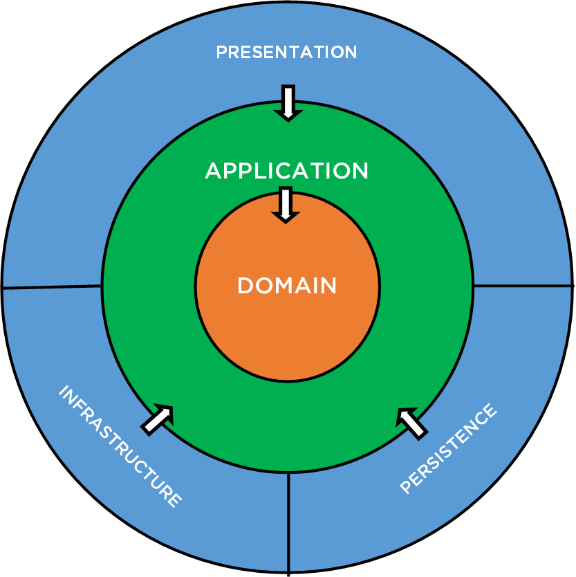
**Newsy**

**Taskovi**

1. **Inicijalni zahtjevi (to do)**
   1. Implementirati backend API koji će podržavati web i mobilnu aplikaciju
   2. Aplikaciju koriste čitaoci I autori koji objavljuju I uređuju članke
   3. Tri funkcionalnosti: Osnovna prijava, Novinski članci, Autori
   4. API da bude napisan u .NET-u
   5. Funkcionalnosti dostupne putem API-ja(REST, GraphQL)
   6. Kreirati kod pogodan za održavanje na stabilnim temeljima
   7. Predvidjeti potrebe frontend-a
   8. Projekat ne bi trebao trajati predugo, ako je nešto predugo, navesti kao buduću implementaciju
   9. Ako se primjeti nešto za doradu, imamo tu mogućnost
2. **v1 – završeni taskovi (done)**
   1. Uvod
      * Implementiran je backend REST API
      * Korištena je InMemory baza zbog lakšeg razvoja, modelovana relaciona baza
   2. Prijava na sistem
      * Implementirana je osnovna prijava na sistem. Samo se autori mogu prijaviti. Da bi vršili operacije nad novinskim člancima moraju biti autorizovani. Za autentifikaciju korišten je Identity Server u kombinaciji sa JWT Bearer tokenom. Metode su izložene Auth endpointom. Implementirane su Register i Login metode.
   3. Novinski članci
      * Implementiran je rad sa novinskim člancima. Podrazumijevamo da čitaoci mogu vidjeti sve članke I detalje o članku. Metode za rad su izložene News endpointom. Pored standardnih CRUD endpointa, tu je i Get endpoint koji vraća listu novinskih članaka korištenjem paginacije.
   4. Autori
      * Autori se mogu prijaviti na sistem. Samo prijavljeni na sistem mogu da rade modifikacije i kreiranje novinskih članaka. Autori su trenutno jedini User-I sistema.
   5. Ostalo
      * Audit model za Novosti, zna se ko ga je kreirao-modifikovao I kada je to uradio
      * ActionFilter za Model State validaciju
      * Global Exception Handler Middleware
      * Korištene Extensions metode
      * DTO za svaki request I response
      * Validacije na modelima
      * CurrentUser servis
      * Data Seed za neke inicijalne podatke, jedan autor I tri njegova oglasa
3. **v2 – buduća poboljšanja (backlog)**
   1. Prijava na sistem
      * Pored autentifikacija, potrebno je dodati I autorizaciju.
      * Omogućiti više rola u sistemu
      * Rola Administrator koja može da upravlja sistemom, briše novosti neprikladnog sadržaja, administrira autore I njihove naloge
      * Rola Čitalac koja može da komentariše,lajkuje,sačuva ili podijeli novinski članak
      * Dodati Logout endpoint potreban za web I mobilnu app
      * Izmjena lozinke
      * Admin panel
      * Claims, Policies
      * Kad je token istekao, novi izdati
   2. Novinski članci
      * Da Text članka bude Html tipa, da se može formatirati, bojite, boldovati,..
      * Dodati nova polja Kategorija, Komentari, Lajkovi, Slika…
      * Proširiti metode za čitanje članaka, GetMyNews, GetTop5News, …
      * Omogućiti pretragu vijesti u kombinaciji sa paginacijom
      * Omogućiti filtriranje vijesti u kombinaciji sa paginacijom
      * Omiljene vijesti, najnovije vijesti, Hronika, Sport, Top5, Najnovije …
      * Tagovi za novinske članke
   3. Autori
      * Autor da vidi sve vijesti
      * Autor da vidi samo svoje vijesti
      * Ocjenjivanje autora
      * Blokiranje autora
   4. Ostalo
      * Dodati verzionisanje API-ja, pogotno u ranim fazama razvoja zbog čestih izmjena
      * Ubaciti testove
      * Dodati logger
      * Dodati domain eventse
      * Omogućiti transakciju u ExceptionHandleru, ako pukne da se stanje konteksta upiše u bazu
      * Sakupljati podatke o korisnicima, šta čitaju najviše, koliko se zadržavaju, ti podaci su ulazni parameter za serviranje reklama
      * Prostor za reklame, kontaktirati nekog oglašivača I nuditi prostor
      * Iako se password čuva heširan i šalje preko https-a. Možda ga heširati odmah na klijentu.
      * Security

**Solucija (projekti)**

1. **Arhitektura solucije**



Slika 1. – Clean architecture

* 1. Uvod
     + Vrijeme je posvećeno više ka samoj arhitekturi, jer je to najbitniji segment održivog koda
     + Izabran je CQRS pattern u kombinaciji sa Mediator-om. Omogućiće stabilan I pogodan kod za održavanje (by loose coupling)
     + Clean arhitecture
  2. Projekti
     + **Domain** – Svi entiteti specifični za domain sloj
     + **Application** – Logika aplikacije. Zavisi od domain sloja, al nema druge zavisnosti. Definisani interfejsi koje implementiraju slojevi koji koriste ovaj projekat. Sve nove funkcionalnosti se prvo dodaju kao interfejs pa onda njegova implementacija
     + **Infrastructure** – Eksterni resursi (baza, kontekst, auth-identity I jwt). Baziran na interfejsima iz app sloja.
     + **API** – interfejs. Zavisi od Application I Infrastructure slojeva. Zavisnost Infrastrukture je preko DI ukljucena u Startup. Tu se radi data seed.

1. **Tehnologije**
   1. ASP.NET Core 5
   2. EntityFrameworkCore 5
   3. InMemory Database
   4. Identity Server(JWT)
   5. MediatR(CQRS)
   6. AutoMapper
   7. Swagger
   8. IOptions
2. **Buduća poboljšanja**
   1. Iskoristiti Fluent validacije
   2. Ubaciti NUnit u Tests I implementirati pokrivenost koda testovima. (Unit, Integration)
   3. Napisati Docker fajl i omogućiti kreiranje Docker Image-a
   4. Napraviti CI/CD
   5. Podići soluciju na neku source kontrolu (GitHub)

**Kredencijali**

Kredencijali korisnika koji je upisan kroz seed:

Username: [marko@mail.com](mailto:marko@mail.com)

Password: Pass#1234