

Visoka škola strukovnih studija za informacione i komunikacione
tehnologije

DOKUMENTACIJA

InkWell

Web Programiranje ASP

Student: Branko Pualić-Radujko

Broj indeksa: 13/21

1. Uvod.....	3
1.1 Korišćeni alati.....	3
1.2 Opis projekta.....	3
1.3 Šema baze podataka	4
1.4 Struktura projekta.....	5
2. Pregled funkcionalnosti	9
2.1 Authentication and Authorization	9
2.1.1 Signup	9
2.1.2 Signin.....	10
2.2 Ostalo	11

1. Uvod

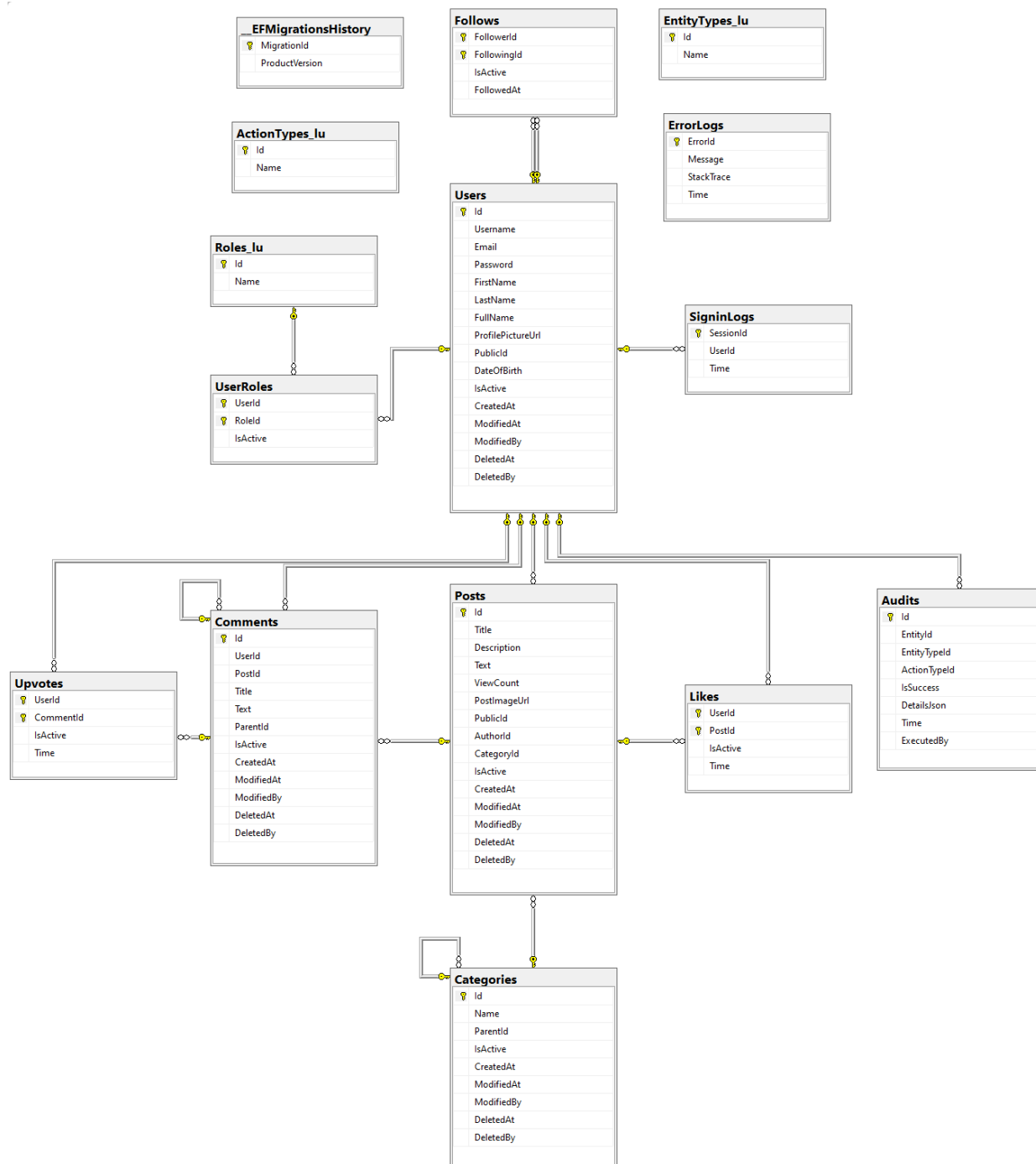
1.1 Korišćeni alati

- **IDE/Editor:** Visual Studio
- **Version Control:** Git
- **Programming Language:** C#, .Net8, SQL Server
- **Test Environment:** Postman

1.2 Opis projekta

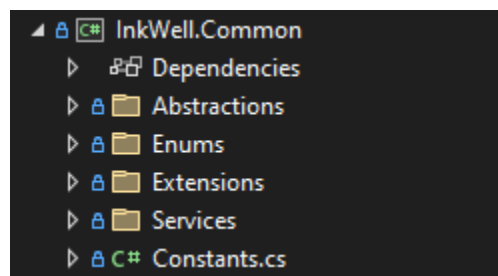
- InkWell projekat predstavlja mogućnosti jedne blog aplikacije. Korisnici mogu da čitaju i pregledaju postove i komentare. Uz pomoć logovanja korisniku se daje mogućnost da lajkuje postove, dodaje komentare, lajkuje komentare, prati druge korisnike, itd.
- Projekat je radjen prateći principe **Clean** arhitekture, takodje koristi **CQRS** pattern za čitanje i upis podataka, i **UnitOfWork**, **Repository** pattern za komunikaciju sa bazom podataka.

1.3 Šema baze podataka

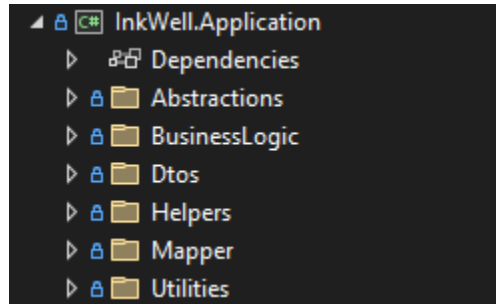


1.4 Struktura projekta

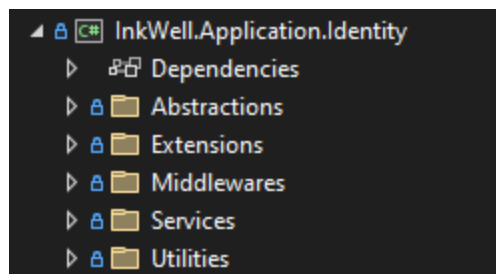
- Projekat je podeljen na sledeće direktorijume i njihove slojeve:
 - **Common**
 - Common
 - **Core**
 - Application
 - Application.Identity
 - Domain
 - Persistence
 - **Infrastructure**
 - Infrastructure
 - DependencyInjection
 - **Presentation**
 - WebApi
 - **Test**
 - Test
- **Common sloj** predstavlja sloj koji obuhvata zajedničke potrebstine aplikacije.



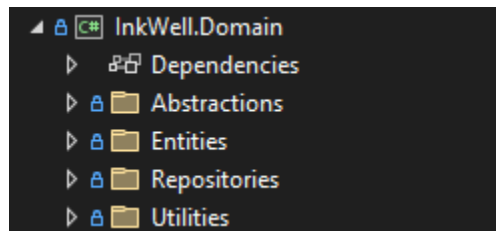
- **Application sloj** obuhvata svu biznis logiku aplikacije. Ovde su smeštene abstrakcije **MediatR** biblioteke koju aplikacija koristi zajedno sa **CQRS** pattern-om za upis i čitanje podataka. Takodje ovde su smeštene i **Dto** klase, **Mapper** klase i **Utility** klase koje se koriste za komunikaciju kroz aplikaciju (Result, Error).



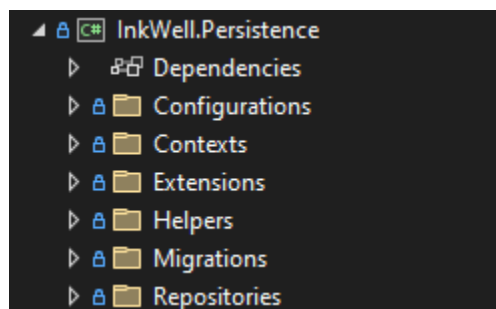
- **Application.Identity sloj** obuhvata logiku koja je vezana za krajnjeg korisnika. Ovde je smeštena logika kreiranja **JWT** tokena, čuvanje state-a korisnika preko **JWT** calim-ova korisnika koji pristupa api-ju, servis za hešovanje i verifikaciju šifre.



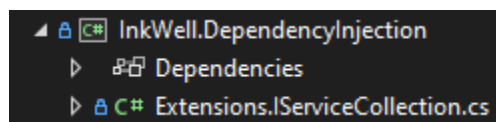
- **Domain sloj** je centar cele aplikacije. Sadrži abstrakcije servisa, entitete aplikacije, abstrakcije **Repositories** i ključne pomoćne klase aplikacije.



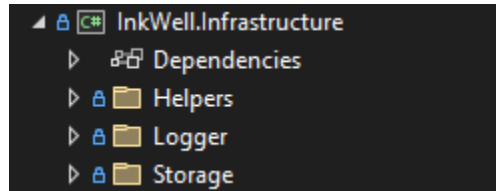
- **Persistence sloj** je zadužen za komunikaciju sa bazom podataka. Sadrži **Context**, **Configurations**, **Migration** klase. Takodje ovde su definisane **Repositories** klase kroz koje **Application** sloj komunicira sa bazom.



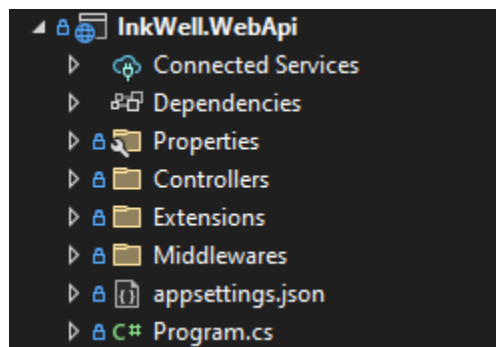
- **DependencyInjection sloj** registruje sve zavisnosti aplikacije.



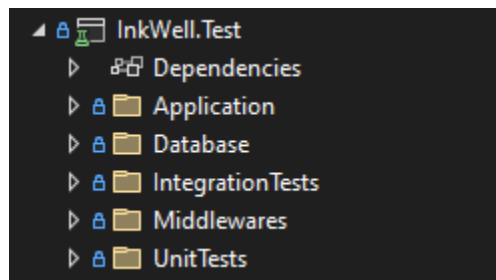
- **Infrastructure sloj** sadrži logiku servisa aplikacije kao što su **Logger** i **Image Cloud Storage**.



- **WebApi sloj** predstavlja sloj koji komunicira sa korisnikom. Obuhvata kontrolere i startup klasu aplikacije.



- **Test sloj** obuhvata testove aplikacije.



2. Pregled funkcionalnosti

- Svi testovi su dostupni u vidu Postman kolekcije u direktorijumu /Other/Postman/
- U nastavku sve funkcionalnosti će se oslanjati na ove testove.

2.1 Authentication and Authorization

2.1.1 Signup

- Registracija prihvata sledece parametre:
 - First Name
 - Last Name
 - Email
 - Username
 - Password
 - Confirm Password
 - Date of Birth
 - Profile Picture (optional)
- Svaki parametar ima svoju validaciju:
 - First Name
 - 3 - 20 karaktera
 - Last Name
 - 3 - 50 karaktera
 - Email
 - Email format
 - Jedinstven email koji nije već korišten
 - Do 60 karaktera
 - Username
 - 4 - 20 karaktera
 - Jedinstven username koji nije već korišten
 - Password

- Minimum 8 karaktera
 - Bar jedno veliko slovo, malo slovo, broj i neki znak
 - Confirm Password
 - Mora da bude isto kao Password parametar
 - Date of Birth
 - Korisnik mora biti stariji od 10 godina
- Uspešna registracija vraća:
 - Username
 - Profile Picture Url
 - Date of Birth
 - Token
- Token poseduje sledeće kleimove (Claims):
 - Issuer
 - Audience
 - Expires
 - Role

2.1.2 Signin

- Logovanje prihvata:
 - Email
 - Password
- Kao i kod registracije prisutni su slični validatori, sem što ovde ispitujemo da li korisnik postoji.
- Postman kolekcija ima već spremno logovanje za svaku rolu u aplikaciji. Ti korisnici su generisani kao seed u konfiguraciji baze podataka.

2.2 Ostalo

- Svaki Get endpoint koji vraća skup podataka poseduje filtriranje putem paginacije, pretrage i sortiranje
- Korisnik može da zaprati druge ulogovane korisnike
- Da lajkuje postove po želji
- Da upvotuje komentare po želji
- Da postavlja komentare i odgovara na komentare
- Upload slika se vrši kod korisnika gde je opcioni parametar i kod postova gde je required
- Slike se postavljaju na cloud storage aplikaciju gde se i čuvaju radi manje obimnosti celokupnog backend-a
- Moguća je pretraga i filtriranje celokupne Audit tabele
- Sve greške u aplikaciji koje za koje nismo spremni se hvataju u Exception Middleware-u gde se ispisuju u konzoli i čuvaju u tabeli Errro logs
- Testovi aplikacije obuhvataju:
 - Arhitektualne testove
 - Imenska pravila command-i i query-ja
 - Test baze podataka
 - Integracioni testovi:
 - Registracija korisnika
 - Exception Middleware test
 - User Context Middleware test
 - Unit test za proveru hashovanja i verifikacije šifre