

1. Kontekst i cilj projekta

Newsman je aplikacija za kolaborativno kreiranje novina od strane grupe korisnika sa ciljem smanjenja cenzure prilikom izveštavanja o događajima i novostima. Svaki korisnik može da kreira neku novu vest, pri čemu ostali korisnici mogu da pregledaju datu vest i po potrebi izmene i na taj način dodaju svoje vidjenje događaja. Svaka vest pored teksta koji opisuje događaj može sadržati i dodatne multimedijalne sadržaje koji je bolje opisuju (fotografija, video, zvuk ...).

2. Arhitekturni zahtevi

1. Arhitekturno značajni slučajevi korišćenja (glavni funkcionalni zahtevi)

- Pregled vesti u svakom trenutku
- Kreiranje nove vesti
- Izmena već postojeće vesti
- Mogućnost prijavljivanja za praćenje vesti
- Vest može sadržati bilo koji multimedijalni sadržaj

2. Ne-funkcionalni zahtevi

- Skalabilnost – aplikacija mora biti skalabilna jer je cilj aplikacije kolaborativno kreiranje vesti i samim tim postoji mogućnost velikog broja korisnika sistema
- Proširljivost – aplikacija treba da bude lako proširljiva za dodavanje novih funkcionalnosti, kreiranja novih tipova klijenata kao i novih tipova sadržaja za opisivanje vesti, ocenjivanje verodostojnosti sadržaja vesti, portabilnost na druge platforme
- Dostupnost – sistem mora biti aktivan 24h/7 dana u nedelji i biti dostupan što većem broju korisnika („on the go“ – u svakom trenutku)
- Modifikabilnost – laka izmena funkcionalnosti sistema
- Pouzdanost –
- Upotrebljivost – aplikacija treba da ima intuitivan korisnički interface

3. Tehnička i poslovna ograničenja

Za izradu sistema potrebno je koristiti open-source biblioteke i okruženja.

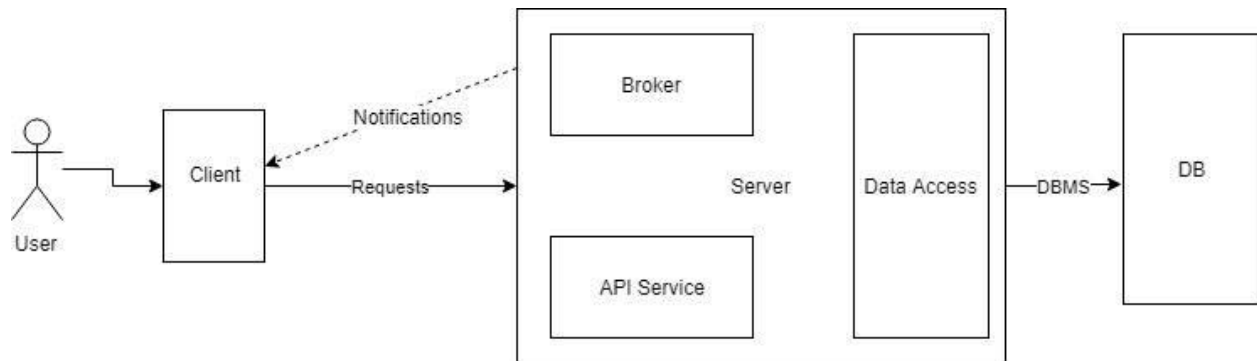
3. Arhitekturni dizajn

1. Arhitekturni obrasci

- Slojevitost arhitektura (Layerd architecture) – sistem je razvijen kao 3-slojna client-server arhitektura radi struktuiranja dizajna i postizanja niskog stepena povezanosti između komponenti (“loosely coupled”)
- MVP (Model-View-Presenter) – ova arhitektura je nametnuta od android-studio framework-a
- Publish-Subscribe – model za asinhronu implicitnu komunikaciju između klijenta i servera, omogućava da se klijenti pretplate na željeni sadržaj i dobijaju obaveštenja kada dođe do izmene tog sadržaja (implementiran od strane Message Broker-a)

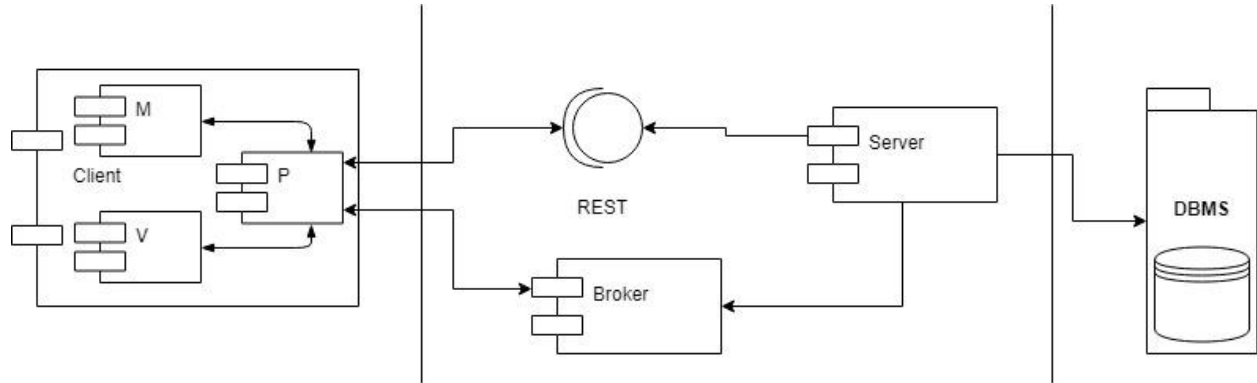
- Skladište (Repository) – svi podatci se čuvaju u bazi podataka

2. Generalna arhitektura

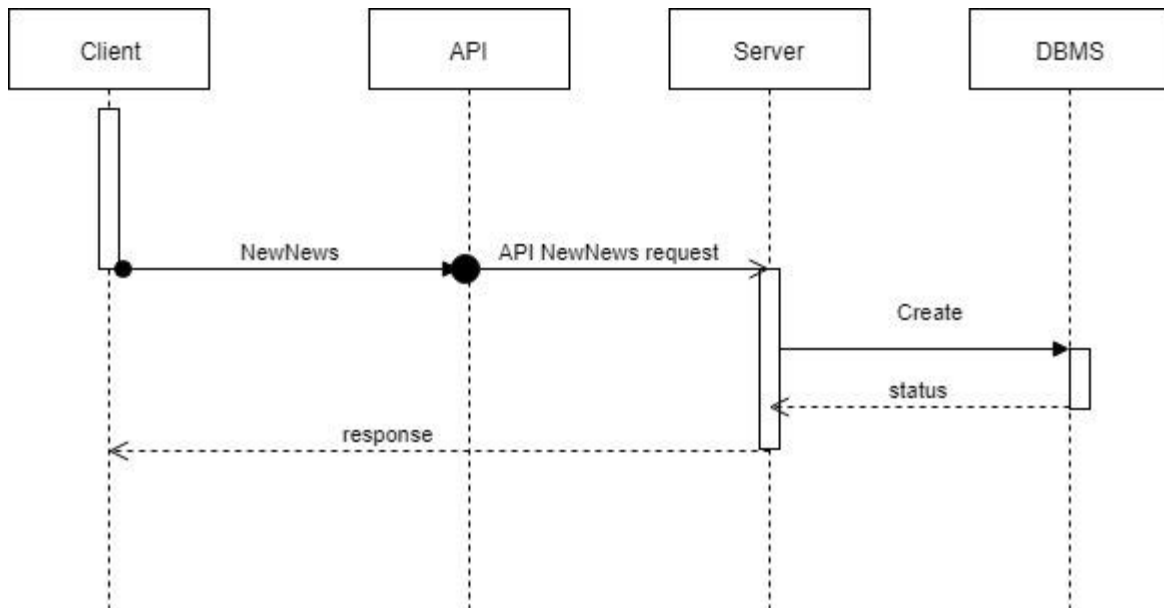


3. Strukturni pogledi

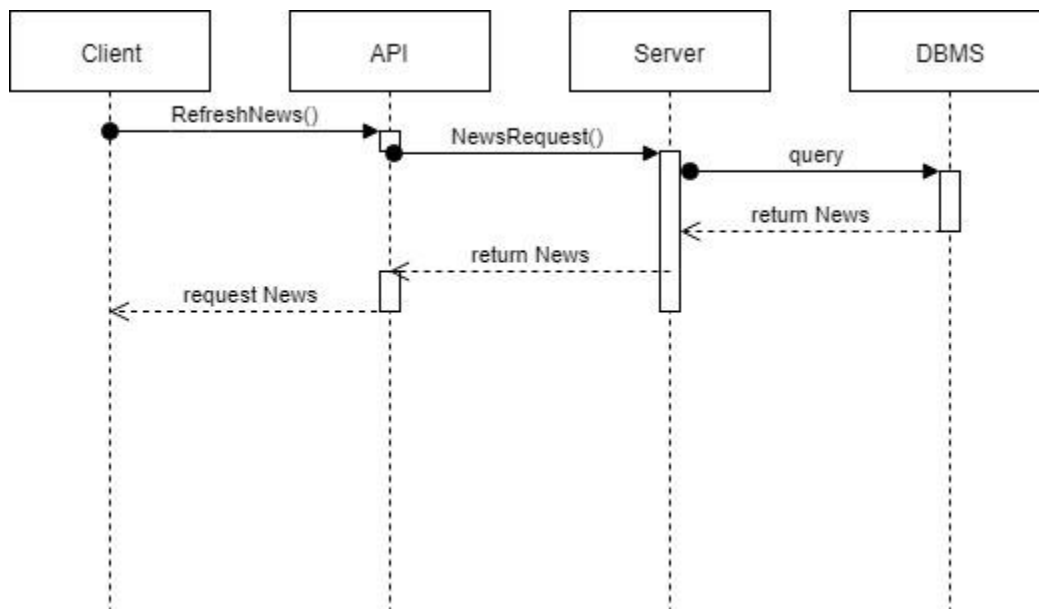
Prikazana je struktura sistema kao i njegove gradivne komponente. Klijentski deo aplikacije je realizovan na osnovu MVP projektnog obrasca. On preko REST servisa sinhrono pribavlja podatke o vestima sa servera. Message Broker služi za asinhronu komunikaciju i obezbeđuje da samo pretplaćeni korisnici dobijaju izmene o vestima. Na Server-u se izvršava API koji komunicira sa bazom podataka.



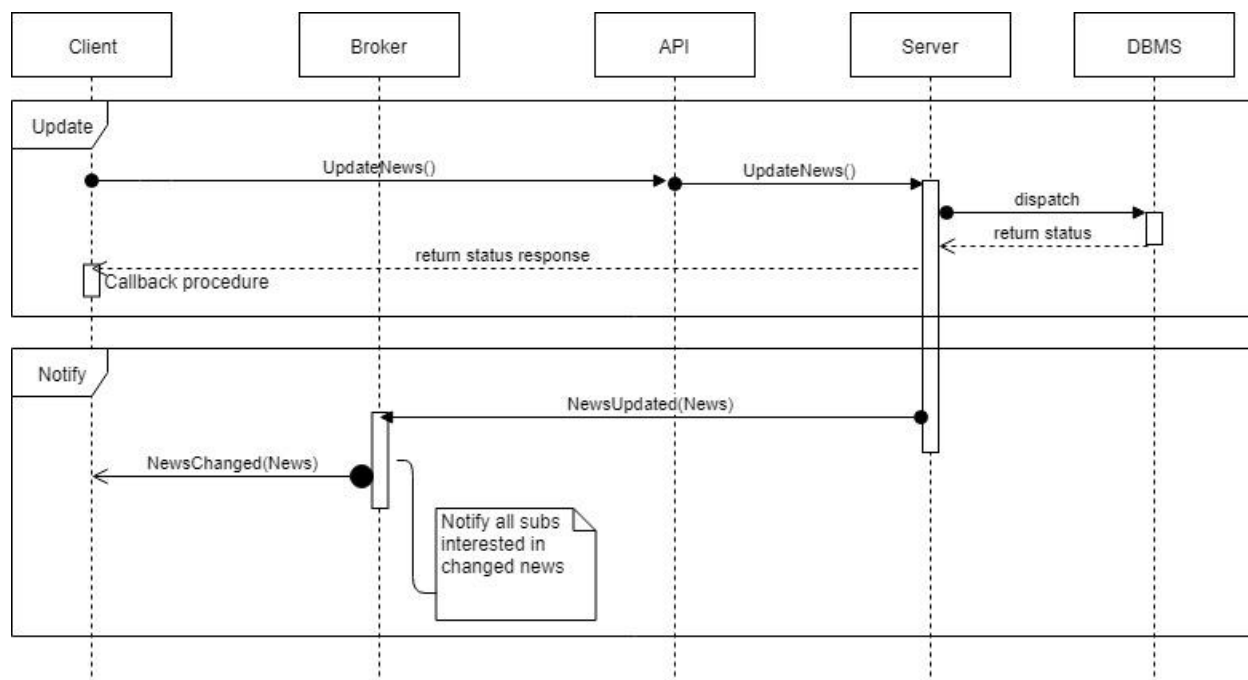
4. Bihevioralni pogledi



1. Prikazuje komunikaciju pri kreiranju nove vesti

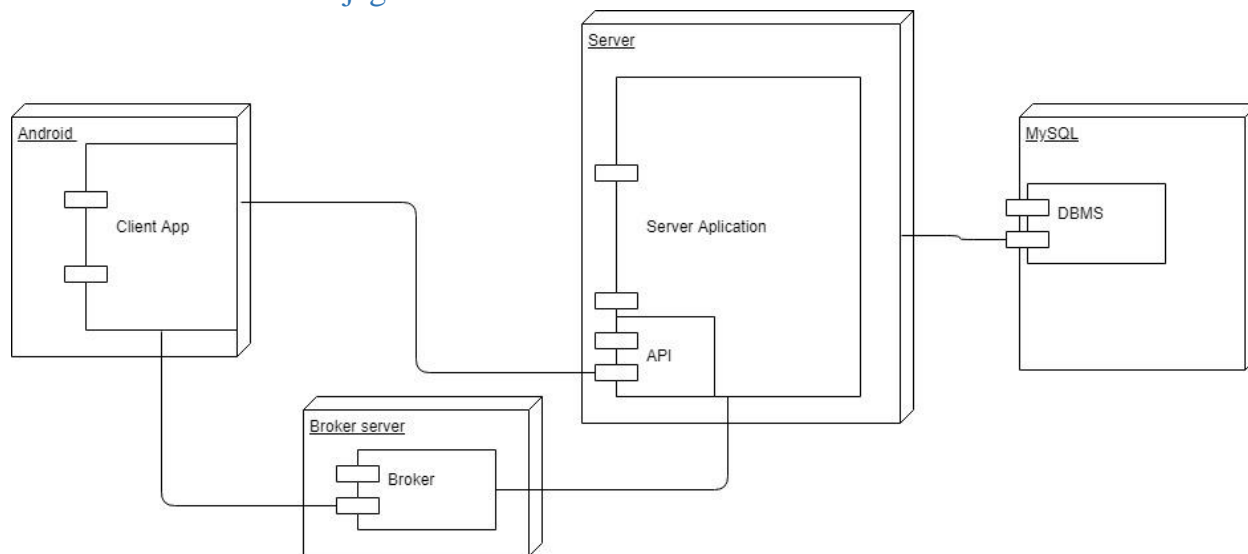


2. Prikazuje pribavljanje vesti od strane klijenta



3. Prikazuje komunikaciju prilikom ažuriranja vec postojeće vesti

5. Alokacioni dijagram



6. Implementaciona pitanja – biblioteke, komponente i okviri (framework) koji će biti korišćeni za implementaciju

Android-studio framework - za izradu klijentske aplikacije

ASP.NET framework - za izradu web API servera

Fluent NHibernate – ORM mapper

MySQL DBMS – baza podataka

RabbitMQ – Message Broker server

RabbitMQ.NET – biblioteka za Message Broker servera

RabbitMQ-Client.jar – biblioteka za Message Broker klijenta