

## Projektni zadatak iz predmeta

### **Računarstvo u oblaku**

*Školska 2022/2023. godina*

Projekat predstavlja web aplikaciju za skladištenje i rukovanje fotografijama i video zapisima, uz mogućnost kreiranja foto-albuma, nalik galeriji mobilnih uređaja. Aplikacija se razvija upotrebom **AWS cloud servisa** prateći **serverless** arhitekturu.

### **Tipovi korisnika**

- **Neautentifikovani korisnik** - Može da se registruje na sistem. Ako poseduje nalog može da se prijavi na sistem.
- **Autentifikovani korisnik** - Korisnik koji se uspešno prijavio na sistem. Ima mogućnost postavljanja novog sadržaja na sistem, kao i izmenu i brisanje postojećeg sadržaja. Rukovodi svojim foto-albumima. Može da omogući pristup svom sadržaju drugom korisniku. Može da ukloni dati pristup. Može da lokalno preuzme okačeni sadržaj.

### **Delovi sistema**

- **Klijentska aplikacija** - Pruža grafički interfejs preko kog korisnici pristupaju funkcionalnostima sistema.
- **Serverska aplikacija** - Serverless aplikacija koja sadrži čitavu poslovnu logiku sistema. Potrebno je osmisлити arhitekturu koja će ispuniti sve funkcionalne i nefunkcionalne zahteve sistema. Potrebno je iskoristiti adekvatne AWS servise.

## **1. Funkcionalni zahtevi**

### **1.1. Registracija korisnika (neautentifikovani korisnik)**

Korisnik se registruje tako što unosi ime, prezime, datum rođenja, korisničko ime (mora biti jedinstveno), email i lozinku.

### **1.2. Prijava na sistem (neautentifikovani korisnik)**

Unosom korisničkog imena i lozinke korisnik može pristupiti svom nalogu i svim funkcionalnostima sistema.

### **1.3. Postavljanje sadržaja (prijavljeni korisnik)**

Sadržaj može biti:

- A. Fotografija
- B. Video zapis

C. Audio zapis

D. Fajl (pdf, html, docx,...)

Pored samog sadržaja, potrebno je skladištiti i prateće informacije o sadržaju. Minimalno je potrebno podržati: naziv datoteke, tip datoteke, veličina datoteke, vreme nastanka i vreme poslednje izmene. Pored navedenih, potrebno je podržati definisanje dodatnih informacija od strane korisnika, kao što je opis, tagovi, i slično.

**Naziv sadržaja mora biti jedinstven u okviru foto-albuma.**

**1.4. Pristup sadržaju (prijavljeni korisnik)**

Korisnik može pristupati sadržaju koji je okačio. Prikaz treba da bude sortiran u redosledu dodavanja sadržaja, počevši od najnovijeg.

**1.5. Izmena sadržaja (prijavljeni korisnik)**

Korisnik može menjati samo sadržaj čiji je vlasnik. Korisnik može menjati sadržaj ili dodatne informacije vezane za njega.

**1.6. Brisanje sadržaja (prijavljeni korisnik)**

Korisnik može brisati samo sadržaj čiji je vlasnik. Prilikom brisanja, brišu se i dodatne informacije vezane za sadržaj.

**1.7. Kreiranje foto-albuma (prijavljeni korisnik)**

Inicijalno postoji jedan foto-album u kom se skladišti sav sadržaj. Korisnik ima mogućnost kreiranja dodatnih foto-albuma. Korisnik može premestiti postojeći sadržaj u foto-album ili dodavati novi direktno u njega.

**1.8. Brisanje foto-albuma (prijavljeni korisnik)**

Korisnik može obrisati bilo koji foto-album sem inicijalnog. Brisanjem foto-albuma se brišu i svi sadržaji (i dodatnih informacija o sadržaju) koji su deo istog.

**1.9. Deljenje sadržaja sa drugim korisnicima (prijavljeni korisnik)**

Korisnik ima mogućnost deljenja sadržaja sa drugim korisnikom. Deljenje je moguće na nivou sadržaja i na nivou foto-albuma. Korisnik kojem je dodeljeno pravo ima read-only pristup sadržaju (može da ga pregleda i preuzme, ali ne da ga menja i briše).

**1.10. Ukidanje deljenja sadržaja sa drugim korisnicima (prijavljeni korisnik)**

Korisnik ima mogućnost ukidanja prava pristupa sadržaju koji je podelio sa drugim korisnikom.

**1.11. Preuzimanje sadržaja (prijavljeni korisnik)**

Korisnik ima mogućnost preuzimanja sadržaja kom ima pristup (kao vlasnik ili kroz deljenje sadržaja).

## **2. Nefunkcionalni zahtevi**

### **2.1. Serverless arhitektura**

Potrebno je modelovati sistem u skladu sa serverless arhitekturom. Neophodno je iskoristiti adekvatne AWS servise za podršku iste.

### **2.2. Odvojeno skladištenje sadržaja i dodatnih informacija**

Sadržaj i dodatnih informacija je potrebno čuvati u adekvatnim skladištima.

### **2.3. Infrastructure as Code**

Instanciranje i konfiguraciju potrebnih servisa neophodno je odraditi upotrebom nekog od Infrastructure as Code alata.

### **2.4. Asinhrona komunikacija**

Komunikacija između komponenti sistema treba realizovati asinhrono.

### **2.5. 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.

### **2.6. Sistem notifikacija**

Korisnik dobija notifikaciju kada se sadržaj uspešno skladišti ili izmeni u okviru sistema. Notifikacija treba da sadrži osnovne informacije o sadržaju. Takođe, potrebno je poslati notifikaciju i u slučaju brisanja sadržaja.

### **2.7. Proces registracije člana porodice**

Potrebno je implementirati proces registracije člana porodice upotrebom adekvatnog AWS servisa. Kako bi se korisnik registrovao, neophodno je da bude "pozvan" od strane postojećeg korisnika. Novi korisnik prilikom registracije, pored osnovnih podataka, navodi i korisničko ime korisnika od koga je dobio poziv. Registracija se sastoji iz koraka validacije unetih podataka, kreiranja korisnika, slanja notifikacije postojećem korisniku i verifikacije sa njegove strane. Nakon uspešne verifikacije, novi korisnik automatski dobija pristup svom sadržaju korisnika koji je uputio poziv. Kao i kod regularnog deljenja sadržaja, vlasnik naknadno može ukinuti deljenje sadržaja i članu porodice.

### **2.8. Problem konzistencije**

Kako se sadržaj i dodatne informacije čuvaju u dva različita skladišta u okviru sistema, potrebno je rešiti problem nekonzistentnog stanja do kog može doći. Neophodno je implementirati rešenje i dokumentovati kako može doći do nekonzistentnog stanja i kako sistem rešava isto.

## Polaganje predmeta

### Kontrolna tačka 1 - obavezna (9 bodova)

**Termin: 3.5. - 6.5.**

Potrebno je implementirati zahteve **1.1, 1.2 i 1.3** uz poštovanje zahteva **2.2**. Neophodno je implementirati API serversku aplikaciju koja će upotrebom AWS SDK komunicirati sa odabranim skladištima. Nije neophodno pratiti serverless arhitekturu i iskoristiti AWS Lambda funkcije.

- Studenti koji ne izađu na kontrolnu tačku gube bodove bez mogućnosti nadoknade.
- Studenti koji izađu na kontrolnu tačku bez pokušaja implementacije zahteva, nemaju pravo nadoknade bodova.
- Studenti koji urade funkcionalnosti, ali predmetni asistent istakne potrebu za izmenom, imaju mogućnost da nadoknade bodove na drugoj kontrolnoj tački ili finalnoj odbrani.

### Kontrolna tačka 2 - konsultativna

**Termin: 7.6. - 10.6.**

Potrebno je pokazati inicijalnu verziju potpune arhitekture rešenja prateći serverless principe i savladane AWS servise.

### Finalna odbrana - obavezna

**Termin: 26.6. - 1.7. ili 3.7 - 8.7.**

**Napomena:** datumi nisu fiksni i može doći do izmene u toku semestra.

## Osnovne informacije

- Projekat se implementira u **timovima od 3 ili 4 člana**.
- Studenti imaju mogućnost odabira bilo kojih tehnologija i programskih jezika za implementaciju projekta. Jedino ograničenje je upotreba [LocalStack](#) tehnologije za simulaciju AWS servisa ili upotreba pravih AWS servisa.
- Dodatni rokovi za finalnu odbranu projekta biće organizovani u zavisnosti od broja studenata i dogovora sa istima. Maksimalan broj bodova koji se može osvojiti će biti skaliran u svakom narednom roku faktorom **0.9**.
- Pitanja vezana za specifikaciju projekta i način polaganja predmeta možete postaviti predmetnom asistentu. Česta pitanja i odgovore će biti objavljena na dnu specifikacije. **Proveravajte spisak pitanja i odgovora.**

## Ocenjivanje

1. Za sve ocene neophodno je kreirati **prateću klijentsku aplikaciju**. Klijentska aplikacija nije fokus predmeta i neće se uzimati u obzir za ocenjivanje.
2. Za ocenu **6** treba implementirati zahteve 1.1, 1.2, 1.3 i 2.1, 2.2.
3. Za ocenu **7** treba implementirati sve navedeno za ocenu 6 i zahteve 1.4, 1.5, 1.6 i 2.3, 2.4.
4. Za ocenu **8** treba implementirati sve navedeno za ocenu 7 i zahteve 1.7, 1.8 i 2.5, 2.6.
5. Za ocenu **9** treba implementirati sve navedeno za ocenu 8 i zahteve 1.9, 1.10 i 2.7.
6. Za ocenu **10** treba implementirati sve navedeno za ocenu 9 i zahteve 1.11 i 2.8.

### Bodovanje po zahtevima

Zahtevi - ocena 6	Broj bodova
1.1. Registracija	3
1.2. Prijava	1
1.3. Postavljanje sadržaja	4
2.1. Serverless arhitektura	1
2.2. Odvojeno skladište sadržaja i dodatnih informacija	1

Zahtevi - ocena 7	Broj bodova
1.4. Pristup sadržaju	1
1.5. Izmena sadržaja	1
1.6. Brisanje sadržaja	1
2.3. Infrastructure as Code	4
2.4. Asinhrona komunikacija	3

Zahtevi - ocena 8	Broj bodova
1.7. Kreiranje foto-albuma	2
1.8. Brisanje foto-albuma	1
2.5. API Gateway	3
2.6. Sistem notifikacija	4

<b>Zahtevi - ocena 9</b>	<b>Broj bodova</b>
1.9. Deljenje sadržaja	2
1.10. Ukidanje deljenja sadržaja	1
2.7. Proces registracije člana porodice	7

<b>Zahtevi - ocena 10</b>	<b>Broj bodova</b>
1.11. Preuzimanje sadržaja	1
2.8. Problem konzistencije	9

## FAQ

1.