

# Online videoteka

**Predmet:** Dizajn i arhitektura softverskih sistema

**Mentor:** dr Amila Akagić

**Studenti:**

Halilović Delila

Halilović Sumejja

Komić Merima

<b>Uvod</b>	<b>3</b>
Svrha	3
Opseg	4
Definicije, akronimi i skraćenice	4
Reference	4
Pregled	4
<b>Reprezentacija arhitekture</b>	<b>5</b>
<b>Ciljevi i ograničenja</b>	<b>6</b>
Sigurnost	6
Zaštita sistema	6
Autorizacija i autentifikacija	6
Perzistencija	6
Pouzdanost/dostupnost	6
Performanse	7
<b>Use-case</b>	<b>7</b>
Akteri	8
Realizacija Use-case dijagrama	8
<b>Logički pogled</b>	<b>9</b>
Pregled	9
<b>Procesni pogled</b>	<b>9</b>
<b>Modul dekompozicije</b>	<b>10</b>
<b>Podatkovni pogled</b>	<b>10</b>
<b>Razvojni pogled</b>	<b>11</b>
<b>Veličina i performanse</b>	<b>12</b>
<b>Atributi kvalitete softvera</b>	<b>12</b>
Performanse i dostupnost	12
Iskoristivost	12
Sigurnost	13
Izmjenjivost	13
<b>Problemi</b>	<b>14</b>

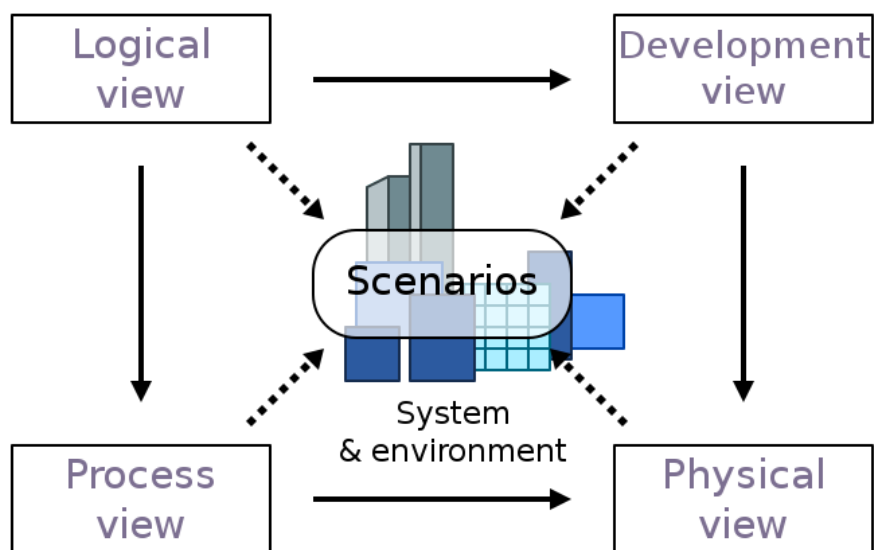
# 1. Uvod

Ovaj dokument objašnjava cjelokupnu arhitekturu Sistema za online videoteku. Objašnjava mogućnosti korisnika da kreira vlastiti profil te da koristi sistem. Dokument pruža opis ciljeva arhitekture na visokom nivou, use-case dijagrame, te alate koji najbolje postižu njihovu izgradnju.

## 1.1. Svrha

Glavnu svrhu ovog dokumenta predstavlja detaljan opis funkcionalnosti softverskog rješenja za online videoteku. Predstavljani su arhitekturni pogledi kako bi se prikazali različiti aspekti sistema.

Kako bi se softver što preciznije opisao, struktura ovog dokumenta je bazirana na Krutchten "4+1 model view" arhitekturi [KRU41].



Slika 1. "4+1" model view arhitektura

Ovakav prikaz pruža interesnim grupama (stakeholderi) sve ono što im je potrebno u softverskoj arhitekturi.

## 1.2. Opseg

Dokument sadrži softversko rješenje za pregled video sadržaja, online, kojeg razvija organizacija HAKO. Ovo softversko rješenje bi trebalo da olakša i unaprijedi pregled video sadržaja.

Dokument ne sadrži detaljan opis implementacije sistema jer je svrha dokumenta da ponudi okvirni pregled svih komponenti sistema i posluži kao globalni opis interakcije među pojedničnim dijelovima sistema, ali i sistema kao jedne cjeline.

## 1.3. Definicije, akronimi i skraćenice

<b>WWW</b>	World Wide Web
<b>Apache</b>	Web Server
<b>ASP.NET</b>	Microsoft web platform
<b>HTTP</b>	Hypertext Transfer Protocol
<b>feedback</b>	Povratna informacija o određenoj stavki sistema

## 1.4. Reference

[PP]: [Project Proposal](#)

[SPMP]: [Software Project Management Plan](#)

[SRS]: Software Requirements Specification

[KRU41]: The “4+1” view model of software architecture, Philippe Kruchten, November 1995,

<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

## 1.5. Pregled

Software Architecture Document sadrži sljedeća poglavlja:

Poglavlje 2: opisuje svrhu

Poglavlje 3: opisuje arhitekture sistema

Poglavlje 4: opisuje funkcionalne zahtjeve i njihov uticaj na arhitekturu

Poglavlje 5: opisuje najvažnije use-case realizacije.

Poglavlje 6: opisuje dizajn s aspekta konkurentnosti

Poglavlje 7: opisuje deploy sistema.

Poglavlje 8: opisuje sve znančajne elemente

Poglavlje 9 opisuje performanse i ograničenja

Poglavlje 10: opisuje attribute kvaliteta (QA)

## 2. Reprezentacija arhitekture

Ovaj dokument koristi poglede definisane "4+1" modelom [KRU41], ali uz upotrebu RUP konvencije imenovanja. Pogledi koji se koriste u dokumentu su:

### Use Case pogled

**Učesnici:** svi stakeholderi, uključujući i krajnje korisnike

**Oblast:** opisuje scenarije i/ili use caseove koji opisuju glavne funkcionalnosti sistema. Opisuje aktere i use case dijagrame za sistem, ovaj pogled prikazuje potrebe korisnika, tok podataka kao i detaljna ograničenja.

**Povezani artefakti:** Use-case modeli, Use-case dokumenti

### Logički pogled

**Učesnici:** Dizajneri

**Oblast:** Funkcionalni zahtjevi (opisuju konkretne funkcionalnosti koje sistem treba da omogući)

**Povezani artefakti:** Dizajn model

### Procesni pogled

**Učesnici:** Integratori

**Oblast:** Nefunkcionalni zahtjevi (karakteristika/osobina koju sistem treba da posjeduje)

**Povezani artefakti:** nije specificirano

### Module decomposition pogled

**Učesnici:** Programeri

**Oblast:** Softverske komponente: opisuju module i podsisteme aplikacije

**Povezani artifakti:** Implementacijski modul,komponente

### **Podatkovni pogled**

**Učesnici:** stručnjaci podataka, bazni administratori

**Oblast:** Dosljednost: opisuje arhitekturne značajne elemente u modelu podataka

**Povezani artifakti:** model podataka.

### **Razvojni pogled**

**Učesnici:** Deploy menadžeri

**Oblast:** Razvojni pogled opisuje kako su dijelovi sistema organizirani u module i komponente. Veoma su korisni za upravljanje nivoima unutar arhitekture sistema.

**Povezani artifakti:** Razvojni model.

## **3. Ciljevi i ograničenja**

### **3.1. Sigurnost**

Sigurnost je proces, a ne krajnje stanje. Svaki neovlašten pristup te svi pokušaja upada na sistem će biti zabilježeni u log fajlu koje administrator analizira na kraju svakog dana te ima mogućnost da brzo zaštiti sistem od takvih pokušaja.

#### **3.1.1. Zaštita sistema**

Zaštita sistema će biti postignuta sljedećim načinima:

- Enkripcija podataka
- Provjera integriteta
- Validacija podataka
- Promjene nad podacima se bilježe u bazi podataka

#### **3.1.2. Autorizacija i autentifikacija**

Na sistemu najveće permisije ima administrator sistema. Svi korisnici će vršiti prijavu koristeći username i password. Zavisno od tipa korisnika, imamo i različite mogućnosti rada na sistemu.

### 3.2. Perzistencija

Podrška postoji 24 sata dnevno svim danima, pomoću mail-a ili putem telefona čije troškove pokriva naručilac sistema. Ako se dogodi neka greška, naša ekipa će izaći na teren i pokušati u što kraćem periodu otkloniti problem. Server baze podataka kao najvažniji dio informacijskog sistema, će posjedovati backup na nekoj drugoj lokaciji . Backup ce se vršiti bar jednom mjesečno.

### 3.3. Pouzdanost/dostupnost

Sistem treba da omogući korisniku stalno slanje upita prema serveru te će biti dostupan korisnicima svaki dan, 0-24h. U nekim slučajevima se dešava da server padne ili je nedostupan na neko vrijeme. Međutim pošto se ovaj problem rješava u relativno kratkom vremenskom periodu, ovo ne bi trebalo da predstavlja veći problem za korisnika. Nakon pojave greške na sistemu, dozvoljeno je sistem bude nedostupan najviše 1 dan.

Srednje vrijeme između dva otkaza ne bi trebao biti čest slučaj i ne može se precizno odrediti dok sistem ne bude u produkciji više od godinu dana.

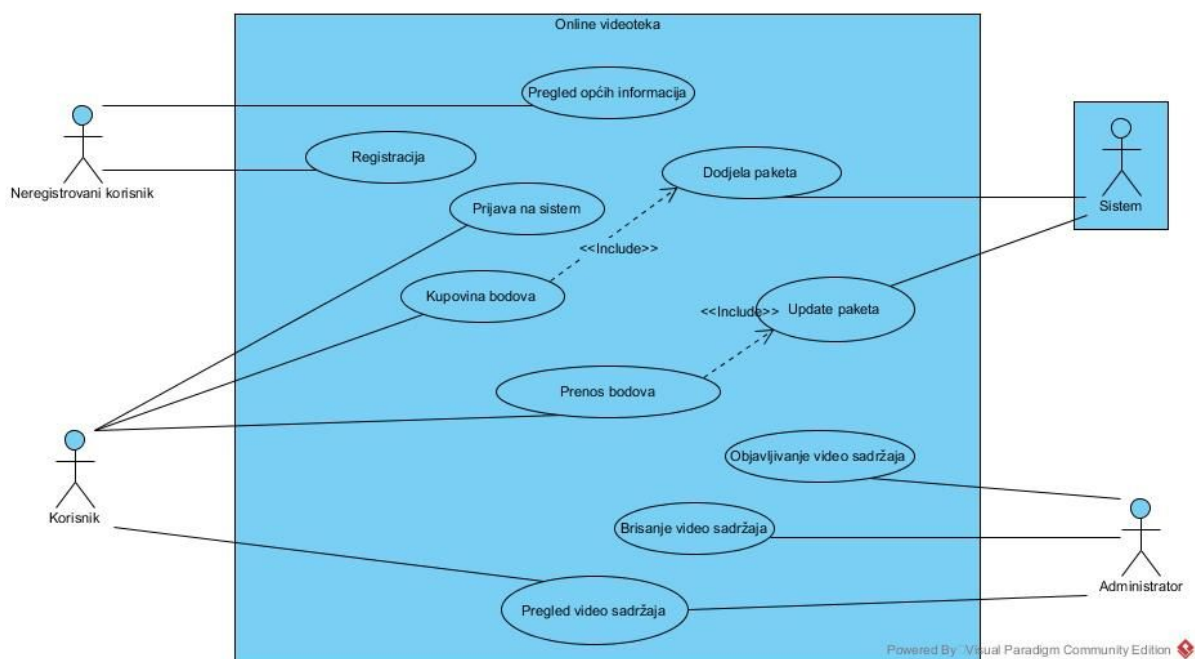
### 3.4. Performanse

Da bi imali zadovoljne korisnike potrebno je voditi računa o performansama sistema. Sistem treba da omogući brz pristup i pretragu za korisnike sistema.

## 4. Use-case

Lista use-case slučajeva koji predstavljaju glavne funkcionalnosti sistema:

- Registracija
- Prijava na sistem
- Odjava sa sistema
- Kupovina bodova
- Prenos bodova
- Pregled video sadržaja
- Dodjela paketa
- Update paketa
- Dodavanje video sadržaja
- Brisanje video sadržaja



Slika 2. Use-case dijagram

### 4.1. Akteri

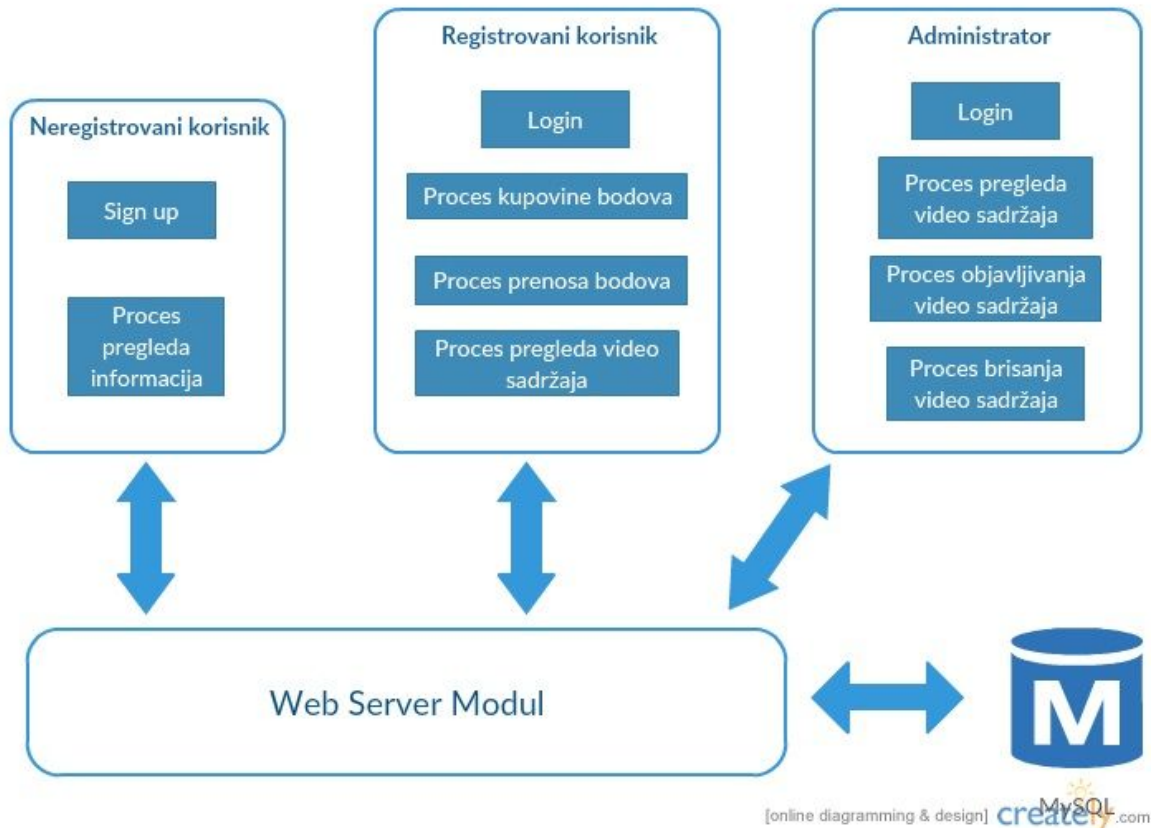
Kao što je navedeno na dijagramu ispod, korisnik može imati tri uloge:

1. **Administrator** ima mogućnosti pregleda, objavljivanja, brisanja video sadržaja



2. **Neregistrovani korisnik** ima mogućnosti registracije i pregled općih informacija o sistemu
3. **Registrovani korisnik** ima mogućnosti prijave na sistem, odjave sa sistema, kupovinu i prenos bodova, kao i pregled video sadržaja
4. **Sistem** je četvrti tip aktera ima mogućnosti dodjele paketa i update paketa

## 4.2. Realizacija Use-case dijagrama



Slika 3. Realizacija Use-case dijagrama

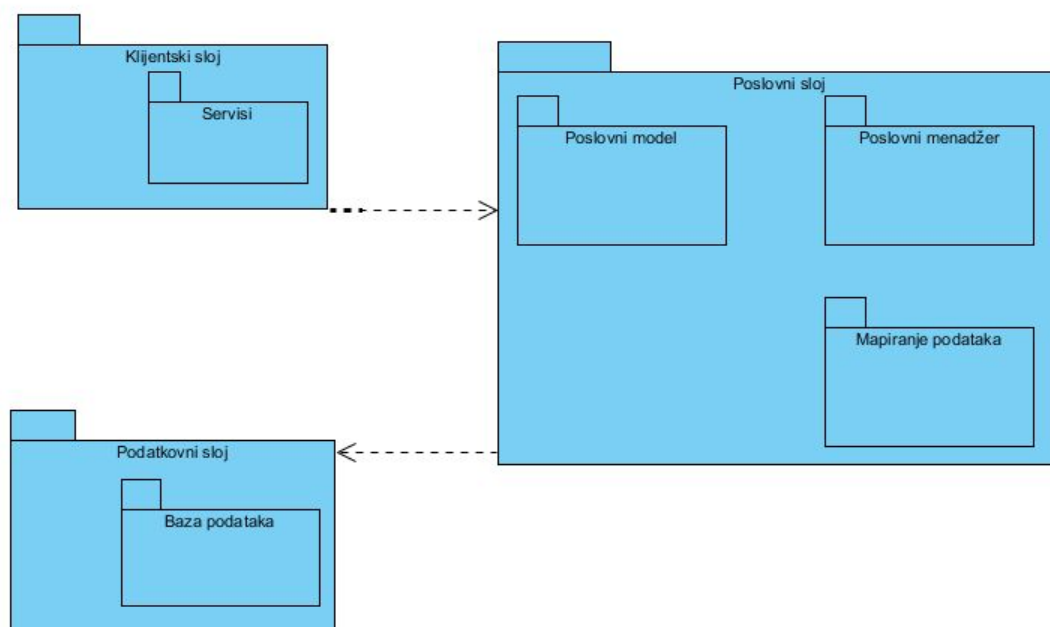
## 5. Logički pogled

### 5.1. Pregled

Lakše razumijevanje sistema postiže se logičkim raslojavanjem dijelova sistema. U logičkom smislu, sistem se temelji na troslojnoj arhitekturi. Slojevi koji čine arhitekturu su:

- klijentski sloj (web browser pomoću kojih klijenti pretražuju sadržaj videoteke)
- poslovni sloj (web server koji opslužuje zahtjeve klijenata i vrši eventualne backupe)
- podatkovni sloj (podaci smješteni u bazi podataka)

Svaki od ovih slojeva reliziran je zasebnim modulima. Ovaj tip raslojavanja odvaja odgovornosti pojedinih dijelova. Nezavisnost odgovornosti među dijelovima sistema olakšava razvoj i održavanje sistema.

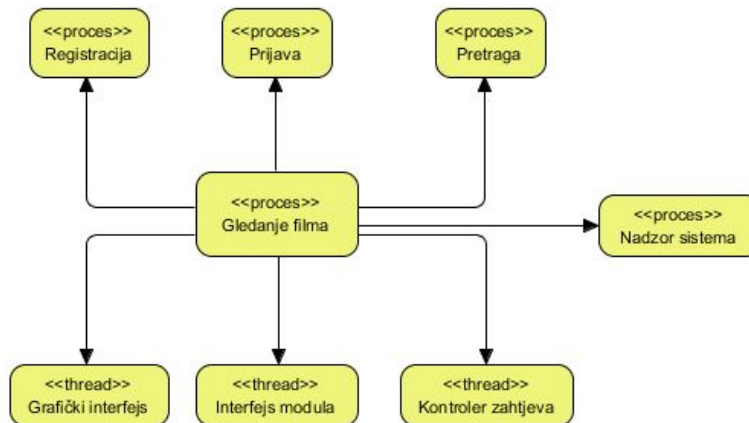


## 6. Procesni pogled

Zadaci (proces i niti) koji se obavljaju unutar sistema:

- Registrovanje
- Prijava
- Pregled dostupnih sadržaja
- Gledanje određenog sadržaja
- Prijava problema (putem e-maila ili telefona)
- Uplata

- Podrška korisnicima u vidu ispravljanja grešaka
- Nadzor sistema



Navedeni procesi se odvijaju između krajnjeg korisnika i administratora servera. Proces koji se odvijaju između administratora servera i baze podataka su: kreiranje, čitanje, ažuriranje i brisanje podataka. U ovom sistemu nisu uzete u obzir različite vrste konkurentnosti i asinhroni događaji koji bi mogli utjecati na izvršenje procesa i niti. Više procesa se izvršava paralelno.

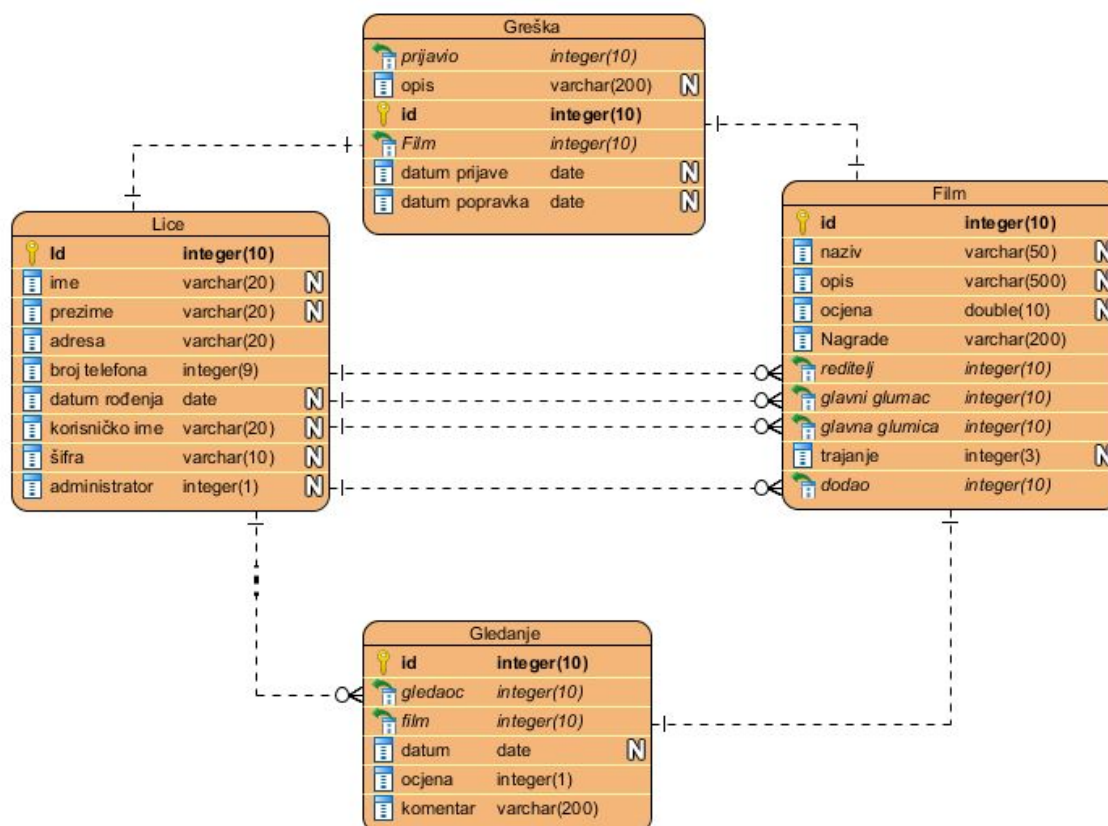
## 7. Modul dekompozicije

Modul dekompozicije vrši rasčlanjivanje sistema na sastavne dijelove. Time se utvrđuje njegova struktura a sam sistem može se lakše i detaljnije upoznati. Prema tome, u analizi strukture sistema primjenjuje se metoda dekompozicije. Grafički prikaz metode dekompozicije jest dijagram dekompozicije.

Metodom dekompozicije sistem se dekomponira na svoje sastavne dijelove (podsisteme). Dekompozicija se provodi od općeg k pojedinačnom što daje hijerarhijski prikaz (opis) sistema. Dekomponirati se mogu: sistemi, podaci, dokumenti, programski moduli ....

## 8. Podatkovni pogled

Podatkovni pogled sistema realizuje se kroz ERD dijagram baze podataka. Sistem ordinira video uredcima pa baza podataka sadrži veliku količinu podataka. Sigurnost i kvaliteta podataka je važna kod podataka o korisnicima, dok se za podatke o videima ne vrše stalne provjere kvalitete i backupi. Pristup podacima za krajnje korisnike se realizuje pomoću web aplikacije i njima je omogućeno samo čitanje podataka, dok se izmjena podataka vrši direktno na glavnom serveru baze podataka. Fleksibilnost u radu baze podataka ostvarena je neznatnom interakcijom krajnjih korisnika sa bazom podataka.



## 9. Razvojni pogled

Sistem online biblioteke je dizajniran kao web aplikacija i podrazumijeva klijent-server organizaciju sistema. Aplikacija se podiže unutar globalne domene .net i nalazi se na adresi videotekahk.net. Razvoj sistema provodi se u tri različita nivoa. Razvoj obuhvata: klijentsku aplikaciju na webu, web server i baza podataka.

Klijentska aplikacija se pokreće korištenjem konkretnog web browsera (npr. google chrome). Komunikacija između klijenta i servera se realizuje korištenjem https protokola (zbog sigurnosti podataka registrovanih korisnika), a olakšana je korištenjem prilagođenog grafičkog interfejsa.

Web server realiziran je korištenjem Stu03 servera. ovaj server predložen je i odobren od strane stručnjaka iz oblasti računarstva sa Univerziteta u Sarajevu. Razvoj aplikacije na webu je podređeno uslovima sigurnosti koje nalaže ovaj server. Obezbijeđena sigurnost je najznačajnija osobina ovog tipa servera. konekcija na bazu podataka se vrši posredstvom localhosta. Pristup fajlovima za čitanje i pisanje je u skladu sa polisijem servera.

Baza podataka koja se koristi je Oracle8i. Server baze podataka sješten je skupa sa web serverom na Stu03 serveru i realizovan korištenjem Windows operativnog sistema. Korištenje i kreiranje tabela unutar baze se odvija isključivo u okviru mogućnosti koje nudi Oracle, odnosno ne postoje eksterne datoteke sa SQL ili MySQL fajlovima. Nije moguće dodavanje ili brisanje tabela unutar baze. Korisnici sa dozvolom pristupa na server mogu dodavati, čitati, mijenjati ili brisati podatke iz baze, kao i kreirati upite nad bazom.

Struktura organizacije fajlova odrađena je uvođenjem foldera. Svi sistemski fajlovi su podijeljeni u tri osnovna foldera: folder za izvorne datoteke, folder za datoteke baze

podataka i folder za testove. Struktura izvornog direktorija sastoji se iz dva foldera koji sadrže serverske te klijentske fajlove.

## 10. Veličina i performanse

### Veličina

- Potrebno je omogućiti svim registrovanim korisnicima da istovremeno koriste sistem. Očekuje se 500 registrovanih korisnika tokom prve godine upotrebe sistema.
- Omogućiti online čuvanje velikih količina podataka (video sadržaja) - 1T

### Performanse

- Odziv sistema treba biti manji od 5 sekundi.

## 11. Atributi kvalitete softvera

### 11.1. Performanse i dostupnost

S obzirom da su performanse i dostupnost usko vezani atributi kvalitete softvera, navodimo ih u istom poglavlju. Rad web aplikacije bi trebao biti neprekidan, dakle aplikacija treba biti dostupna 24/7, međutim zbog određenih grešaka koje se mogu javiti, očekivana dostupnost je 93.2%.

Svi korisnici bi trebali biti u mogućnosti prijaviti se i gledati video-sadržaje bilo kada. Na sistem može biti prijavljeno najviše 300 korisnika u jednom trenutku. Pokretanje videa treba biti u okviru vremena od 30 sekundi, a dobavljanje novih kilobajta videa sa servera se obavlja svakih 10 milisekundi.

Kako bi se postigla očekivna dostupnost, za detekciju grešaka, lakše nadgledanje sistem i otkrivanje DOS napada koristit će se monitoring i heartbeat.

Za oporavak sistema od greške, koristit će se Aktivna redundancija, što znači da će sve komponente u sistemu u isto vrijeme reagovati na događaje. Sve komponente će biti u istom stanju. Ovo stanje zavisi od reagovanja jedne komponente, koja je obično prva komponenta koja reaguje na grešku.

### 11.2. Iskoristivost

Sistem treba biti jednostavan za upotrebu te treba obavljati zadatke na što jednostavniji i brži način. Kako bi se poboljšala iskoristivost sistema izbjegavaju se situacije u kojima korisnik može izazvati kvar te se smanjuje interakcija korisnika sa serverom.

Taktike koje će omogućiti navedeno se sastoje iz:

- razdvajanja interfejsa od ostatka aplikacije - s obzirom da se očekuje česte promjene interfejsa i tokom i nakon razvoja sistema poželjno je držati kod koji se tiče interfejsa odvojeno kako bi bilo lakše locirati promjene

- podrške korisničke inicijative (cancel, undo, aggregate) - sistem će konstantno osluškivati korisničku inicijativu te ukoliko korisnik zatraži prekid neke akcije (cancel) omogućit će to u što kraćem roku
- podrške systemske inicijative - na osnovu raznih modela (korisničkog, systemskog) sistem će pratiti akcije korisnika kako bi mogao dati odgovarajuću povratnu informaciju (eng. feedback) korisniku

### 11.3. Sigurnost

Kako bi se zaštitili video sadržaji te obezbijedila sigurnost pružanje otpora će se vršiti pomoću autorizacije i autentifikacije. Otkrivanje napada se može vršiti praćenjem saobraćaja, a za oporavak od napada neophodno je provjeriti podatke te ih vratiti na stanje prije napada. Jedan od načina jeste držanje kopija informacija o sistemu te ažuriranje tih informacija u zadanom vremenskom roku.

### 11.4. Izmjenjivost

Izmjena hardveskih i softverskih dijelova se vrši samo ako se profit povećava za najmanje 5%. Vremenski rok za veće izmjene je mjesec dana, a za manje je jedan dan. Sa ciljem olakšavanja izmjena sistema, tokom rada bit će rađen refactoring te će se na taj način promjene locirati brže. Također, od velikog je značaja odgoditi što je više moguće vrijeme vezivanja vrijednosti varijable za tu varijablu jer često problem bude upravo u dodanim vrijednostima.

Kako bi se postigli navedeni atributi kvalitete softvera koristi će se Client-Server šablon. On je poželjanz auptrebu u situacijama gdje postoji nekoliko dijeljenih resursa i servisa kojima pristupa velik broj distribuiranih klijenata. Kako obezbijediti kvalitet usluge i uspostaviti kontrolu pristupa? Rješenje je omogućiti klijentima slanje zahtjeva serverima, koji imaju nekoliko servisa u ponudi. Nije još odlučeno da li će biti samo jedan centralni server ili nekoliko distribuiranih servera.

S obzirom da Client-Server šablon možda neće biti dovoljan da pokrije sve probleme, koristit će se i Broker šablon. Njega koristimo kako bismo napravili strukturu distribuiranog softvera tako da korisnici servisa ne znaju prirodu i lokaciju proizvođača usluge. Dakle, Broker šablon pravi podjelu između korisnika usluge(klijenta) i proizvođača usluge (servera) tako što dodaje dodatni sloj koji se naziva broker. Kada klijent treba uslugu, šalje zahtjev brokeru preko ulaznog interfejsa. Broker proslijeđuje zahtjev serveru koji ga zatim procesira.

## 12. Problemi

Problemi koji se mogu javiti su vezani za objavljivanje i spremanje video sadržaja - da li je moguće čuvati video sadržaje u binarnom obliku i koja je gornja granica za veličinu videa? Kako omogućiti korisnicima brzo učitavanje filma, a pri tome treba imati u vidu da se radi o velikim količinama podataka.