Projektni zadatak iz predmeta

Servisno orijentisane arhitekture i NoSQL baze podataka

Školska 2022/2023. godina

Projekat predstavlja društvenu mrežu za postavljanje i pregled kratkih objava, nalik na Twitter. Aplikacija se razvija prema principima mikroservisne arhitekture i podaci se skladište u različitim tipovima NoSQL baza.

Tipovi korisnika

- **Neautentifikovani korisnik** - Može da se registruje na sistem čime dobija običan ili biznis nalog. Ako poseduje nalog može da se prijavi na sistem.
- **Običan prijavljeni korisnik** - Korisnik koji se prijavio na običan profil. Postavlja, ritvituje i lajkuje objave. Može da prati druge korisnike i pregleda njihove profile. Podešava privatnost profila. Dobija preporuke za nova praćenja.
- **Biznis korisnik** - Korisnik koji se prijavio na biznis profil. Ima pristup svim funkcionalnostima kao i običan korisnik. Pored toga, može da postavlja reklame u vidu objava i da za svaku od njih bira ciljnu grupu kojoj će reklama biti prikazana. Ima uvid u dnevne i mesečne izveštaje o statistikama reklama kao što su broj lajkova, odlazaka na stranicu objave i odlazaka na profil biznis korisnika nakon pregleda reklame.

Delovi sistema

- **Klijentska aplikacija** - Pruža grafički interfejs preko kog korisnici pristupaju funkcionalnostima sistema.
- **Serverska aplikacija** - Mikroservisna aplikacija koja sadrži čitavu poslovnu logiku sistema. Sastoji se iz sledećih servisa:
  - **Auth** - Čuva kredencijale korisnika i njihove uloge u sistemu. Zadužen za proces registracije i prijave korisnika. **Podatke čuva u bazi tipa po izboru.**
  - **Profile** - Sadrži informacije o osnovnim podacima korisnika kao što su ime, prezime, pol, starost, imejl itd. Podržava funkcionalnost izmene privatnosti profila. **Podatke čuva u document bazi.**
  - **Tweet** - Skladišti sve objavljene tweet-ove, retweet-ove, kao i njihove lajkove. Pruža prikaz home i profile feed-a. **Podatke čuva u wide-column bazi.**
  - **Social Graph** - Čuva informacije o međusobnim praćenjima među profilima i o vezama koje postoje između biznis profila i ciljnih grupa za

reklame. Pruža preporuku profila i formira skup profila koji pripadaju nekoj ciljnoj grupi. **Podatke čuva u graf bazi.**

- **Ads** - Beleži koja reklama (objava) treba da se prikaže kojoj ciljnoj grupi korisnika. Formira i čuva statistike o reklamama. **Podatke čuva u bazi tipa po izboru.**

1. Funkcionalni zahtevi

1.1. Registracija profila (neautentifikovani korisnik)

A. Običan profil se registruje tako što se unosi ime, prezime, pol, starost, mesto stanovanja, korisničko ime (mora biti jedinstveno) i lozinka.
B. Biznis profil se registruje tako što se unosi naziv kompanije, imejl, veb sajt, korisničko ime (mora biti jedinstveno) i lozinka.

1.2. Prijava na sistem (neautentifikovani korisnik)

Unosom korisničkog imena i lozinke korisnik može pristupiti svom nalogu i funkcionalnostima koje odgovaraju tipu korisnika kom pripada nalog (običan ili biznis).

1.3. Postavljanje tweet-a (običan i biznis korisnik)

Tweet može da sadrži:
A. Tekstualni sadržaj
B. Sliku
C. Kombinaciju prethodna dva tipa sadržaja

1.4. Lajkovanje tweet-a (običan i biznis korisnik)

Svaki tweet može biti lajkovan maksimalno jednom od strane jednog profila. Ukoliko je korisnik već lajkovao objavu i ponovo pokuša to da odradi, akcija neće imati nikakvog efekta. Moguće je i ukloniti lajk sa objave. Potrebno je za svaki tweet obezbediti prikaz liste korisnika koji su ga lajkovali.

1.5. Pregled profila (običan i biznis korisnik)

Stranica profila treba da sadrži osnovne informacije o profilu i listu tweet-ova koje je taj korisnik objavio.

1.6. Podešavanje privatnosti profila (običan i biznis korisnik)

Profil je nakon kreiranja inicijalno privatan. Sadržaj profila koji su privatni nije dostupan profilima koji ga ne prate, dok je sadržaj javnih profila dostupan svima. Potrebno je omogućiti korisniku da bira da li je njegov profil javan ili privatan.

1.7. Zapraćivanje profila (običan i biznis korisnik)

Ukoliko je profil javan može se automatski zapratiti. Ako je privatan prvo se kreira zahtev za praćenje koji može biti potvrđen ili odbijen. Kada se profil prati, sve njegove objave su vidljive bez obzira na to da li je javan ili privatan.

Na profilu korisnika potrebno je prikazati listu svih korisnika koje profil prati i listu profila koji prate njega.

1.8. Home feed (običan i biznis korisnik)

Home feed treba da sadrži listu tweet-ova tog korisnika i svih ostalih korisnika koje taj profil prati. Tweet-ove treba sortirati opadajuće prema datumu i vremenu objave.

1.9. Retweet (običan i biznis korisnik)

Korisnik može da retweet-uje svaki tweet kom može da pristupi. Kada to odradi, retweet se tretira kao i svaki drugi tweet tog korisnika, ali sa naznakom da je retweet. Mora se navesti ko je korisnik koji ga je originalno objavio. Ukoliko je profil koji je originalno objavio tweet privatan i korisnik kom se prikazuje retweet ne prati taj profil, potrebno je sakriti sadržaj retweet-a, ali ne i ime osobe koja ga je objavila.

1.10. Preporuka profila (običan i biznis korisnik)

Na osnovu profila koje korisnik prati, treba mu dati predlog profila koje trenutno ne prati, ali bi mu mogli biti interesantni (na primer profili koje zapraćeni profili prate). Treba pokriti slučaj kada korisnik ne prati nikoga.

1.11. Postavljanje reklama (biznis korisnik)

Prilikom kreiranja tweet-a korisniku se nudi opcija da odabere da li će se tweet tretirati kao obična objava ili kao reklama. Ukoliko odabere reklamu, treba da unese ciljnu grupu koja se sastoji iz pola, starosnih granica i mesta stanovanja. Svim korisnicima koji pripadaju odabranoj ciljnoj grupi reklama će biti prikazana u okviru home feed-a, nezavisno od toga da li prate profil koji je objavio reklamu ili ne. Ako neki korisnik prati biznis profil reklama će mu biti prikazana na home feed-u nezavisno od toga da li pripadaju ciljnoj grupi te reklame ili ne. Potrebno je naznačiti da je u pitanju reklama.

1.12. Prikaz statistika reklama (biznis korisnik)

Za svaku reklamu treba prikazati dnevni i mesečni izveštaj koji sadrži broj lajkova ostvarenih u tom periodu, broj lajkova koji su uklonjeni, prosečno vreme zadržavanja na stranici detaljnog prikaza tweet-a i broj puta da je neko sa stranice detaljnog prikaza tweet-a prešao na stranicu biznis profila.

2. Nefunkcionalni zahtevi

2.1. API gateway

Predstavlja ulaznu tačku u sistem i sva komunikacija između serverske i klijentske aplikacije obavlja se putem nje. API gateway klijentima nudi REST API za komunikaciju.

2.2. Kontejnerizacija

Sve mikroservise, API Gateway i baze podataka potrebno je pokrenuti kao Docker kontejnere i koristiti Docker Compose alat.

2.3. Circuit breaker

U jednom mikroservisu treba implementirati circuit breaker šablon.

2.4. Tracing

Pomoću Jeager alata implementirati tracing u celoj mikroservisnoj aplikaciji.

2.5. Keširanje

Slike koje se objavljuju kao deo tweet-a treba keširati u Redis-u.

2.6. Saga

Za jednu komandu koja menja podatke u više mikroservisa implementirati sagu. Na odbrani je potrebno demonstrirati uspešan tok, ali i sve verzije neuspešnih tokova.

2.7. Event sourcing i CQRS

Prikupljanje i prikaz statistika o reklamama treba implementirati upotrebom event sourcing i CQRS šablona. Svaka relevantna akcija se čuva kao događaj u event store-u po izboru. Pogledi u vidu dnevnih i mesečnih izveštaja mogu se čuvati u istoj ili u drugačijoj bazi.

Ocenjivanje

- Za ocenu **6** treba implementirati zahteve 1.1, 1.2, 1.3A, 1.4, 1.5, 2.1 i 2.2
- Za ocenu **7** treba implementirati sve navedeno za ocenu 6 i zahteve 1.6, 1.7, 1.8, i 2.3
- Za ocenu **8** treba implementirati sve navedeno za ocenu 7 i zahteve 1.9, 1.10 i 2.4
- Za ocenu **9** treba implementirati sve navedeno za ocenu 8 i zahteve 1.3B, 1.3C i 2.5
- Za ocenu **10** treba implementirati sve navedeno za ocenu 9 i zahteve 1.11 i 2.6
- Za ocenu **10+** treba implementirati sve navedeno za ocenu 10 i zahteve 1.12 i 2.7

Pravila polaganja - Servisno orijentisane arhitekture i NoSQL baze podataka

- Projekat se radi u timovima od po 4 člana. Svi članovi tima moraju slušati vežbe u istom terminu. Ako želite da formirate tim, a članovi su u različitim grupama, potrebno je da izvršite zamenu grupe tako da ste na kraju svi u istoj grupi.
- Za implementaciju serverske i klijentske aplikacije možete koristiti programske jezike i radne okvire po želji. Ukoliko odaberete tehnologije koje se razlikuju od onih koje su pokrivene na vežbama, pomoć koju asistenti mogu pružiti pri rešavanju problema je ograničena.
- Klijentska aplikacija služi da demonstrirate rad sistema i ne ocenjuje se.
- Ako implementirate zahteve navedene za ocenu 10+ oslobođeni ste polaganja teorijskog dela ispita na predmetima.
- Sredinom decembra održaće se kontrolna tačka za koju je neophodno implementirati funkcionalnosti navedene za ocenu 6. Odrađene funkcionalnosti neophodno je demonstrirati kroz klijentsku aplikaciju (nije dovoljna upotreba alata poput Postman-a ili cURL-a).
- Ko izađe na kontrolnu tačku i bude zadovoljan ocenom, ne mora da dolazi na finalnu odbranu, koja će se održati početkom februara.
- Ako ne izađete na kontrolnu tačku, **ocena vam se smanjuje za jednu na finalnoj odbrani** (na primer morate odraditi funkcionalnosti za ocenu 10 kako biste dobili ocenu 9).
- U septembru će biti održan još jedan termin finalne odbrane, na kom vam se **ocena smanjuje za jednu**.

Informaciona bezbednost-zahtevi

1 BEZBEDNOST SISTEMA I ZAŠTITA PODATAKA

Razvoj bezbednog softvera podrazumeva ugrađivanje bezbednosnih kontrola u softver tokom njegovog razvoja, prateći koncept poznat kao *built-in security*. Trošak ugrađivanja bezbednosti na ovaj način je najmanji i, po pravilu, bezbednost je implementirana najkvalitetnije, jer se koncipira bezbedan dizajn koji će kod poštovati, umesto da se bezbednost *ad hoc* prilagođava već napisanom kodu.

Potrebno je implementirati sledeće bezbednosne mehanizme:

1.1 Validaciju podataka

- Sprečiti relevantne *Injection* napade;
- Sprečiti XSS napade;
- Izvršiti validaciju podatka, koristeći kriterijume validacije definisane po najboljim praksama za pisanje bezbednog koda.

1.2 HTTPS komunikaciju

- Potrebno je demonstrirati bezbednu komunikaciju između API gateway-a i klijentske aplikacije

1.2a Demonstracija bezbedne komunikacije između servisa

1.3 Autentifikaciju i kontrolu pristupa

- Omogućiti mehanizme za potvrdu naloga, oporavak lozinke i promenu lozinke;
- Kontrolisanje pristupa *endpoint*-ima po RBAC modelu;
- Kontrola pristupa frontend dela (detalji implementacije prepušteni studentima);
- Testirati i demonstrirati da sve kontrole pristupa rade (pozitivan i negativan ishod);

1.4 Zaštita podataka

Osetljivi podaci sa kojim aplikacija radi treba da budu obezbeđeni u skladištu, u transportu i tokom upotrebe. Identifikovati osetljive podatke, definisati i implementirati prikladne bezbednosne kontrole. Podaci čije skladištenje se ne može izbeći treba da budu šifrovani ili heširani ukoliko je to prikladno. Poruke u internoj komunikaciji treba da imaju očuvanu poverljivost, integritet i neporecivost, kao i da budu zaštićene od *replay* napada.

2 LOGGING I RANJIVE KOMPONENTE

Log zapisi koje generišu aplikacije i operativni sistemi nekog postrojenja su veoma korisni, kako sa aspekta debagovanja problema, tako i za potrebe bezbednosti. Log zapisi predstavljaju osnovni mehanizam za postizanje neporecivosti. Dodatno, kolekcije log zapisa se mogu slati alatima za *monitoring*, poput SIEM alata, čiji zadatak je da prati događaje u sistemu i da okine alarm svaki put kada se sumnjivo ponašanje detektuje. U sklopu odvojene priče, današnji softverski sistemi značajno zavise od komponenti koje nisu dizajnirali i programirali inženjeri originalnog sistema. Od infrastrukture (operativni sistem, baza podataka) do alata (radni okvir, biblioteke), značajan deo koda nije pod našom kontrolom. Međutim, to ne smanjuje našu odgovornost kada nam softver bude eksploatisan zbog ranjivosti u nekoj *third-party* komponenti jer iako nismo pravili tu komponentu, svesno smo je integrisali u naše rešenje.

Potrebno je implementirati logging mehanizam koji ispunjava sledeće zahteve:

1. **Kompletnost** – log zapis mora da sadrži dovoljno informacija da dokaže neporečivost i svaki događaj za koji je neporečivost potrebna treba da bude zabeležen. Dodatno, svaki *security-related* događaj, interesantan za potrebe *monitoring*-a, treba da bude zabeležen.

2. **Pouzdanost** – logging mehanizam treba da bude pouzdan, što podrazumeva dostupnost samog mehanizma (gde je neophodno voditi računa o memorijskom zauzeću log datoteka i napraviti mehanizam za rotaciju logova), kao i integritet log datoteka. Dodatno, dizajnirati kod tako da aplikacija nastavi sa radom u slučaju da logging mehanizam otkaže.

3. **Konciznost** – logging mehanizam treba da proizvodi najmanju količinu zapisa koji su potrebni da ispuni svoju svrhu. Dodatno, optimizovati svaki zapis da sadrži sve informacije, a zauzima najmanju količinu memorije.

2.1 Ranjivosti

Neophodno je:

- Definisati alate koji će se koristiti za proveru različitih skupova komponenti.
- Isproveravati svaku od komponenti i sakupiti listu ranjivosti.
- Analizirati ozbiljnost ranjivosti i mogućnost eksploatacije.
- Definisati i izvršiti strategiju za razrešenje mogućih rizika.
- Kreirati izveštaj neke vrste, koji će istaći temeljnost analize i krajnje rezultate. Format i tačan sadržaj je proizvoljan.

Zahteve navedene iznad je moguće formalno ispuniti bez istraživanja i mnogo truda, no to rešenje neće biti kvalitetno. Kvalitet je upravo ono što se ocenjuje, i da bi se date stavke

ispunile neophodno je razmotriti savete i najbolje prakse koje možete pronaći online, poput onih navedenih u OWASP ASVS standardu.

Napomena:

- Obratiti pažnju na *best practice* konfiguraciju bezbednosnih funkcija koje koristite.
- Potrebno je implementirati funkcionalnosti tek toliko da se podrži smislena demonstracija bezbednosnih kontrola.
- Pojedine tačke je moguće rešiti uz pomoć tehnologije za implementaciju softvera (jezika, radnog okvira) ili alata – ovo je dozvoljeno, no neophodno je razumeti kako tehnologija rešava problem i o čemu treba voditi računa da se pružena bezbednosna kontrola „ne pokvari“.
- Prilikom istraživanja i implementacije kontrola, neophodno je voditi računa o bezbednoj konfiguraciji kontrole – skup parametara koje kontrola ima i njihova *best practice* vrednost.

Ocenjivanje IB

Za ocenu 6 neophodno je implementirati sledeće zahteve:

- **1.1**
- **1.3**
- **1.4**

Za ocenu 7 neophodno je implementirati sve za ocenu 6 i sledeće zahteve:

- **1.2**
- **integrisati postojeći sistem sa nekim od alata za statičku analizu koda (SonarCloud, SonarCube, …). Analizirati i rešiti ranjivosti u skladu sa preporukama alata.**

Za ocenu 8 neophodno je implementirati sve za ocenu 7 i sledeće zahteve:

- **2**

Za ocenu 9 neophodno je implementirati sve za ocenu 8 i sledeće zahteve:

- **2.1**

Za ocenu 10 neophodno je implementirati sve za ocenu 9 i sledeće zahteve:

- **1.2a**

Kontrolne tačke IB

U toku semestra rad na projektu biće proveravan 2 puta. Prva kontrolna tačka će se održati sredinom semestra (okvirno kraj novembra), a druga kontrolna tačka ujedno će predstavljati i finalnu odbranu projekta (okvirno januar). Od ukupno 35 bodova koliko nosi KT1, na finalnoj odbrani-KT2 može se nadoknaditi maksimalno 10 bodova.

KT1 (35 bodova)

- 1.1 za implementirani deo projekta na drugim predmetima
- 1.4 za implementirani deo projekta na drugim predmetima
- 1.3
- 1.2

KT2 (36 bodova)

- 1.1 za ceo projekat
- 1.4 za ceo projekat
- 1.2a
- 2
- 2.1

Usmeni ispit IB

- Usmeni ispit **je obavezan za ocene 9 i 10**.
- Usmeni ispit **nije obavezan za ocene 6,7,8**.
- Ko na predispitim obavezama ispuni zahteve za ocene 6 i 7 može izaći na usmeni maksimalno za ocenu 8.
- Nakon položenih predispitnih obaveza, ko ispunjava uslov za usmeni ispit i želi da izađe na isti neophodno je da prijavi ispit i prijavi se email na stojkovm@uns.ac.rs da dobije svoj termin.

Dodatni rokovi IB

Pored odbrane na KT2 u januaru, organizovaće se još dva termina odbrane predispitnih obaveza, u junu i septembru. Odbrane će biti održane maksimalno za ocenu 6, bez mogućnosti izlaska na usmeni ispit.