

2026

# Bomb Rush

MODEL PODATAKA I PERZISTENCIJE

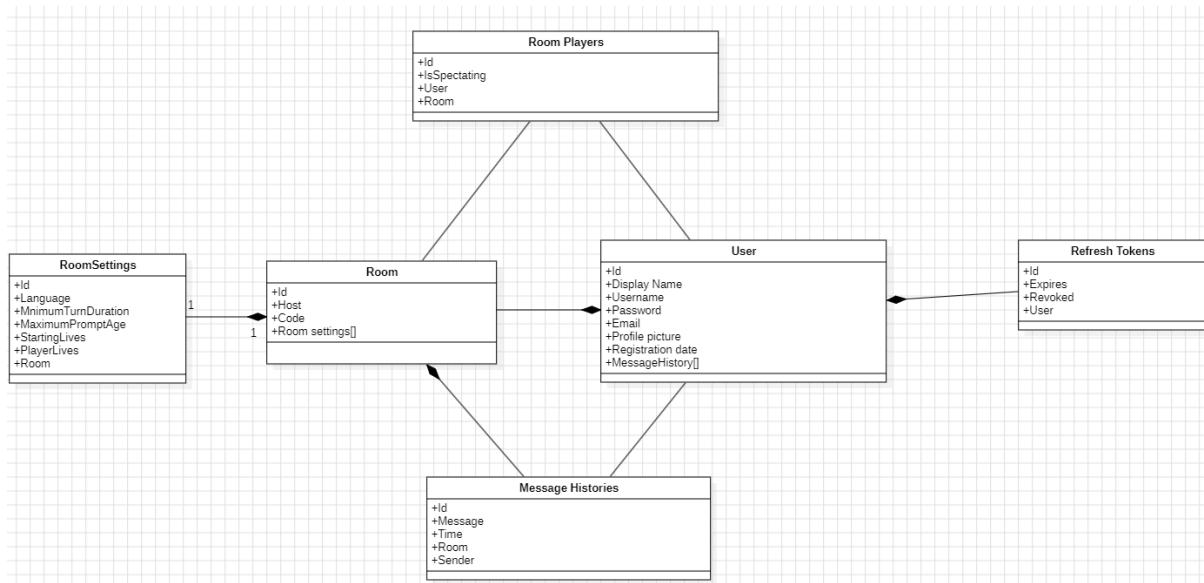
STEFAN CVETKOVIĆ 19461 | LAZAR STANKOVIĆ 19375

## Contents

MODEL PODATAKA.....	2
Dijagram klasa .....	2
Klasa User .....	2
Klasa Room .....	2
Klasa RoomSettings .....	2
Klasa MessageHistories.....	3
Klasa RoomPlayers.....	3
Klasa RefreshTokens.....	3
MODEL PERZISTENCIJE .....	4
Dijagram entiteta.....	4
Entitet Users.....	4
Entitet Refresh Tokens .....	4
Entitet Rooms .....	5
Entitet RoomSettings.....	5
Entitet MessageHistories .....	5
Entitet RoomPlayers .....	5
MAPIRANJE.....	6

# MODEL PODATAKA

## Dijagram klasa



## Klasa User

Klasa User nam predstavlja glavnu klasu samog sistema. Ova je klasa koja će biti direktno povezana sa igračem igre. Ova klasa će kao dužnost imati kreiranje svojih „prostorija“ u kojima će se vršiti igranje igre između različitih igrača.

## Klasa Room

Klasa Room nam predstavlja klasu kreirane od strane korisnika, igrača. Svaki igrač može kreirati samo jednu prostoriju i toj prostoriji se mogu pridružiti ostali igrači. Nakon kreiranje sobe se generiše kod te prostorije preko kojeg se drugi igrači pridružuju.

## Klasa RoomSettings

Klasa RoomSettings služi za podešavanje same igre u nekoj prostoriji. Tu klasu može menjati samo kreator.

## Klasa MessageHistories

Klasa MessageHistories čuva sve poruke u globalnom chat-u neke prostorije. Prilikom igranja igre igrači imaju pristup četu.

## Klasa RoomPlayers

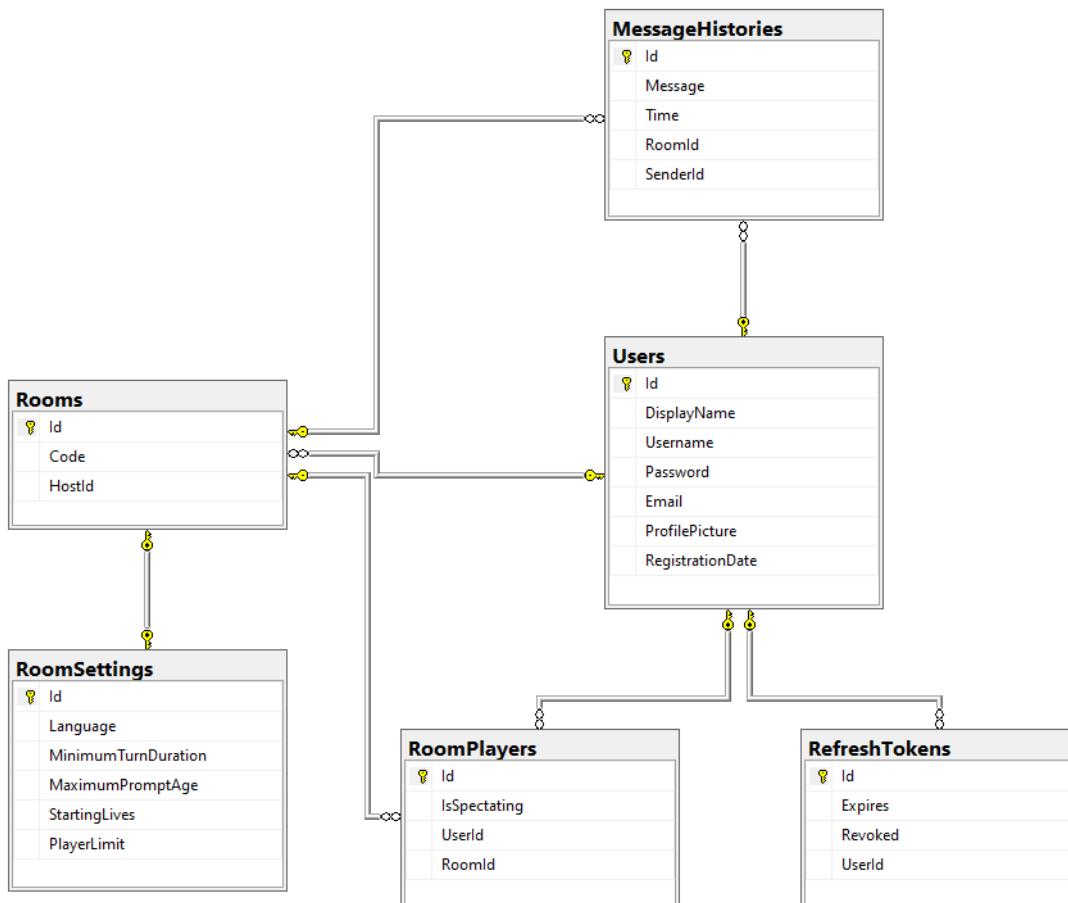
Klasa RoomPlayers predstavlja direktnu vezu između igrača i prostorija.

## Klasa RefreshTokens

Klasa RefreshTokens nam čuva tokene igrača koji se čuvaju pomoću SecureCookies. Na ovaj način možemo sačuvati sesiju korisnika nakon isključenja sajta ili refresh-a.

# MODEL PERZISTENCIJE

## Dijagram entiteta



## Entitet Users

Entitet koji pamti podatke o korisnicima (igračima). O igračima se pamte informacije za prijavu (username/email i password), display ime koje će se prikazivati drugim igračima, profilna slika koja nije obavezna i datum registracije. Šifra će se pamtititi kao Hash i pri prijavljivanju će se upoređivati Hash unete šifre sa Hash-om koji se čuva u bazi.

## Entitet Refresh Tokens

Entitet Refresh Tokens čuva podatke o tokenu. On čuva datum isteka toktena (Expires), status da li je token prekinut (Revoked) i povezanost sa igračem. Klikom na logout dugme se postavlja vrednost svih tokena igrača revoked na true.

## Entitet Rooms

Entiter Rooms skladišti podakte o prostorijama korisnika. Od atributa sadrži samo kod sobe i Id kreator. Sva ostala podešavanja igre se čuvaju u RoomSettings.

## Entitet RoomSettings

Entitet RoomSettings skladišti podatke o podešavanjima igre. On čuva jezik igre (u bazi je short dok se u programu koristi enum eLanguage), MinimumTurnDuration preko kojeg se podešava koliko najmanje traje potez, preko MaximumPromptAge se podešava posle koliko promašenih reči se menjaju slova na bombi, StartingLives pamti broj početnih života igrača i PlayerLimit pamti koliko najviše igrača može igrati igru u toj sobi.

## Entitet MessageHistories

Ovaj entitet čuva sve poruke koje su poslate u određenoj sobi. On pamti datum slanja poruke, sadržaj poruke i informacije o tome ko je poslao poruku i gde.

## Entitet RoomPlayers

Ovaj entitet pamti informacije o tome koji igrač se nalazi u kojoj prostoriji i da li taj igrač samo posmatra igra ili učestvuje u njoj.

## MAPIRANJE

Bomb Rush kao bazu podataka koristi SQL Server uz koju dolazi SQL Server Management Studio koji se može koristiti za proveru stanja samih podataka u bazi.

API samog projekta je pisan u C# jeziku koristeći .NET API. Za mapiranje podataka iz baze i samu komunikaciju sa njom se koristi Entity Core Framework. On nam omogućava da pišemo SQL upite pomoću LINQ-a, bez pisanja samih SQL query-a. U folderu models se čuvaju sve tabele iz baze podataka koje se mapiraju u ApplicationDbContext-u. Da bi se izvršila direktna promena neke tabele u bazi potrebno je samo izmeniti model te tabele i u package manager konzoli napisati „Add-Migration Naziv1“ i nakon toga „Update-Database“. Na ovaj način se može i kreirati baza podataka ako već ne postoji u sistemu.

Koristiće se DTO (Data Transfer Objects) za lakše i preglednije prenošenje podataka između baze i kontrolera. Preko DTO možemo lepo prilagoditi da nam se podaci šalju klijentskoj strani u što preglednijem i jednostavnijem obliku.