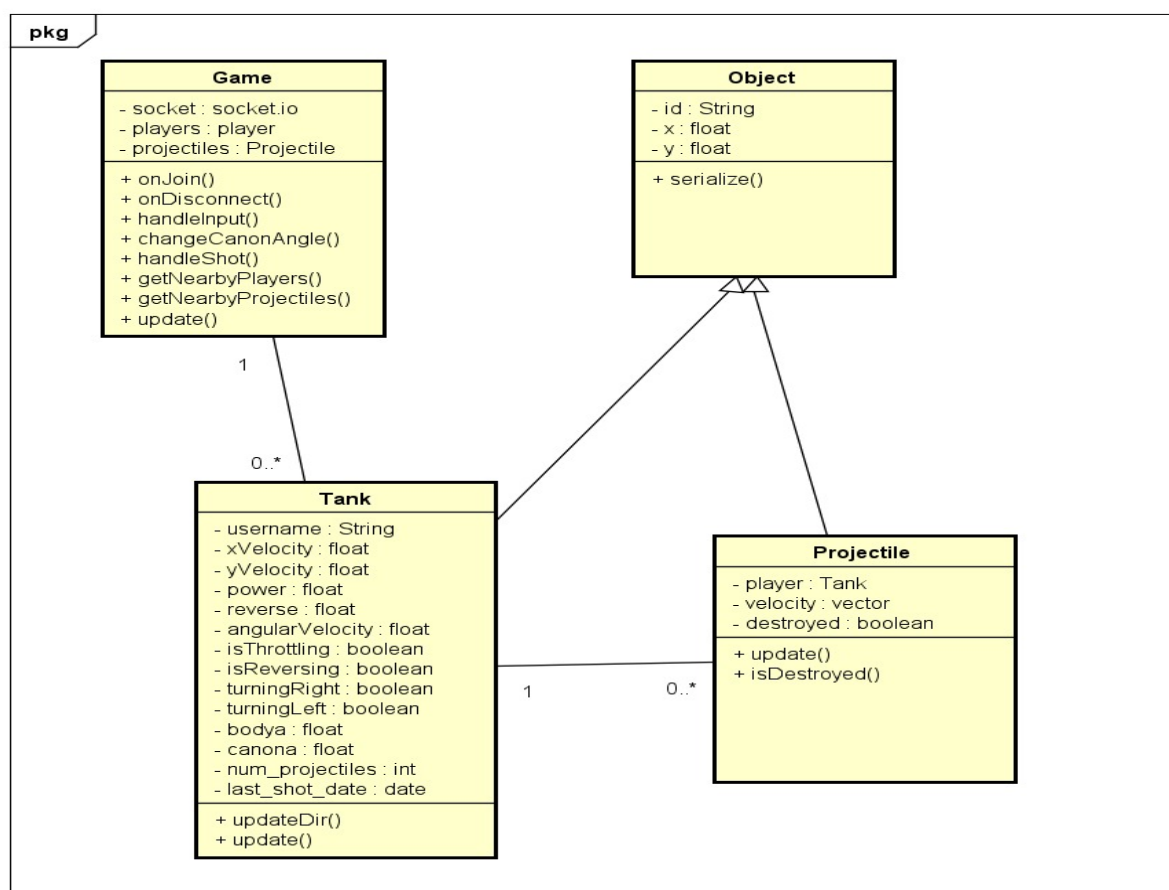
4.1.2 Dijagram baze podataka



Slika 4.1: Dijagram baze podataka

4.2 Dijagram razreda

Na slici 4.2 prikazani su razredi koji pripadaju *backend* dijelu MVC arhitekture. Razred Objekt je razred kojeg nasljeđuju svi objekti koji se pojavljuju na igračem polju. Trenutno su to samo razredi Projectile i Tank. Za svaki tenk je vezana kolekcija svih projektila koje je ispalio, a da su još uvijek u igračem polju. Također, postoji i razred Game koji reprezentira jednu igru. Budući da je moguće imati više igara u tijeku, moguće je imati i više instanci tog razreda. U jednoj igri može biti više igrača(tenkova) dok jedan igrač može trenutno biti u samo jednoj igri. U budućim iteracijama planira se više razreda objekata na mapi koji nasljeđuju razred Object. Budući da JavaScript ne razlikuje privatne, zaštićene i javne metode i attribute oni nisu ni navedeni.



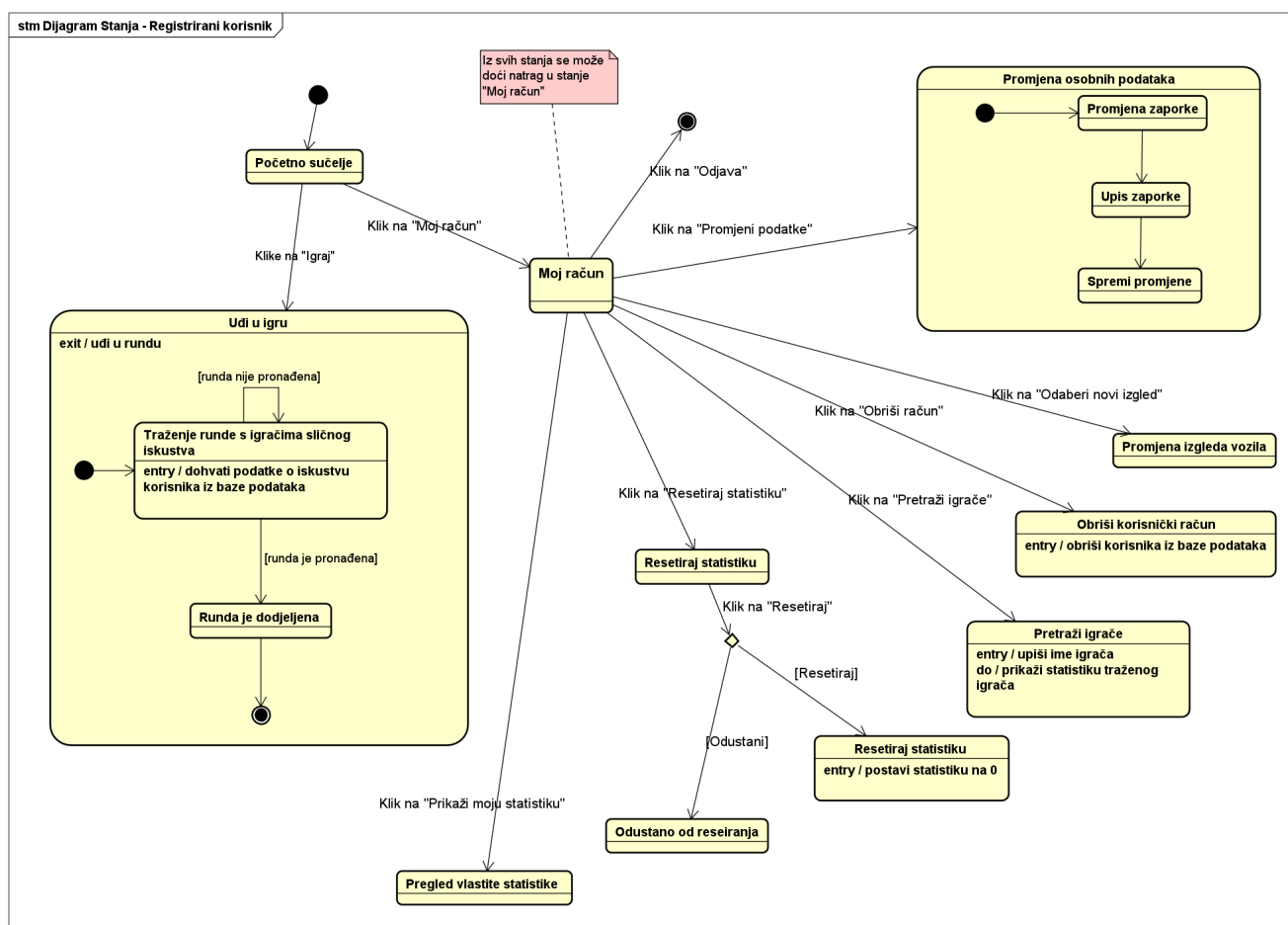
Slika 4.2: Dijagram razreda

dio 2. revizije

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

4.3 Dijagram stanja

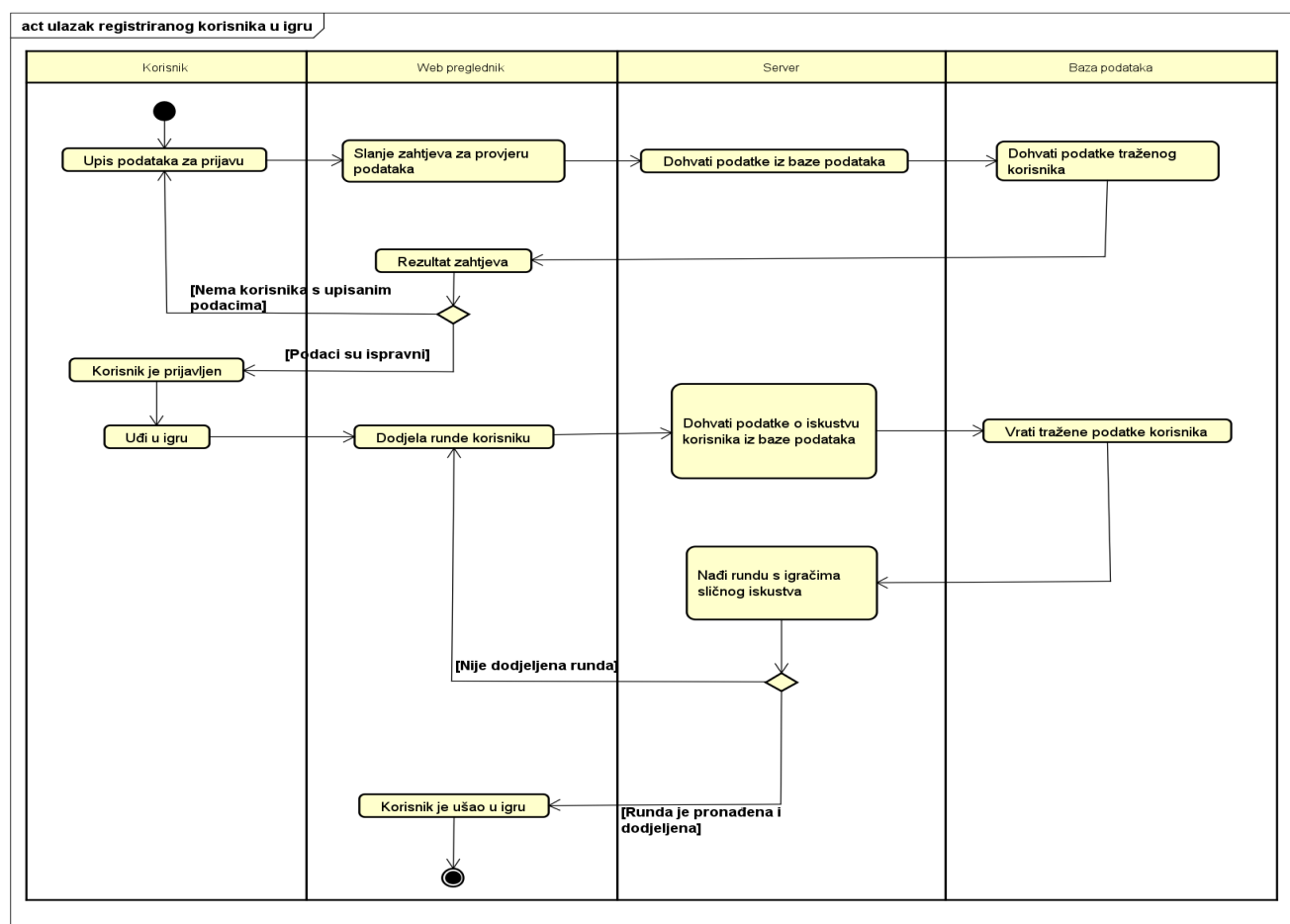
UML dijagram stanja opisuje dinamičko ponašanje dijela sustava. Prikazuje stanja te prijelaze u neka druga stanja temeljene na nekim događajima. U ovom slučaju događaje određuje registrirani korisnik. Najprije, klikom na "Moj račun" korisniku se otvara izbornik s različitim mogućnostima koje može odabrati sve dok ne odluči kliknuti "Odjava". Korisnik može pregledati vlastitu statistiku ili resetirati statistiku. Kada će potvrditi resetiranje statistike, ona će se u bazi podataka postaviti na 0. Također, korisnik može pretražiti igrače. Kada odabere opciju "Pretraži igrače", automatski se pojavi polje za upis imena željenog igrača. U izborniku se nudi i promjena izgleda vozila koje je korisnik otključao. Klikom na "Promjeni podatke", korisnik može promijeniti svoje podatke tako da upiše novu zaporku i zatim spremi podatke. Korisnik može i obrisati svoj račun klikom na gumb "Obrši račun" čime će se iz baze podataka automatski korisnik obrisati. Nakon mogućih opcija koje korisnik može odabrati na "Moj račun", on također ulazi u samo igru klikom na gumb "Igraj". Nakon toga, server će tražiti rundu, ali odmah po ulasku u stanje traženja runde iz baze podataka će dobiti podatke o iskustvu kako bi pronašao odgovarajuću rundu. Ako runda nije pronađena, postupak će se ponoviti, a u suprotnom korisnik ulazi u rundu.



Slika 4.3: Dijagram stanja

4.4 Dijagram aktivnosti

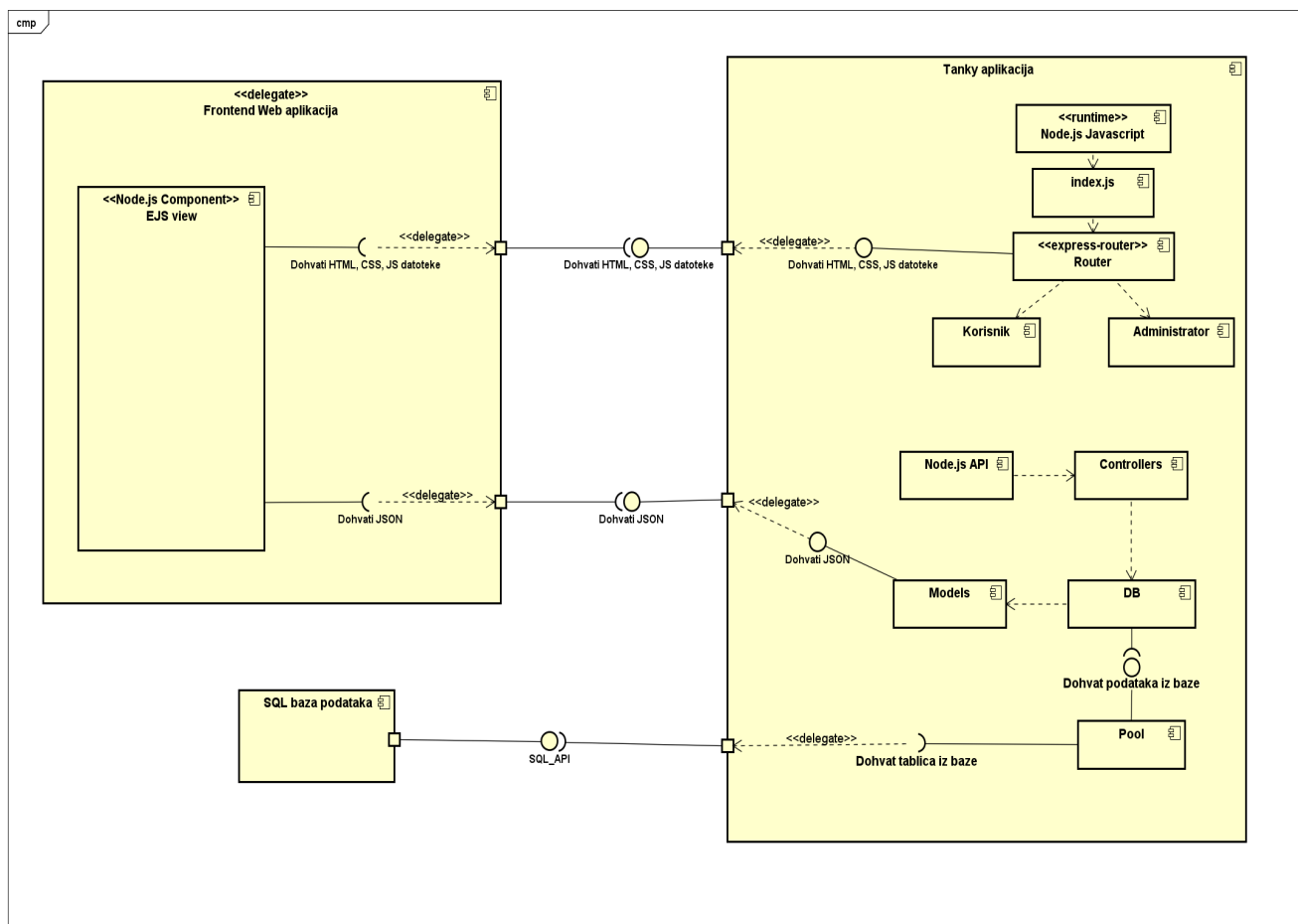
UML dijagram aktivnosti modelira ponašanja nizom aktivnosti, ali pritom se prijelaz iz jedne aktivnosti u drugu ne potiče nekim događajem. Priloženi dijagram aktivnosti prikazuje ulazak registriranog korisnika u igru. Korisnik se najprije prijavljuje u sustav sa svojim podacima. Akcije koje slijede su postupak prijave korisnika sve dok baza podataka ne vrati tražene podatke ili ih ne pronađe što znači da prijava nije uspjela. Nakon prijave, korisnik će ući u igru. Web preglednik će od servera tražiti dodjelu runde. Na serveru će se dogoditi dvije akcije. Prvo će iz baze podataka dohvatiti razinu iskustva tog korisnika, a zatim pronaći rundu s igračima slične razine iskustva. Ukoliko pronađe rundu, korisnik će uspješno ući u igru, a u suprotnom će se ponoviti traženje runde.



Slika 4.4: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti prikazuje specifikaciju arhitekture programske potpore. Vizualizira organizaciju i međuovisnosti između implementacijskih komponenata.



Slika 4.5: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Projektni tim je tijekom izrade projekta komunicirao putem Microsoft Teams(1) i WhatsApp(2) aplikacija. Pri izradi UML dijagrama korišten je alat Astah UML(3). Kao sustav za upravljanje izbornim kodom korišten je Git(4), a web platforma GitLab(5) koristila se kao udaljeni repozitorij. Visual Studio Code(6) koristio se kao razvojna okolina za izradu projekta. Radni okvir Express(7) u kombinaciji sa serverskim okruženjem Node.js(8) korišten je za izradu aplikacije na poslužiteljskoj strani, a na klijentskoj strani korišteni su HTML(9), CSS(9) te jezik Javascript(10). Express je brz minimalistički web radni okvir za Node.js, a značajke su mu: robusno preusmjeravanje, fokus na visoke performanse i http pomoćne metode. Sustav za upravljanje bazom podataka korišten na ovom projektu je PostgreSQL(11). Kao pomoć pri razmještanju korišten je Digital Ocean(12).

(1) <https://www.microsoft.com/>

(2) <https://web.whatsapp.com/>

(3) <https://astah.net/products/astah-uml/>

(4) <https://git-scm.com/>

(5) <https://gitlab.com/>

(6) <https://code.visualstudio.com/>

(7) <https://expressjs.com/>

(8) <https://nodejs.org/en/>

(9) <https://html.com/>

(10) <https://www.javascript.com/>

(11) <https://www.postgresql.org/>

(12) <https://www.digitalocean.com/>

5.2 Ispitivanje programskog rješenja

Svi unit testovi napravljeni su nad klasom Tank koja implementira najvažnije funkcionalnosti. Napravljeni su nad funkcijama koje izazivaju moguće poteškoće u samom kodu i tokom igranja igrice. Funkcije se odnose na koliziju tanka sa objektima na mapi. Na priloženoj slici vidimo da su svi testovi uspješno prošli.

5.2.1 Ispitivanje komponenti

```
1  const tank = require('./tank.js');
2  const Constants = require('../const/constants');
3
4  var t = new tank(1, 'asd', 1, 1, 1, 1, 1, 1)
5
6  test('VertexEdgeRelation returns -1000', () => {
7    const d = t.vertexEdgeRelation(100, 100, 0, 0, 10, 0);
8    expect(d).toBe(-1000)
9  })
10
11 test('onSegment returns false', () => {
12   const d = t.onSegment([0,0], [0,10], [10, 0]);
13   expect(d).toBe(false)
14 })
15
16 test('onSegment returns true', () => {
17   const d = t.onSegment([0,0], [0,10], [5, 10]);
18   expect(d).toBe(true)
19 })
20
21 test('orientation returns 1', () => {
22   const d = t.orientation([50, 20], [25, 50,], [100, 10]);
23   expect(d).toBe(1)
24 })
25
26 test('orientation returns 2', () => {
27   const d = t.orientation([100, 10], [25, 50,], [50, 20]);
28   expect(d).toBe(2)
29 })
```

Slika 5.1: Unit testovi 1.dio

```
30
31 test('orientation returns 0', () => {
32   const d = t.orientation([0,0], [0,10], [0, 20]);
33   expect(d).toBe(0)
34 })
35
36 test('intersect returns true', () => {
37   const d = t.intersect([50, 10], [20, 10], [40, 20], [20, 5]);
38   expect(d).toBe(true)
39 })
40
41 test('intersect returns false', () => {
42   const d = t.intersect([50, 10], [20, 10], [40, 20], [20, 20]);
43   expect(d).toBe(false)
44 })
```

Slika 5.2: Unit testovi 2.dio

```
PASS ./project.test.js
✓ VertexEdgeRelation returns -1000 (2 ms)
✓ onSegment returns false
✓ onSegment returns true
✓ orientation returns 1
✓ orientation returns 2
✓ orientation returns 0
✓ intersect returns true
✓ intersect returns false

Test Suites: 1 passed, 1 total
Tests: 8 passed, 8 total
Snapshots: 0 total
Time: 1.387 s
Ran all test suites.
```

Slika 5.3: Rezultati unit testova

5.2.2 Ispitivanje sustava

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

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

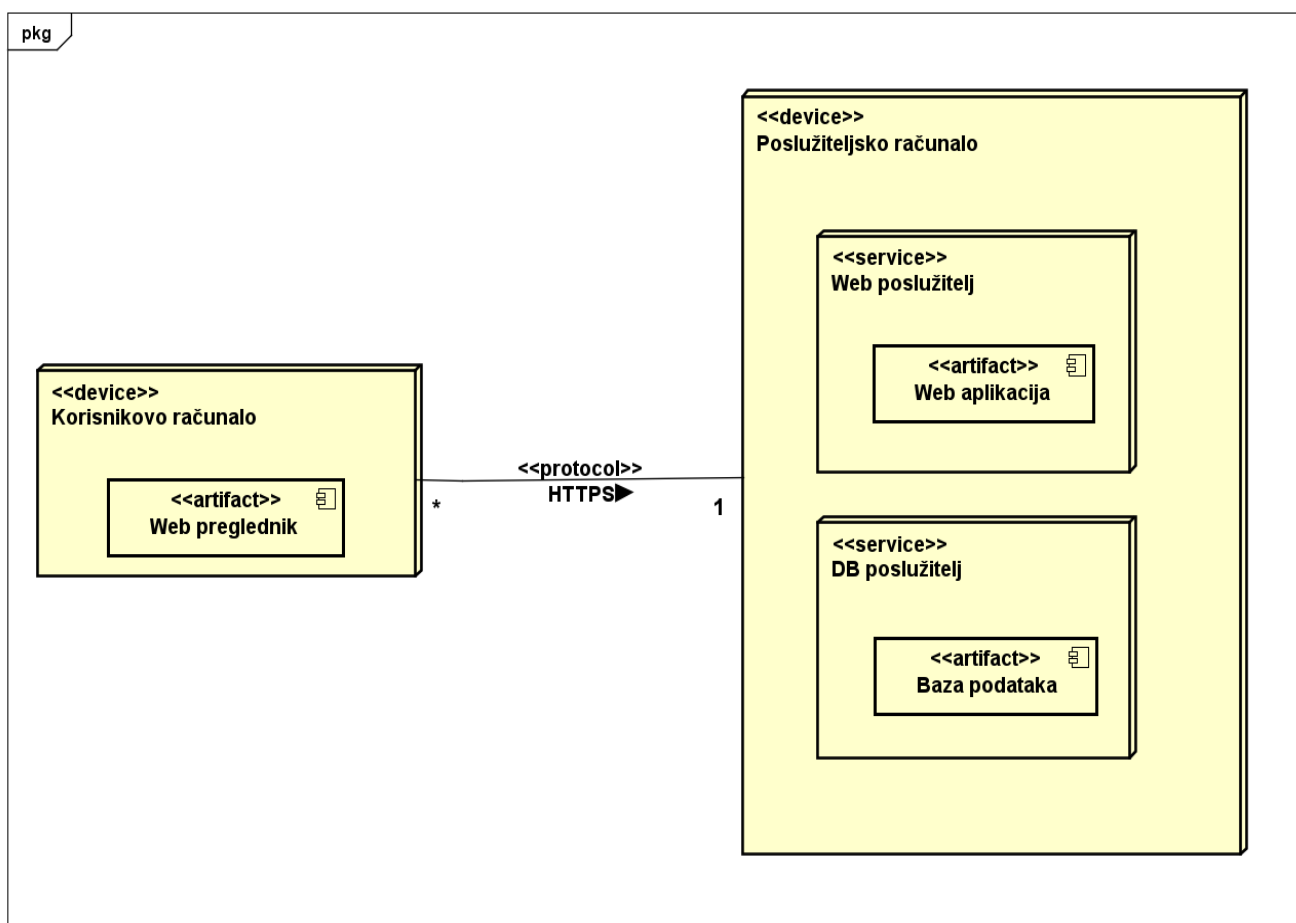
- dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita
- **Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sustava i odnos sklopovskih i programskih dijelova. Na poslužiteljskom računalu se nalaze web poslužitelj i poslužitelj baze podataka. Klijenti preko web preglednika na svom računalu pristupaju web aplikaciji. Sustav je baziran na arhitekturi "klijent – poslužitelj", a komunikacija između računala korisnika i poslužitelja odvija se preko HTTPS veze. Dakle, korisnik putem web preglednika šalje zahtjev web poslužitelju koji pokreće web aplikaciju i proslijeđuje joj zahtjev.



Slika 5.4: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Za puštanje aplikacije u pogon potreban je korisnički račun na usluzi Digital Ocean. Zapošlebe projekta, koristimo besplatnu verziju korisničkog računa. Za potrebe projekta korištena je baza podataka PostgreSQL. Konfiguracijska datoteka za bazu podataka nalazi se u datoteci izvorni_kod/db/the_backup.sql. Prazna baza podataka pokreće se korištenjem naredbe “psql the_backup.sql”. Nakon toga na bazu se spaja koristeći datoteku izvorni_kod/db/inde.js gdje se unosi korisničko ime za bazu podataka, lozinka i port. Za puštanje aplikacije u rad potrebno je klonirati repozitorij na server te pokrenuti naredbom “npm start” uz prethodno postavljanje node.js.

6. Zaključak i budući rad

Prva faza projekta započela je okupljanjem projektnog tima i definicijom vlastitog projektnog zadatka. Tim je podijeljen na manje podtimove koji su bili specijalizirani za različita područja razvoja. Izrađeni su početni konceptualni prototipi. Prototipi su provjereni te se nastavio razvoj na najboljem. Sama igrica koja je razvijena je postavljena na Digital Ocean server te se mogla slobodno isprobati.

U drugoj fazi razvoj igrice je bio u fokusu te su se dodavale nove značajke. Paralelno se igricu testiralo te su se ispravljali pronađeni problemi. Također, u drugoj fazi je dodano i korisničko sučelje za ulaz u igru, te grafika za prijavljivanje i registriranje.

U trećoj fazi izgrađena je baza podataka te je izgrađena poslužiteljska strana aplikacije. U ovoj fazi je provedeno i testiranje te su dodane završne značajke za samu igru.

Potencijalna nadogradnja aplikacije je dodavanje drugih modova igre te podrška za druge jezike.

U radu na ovom projektu susreli smo se s raznim problemima vezanim uz neiskustvo, manjak vremena članova tima te raznovrsnosti korištenih tehnologija. Rad na ovom projektu ponudio nam je nove vještine razvoja web aplikacija te komunikacijske vještine nužno potrebne za rad u timu.

Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.

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.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Razvoj programske potpore za web i pokretne uređaje, <https://www.fer.unizg.hr/predmet/rpp>

Indeks slika i dijagrama

2.1	Početni izbornik	5
2.2	AZ Tank Game	6
2.3	Slither.io	7
3.1	Dijagram obrazaca uporabe, funkcionalnosti korisnika	16
3.2	Dijagram obrazaca uporabe, funkcionalnosti administratora	17
3.3	Sekvencijski dijagram za UC12	18
3.4	Sekvencijski dijagram za UC3	19
4.1	Dijagram baze podataka	24
4.2	Dijagram razreda	25
4.3	Dijagram stanja	28
4.4	Dijagram aktivnosti	29
4.5	Dijagram komponenti	30
5.1	Unit testovi 1.dio	32
5.2	Unit testovi 2.dio	33
5.3	Rezultati unit testova	33
5.4	Dijagram razmještaja	35

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 2. listopada 2020.
- Prisustvovali: svi članovi
- Teme sastanka:
 - brainstorming

2. sastanak

- Datum: 5. listopada 2020.
- Prisustvovali: F.Čutura, H.Kristić, B.Kušen, L.Mutvar, T.Pavić, L.Pavlović
- Teme sastanka:
 - 1. laboratorijska vježba s asistentom

3. sastanak

- Datum: 6. listopada 2020.
- Prisustvovali: F.Čutura, H.Kristić, L.Mutvar, T.Pavić, L.Pavlović
- Teme sastanka:
 - oblikovanje dokumenta vlastitog prijedloga zadatka

4. sastanak

- Datum: 7. listopada 2020.
- Prisustvovali: F.Čutura, L.Pavlović
- Teme sastanka:
 - revizija dokumenta 6. listopada

5. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: B.Kušen, F.Čutura, H.Kristić, L.Mutvar, T.Pavić, L.Pavlović
- Teme sastanka:
 - konfiguriranje git-a
 - rasprava o tehnologijama i raspored u radne skupine

6. sastanak

- Datum: 19. listopada 2020.
- Prisustvovali: svi članovi
- Teme sastanka:
 - laboratorijska vježba s asistentom
 - dogovor o Latex okviru za dokumentaciju te dnevniku sastanaka

7. sastanak

- Datum: 22. listopada 2020.
- Prisustvovali: B.Kušen, F.Čutura, L.Mutvar, L.Pavlović
- Teme sastanka:
 - sastanak uživo
 - razrada daljnjeg razvoja
 - prikaz koncepata igrice

8. sastanak

- Datum: 23. listopada 2020.
- Prisustvovali: F.Čutura, H.Kristić, T.Pavić, L.Pavlović
- Teme sastanka:
 - dogovor za grafiku
 - demonstracija radne verzije

9. sastanak

- Datum: 4.studeni 2020.
- Prisustvovali: T.Pavić, L.Pavlović
- Teme sastanka:
 - zajednički rad na dokumentaciji

10. sastanak

- Datum: 6. studeni 2020.
- Prisustvovali: B.Kušen, F.Čutura, H.Kristić, L.Mutvar, T.Pavić, L.Pavlović
- Teme sastanka:
 - planovi za daljnji rad
 - raspodjela zadataka oko dovršetka dokumentacije
 - prijenos znanja

11. sastanak

- Datum: 9. studeni 2020.
- Prisustvovali: B.Kušen, F.Čutura, H.Kristić, L.Mutvar, T.Pavić, L.Pavlović
- Teme sastanka:
 - obavezni sastanak s asistentom i stress test implementacije

12. sastanak

- Datum: 11. studeni 2020.
- Prisustvovali: B.Kušen, F.Čutura, L.Mutvar, T.Pavić, L.Pavlović
- Teme sastanka:
 - rasprava o dijagramu razreda i budućnosti razvoja razreda

13. sastanak

- Datum: 1. prosinca 2020.
- Prisustvovali: B.Kušen, F.Čutura, L.Mutvar, H.Kristić, T.Pavić, L.Pavlović
- Teme sastanka:
 - izvještaj o dosadašnjem napretku, dogovor o daljnjem radu

14. sastanak

- Datum: 30. prosinca 2020.
- Prisustvovali: F.Čutura, L.Mutvar, H.Kristić, T.Pavić, L.Pavlović
- Teme sastanka:
 - raspodjela poslova za daljnji rad

15. sastanak

- Datum: 6. siječnja 2021.
- Prisustvovali: B.Kušen, F.Čutura, L.Mutvar, H.Kristić, T.Pavić, L.Pavlović
- Teme sastanka:
 - pregled napravljenog i dogovor za daljnji rad

16. sastanak

- Datum: 8. siječnja 2021.
- Prisustvovali: svi članovi
- Teme sastanka:
 - pregled napravljenog
 - dogovor za daljnji rad
 - laboratorijska vježba s asistentom

17. sastanak

- Datum: 14. siječnja 2021.
- Prisustvovali: B.Kušen, F.Čutura, H.Kristić, T.Pavić, L.Pavlović
- Teme sastanka:
 - pregled napravljenog i dogovor za završetak rada

Tablica aktivnosti

	Luka Pavlović	Fran Čutura	Benjamin Kušen	Tihomir Pavić	Lovro Mutvar	Hrvoje Kristić	Niko Bucalo
Upravljanje projektom	2.5h	2.5h					
Opis projektnog zadatka	2.5h	2.5h	1h	1h	3h	1h	1h
Funkcionalni zahtjevi				3h			
Opis pojedinih obrazaca				4h			
Dijagram obrazaca				3h			
Sekvencijski dijagrami				3h			
Opis ostalih zahtjeva						2h	
Arhitektura i dizajn sustava						1h	
Baza podataka	3h						
Dijagram razreda					2h		
Dijagram stanja				2h			
Dijagram aktivnosti				1h			
Dijagram komponenti	1h			1h			
Korištene tehnologije i alati	0.5h			0.5h			
Ispitivanje programskog rješenja			1h				
Dijagram razmještaja				1h			
Upute za puštanje u pogon							
Dnevnik sastajanja	0.5h						
Zaključak i budući rad	0.5h						
Popis literature				0.5h			
Grafika, HTML, CSS				2h	1h		15h
Izrada baze podataka	12h						
Routing						6h	
Back end	10h					10h	
Izrada igrice		60h	60h		5h		

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.