

# Digitalizacija arhivskih filmova

## Opis projektnog zadatka

Projekt **Digitalizacija arhivskih filmova** ima za cilj razviti aplikaciju koja podržava proces digitalizacije starih filmskih zapisa pohranjenih u fizičkim arhivima. Kroz ovaj projekt omogućit će se efikasno upravljanje, organizacija i evidencija digitalizacije, čime se olakšava očuvanje kulturne baštine i dostupnost ovih materijala budućim generacijama.

---

## 1. Uvod

U današnje vrijeme, uz napredak multimedijske tehnologije, većina audio i video materijala pohranjuje se u digitalnom formatu. Međutim, značajan dio kulturne baštine, poput starih arhivskih filmova, još uvijek postoji samo u fizičkom obliku. Ovi zapisi, koji sadrže povijesne i kulturno značajne informacije, suočeni su s propadanjem zbog fizičkog starenja materijala te ograničenih resursa za održavanje uređaja potrebnih za reprodukciju.

Digitalizacija arhivskih filmova predstavlja ključno rješenje za očuvanje ovih zapisa. Digitalni formati omogućuju dugotrajno očuvanje podataka i olakšavaju pristup korisnicima koji istražuju kulturnu baštinu. Proces digitalizacije uključuje upravljanje fizičkim materijalima, očitavanje meta podataka i optimizaciju rada digitalizacijskih uređaja s ograničenjem trajanja postupka.

Projekt “Digitalizacija arhivskih filmova” razvija aplikaciju koja omogućava djelatnicima jednostavan unos i upravljanje podacima o arhivskim zapisima, uključujući očitavanje putem bar koda i evidenciju meta podataka. Aplikacija će također omogućiti voditeljima digitalizacije pregled aktivnosti, optimizaciju procesa i generiranje potrebne dokumentacije za evidenciju premještanja materijala.

---

## 2. Specifični ciljevi projekta

- **Digitalizacija filmova u segmentima od maksimalno 45 minuta:** S obzirom na ograničenja uređaja, aplikacija će grupirati filmske zapise u segmente kako bi osigurala optimalnu upotrebu uređaja i kontinuirani proces digitalizacije.

- **Osiguravanje trajne pohrane digitaliziranih zapisa:** Po završetku digitalizacije, svi podaci bit će trajno pohranjeni u digitalnom formatu, čime se smanjuje potreba za fizičkim skladištenjem i povećava sigurnost i dostupnost materijala.
  - **Učinkovito upravljanje meta podacima:** Kroz integraciju s bazama podataka i podršku za XML format, aplikacija omogućuje unos, pohranu i izmjenu metan podataka.
  - **Optimizacija i automatizacija procesa digitalizacije:** Aplikacija će omogućiti automatsko prepoznavanje i učitavanje podataka putem bar koda, ručno unošenje podataka te generiranje PDF dokumentacije za evidenciju iznošenja i vraćanja materijala iz skladišta.
  - **Praćenje statusa i aktivnosti tijekom digitalizacije:** Voditelji digitalizacije imat će uvid u status svakog arhivskog materijala, uključujući statističke podatke o količini digitaliziranih materijala, čime se osigurava praćenje aktivnosti i analiza učinkovitosti procesa.
- 

### 3. Opis problema

Projekt “Digitalizacija arhivskih filmova” suočava se s nekoliko izazova:

1. **Različita duljina filmskih traka:** Arhivski materijali često dolaze u segmentima različitog trajanja, od nekoliko minuta do nekoliko sati. Uređaj za digitalizaciju ima ograničenje od maksimalno 45 minuta po sesiji, što znači da dulje filmske trake treba podijeliti, dok kraće trake treba grupirati u optimalne segmente. Ovaj problem zahtijeva precizno planiranje i optimizaciju kako bi se maksimalno iskoristili kapaciteti digitalizacijskog uređaja, bez prekoračenja limita od 45 minuta po sesiji.
  2. **Upravljanje meta podacima:** Ako arhivske filmske kutije imaju bar kod, onda se meta podaci učitavaju iz arhivske baze podataka prema jedinstvenom bar kodu. Ako se nije moguće povezati na arhivsku bazu podataka, meta podaci učitavaju se iz XML datoteke. Ako arhivska filmska kutija nema bar kod, podaci se unose ručno preko zadane forme u sučelju.
  3. **Praćenje fizičkog kretanja arhivskih materijala:** Prilikom iznošenja materijala na digitalizaciju i njegovog vraćanja u skladište, potrebno je zabilježiti točan datum, vrijeme i odgovornu osobu za svako premještanje. Sustav za digitalizaciju mora generirati PDF dokumente s evidencijom iznošenja i vraćanja materijala, koji će biti ovjereni potpisom djelatnika, čime se osigurava jednoznačna evidencija svakog fizičkog premještanja.
-

## 4. Potencijalna korist projekta

- **Djelatnici:** Dobivaju podršku kroz sustav koji olakšava unos i upravljanje podacima o arhivskim materijalima. Aplikacija im omogućuje automatsko očitavanje bar kodova i preuzimanje meta podataka, čime se smanjuje potreba za ručnim unosom i minimizira mogućnost pogrešaka.
  - **Voditelji digitalizacije:** Pristup cjelokupnom pregledu i praćenju procesa digitalizacije kroz aplikaciju, uključujući nadzor nad aktivnostima djelatnika i statusom digitaliziranih materijala.
  - **Šira javnost:** Digitalizirani zapisi omogućit će lakši pristup povijesnim materijalima, osiguravajući očuvanje kulturnih artefakata za obrazovne, istraživačke i kulturne svrhe.
- 

## 5. Postojeća slična rješenja

1. **Archivematica:** Alat za digitalizaciju i dugotrajnu pohranu različitih tipova dokumenata i multimedije. Fokusiran je na pohranu digitalnih zapisa, ali ne podržava specifične zahtjeve poput fizičkog praćenja materijala.
  2. **Kultura:** Platforma za distribuciju digitalnih video zapisa. Nudi alate za pohranu digitalnih medijskih datoteka, ali nije prilagođena za rad s meta podacima iz XML-a.
  3. **Preservica:** Rješenje za dugotrajno čuvanje kulturnih resursa koje nudi automatizirano pohranjivanje i migraciju formata. Međutim, ne podržava digitalizaciju filmova s ograničenim trajanjem sesije.
  4. **MediaInfo:** Alat za analizu i pregled meta podataka, bez mogućnosti upravljanja fizičkim arhivima ili podrške za digitalizaciju u segmentima od 45 minuta.
- 

## 6. Opseg projektnog zadatka

Projekt obuhvaća razvoj cjelovitog softverskog rješenja za digitalizaciju filmskih zapisa, uključujući:

1. **Unos i pregled podataka o arhivskim filmovima:** Omogućavanje automatskog očitavanja bar koda za preuzimanje meta podataka i ručni unos kada bar kod ne postoji.
2. **Grupiranje filmova za digitalizaciju:** Grupiranje filmskih zapisa u segmente od najviše 45 minuta.
3. **Generiranje i evidencija PDF dokumenata:** Za evidenciju iznošenja i vraćanja materijala.

4. **Praćenje statusa materijala:** Evidencija statusa tijekom postupka digitalizacije i povratka.
  5. **Upravljanje korisničkim ulogama:** Definiranje različitih uloga s posebnim pravima pristupa.
  6. **Statistički podatci i izvještavanje:** Generiranje statističkih podataka za nadzor korisničke aktivnosti i procesa digitalizacije.
  7. **Prilagodljivost i podrška za različite formate:** Kompatibilnost s formatima (XML, PDF) i responzivni dizajn.
  8. **Sigurnost i pohrana podataka:** Autentifikacija korisnika putem OAuth2 za siguran pristup.
- 

## 7. Analiza korisnika i prilagodljivost

### Skup korisnika

- **Djelatnici:** Unos podataka i upravljanje procesom digitalizacije.
- **Voditelji digitalizacije:** Uređivanje unesenih podataka te nadzor procesa digitalizacije.
- **Administratori sustava:** Upravljanje korisnicima i kontrola pristupa.

### Prilagodljivost rješenja

Aplikacija je responzivna, optimizirana za različite uređaje i mrežne uvjete, osiguravajući nesmetan rad na računalima, tabletima i mobilnim uređajima.

---

## 8. Tehnološka rješenja

Projekt koristi razne tehnologije kako bi osigurao optimalnu funkcionalnost, sigurnost i prilagodljivost aplikacije. Ključna tehnološka rješenja uključuju:

- **API za povezivanje s bazom podataka:** Aplikacija koristi REST API za efikasnu komunikaciju s bazom podataka, omogućujući dohvat i ažuriranje podataka o filmskim zapisima kada je dostupan mrežni pristup.
- **Autentifikacija korisnika putem OAuth2:** OAuth2 protokol osigurava siguran način prijave i kontrolu pristupa podacima. Na ovaj način aplikacija štiti privatnost i sigurnost korisničkih podataka, upravlja pristupima i definira razine ovlasti za različite korisničke uloge.

- **Podrška za različite formate datoteka (PDF, XML):** Aplikacija omogućuje rad s više formata datoteka, što olakšava prijenos podataka i generiranje izvještaja:
  - **PDF:** Za generiranje dokumentacije o iznošenju i vraćanju materijala te statističke izvještaje o aktivnostima zaposlenika i procesu digitalizacije.
  - **XML:** Za prijenos podataka kada baza podataka nije dostupna.
- **Responzivni dizajn:** Aplikacija je razvijena koristeći principe responzivnog dizajna, prilagođavajući sučelje različitim uređajima i veličinama zaslona. Ovo osigurava da aplikacija može raditi na računalima, tabletima i mobilnim uređajima, čime se omogućuje rad u različitim okruženjima i uvjetima.

Ova tehnološka rješenja osiguravaju fleksibilnost, sigurnost i interoperabilnost aplikacije, prilagođene potrebama različitih korisnika.

---

## 9. Dokumentacija projekta

Projekt uključuje sveobuhvatnu dokumentaciju koja pokriva tehničke i korisničke aspekte aplikacije:

- **Tehnička dokumentacija:** Sadrži detalje o arhitekturi aplikacije, tehnologijama, bazama podataka, API specifikacijama, sigurnosnim protokolima i tehničkim zahtjevima.
- **Korisnička dokumentacija:** Namijenjena krajnjim korisnicima, sadrži upute za korištenje aplikacije, upravljanje podacima i izvođenje osnovnih operacija.
- **Upute za održavanje:** Dokumentacija za administratore i tim za podršku s uputama za ažuriranja, backup podataka, upravljanje korisničkim pristupima i rješavanje potencijalnih problema.

Detaljna dokumentacija osigurava jednostavnije održavanje aplikacije i lakše uvođenje novih korisnika i članova tima.

## 10. Arhitektura sustava

Projekt je strukturiran u modularnoj arhitekturi kako bi se olakšalo održavanje, skaliranje i daljnji razvoj aplikacije. Ključne komponente arhitekture uključuju:

- **Frontend komponenta:** Zadužena za korisničko sučelje, omogućava interakciju korisnika s aplikacijom putem responzivnog dizajna.
- **Backend komponenta:** Sadrži logiku poslovnog sustava, upravlja podacima i omogućava pristup bazi podataka putem REST API-ja.
- **Radna baza podataka:** Centralno pohranjuje sve relevantne podatke o arhivskim materijalima, korisnicima i meta podacima.
- **Arhivska baza podataka:** Sadrži meta podatke o filmovima koji se učitavaju prema jedinstvenom bar kodu.
- **Autentifikacijski modul:** Upravljanje korisničkim pristupima i provjera autentičnosti putem OAuth2 protokola.

Ova arhitektura osigurava pouzdanost, sigurnost i jednostavnost proširenja sustava prema potrebama korisnika.

---

## Zaključak

Projekt “Digitalizacija arhivskih filmova” predstavlja značajan korak u očuvanju kulturne baštine i modernizaciji rada arhivskih ustanova. Razvoj aplikacije prilagođene digitalizaciji filmskih zapisa omogućit će učinkovitije upravljanje arhivskim materijalima, dugotrajnu pohranu vrijednih povijesnih zapisa i njihovu veću dostupnost istraživačima i široj javnosti.

Ova digitalna transformacija arhivskih procesa donosi višestruke koristi korisnicima, ustanovama i istraživačima, omogućujući brži pristup informacijama, smanjenje troškova fizičkog skladištenja te dugoročnu zaštitu podataka. Uz mogućnost daljnjeg proširenja funkcionalnosti, poput napredne analitike, projekt ima potencijal postati ključni resurs za digitalizaciju i očuvanje kulturnih resursa.

## Funkcionalni zahtjevi

ID	Opis	Prioritet	Izvor	Kriteriji prihvatanja
FZ-001	Aplikacija podržava čitanje bar koda.	Visok	Funkcionalan	Djelatnik može uspješno skenirati bar kod.
FZ-002	Na temelju skeniranog bar koda dohvaćaju se meta podatci o filmu iz arhivske baze podataka.	Visok	Funkcionalan	Arhivska baza podataka sadrži skenirani bar kod.
FZ-003	Ako bar kod nije dostupan u arhivskoj bazi podataka, aplikacija podržava učitavanje podataka iz XML datoteke.	Visok	Funkcionalan	Djelatnik može učitati meta podatke iz XML datoteke.
FZ-004	Ako bar kod ne postoji, aplikacija omogućuje ručni unos meta podataka o filmu.	Visok	Funkcionalan	Djelatnik može ručno unijeti podatke.
FZ-005	Nakon učitavanja podataka, omogućena je ručna izmjena i nadopuna podataka.	Srednji	Funkcionalan	Voditelj može uspješno izmijeniti podatke koje je djelatnik prethodno unio.
FZ-006	Aplikacija optimalno grupira unesene filmove te dinamički stvara grupe koje djelatnik može pregledati i poslati na digitalizaciju.	Visok	Funkcionalan	Djelatnik unosi podatke o filmovima koje aplikacija dinamički grupira prije slanja na digitalizaciju.
FZ-007	Aplikacija vodi statističku evidenciju o grupama koje su trenutno na digitalizaciji i koje su završene te generira PDF dokument.	Srednji	Funkcionalan	Voditelj ima pristup statistici postupka digitalizacije.
FZ-008	Aplikacija vodi statističku evidenciju o radu djelatnika te generira PDF dokument.	Srednji	Funkcionalan	Voditelj ima pristup statistici koliko je koji djelatnik/voditelj iznio filmskih grupa na digitalizaciju i koliko ih je vratio s postupka digitalizacije.

ID za- ht- jeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
FZ-009	Djelatnici, voditelji i administratori imaju različite korisničke uloge s definiranim pravima pristupa.	Visok	Funkcionalni	Sustav omogućava različite razine pristupa za djelatnike, voditelje digitalizacije i administratore prema njihovoj ulozi.
FZ-010	Aplikacija generira PDF dokument prilikom iznošenja i vraćanja filmskih grupa s postupka digitalizacije.	Srednji	Funkcionalni	Djelatnik može preuzeti PDF dokument radi evidencijskog potpisa.
FZ-011	Aplikacija omogućuje prikaz svih korisnika aplikacije te generira PDF dokument.	Srednji	Funkcionalni	Administrator ima uvid u sve korisnike aplikacije.
FZ-012	Administrator ima mogućnost brisanja i promjene uloge korisnika aplikacije.	Visok	Funkcionalni	Sustav omogućava korisnicima aplikacije čijim podacima može upravljati administrator.
FZ-013	Autentifikacija korisnika omogućena je putem OAuth2 servisa pri registraciji i prijavi.	Visok	Funkcionalni	Korisnik se može prijaviti i registrirati uz sigurnosne provjere putem OAuth2 autentifikacijskog protokola.
FZ-014	Aplikacija je prilagođena radu na različitim uređajima.	Niski	Funkcionalni	Korisničko sučelje prilagođeno za rad na različitim veličinama zaslona i uređajima.

## Ostali zahtjevi

## Nefunkcionalni zahtjevi



ID	Opis	Prioritet	Izvor	Kriteriji prihvatanja
NFZ-001	Aplikacija mora grupirati arhivske snimke do maksimalnog trajanja od 45 minuta (naglasak na vremensko ograničenje).	Visok	Nefunkcionalni	Aplikacija uspješno grupira arhivske snimke unutar zadanog vremenskog ograničenja.
NFZ-002	Vrijeme odziva pri učitavanju podataka iz XML datoteka ili baze podataka mora biti unutar 5 sekundi.	Visok	Nefunkcionalni	Vrijeme učitavanja podataka iz XML-a ili baze uvijek je ispod 5 sekundi tijekom testiranja sustava.
NFZ-003	Aplikacija osigurava trajno čuvanje svih unesenih podataka te otpornost na padove ili gubitak veze s bazom.	Visok	Nefunkcionalni	Kršnja pada sustava ili prekida veze, svi podaci su trajno sačuvani i integritet podataka nije ugrožen.
NFZ-004	Aplikacija je spremna na povećanje broja korisnika, količine podataka i proširenje funkcionalnosti.	Srednji	Nefunkcionalni	Aplikacija nastavlja normalno raditi i nakon povećanja opterećenja, bez gubitka performansi ili stabilnosti.
NFZ-005	Korisničko sučelje je intuitivno, prilagođeno svakodnevnoj upotrebi s jasno definiranim ulogama korisnika.	Srednji	Nefunkcionalni	Korisnici lako razumiju sučelje i obavljaju zadatke unutar uloga bez dodatne obuke.
NFZ-006	Za kodiranje se koristi isti stil i konvencija za sve članove tima radi čitljivosti i jednostavnosti koda.	Niski	Nefunkcionalni	Kod prati uspostavljene stilove i konvencije kako bi bio razumljiv svim članovima tima.
NFZ-007	Dokumentacija koda osigurava da su sve funkcionalnosti i koraci projekta detaljno objašnjeni.	Visok	Nefunkcionalni	Ukupna dokumentacija projekta pokriva sve funkcionalnosti s detaljnim uputama za implementaciju i održavanje.

ID	Opis	Prioritet	Izvor	Kriteriji prihvatanja
008	Aplikacija omogućava integraciju sa SUBP-om te podržava XML i PDF formate datoteka.	Srednji	Ne funkcioniše	Sustav podržava navedene formate datoteka, a integracija sa SUBP-om funkcioniše bez grešaka.

## Dionici

1. Djelatnici
2. Voditelj postupka digitalizacije
3. Administrator sustava
4. Razvojni tim

## Aktori i njihovi funkcionalni zahtjevi:

1. Djelatnici (inicijator): >- unos podataka »- skeniranje bar koda »- ručni unos meta podataka »- učitavanje meta podataka iz XML datoteke >- pokretanje digitalizacije >- preuzimanje PDF dokumenta radi evidencije iznesenih filmskih grupa na digitalizaciju >- preuzimanje PDF dokumenta radi evidencije vraćenih filmskih grupa s digitalizacije
2. Voditelj postupka digitalizacije (inicijator): >- pregled statistike aktivnosti djelatnika (izvoz statistike u PDF dokumentu) >- pregled statistike postupka digitalizacije (izvoz statistike u PDF dokumentu) >- upravljanje unesenim podacima o filmovima »- pregled podataka »- izmjena podataka
3. Administrator sustava (inicijator): >- upravljanje korisnicima »- promjena uloge korisnika »- brisanje korisnika iz sustava
4. Arhivska baza podataka (sudionik): >- Pohrana podataka o arhivskim filmskim zapisima
5. Baza podataka (sudionik): >- Pohrana podataka o filmovima, filmskim grupama, korisnicima, aktivnostima korisnika i procesu digitalizacije
6. XML datoteka (sudionik): >- Izvoz podataka u XML formatu za učitavanje meta podataka u radnu bazu
7. OAuth2 servis autentifikacije (sudionik): >- Autentifikacija korisnika pri registraciji i prijavi u sustav

## Obrasci uporabe

### Dijagrami obrazaca uporabe

#### 1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

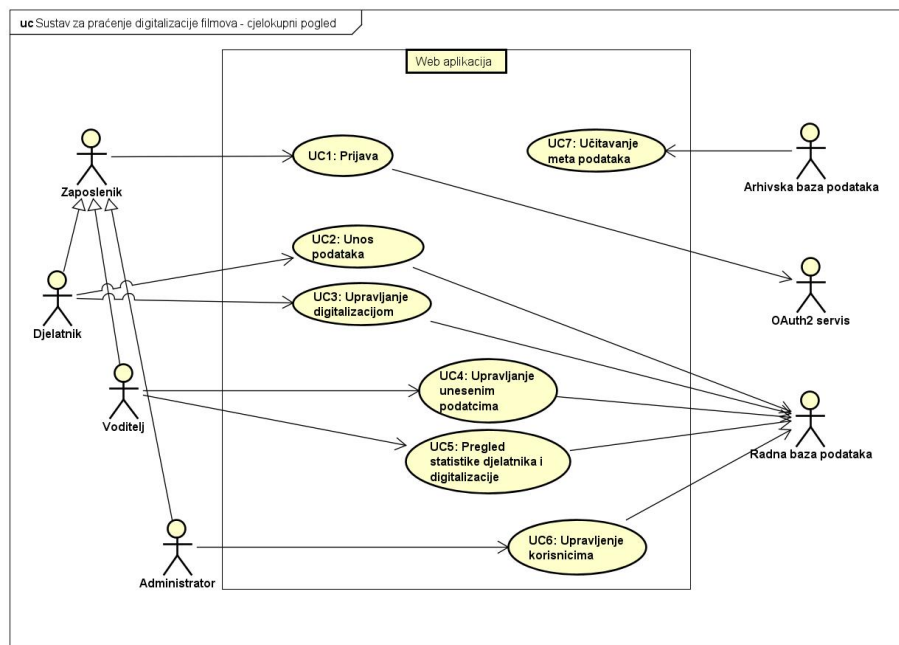


Figure 1: Cjelokupni pogledPRAVI

#### 2. dijagram obrazaca uporabe za učitavanje podataka/kritične sustave i integracije

#### 3. dijagram obrazaca uporabe za djelatnika/osnovne poslovne procese

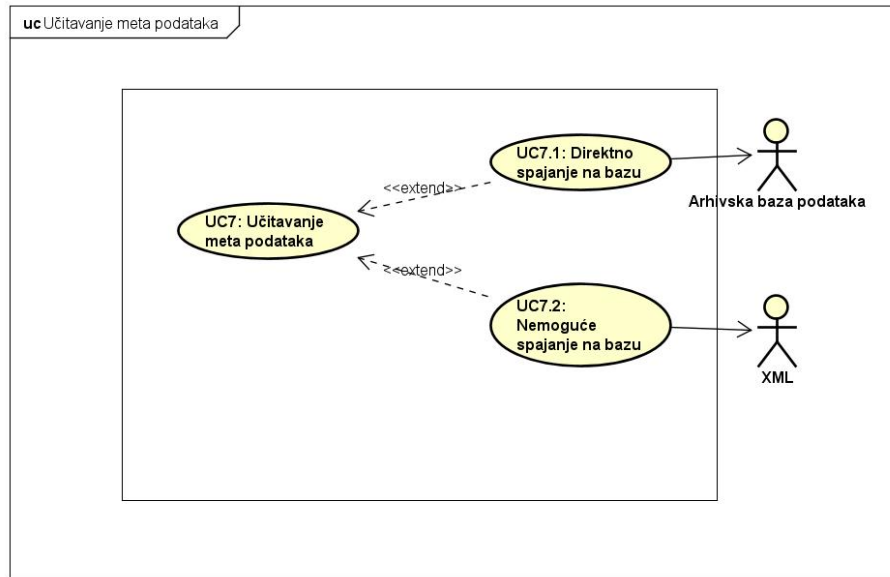


Figure 2: ucitavanjeMetaPodataka

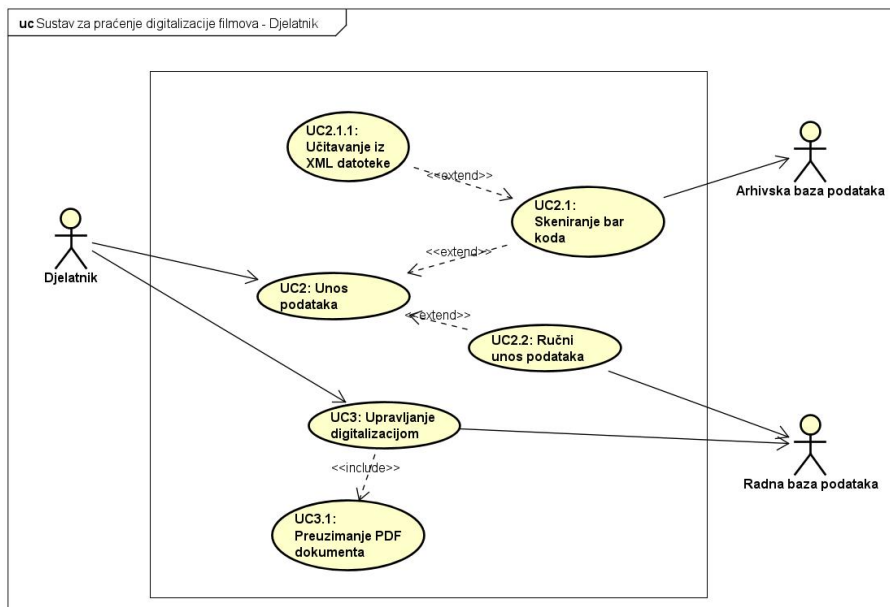


Figure 3: Djelatnik

#### 4. dijagram obrazaca uporabe za voditelja

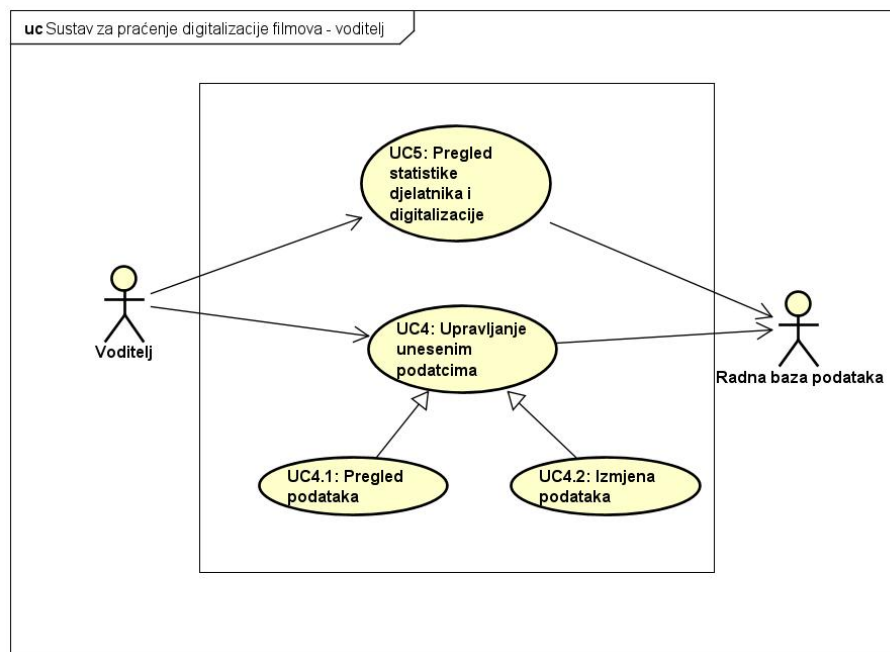


Figure 4: voditelj

#### 5. dijagram obrazaca uporabe za administratora

##### Opis obrazaca uporabe

##### UC1 - Prijava

- Glavni sudionik: zaposlenik
- Cilj: Omogućiti zaposleniku pristup sustavu uz autentifikaciju.
- Sudionici: zaposlenik, sustav za autentifikaciju
- Preduvjet: zaposleniku je dodijeljena važeća @fer.unizg.hr mail adresa
- Opis osnovnog tijeka: > 1. Zaposlenik otvara aplikaciju i dolazi na stranicu za prijavu. > 2. Zaposlenik unosi svoje korisničko ime i lozinku. > 3. Sustav autentificira zaposlenika. > 4. Sustav preusmjerava zaposlenika na početnu stranicu aplikacije prema njegovim pravima pristupa.

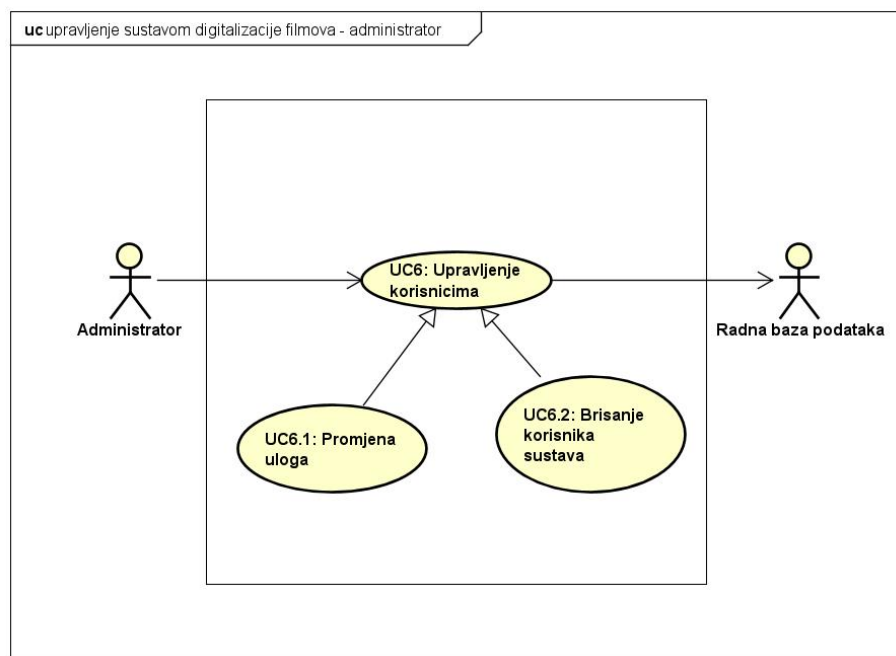


Figure 5: Administrator

- Opis mogućih odstupanja: > 1. Zaposlenik unosi pogrešno korisničko ime ili lozinku. »\* Sustav omogućava zaposleniku ponovnu prijavu.

## UC2 - Unos podataka

- Glavni sudionik: djelatnik
- Cilj: Omogućiti djelatniku skeniranje bar koda, učitavanje meta podataka iz XML datoteke te ručni unos meta podataka.
- Sudionici: djelatnik, arhivska baza podataka, radna baza podataka, XML datoteka
- Preduvjet: -
- Opis osnovnog tijeka: > 1. Djelatnik otvara aplikaciju i dolazi na stranicu za skeniranje bar koda. > 2. Djelatnik skenira bar kod s kutije. > 3. Sustav učitava meta podatke iz arhivske baze podataka.
- Opis mogućih odstupanja: > 1. Ne postoji bar kod na kutiji. »\* Sustav omogućava djelatniku ručni unos meta podataka. > 2. Nije moguće spajanje na arhivsku bazu podataka. »\* Sustav učitava meta podatke iz XML datoteke.

## UC3 - Upravljanje digitalizacijom

- Glavni sudionik: djelatnik
- Cilj: Omogućiti djelatniku slanje filmskih grupa na postupak digitalizacije te njihovo vraćanje nakon dovršetka postupka digitalizacije uz preuzimanje odgovarajućih PDF potvrda.
- Sudionici: djelatnik, radna baza podataka
- Preduvjet: Filmske grupe su formirane.
- Opis osnovnog tijeka: > 1. Sustav dinamički grupira filmove kako bi se maksimalno iskoristilo dostupnih 45 minuta za svaku grupu. > 2. Djelatnik šalje grupe na digitalizaciju te preuzima PDF dokument kojim se evidentira koje grupe je odnio na digitalizaciju. > 3. Nakon završetka postupka digitalizacije, djelatnik vraća filmske grupe te preuzima PDF dokument kojim se evidentira koje grupe je vratio s digitalizacije.

## UC4 - Upravljanje unesenim podacima

- Glavni sudionik: voditelj
- Cilj: Omogućiti voditelju postupka digitalizacije pregled i izmjenu podataka.
- Sudionici: voditelj, radna baza podataka
- Preduvjet: djelatnici unose podatke o filmovima
- Opis osnovnog tijeka: > 1. Voditelj pristupa stranici na kojoj se nalaze svi podatci o filmovima koje su unijeli djelatnici te sve kreirane grupe i

njihov status digitalizacije. > 2. Voditelj pregledava podatke i po potrebi ih izmjenjuje.

#### **UC5 - Pregled statistike djelatnika i digitalizacije**

- Glavni sudionik: voditelj
- Cilj: Omogućiti voditelju postupka digitalizacije pregled statistike o aktivnostima svih djelatnika i o digitalizaciji.
- Sudionici: voditelj, radna baza podataka
- Preduvjet: djelatnici unose podatke o filmovima i šalju ih na digitalizaciju
- Opis osnovnog tijeka: > 1. Voditelj pristupa stranici na kojoj se nalazi odgovarajuća statistika. > 2. Voditelj preuzima PDF dokument te pregledava statistike.

#### **UC6 - Upravljanje korisnicima**

- Glavni sudionik: administrator
- Cilj: Omogućiti administratoru promjenu uloga drugim korisnicima sustava i brisanje korisnika iz sustava.
- Sudionici: administrator, radna baza podataka
- Preduvjet: Postoje drugi korisnici u sustavu.
- Opis osnovnog tijeka: > 1. Administrator pristupa stranici na kojoj mu je dostupna lista svih korisnika sustava koju može preuzeti u PDF formatu. > 2. Administrator po potrebi mijenja uloge te briše korisnike iz sustava.

#### **UC7 - Učitavanje meta podataka**

- Glavni sudionik: arhivska baza podataka
- Cilj: Omogućiti učitavanje meta podataka o filmovima.
- Sudionici: arhivska baza podataka, XML
- Preduvjet: Skenirani bar kod postoji u arhivskoj bazi podataka.
- Opis osnovnog tijeka: > 1. Učitavanje meta podataka direktno iz arhivske baze podataka.
- Opis mogućih odstupanja: > 1. Nije moguće spajanje na arhivsku bazu podataka. »\* Sustav učitava meta podatke iz XML datoteke.

## **Sekvencijski dijagrami**

### **1. sekvencijski dijagram za upravljanje podacima**



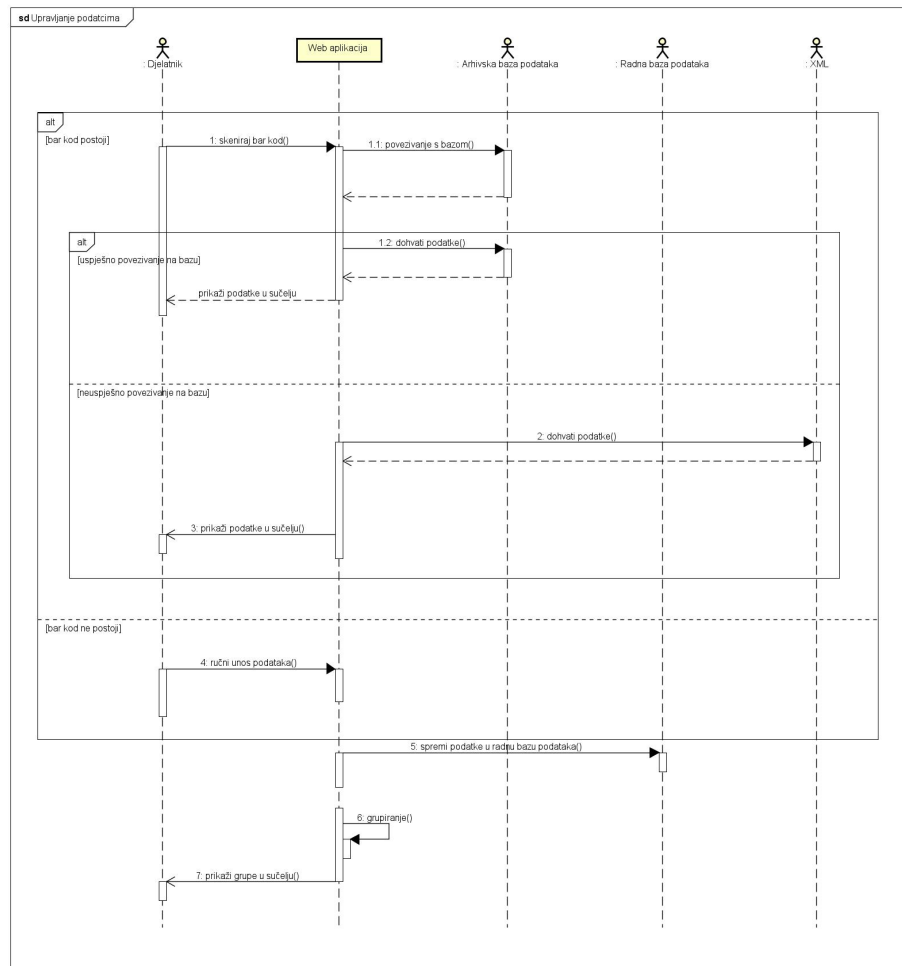
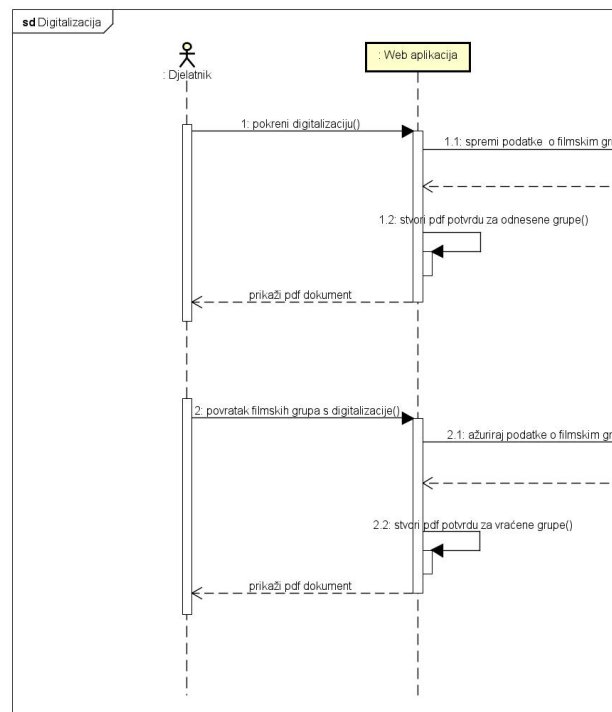


Figure 6: Upravljanje podacima

**Opis:** Dijagram prikazuje glavne scenarije vezane za upravljanje podacima u arhivskoj bazi podataka, radnoj bazi podataka te XML datotekama.

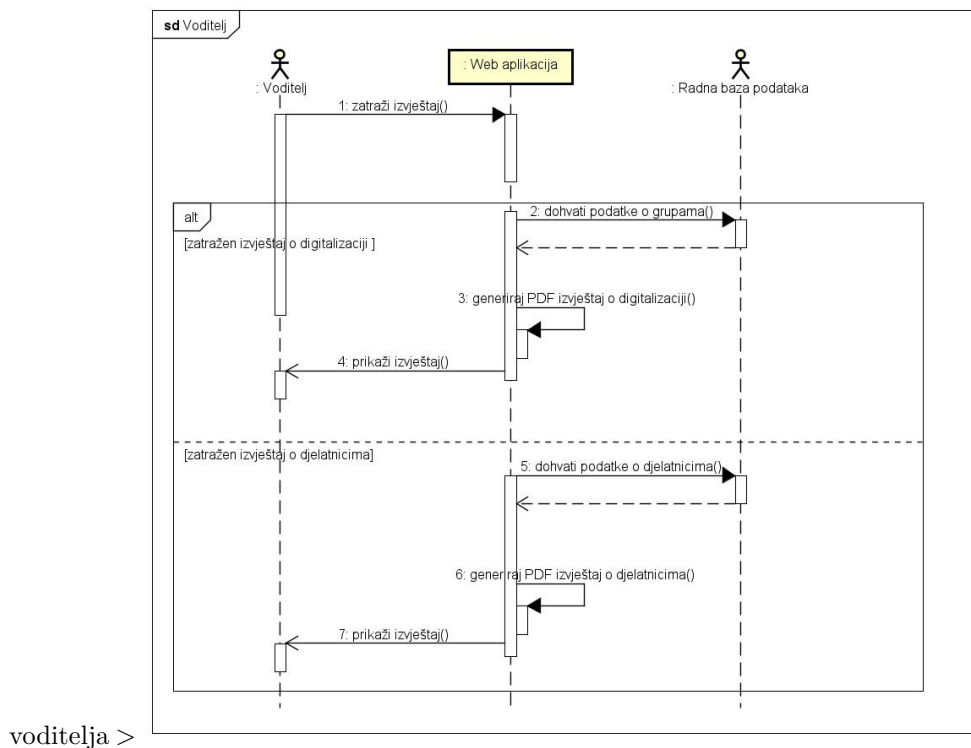
1. Dijagram započinje skeniranjem bar koda (pod uvjetom da bar kod postoji) nakon čega web aplikacija uspostavlja vezu s arhivskom bazom podataka.
- Ako je povezivanje s arhivskom bazom podataka uspješno, web aplikacija dohvaća meta podatke o filmskim zapisima iz arhivske filmske baze, prikazuje ih u korisničkom sučelju.
  - Ako je povezivanje s arhivskom bazom podataka neuspješno, web aplikacija dohvaća meta podatke o filmskim zapisima iz XML datoteke, prikazuje ih u korisničkom sučelju.
2. U slučaju da bar kod ne postoji, djelatnik unosi podatke o filmovima ručno putem web aplikacije.
3. Nakon unosa podataka, aplikacija podatke sprema u radnu bazu podataka.
4. Web aplikacija grupira unesene podatke te prikazuje filmske grupe u korisničkom sučelju.

> > > # 2. sekvencijski dijagram za digitalizaciju >



**Opis:** Dijagram prikazuje akcije vezane za digitalizaciju filmskog materijala.

1. Nakon postupka grupiranja, djelatnik pokreće postupak digitalizacije pri čemu se podatci o filmskim grupama spremaju u radnu bazu podataka i generira se PDF dokument kao potvrda da je korisnik odnio odabrane grupe na digitalizaciju.
2. Nakon završetka postupka digitalizacije, u bazi se ažuriraju podatci o filmskim grupama i generira se PDF dokument kao potvrda da je korisnik vratio odabrane grupe s digitalizacije.



**Opis:** Dijagram prikazuje proces dobivanja izvještaja o digitalizaciji i o aktivnostima djelatnika.

1. Voditelj šalje zahtjev za generiranje izvještaja pri čemu može odabrati jednu od ove dvije opcije:
  - izvještaj o digitalizaciji (koliko je filmskih grupa trenutno na digitalizaciji i koliko ih je digitalizirano)
  - izvještaj o djelatnicima (koliko je koji djelatnik/voditelj iznio filmskih grupa na digitalizaciju, odnosno vratio s digitalizacije)
2. Iz baze se dohvaćaju odgovarajući podatci.
3. Generira se i prikazuje odgovarajući izvještaj.

## Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Naziv Dijagrama	Funkcionalni Zahtjevi
UC1 - Prijava	FZ-009: Djelatnici, voditelji i administratori imaju različite korisničke uloge s definiranim pravima pristupa. FZ-013: Autentifikacija korisnika omogućena je putem OAuth2 servisa pri registraciji i prijavi.
UC2 - Unos podataka	FZ-001: Aplikacija podržava čitanje bar koda.  FZ-004: Ako bar kod ne postoji, aplikacija omogućuje ručni unos meta podataka o filmu.
UC3 - Upravljanje digitalizacijom	FZ-006: Aplikacija optimalno grupira unesene filmove te dinamički stvara grupe koje djelatnik može pregledati i poslati na digitalizaciju. FZ-010: Aplikacija generira PDF dokument prilikom iznošenja i vraćanja filmskih grupa s postupka digitalizacije.
UC4 - Upravljanje unesenim podatcima	FZ-005: Nakon učitavanja podataka, omogućena je ručna izmjena i nadopuna podataka.
UC5 - Pregled statistike djelatnika i digitalizacije	FZ-007: Aplikacija vodi statističku evidenciju o grupama koje su trenutno na digitalizaciji i koje su završene te generira PDF dokument.  FZ-008: Aplikacija vodi statističku evidenciju o radu djelatnika te generira PDF dokument.
UC6 - Upravljanje korisnicima	FZ-011: Aplikacija omogućuje prikaz svih korisnika aplikacije te generira PDF dokument. FZ-012: Administrator ima mogućnost brisanja i promjene uloge korisnika aplikacije.
UC7 - Učitavanje meta podataka	FZ-002: Na temelju skeniranog bar koda dohvaćaju se meta podatci o filmu iz arhivske baze podataka. FZ-003: Ako bar kod nije dostupan u arhivskoj bazi podataka, aplikacija podržava učitavanje podataka iz XML datoteke.

# Arhitektura sustava

## Opis arhitekture

### Stil arhitekture

Arhitektura sustava za **Digitalizaciju arhivskih filmova** temelji se na *klijent-poslužitelj* arhitektonskom stilu. Taj je stil odabran zbog potrebe za jasnom podjelom odgovornosti između klijentskog dijela (frontend aplikacije) i poslužiteljskog dijela (backend aplikacije), čime se osigurava skalabilnost, modularnost te lakše održavanje.

Klijent-poslužitelj arhitektura omogućuje frontend dijelu sustava preuzimanje podataka s poslužitelja preko HTTP protokola i prikazuje ih korisnicima na jednostavan i intuitivan način. Backend, s druge strane, obavlja glavnu logiku i rukuje podacima, osiguravajući integritet i kontrolu nad procesima.

### Podsustavi

- **Frontend:** Odgovoran za korisničko sučelje i interakciju s korisnikom. Omogućuje prikaz digitaliziranih materijala, rad s barkodovima i unos novih podataka. Implementiran je u React-u i koristi HTML, CSS, te JavaScript.
- **Backend:** Zadužen za poslovnu logiku i pristup podacima u bazi. Obraduje zahtjeve s klijentskog dijela i komunicira s bazom podataka za spremanje i dohvaćanje podataka o arhivskim filmovima. Backend je implementiran u Spring Boot frameworku.
- **Baza podataka:** Spremište za sve podatke o arhivskim filmovima, metapodacima i korisničkim podacima. Koristi se relacijska baza podataka PostgreSQL koja podržava sve potrebne funkcionalnosti za projekt.

### Spremišta podataka

Podaci će se pohranjivati u relacijskoj bazi podataka **PostgreSQL**. PostgreSQL je odabran zbog svoje robusnosti, skalabilnosti i podrške za složene upite, što omogućava pouzdano pohranjivanje i dohvaćanje podataka o arhivskim filmovima i metapodacima. Ova baza podataka nudi visoku razinu sigurnosti i može se integrirati sa Spring Boot aplikacijom bez dodatnih problema.

### Mrežni protokoli

Komunikacija između frontend i backend dijela odvijat će se putem **HTTP** protokola za slanje zahtjeva i odgovora. Također, za autentifikaciju korisnika

koristi se OAuth2 protokol, koji osigurava sigurnu prijavu korisnika kroz vanjski autentifikacijski servis.

### Globalni upravljački tok

Podaci teku kroz sustav od korisnika koji koristi frontend aplikaciju za skeniranje ili unos barkodova. Nakon unosa, podaci se šalju backendu koji obavlja validaciju i pohranu u bazu podataka. Backend potom šalje potvrdu ili relevantne podatke natrag prema korisniku na frontend aplikaciji.

## Obrazloženje odabira arhitekture

### Izbor arhitekture temeljen na principima oblikovanja

Odabrana arhitektura klijent-poslužitelj omogućuje jasno razdvajanje odgovornosti između frontend i backend sustava, što donosi nekoliko prednosti za projekt **Digitalizacija arhivskih filmova**:

- **Visoka kohezija:** Svaki dio sustava ima jasno definiran zadatak – frontend za korisničku interakciju, a backend za obradu i pohranu podataka. Ova organizacija olakšava razvoj i održavanje svakog dijela sustava.
- **Niska povezanost:** Klijent-poslužitelj model smanjuje povezanost između komponenti jer frontend šalje HTTP zahtjeve backendu bez direktne ovisnosti o internim procesima na poslužiteljskoj strani. Ovo smanjuje rizik od kvarova u jednom dijelu sustava koji bi mogli utjecati na cijeli sustav.
- **Fleksibilnost:** Arhitektura omogućuje jednostavno proširenje funkcionalnosti, dodavanje novih značajki i mogućnost prilagodbe sučelja bez značajnih promjena u backend sustavu.
- **Sigurnost:** Uvođenjem OAuth2 autentifikacijskog protokola omogućena je sigurna prijava korisnika, čime se osigurava zaštita podataka i autentifikacija kroz vanjski servis.

Ovi principi oblikovanja osiguravaju da sustav bude održiv, jednostavan za nadogradnju i skalabilan, čime se zadovoljavaju ključni ciljevi projekta.

## Organizacija sustava na visokoj razini

### Klijent-poslužitelj

Sustav **Digitalizacija arhivskih filmova** organiziran je prema klijent-poslužitelj arhitekturi. Klijentski dio čini frontend aplikacija koja pruža

korisničko sučelje i omogućava interakciju s korisnicima putem web preglednika. Frontend šalje zahtjeve poslužitelju (backendu) putem HTTP protokola. Backend obrađuje zahtjeve, komunicira s bazom podataka i vraća odgovore frontend aplikaciji, osiguravajući tako obradu i pohranu podataka.

## Baza podataka

Sustav koristi **relacijsku bazu podataka** (PostgreSQL baza podataka) za pohranu svih važnih podataka o arhivskim filmovima, uključujući metapodatke kao što su trajanje, naziv i status digitalizacije. PostgreSQL baza odabrana je zbog svoje brzine i fleksibilnosti u razvoju, kao i mogućnosti integracije u Java Spring okruženje. Podaci su organizirani u strukturirane tablice, što omogućuje jednostavno dohvaćanje i manipulaciju podacima, čime se osigurava stabilnost i dosljednost podataka unutar sustava.

## Datotečni sustav

U trenutnoj verziji projekta, sustav se ne oslanja na zaseban datotečni sustav za pohranu podataka. Svi relevantni podaci nalaze se u relacijskoj bazi podataka. Međutim, PDF dokumenti generirani tijekom procesa digitalizacije mogu se privremeno pohranjivati i preuzimati na strani klijenta, ovisno o zahtjevima specifične funkcionalnosti.

## Grafičko sučelje

Korisničko sučelje implementirano je kao **web aplikacija** koristeći moderne tehnologije poput Reacta za frontend dio. Sučelje omogućava djelatnicima i voditeljima jednostavan i intuitivan pristup funkcionalnostima sustava, uključujući skeniranje bar kodova, pregled metapodataka i generiranje izvještaja. Frontend je direktno povezan s backend dijelom putem API poziva, omogućavajući brzu i sigurnu komunikaciju. Ova arhitektura olakšava korisnicima interakciju s aplikacijom, čineći procese digitalizacije efikasnijima.

## Organizacija aplikacije

### Frontend i Backend slojevi

Aplikacija **Digitalizacija arhivskih filmova** organizirana je u dva glavna sloja:

- Frontend: Frontend je odgovoran za prikaz korisničkog sučelja i interakciju s korisnicima. Razvijen je koristeći ‘React’, pruža prikaz podataka te

omogućava korisnicima navigaciju kroz aplikaciju, pregled arhivskih podataka i skeniranje bar kodova.

- Komponente za skeniranje: Uključuje komponentu za skeniranje bar kodova koja omogućuje očitavanje filmova.
  - Lista bar kodova: Sadrži komponentu za pregled liste skeniranih bar kodova, što korisnicima omogućava grupiranje i pregled metapodataka o filmskim trakama.
  - Komponenta za autentifikaciju: Omogućuje prijavu i odjavu korisnika koristeći OAuth2 autentifikacijski protokol.
- Backend: Backend je razvijen u Spring Bootu i pruža API za upravljanje podacima, autentifikaciju i komunikaciju s bazom podataka. Backend obrađuje zahtjeve iz frontenda, upravlja poslovnom logikom i dohvaća podatke iz baze.
    - Servisni sloj: Obraduje logiku vezanu uz upravljanje filmskim zapisima i korisnicima.
    - Kontroleri: Primaju zahtjeve s frontenda i komuniciraju s odgovarajućim servisima za obradu i povrat podataka.
    - Modeli podataka: Predstavljaju strukturu podataka pohranjenu u bazi, uključujući filmske zapise i korisničke podatke.

## MVC arhitektura

Aplikacija koristi **MVC (Model-View-Controller)** arhitekturu na backendu:

\* Model: Sadrži podatke o arhivskim filmovima, korisnicima, grupama za digitalizaciju i statistikama. \* View: Integrira JSON odgovore koje backend šalje prema frontend sučelju. \* Controller: Obraduje zahtjeve iz frontenda, koristi servise za dohvat podataka iz baze i vraća podatke prema potrebama aplikacije.

Ova arhitektura omogućava odvajanje korisničkog sučelja, poslovne logike i podataka, čime se postiže fleksibilnost i olakšava održavanje koda.

## Reference

- Architect modern web applications with ASP.NET Core and Azure - .NET | Microsoft Learn. Dostupno: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/>.

## Baza podataka

Tijekom razvoja i testiranja aplikacije koristili smo H2 bazu podataka (embedded). Za produkcijsko okruženje, prilikom postavljanja aplikacije u stvarni radni sustav, prešli smo na PostgreSQL bazu podataka. > Glavne komponente:



\* Shema podataka koja se sastoji od entiteta koji su predstavljeni tablicom s jasno definiranim atributima i relacijama. \* Svaka tablica ima primarni ključ za identifikaciju redaka unutar tablice, a za povezivanje tablica koriste se strani ključevi što osigurava referencijski integritet među povezanim tablicama. \* PostgreSQL omogućava potpunu podršku za ACID transakcije, čime se osigurava pouzdanost prilikom izvršavanja više povezanih upita. \* PostgreSQL podržava različite metode autentifikacije i omogućava precizno upravljanje pristupima na razini korisnika i tablica.

## Opis tablica

### FilmskaTraka

Atribut	Tip podatka	Opis varijable
IDEmisije	Long	Jedinstveni identifikator emisije (PK)
OriginalniNaslov	String	Originalni naslov filmske trake
JezikOriginala	String	Jezik originala filmske trake
Ton	String	Informacija o tonu (mono, stereo, itd.)
VrstaSadrzaja	String	Vrsta sadržaja (npr. dokumentarni, igrani)
Porijeklo_ZemljaProizvodnje	String	Zemlja porijekla/proizvodnje
Licenca	String	Informacije o licenci

Atribut	Tip podatka	Opis varijable
GodinaProizvodnje	Integer	Godina kada je filmska traka proizvedena
Duration	LocalTime	Trajanje filmske trake
IDGrupeZaDigitalizaciju	String	Grupe u kojima se film nalazi

#### **GrupaZaDigitalizaciju**

Atribut	Tip podatka	Opis varijable
IDGrupe	Long	Jedinstveni identifikator grupe za digitalizaciju (PK)
VrijemePocetka	LocalDateTime	Vrijeme početka digitalizacije
VrijemeZavrsetka	LocalDateTime	Vrijeme završetka digitalizacije
IznioIzSkladistaKorisnikId	Long	ID korisnika koji je iznio grupu za digitalizaciju

Atribut	Tip podatka	Opis varijable
VratioUSkladisteKorisnikIdLong		ID korisnika koji je vratio grupu u skladište
StatusDigitalizacije	Integer	Status procesa digitalizacije

### StatusDigitalizacije

Atribut	Tip podatka	Opis varijable
IDStatusDigitalizacije	Integer	Jedinstveni identifikator statusa procesa digitalizacije (PK)
NazivStatusa	String	Oznaka statusa procesa digitalizacije za grupu (na_Cekanju, na_Digitalizaciji, zavrsono)

### Korisnik

Atribut	Tip podatka	Opis varijable
IDKorisnika	Long	Jedinstveni identifikator korisnika (PK)
Ime	String	Ime korisnika
Prezime	String	Prezime korisnika

Atribut	Tip podatka	Opis varijable
Email	String	Email adresa korisnika
Uloga	Integer	Uloga korisnika

### Uloga

Atribut	Tip podatka	Opis varijable
IDUlogeKorisnika	Integer	Jedinstveni identifikator uloge korisnika (PK)
NazivUlogeKorisnika	String	Naziv uloge korisnika (djelatnik, voditelj, administrator)

### Grupa\_FilmskeTrake

Atribut	Tip podatka	Opis varijable
filmskaTraka_originalniNaslov	String	Originalni naslov filma
idGrupaZaDigitalizaciju	Long	ID grupe kojoj film pripada (FK)

### Korisnik\_GrupaZaDigitalizaciju\_iznio

Atribut	Tip podatka	Opis varijable
idIznesenihGrupaZaDigitalizaciju	Long	ID iznesene grupe

Atribut	Tip podatka	Opis varijable
iznioIzSkladistaKorisnikId	Long	ID korisnika koji je iznio grupu (FK)

#### Korisnik\_GrupaZaDigitalizaciju\_vratio

Atribut	Tip podatka	Opis varijable
idVracenihGrupaZaDigitalizaciju	Long	ID vraćene grupe
vratioUSkladisteKorisnikId	Long	ID korisnika koji je vratio grupu (FK)

#### FilmskaTrakaArhiva - ARHIVSKA BAZA PODATAKA

Atribut	Tip podatka	Opis varijable
BarKod	String	Jedinstveni barkod koji identifikira filmsku traku (PK)
OriginalniNaslov	String	Originalni naslov filmske trake

Atribut	Tip podatka	Opis varijable
RadniNaslov	String	Radni naslov filmske trake
JezikOriginala	String	Jezik originala filmske trake
Ton	String	Informacija o tonu (mono, stereo, itd.)
Emisija	String	Naziv emisije
VrstaSadrzaja	String	Vrsta sadržaja (npr. dokumen-tarni, igrani)
Porijeklo_ZemljaProizvodnje	String	Zemlja porijekla/proizvodnje
Licenca	String	Informacije o licenci
GodinaProizvodnje	Integer	Godina kada je filmska traka proizve-dena
MarkIN	LocalTime	Početak (mark-in) vremena unosa
MarkOUT	LocalTime	Kraj (mark-out) vremena unosa

Atribut	Tip podatka	Opis varijable
Duration	LocalTime	Trajanje filmske trake
BrojMedija	String	Broj medija na kojem se nalazi traka

## Dijagram baze podataka

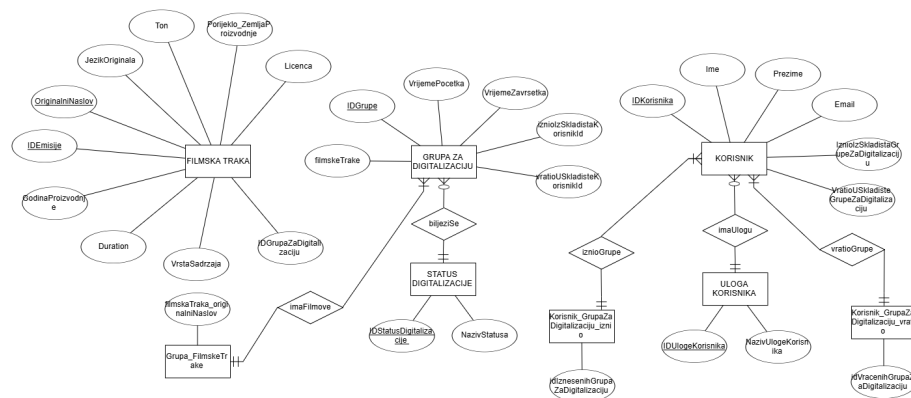


Figure 7: BazaPodataka-ERDmodel3

## ER dijagram radne baze podataka

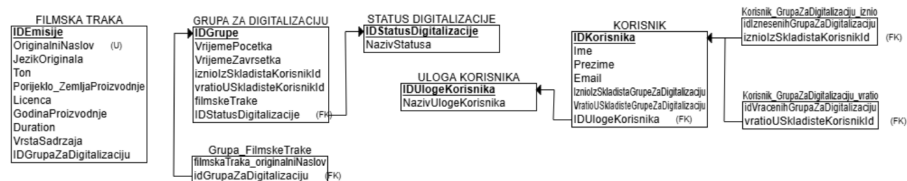


Figure 8: bazaPodataka-relacijskiModel3

## Relacijski dijagram radne baze podatka

## Dijagram razreda - dio Models

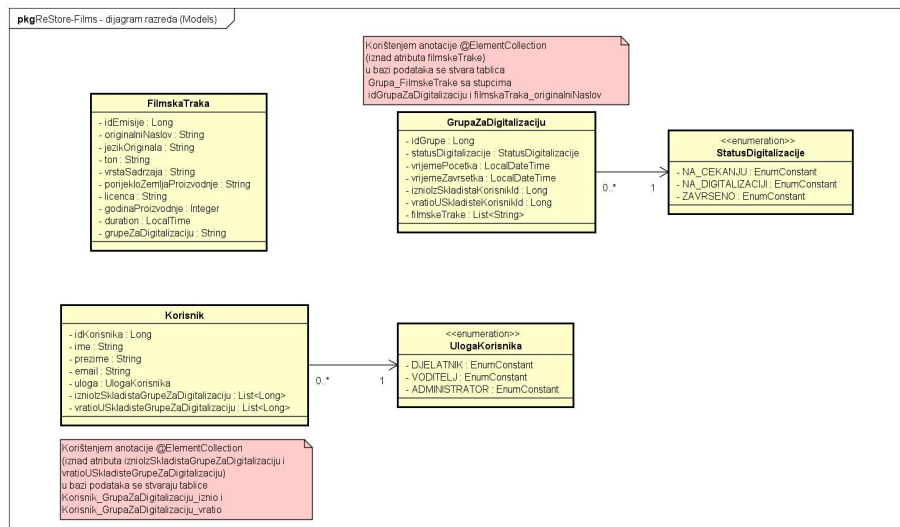


Figure 9: dijagramRazredaModels3

Ovaj dijagram razreda prikazuje model digitalizacije filmskih traka, korisnike s ulogama, grupe za digitalizaciju i statuse digitalizacije.

Klase:

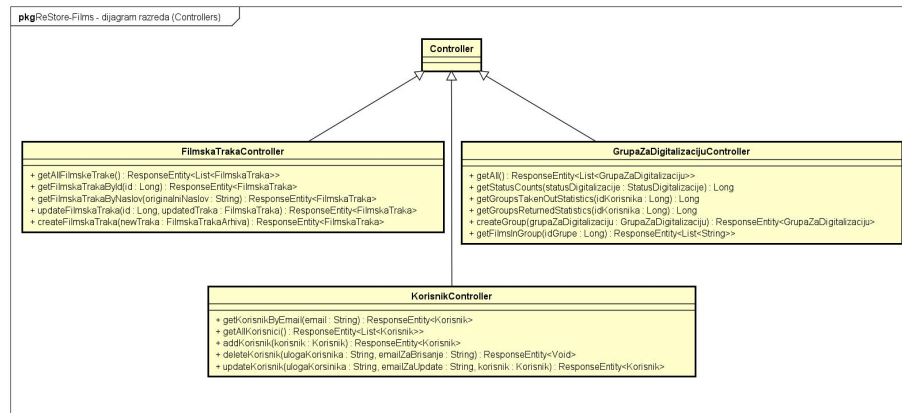
- FilmskaTraka: Sadrži osnovne informacije o traci.
- GrupaZaDigitalizaciju: Sadrži podatke o grupi traka, uključujući status digitalizacije i korisnike odgovorne za skladištenje.
- Korisnik: Sadrži podatke o korisniku te njegovu statistiku i ulogu. Također ima uloge u digitalizaciji (iznio/vratio).
- StatusDigitalizacije i UlogaKorisnika: Enumeracije za statuse digitalizacije i uloge korisnika.

Dodatni opis: \* U klasi FilmskaTraka u atribut grupeZaDigitalizaciju zapisuje se ID grupe/grupa kojima pripada filmska traka (jedna filmska traka se može nalaziti u više grupa za digitalizaciju). \* U klasi GrupaZaDigitalizaciju korištenjem anotacije @ElementCollection (iznad atributa filmskeTrake) u bazi podataka se stvara tablica Grupa\_FilmskeTrake sa stupcima idGrupaZaDigitalizaciju i filmskaTraka\_originalniNaslov koja služi za pregled naslova filmova po grupama. \* U klasi Korisnik korištenjem anotacije @ElementCollection (iznad atributa iznioIzSkladistaGrupeZaDigitalizaciju i vratioUSkladistaGrupeZaDigitalizaciju) u bazi podataka se stvaraju tablice Korisnik\_GrupaZaDigitalizaciju\_iznio i Korisnik\_GrupaZaDigitalizaciju\_vratio



koje služe za evidenciju statistike o djelatnicima (broj iznesenih/vraćenih grupa svakog djelatnika).

## Dijagram razreda - dio Controllers



Ovaj dijagram razreda prikazuje strukturu kontrolera u sustavu. Svi kontroleri nasljeđuju osnovni razred Controller, što omogućuje zajedničke funkcionalnosti među kontrolerima.

Razredi i njihove metode:

- FilmskaTrakaController
  - getAllFilmskeTrake(): Vraća popis svih filmskih traka u obliku liste.
  - getFilmskaTrakaById(Long id): Dohvaća specifičnu filmsku traku prema njenom ID-u.
  - getFilmskaTrakaByNaslov(String originalniNaslov): Vraća filmsku traku prema naslovu.
  - updateFilmskaTraka(Long id, FilmskaTraka updatedTraka): Omogućuje ažuriranje postojeće filmske trake.
  - createFilmskaTraka(FilmskaTrakaArhiva newTraka): Dodaje novu filmsku traku.
- GrupaZaDigitalizacijuController
  - getAll(): Vraća sve grupe za digitalizaciju.
  - getStatusCounts(StatusDigitalizacije statusDigitalizacije): Broji grupe prema određenom statusu.
  - getGroupsTakenOutStatistics(Long idKorisnika): Vraća statistiku o grupama koje je korisnik iznio.
  - getGroupsReturnedStatistics(Long idKorisnika): Vraća statistiku o grupama koje je korisnik vratio.
  - createGroup(GrupaZaDigitalizaciju grupaZaDigitalizaciju): Omogućuje kreiranje nove grupe za digitalizaciju.

- getFilmsInGroup(Long idGrupe): Vraća nazive filmova iz grupe.
- KorisnikController
  - getKorisnikByEmail(String email): Dohvaća korisnika pomoću e-mail adrese.
  - getAllKorisnici(): Vraća popis svih korisnika.
  - addKorisnik(Korisnik korisnik): Dodaje novog korisnika u sustav.
  - deleteKorisnik(String ulogaKorisnika, String emailZaBrisanje): Briše korisnika prema njegovoj e-mail adresi.
  - updateKorisnik(String ulogaKorisnika, String emailZaUpdate, Korisnik korisnik): Ažurira korisnika prema njegovoj e-mail adresi.

## Dinamičko ponašanje aplikacije

### UML dijagram stanja

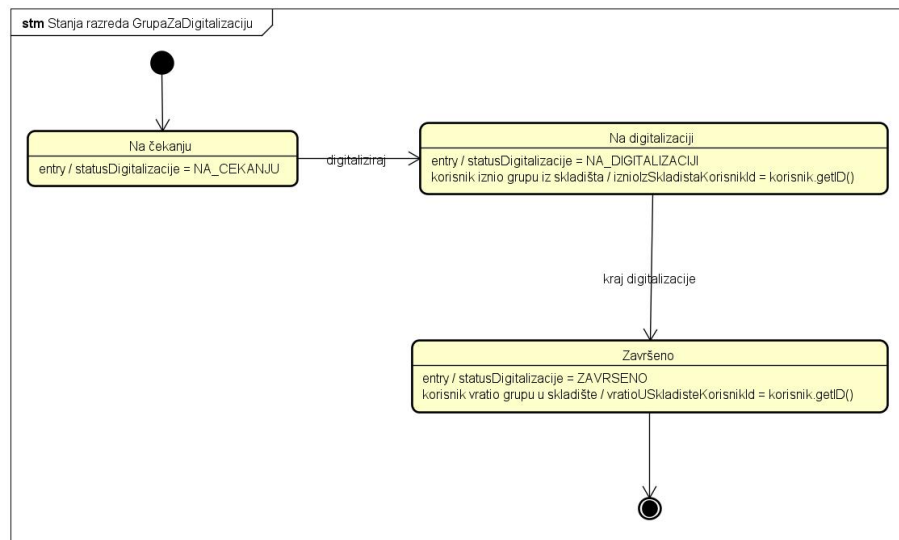


Figure 10: Stanja razreda GrupaZaDigitalizaciju

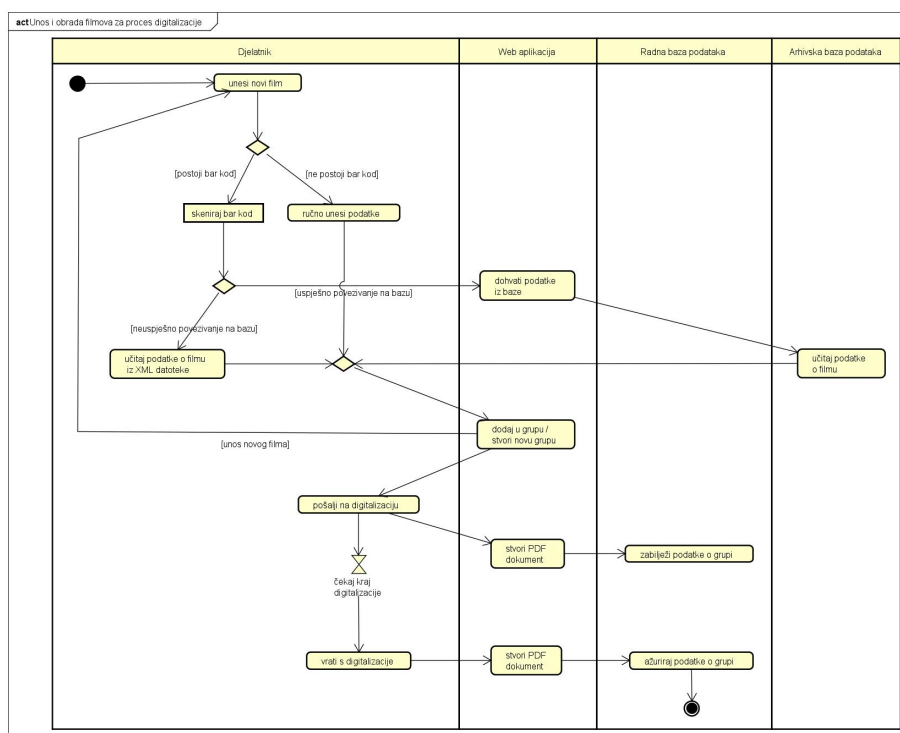
Ovaj dijagram stanja prikazuje stanja kroz koja prolazi objekt klase `GrupaZaDigitalizaciju` u procesu digitalizacije.

Opis stanja: \* Na čekanju: \* Ovo je početno stanje grupe. \* Kada se grupa kreira, postavlja se atribut `statusDigitalizacije` na vrijednost `NA_CEKANJU`.

- Na digitalizaciji:

- Događaj početka digitalizacije uzrokuje prijelaz u ovo stanje.
- Atribut statusDigitalizacije mijenja se na vrijednost NA\_DIGITALIZACIJI.
- U atribut iznioIzSkladistaKorisnikId zabilježi se IDKorisnika koji je iznio filmsku grupu na postupak digitalizacije.
- Završeno:
  - Događaj završetka digitalizacije uzrokuje prijelaz u ovo stanje.
  - Atribut statusDigitalizacije mijenja se na vrijednost ZAVRSENO.
  - U atribut vratioUSkladisteKorisnikId zabilježi se IDKorisnika koji je vratio filmsku grupu s postupka digitalizacije.

## UML dijagram aktivnosti



\* Djelatnik unosi film. \* Ako postoji bar kod, skenira ga i spaja se na arhivsku bazu podataka za učitavanje meta podataka o filmu. \* Ako nije moguće spajanje na arhivsku bazu podataka, meta podatci se učitavaju iz XML datoteke. \* Ako bar kod ne postoji, djelatnik ručno unosi meta podatke o filmu.

- Nakon što je djelatnik učitao podatke o filmu, aplikacija po trajanju filma dodaje film u postojeću grupu ili kreira novu grupu prema sljedećem algoritmu:

- Filmovi se sortiraju po trajanju od najvećeg prema najmanjem prije grupiranja.
- Grupiranje prema trajanju:
  - \* Filmovi se grupiraju tako da ukupno trajanje svake grupe ne prelazi 45 minuta.
  - \* Ako trenutni film ne može stati ni u jednu postojeću grupu, stvara se nova grupa.
  - \* Filmovi čije trajanje prelazi maksimalno dopušteno trajanje od 45 minuta dijele se na manje segmente. Svakom segmentu se dodaje oznaka dijela (npr. 1. dio, 2. dio). Ostatak koji je manji od 45 minuta se grupira s ostalim filmovima.
- Nakon što je film grupiran, djelatnik započinje proces digitalizacije ili unosi još filmova.
- Prilikom pokretanja procesa digitalizacije djelatnik preuzima PDF potvrdu o iznesenim filmskim grupama te se podatci spremaju u radnu bazu.
- Nakon završetka postupka digitalizacije, djelatnik preuzima potvrdu o vraćenim filmskim grupama s procesa digitalizacije te se podatci iz radne baze ažuriraju.

## Dijagram komponenata

UML dijagram komponenata prikazuje arhitekturu aplikacije ReStore-Films. Sastoji se od sljedećih dijelova: 1. **Web preglednik**: Služi za interakciju s aplikacijom. 2. **Web aplikacija (frontend)**: Implementirana u React.js-u, odgovorna za prikaz i navigaciju. 3. **Spring Boot backend**: Rješava poslovnu logiku i upravlja pristupom bazi podataka. 4. **PostgreSQL baza podataka**: Pohranjuje podatke aplikacije.

## Komponente

### 1. Web preglednik

- Korisnici šalju zahtjeve za dohvatom vizualnog sadržaja preko frontenda.

### 2. Web aplikacija (Frontend)

Frontend je implementiran u **React.js-u** i obuhvaća: - **React frontend**: Glavna komponenta koja upravlja korisničkim sučeljem. - **main.jsx**: Ulazna točka React aplikacije koja povezuje komponente. - **React Router**: Upravlja navigacijom između stranica. - **Korisničke uloge**: - **DJELATNIK**: Sučelje

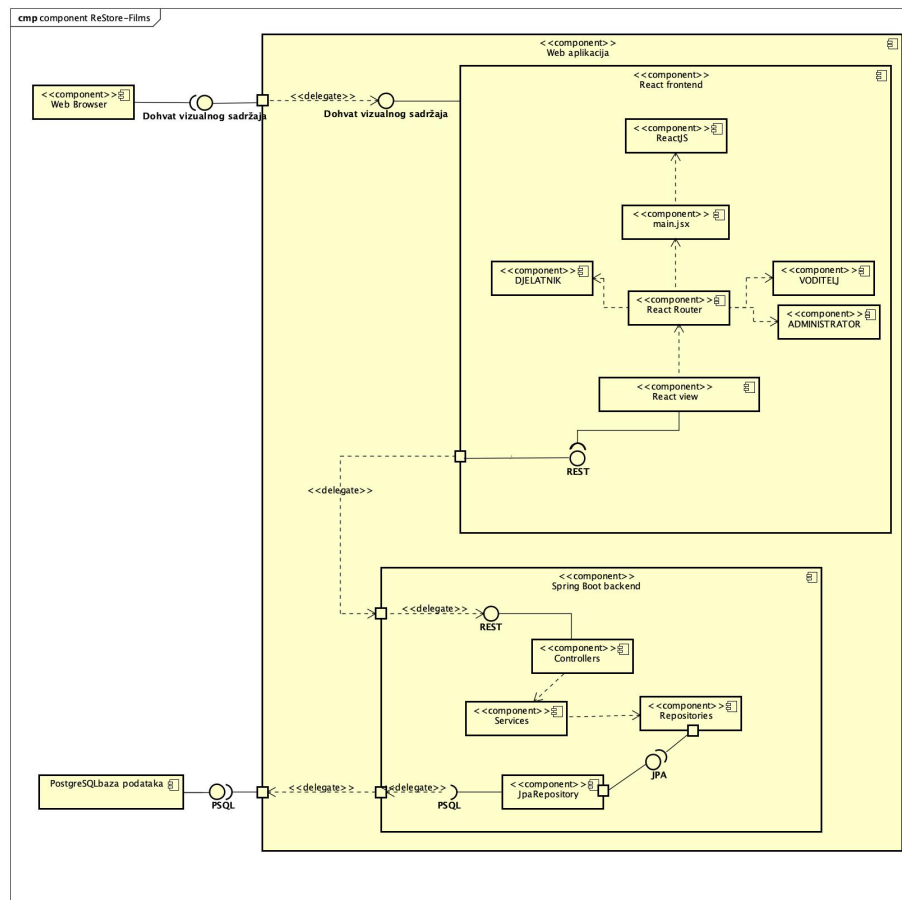


Figure 11: image

za zaposlenike. - **VODITELJ**: Sučelje za voditelje. - **ADMINISTRATOR**: Sučelje za administratore. - **React view**: Komponenta odgovorna za prikazivanje vizualnog sadržaja.

Frontend šalje REST zahtjeve prema backendu za dohvat podataka.

### 3. Spring Boot backend

Backend je implementiran u **Spring Boot-u** okruženju i obuhvaća: - **Controllers**: Prima REST zahtjeve od frontenda i proslijeđuje ih servisnom sloju. - **Services**: Implementira poslovnu logiku aplikacije. - **Repositories**: Upravlja pristupom podacima u bazi putem JPA (Java Persistence API).

### 4. PostgreSQL baza podataka

- Služi za trajnu pohranu podataka.
- Backend pristupa bazi koristeći JPA Repositories za izvršavanje upita.

## Interakcije

1. **Korisnik preko web preglednika** šalje zahtjev aplikaciji putem korisničkog sučelja u web pregledniku.
2. **Frontend (React)** prima korisnički zahtjev i kreira odgovarajući REST zahtjev prema backendu putem HTTP protokola.
3. **Backend (Spring Boot)** obrađuje zahtjev kroz kontrolere i proslijeđuje ga servisnom sloju gdje se provodi poslovna logika.
4. **Servisni sloj backend-a** koristi repozitориjski sloj za dohvat podataka ili izvršavanje operacija nad **PostgreSQL bazom podataka**.
5. **PostgreSQL baza podataka** vraća podatke ili potvrdu o izvršenju operacije natrag repozitориjskom sloju.
6. **Backend** formatira i priprema odgovor te ga vraća frontendu kroz REST API u obliku JSON odgovora.
7. **Frontend** prima podatke, ažurira korisničko sučelje i prikazuje informacije korisniku.

## Dijagram razmještaja instanci

Aplikacija **ReStore-Films** postavljena je na Render platformi, organizirana u **trorazinsku arhitekturu**:

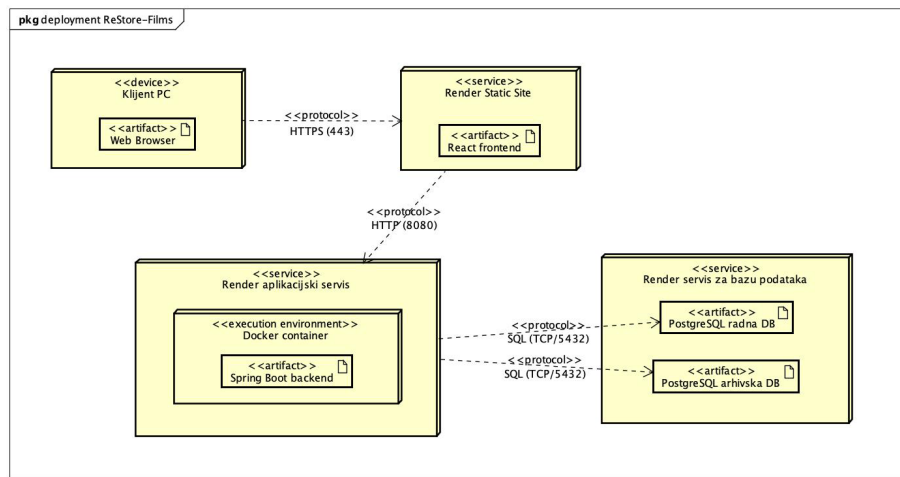


Figure 12: image

## 1. Prezentacijski sloj (Frontend)

- Implementiran u **Reactu** i smješten na **Render Static Site** servisu.
- Globalno je dostupan putem **HTTPS protokola** na portu **443**.
- Prezentacijski sloj šalje **HTTP zahtjeve** prema backendu.

## 2. Poslovni sloj (Backend)

- Backend je implementiran u **Spring Bootu** i pokreće se unutar **Docker kontejnera** na Render servisu.
- Odgovara na zahtjeve koje prima od frontenda putem **HTTP protokola** na portu **8080**.
- Poslovni sloj obrađuje složene logičke procese.

## 3. Podatkovni sloj (Baze podataka)

- Sastoji se od **dvije PostgreSQL baze podataka**: radne i arhivske baze, obje smještene na Render servisu.
- Backend komunicira s bazama podataka putem **SQL upita**, koristeći **TCP protokol** na portu **5432**.

## Ispitivanje komponenti

### 1. Dodavanje korisnika (redovan slučaj)

- **Funkcionalnost:** Dodavanje korisnika u sustav.
- **Ulazni podaci:**
  - Ime: "Ivan"
  - Prezime: "Ivic"
  - E-mail: "ii12345@fer.hr"
- **Očekivani rezultat:** Korisnik je uspješno dodan u sustav.
- **Dobiveni rezultat:** Test prošao ( ).

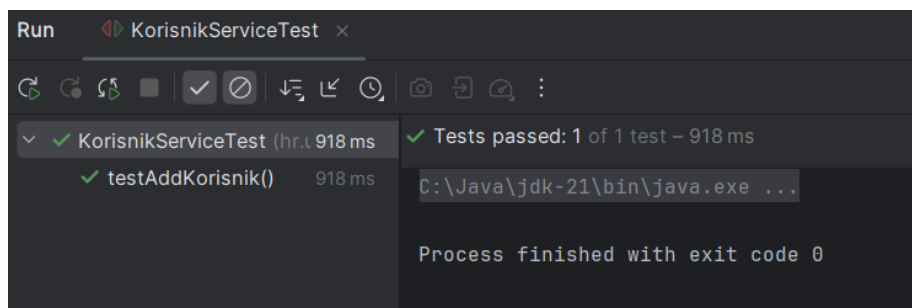


Figure 13: image

- **Opis postupka:**
  1. Simuliran unos podataka za korisnika.
  2. Pozvana metoda `addKorisnik` servisa.
  3. Provjereno da je korisnik dodan u bazu.

### 2. Dodavanje korisnika s dupliciranim e-mailom

**Funkcionalnost:** Onemogućiti dodavanje korisnika s postojećim e-mailom.

**Ulazni podaci:** - Postojeći korisnik u sustavu: - E-mail: `ii12345@fer.hr` - Ime: Ivan - Prezime: Ivić - Novi korisnik: - E-mail: `ii12345@fer.hr` - Ime: Ana - Prezime: Anić

**Očekivani rezultat:** - Izbacuje se iznimka s porukom: "Vec postoji korisnik s tim emailom!!!".

**Dobiveni rezultat:** - Test **nije prošao** ( ): Iznimka je izbačena s očekivanom porukom.



**Postupak provođenja ispitivanja:** 1. Simulirano je da u repozitoriju postoji korisnik s e-mailom `ii12345@fer.hr`. 2. Pokušaj dodavanja novog korisnika s istim e-mailom. 3. Provjereno je da metoda `save` nije pozvana. 4. Test potvrđuje da implementacija ispravno sprječava duplikate.

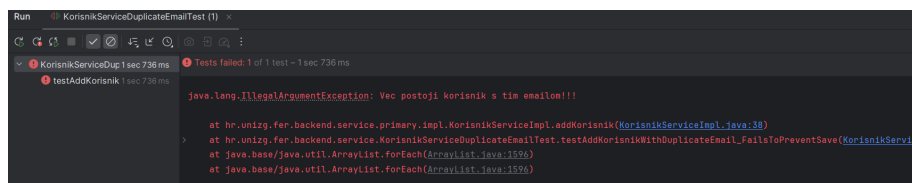


Figure 14: image

### 3. Dohvaćanje korisnika po nepostojećem e-mailu (iznimka)

- **Funkcionalnost:** Dohvaćanje korisnika po e-mailu koji ne postoji u sustavu.
- **Ulazni podaci:**
  - E-mail: `no12345@fer.hr` (nepostojeći u bazi).
- **Očekivani rezultat:**
  - Poruka: "Korisnik s e-mailom `no12345@fer.hr` ne postoji."
- **Dobiveni rezultat:**
  - Test **nije prošao** ( ): izbacuje iznimku `NoSuchElementException`.

#### Postupak provođenja ispitivanja

1. Simuliramo scenarij gdje servis za korisnike (`KorisnikService`) izbacuje iznimku `NoSuchElementException` za nepostojeći e-mail.
2. Pozivamo kontroler metodu `getKorisnikByEmail` s nepostojećim e-mailom.
3. Provjeravamo:
  - Poruka odgovara očekivanom sadržaju: "Korisnik s e-mailom `no12345@fer.hr` ne postoji."
4. Test validira da aplikacija ispravno obrađuje zahtjeve za nepostojeće korisnike.

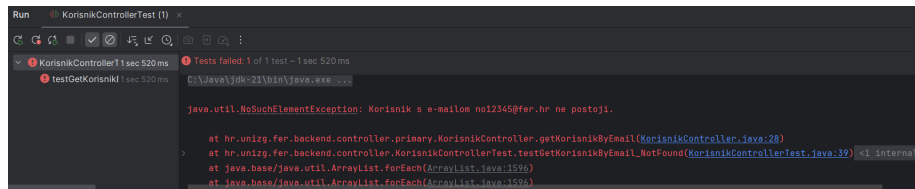


Figure 15: image

#### 4. Ispitni slučaj: Dohvaćanje filmske trake po ID-u (redovan slučaj)

- **Funkcionalnost:** Dohvaćanje filmske trake na temelju ID-a.
- **Ulazni podaci:**
  - ID: 1
- **Očekivani rezultat:** Filmska traka s odgovarajućim ID-om uspješno dohvaćena iz baze podataka.
- **Dobiveni rezultat:** Test prošao ( ).

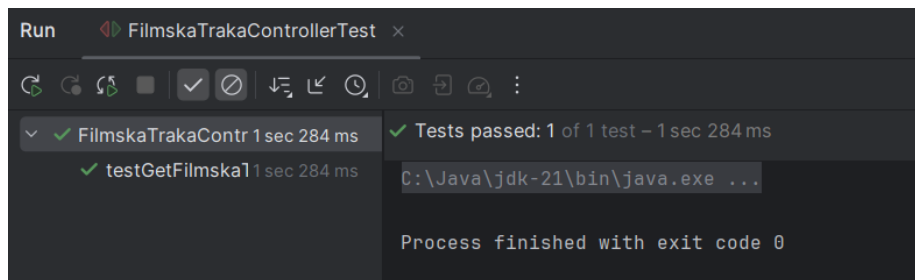


Figure 16: image

- **Opis postupka:**
  1. Pripremljena simulacija za servis `getFilmskaTrakaById` koji vraća filmsku traku s ID-om 1.
  2. Pozvana metoda kontrolera `getFilmskaTrakaById`.
  3. Provjereno je da je odgovor HTTP 200 OK te da vraćeni objekt odgovara očekivanom.

#### 5. Test za dodavanje grupe s nepostojećim filmom

##### Funkcionalnost

Testira stvaranje grupe za digitalizaciju kada grupa sadrži film koji ne postoji u bazi podataka.

## Ispitni slučaj

- **Ulazni podaci:**
  - Naziv filma: "Nepostojeći Film" (film koji nije prisutan u bazi).
- **Očekivani rezultat:**
  - Test baca iznimku `NoSuchElementException` s porukom: "Ne postojeći film u bazi imena: Nepostojeći Film !".
- **Dobiveni rezultat:**
  - Test je pao ( ) jer je metoda `createGroup` bacila iznimku, što je očekivano ponašanje za ovaj ispitni slučaj.

## Postupak provođenja ispitivanja

- Priprema ulaznih podataka:**
  - Kreirana je grupa za digitalizaciju koja sadrži naziv filma "Nepostojeći Film".
- Izvršavanje metode:**
  - Pozvana je metoda `createGroup` iz servisa.
  - Metoda je pokušala provjeriti postojanje filma i bacila iznimku `NoSuchElementException`.

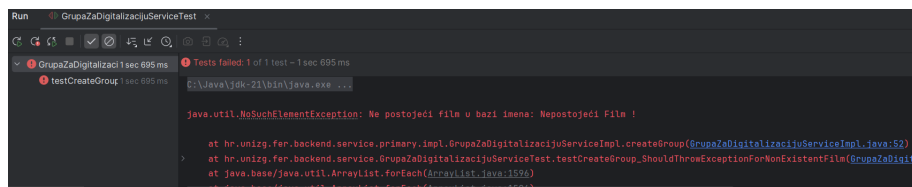


Figure 17: image

## 6. Test za dodavanje filmske trake s neispravnim podacima

### Funkcionalnost

Testira ponašanje metode `addFilmskaTraka` kada se pokušava dodati filmska traka s neispravnim podacima.

## Ispitni slučaj

- **Ulazni podaci:**
  - BarKod: null
  - Originalni naslov: "" (prazno)
  - Radni naslov: null
  - Godina proizvodnje: 1800
  - Trajanje: 00:00 (neispravno trajanje)
- **Očekivani rezultat:**
  - Iznimka `IllegalArgumentException` s porukom: "Jedan/vise argumenata -> NEDOZVOLJENA NULL vrijednost".
- **Dobiveni rezultat:**
  - Test pada i baca iznimku prilikom pokretanja.

## Postupak provođenja ispitivanja

1. **Priprema ulaznih podataka:**
  - Kreirana je instanca klase `FilmskaTrakaArhiva` s neispravnim podacima.
2. **Izvršavanje metode:**
  - Pozvana je metoda `addFilmskaTraka` s pripremljenim ulaznim podacima.

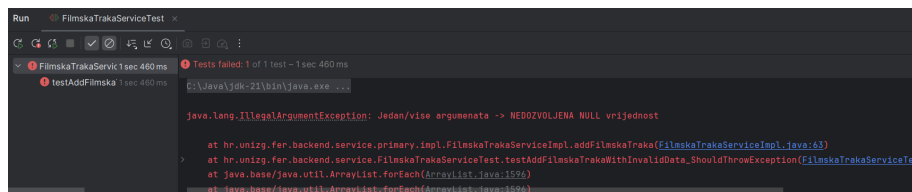


Figure 18: image

# Ispitivanje sustava

## Selenium E2E testovi

### 1. Uvod

Za testiranje prijave korisnika putem Microsoft OAuth2 autentifikacije korišten je Selenium WebDriver. Testovi su osmišljeni kako bi pokrili osnovne funkcionalnosti prijave, rubne uvjete te kako sustav reagira na neispravne ili granične ulaze.

### Alati za pokretanje testa

- **Selenium WebDriver:** Koristi se za simulaciju korisničkih akcija u pregledniku.
- **TestNG:** Za organizaciju i izvođenje testnih slučajeva.
- **ChromeDriver:** Potreban za pokretanje testova na pregledniku Google Chrome.

### Postavke

#### 1. Preuzimanje ChromeDriver-a:

- Preuzmite odgovarajući ChromeDriver za svoj operacijski sustav.
- Postavite stazu do `chromedriver` izvršne datoteke.

#### 2. Preuzimanje IntelliJ IDEA:

- Download IntelliJ IDEA i slijedite upute za instalaciju.

#### 3. Pokretanje testova:

- Idite do direktorija u kojem je ChromeDriver.
- Pokrenite ChromeDriver: Na Macu ili Linuxu, možete pokrenuti ChromeDriver s naredbom: `./chromedriver`. Na Windowsu u terminalu upišite: `chromedriver.exe`.
- Pokrenite testove u IntelliJ IDEA: Desni klik na testnu klasu (npr. `BarcodeScannerTest`) u projektu.
- Odaberite Run 'BarcodeScannerTest' ili kliknite ikonu za pokretanje (zelena strelica) pored naziva testne klase.

## 2. Testni slučajevi

### LoginTest

#### Ispravna prijava

**Opis:** Test provjerava uspješnu prijavu korisnika uz točno korisničko ime i lozinku.

**Ulazi:** - Korisničko ime: `pm54174@fer.unizg.hr` - Lozinka: `*****`

Lozinka je zaštićena zbog implementacije Microsoft Auth 2.0 autentifikacije pa je korisnički unos lozinke siguran i neće biti izložen u javnim uputama. Podaci se obrađuju putem sigurne autentifikacije čime se jamči privatnost i sigurnost korisnika.

**Koraci:** 1. Otvoriti početnu stranicu aplikacije: `https://restore-films-frontend.onrender.com/`. 2. Kliknuti na gumb “Sign in”. 3. Unijeti korisničko ime i potvrditi klikom na gumb “Next”. 4. Unijeti lozinku i potvrditi klikom na gumb “Sign in”. 5. Provjera za dvostruku autentifikaciju (2FA) 6. Provjeriti je li korisnik preusmjeren na `/home`.

#### Očekivani rezultat:

Korisnik je uspješno prijavljen i preusmjeren na početnu stranicu s URL-om `https://restore-films-frontend.onrender.com/home`.

#### Dobiveni izlaz:

Test je prošao, prikazuje se poruka “**Test Passed: Login successful.**”.

### Rubni uvjet – nevažeće korisničko ime

**Opis:** Test simulira unos neispravnog korisničkog imena. Google Auth 2.0 autentifikacija prvo zahtijeva unos ispravnog korisničkog imena. Ako uneseno korisničko ime nije prepoznato ili je pogrešno, sustav ne dopušta daljnji napredak u procesu prijave. U tom slučaju, test će biti označen kao neuspješan, jer autentifikacija ne dopušta unos lozinke bez prethodnog valjanog korisničkog imena.

**Ulazi:** - Korisničko ime: `pogresno@fer.unizg.hr`

**Koraci:** 1. Otvoriti početnu stranicu aplikacije. 2. Kliknuti na gumb “**Sign in**”. 3. Unijeti nevažeće korisničko ime i kliknuti na “**Next**”. 4. Provjeriti je li prikazana poruka o grešci.

#### Očekivani rezultat:

Poruka: “**To korisničko ime možda nije ispravno.**”.

#### Dobiveni izlaz:

Test nije prošao, prikazuje se poruka “**Test Failed: To korisničko ime možda nije ispravno.**” i baca se odgovarajući “*AssertionError*”.

## Rubni uvjet – nevažeća lozinka

**Opis:** Test provjerava reakciju sustava na unos neispravne lozinke uz točno korisničko ime.

**Ulazi:** - Korisničko ime: `pm54174@fer.unizg.hr` - Lozinka: `pogresnaLozinka`

**Koraci:** 1. Otvoriti početnu stranicu aplikacije. 2. Kliknuti na gumb “**Sign in**”. 3. Unijeti ispravno korisničko ime i kliknuti na “**Next**”. 4. Unijeti nevažeću lozinku i kliknuti na “**Sign in**”. 5. Provjeriti je li prikazana poruka o grešci.

**Očekivani rezultat:**

Poruka: “**Račun ili lozinka nisu točni.**”

**Dobiveni izlaz:**

Test nije prošao, prikazuje se poruka “**Test Failed: Račun ili lozinka nisu točni.**” i baca se odgovarajući “*AssertionError*”.

## BarcodeScannerTest

### Dodavanje novog filma s jedinstvenim imenom

**Opis testa:** U ovom testnom slučaju provodimo dodavanje filma s jedinstvenim nazivom koji još nije prisutan u bazi podataka. Očekivani je rezultat uspješno dodavanje filma u sustav i prikaz dodanog filma u listi filmova.

**Koraci ispitivanja:**

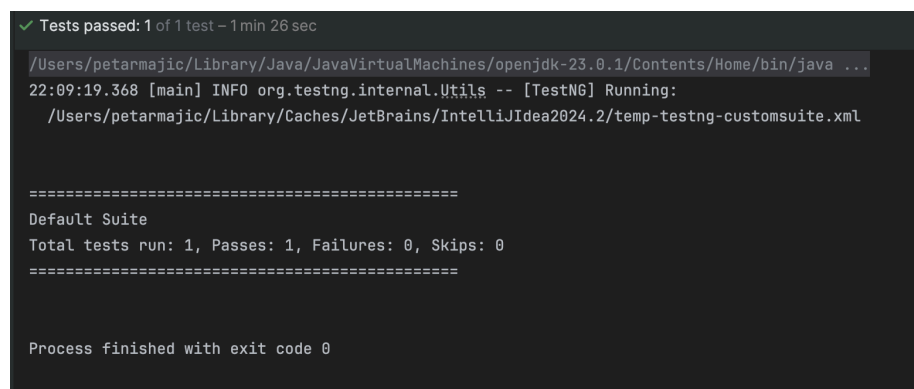
1. Otvoriti početnu stranicu aplikacije: `https://restore-films-frontend.onrender.com`.
2. Proći proces autentifikacije.
3. Pričekati da se stranica učita i URL sadrži `/home`.
4. Kliknuti na gumb “**Scan Barcode**” na stranici `/home`.
5. Kliknuti na gumb “**Manual input**” za ručni unos podataka o filmu.
6. Unijeti podatke o filmu:
  - **Originalni naslov:** Unique name
  - **Originalni jezik:** Croatian
  - **Zemlja podrijetla:** Croatia
  - **Godina proizvodnje:** 2023
  - **Trajanje:** 02:30:00
7. Kliknuti na gumb “**Add**” za dodavanje filma.
8. Provjeriti prikaz filma “**Unique name**” na popisu filmova.

#### Očekivani izlaz:

- Film “**Unique name**” uspješno se dodaje i prikazuje u popisu filmova.

#### Dobiveni izlaz:

- Test je prošao. Film “**Novi test film 2**” je uspješno dodan i vidljiv u popisu filmova.



```
✓ Tests passed: 1 of 1 test – 1 min 26 sec
/Users/petarmajic/Library/Java/JavaVirtualMachines/openjdk-23.0.1/Contents/Home/bin/java ...
22:09:19.368 [main] INFO org.testng.internal.Utils -- [TestNG] Running:
  /Users/petarmajic/Library/Caches/JetBrains/IntelliJ IDEA 2024.2/temp-testng-customsuite.xml

=====
Default Suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====

Process finished with exit code 0
```

Figure 19: image

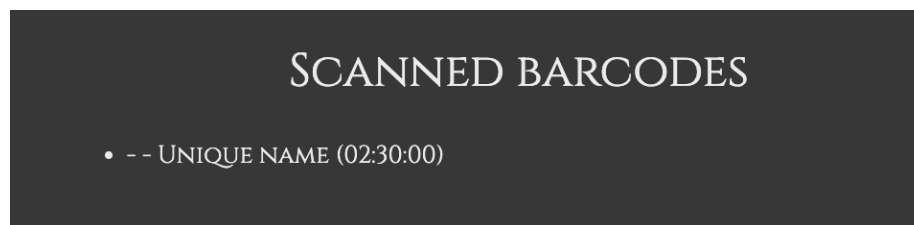


Figure 20: image

#### Pokušaj dodavanja filma s postojećim imenom

**Opis:** U ovom testnom slučaju provodimo pokušaj dodavanja filma s imenom koje postoji u bazi podataka. Očekivani je rezultat neuspješno dodavanje filma u sustav uz poruku o grešci.



### Koraci ispitivanja:

1. Otvoriti početnu stranicu aplikacije: <https://restore-films-frontend.onrender.com>.
2. Proći proces autentifikacije.
3. Pričekati da se stranica učita i URL sadrži `/home`.
4. Kliknuti na gumb **“Scan Barcode”** na stranici `/home`.
5. Kliknuti na gumb **“Manual input”** za ručni unos podataka o filmu.
6. Unijeti podatke o filmu:
  - **Originalni naslov:** Unique name (ime koje postoji u bazi)
  - **Originalni jezik:** English
  - **Zemlja podrijetla:** England
  - **Godina proizvodnje:** 2022
  - **Trajanje:** 02:00:00
7. Kliknuti na gumb **“Add”** za pokušaj dodavanja filma.
8. Provjeriti je li prikazana poruka o grešci **“Failed to add film. Please try again.”**.

### Očekivani izlaz:

- Prikazuje se poruka o grešci **“Duplicate film detected on screen”**.

### Dobiveni izlaz:

- Test nije prošao. Poruka o grešci **“Test failed: Duplicate film detected on screen”** je prikazana, a film nije dodan.

## DigitalizationTest

### Pregled:

Ovaj test provodi testiranje funkcionalnosti statusa digitalizacije filma unutar aplikacije koji je jedna od ključnih komponenti za ispravan rad sustava. Test se fokusira na odabir opcija u dropdownu za filtriranje filmova prema statusu digitalizacije što osigurava ispravno reagiranje sustava na promjene statusa.

## Filter: On Digitalization

### Koraci:

1. Otvoriti aplikaciju na URL-u: <https://restore-films-frontend.onrender.com>.

```
✖ Tests failed: 1 of 1 test – 46 sec 954 ms

/Users/petarmajic/Library/Java/JavaVirtualMachines/openjdk-23.0.1/Contents/Home/bin/java ...
22:19:23.542 [main] INFO org.testng.internal.Util$ -- [TestNG] Running:
/Users/petarmajic/Library/Caches/JetBrains/IntelliJ IDEA 2024.2/temp-testng-customsuite.xml

java.lang.AssertionError: Test failed: Duplicate film detected on screen

> at hr.unizg.fer.backend.e2e.BarcodeScannerTest.testAddFilm(BarcodeScannerTest.java:72) <10 internal lines>
> at java.base/java.util.ArrayList.forEach(ArrayList.java:1597) <6 internal lines>
> at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
> at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95) <4 internal lines>
> at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:65)
> at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:105)

=====
Default Suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====
```

Figure 21: image

2. Proći proces autentifikacije.
3. Pričekati da se učita stranica /home.
4. Kliknuti na gumb **Digitization**.
5. Kliknuti na dropdown meni za odabir statusa.
6. Odabrati opciju **On Digitalization**.
7. Proveriti je li opcija **On Digitalization** pravilno odabrana.

#### Očekivani izlaz:

- Odabrati status “On Digitalization” i provesti filtriranje.

#### Dobiveni izlaz:

- Test je prošao, aplikacija je odabrala opciju “On Digitalization” i provela filtriranje prema toj opciji.

#### Filter: Finished

#### Koraci:

1. Otvoriti aplikaciju na URL-u: <https://restore-films-frontend.onrender.com>.
2. Proći proces autentifikacije.
3. Pričekati da se učita stranica /home.
4. Kliknuti na gumb **Digitization**.



Figure 22: image

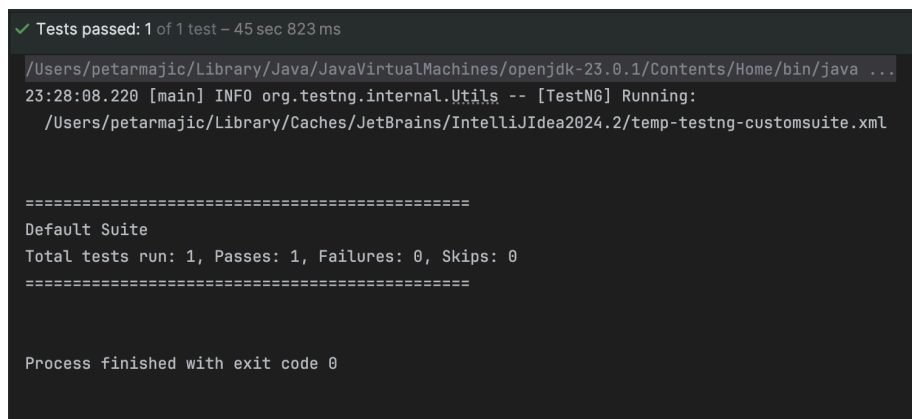


Figure 23: image

5. Kliknuti na dropdown meni za odabir statusa.
6. Odabrati opciju **Finished**.
7. Provjeriti je li opcija **Finished** pravilno odabrana.

**Očekivani izlaz:**

- Odabrati status “Finished” i provesti filtriranje.

**Dobiveni izlaz:**

- Test je prošao, aplikacija je odabrala opciju “Finished” i provela filtriranje prema toj opciji.

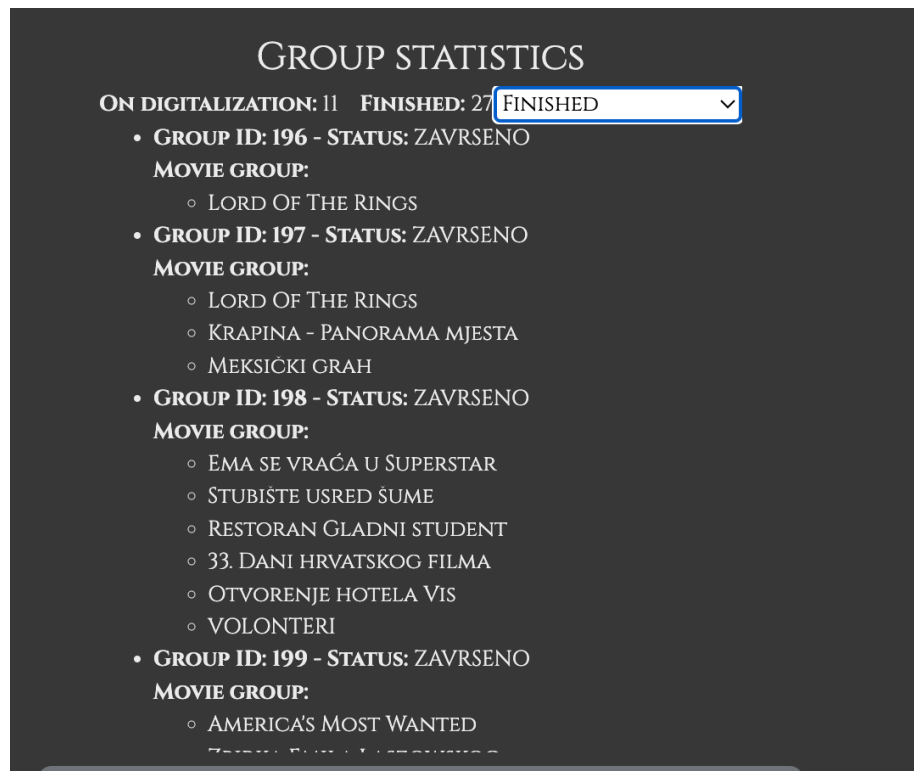


Figure 24: image

```
✓ Tests passed: 1 of 1 test – 33 sec 957 ms

/Users/petarmajic/Library/Java/JavaVirtualMachines/openjdk-23.0.1/Contents/Home/bin/java ...
23:22:37.756 [main] INFO org.testng.internal.Utils -- [TestNG] Running:
  /Users/petarmajic/Library/Caches/JetBrains/IntelliJ IDEA2024.2/temp-testng-customsuite.xml

=====
Default Suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====

Process finished with exit code 0
```

Figure 25: image

## AdminUserTest

### Promjena uloge korisniku sustava

#### Koraci:

1. Otvoriti aplikaciju na URL-u: <https://restore-films-frontend.onrender.com>.
2. Proći proces autentifikacije.
3. Čekati da se stranica učita i kliknuti na gumb “Users”.
4. Provjeriti da je stranica učitana i prikazuje listu korisnika.
5. Potražiti korisnika “Petar Jakuš” u listi korisnika.
6. Kliknuti na padajući izbornik za korisnika “Petar Jakuš” i promijeniti mu ulogu.
7. Odabrati opciju “VODITELJ”.
8. Provjeriti da je opcija “VODITELJ” uspješno odabrana.

#### Očekivani izlaz:

- Uloga korisnika “Petar Jakuš” je uspješno promijenjena u “VODITELJ”.

#### Dobiveni izlaz:

- Test je prošao. Uloga korisnika “Petar Jakuš” uspješno je promijenjena u “VODITELJ”.

## Namjerno izazivanje pogreške: Korisnik ne postoji u sustavu

### Koraci:

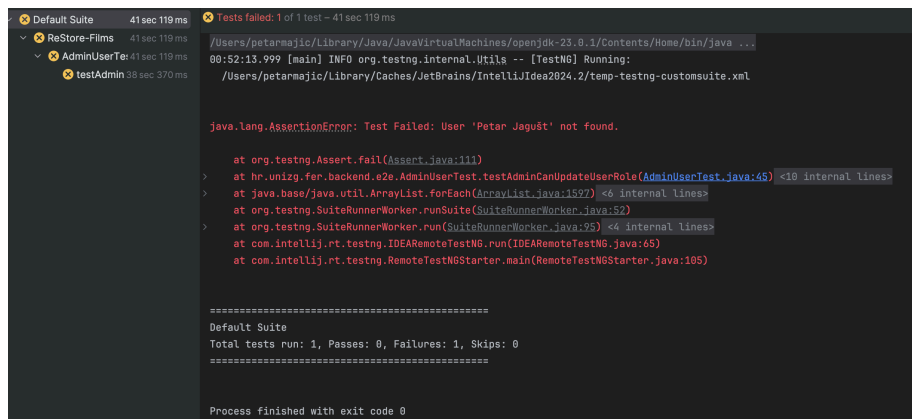
1. Otvoriti aplikaciju na URL-u: `https://restore-films-frontend.onrender.com`.
2. Proći proces autentifikacije.
3. Čekati da se stranica učita i kliknuti na gumb “Users”.
4. Provjeriti da je stranica učitana i prikazuje listu korisnika.
5. Pronaći korisnika “Petar Jaguš” u listi korisnika.
6. Ako korisnik “Petar Jaguš” nije pronađen, sustav dojavljuje odgovarajuću poruku o grešci.

### Očekivani izlaz:

- Sustav prikazuje poruku o grešci da korisnik nije pronađen.

### Dobiveni izlaz:

- Test nije prošao. Uloga za korisnika sustava “Petar Jaguš” nije promijenjena jer korisnik ne postoji u sustavu i prikazuje se poruka “**Test failed: User ‘Petar Jaguš’ not found.**”.



```
Default Suite 41 sec 119 ms
  ReStore-Films 41 sec 119 ms
  AdminUserTest 41 sec 119 ms
    testAdmin 38 sec 370 ms

Tests failed: 1 of 1 test - 41 sec 119 ms

/Users/petarmajic/Library/Java/JavaVirtualMachines/openjdk-23.0.1/Contents/Home/bin/java ...
00:52:13.999 [main] INFO org.testng.internal.Utils -- [TestNG] Running:
/Users/petarmajic/Library/Caches/JetBrains/IntelliJ2024.2/temp-testng-customsuite.xml

java.lang.AssertionError: Test Failed: User 'Petar Jaguš' not found.
    at org.testng.Assert.fail(Assert.java:111)
    at hr.unizg.fer.backend.e2e.AdminUserTest.testAdminCanUpdateUserRole(AdminUserTest.java:45) <10 internal lines>
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1577) <6 internal lines>
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95) <4 internal lines>
    at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:65)
    at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:105)

=====
Default Suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====

Process finished with exit code 0
```

Figure 26: image

## 4. Rezultati ispitivanja

Test slučaj	Očekivani rezultat	Dobiveni rezultat	Status
Prijava s točnim podacima	Korisnik preusmjeren na /home	Korisnik preusmjeren na /home	Prošao
Prijava s netočnim korisničkim imenom	Poruka o grešci	Prikazana poruka o grešci	Očekivana greška
Prijava s netočnom lozinkom	Poruka o grešci	Prikazana poruka o grešci	Očekivana greška
Dodavanje filma s jedinstvenim imenom	Film dodan na popis	Film dodan na popis	Prošao
Dodavanje filma s dupliciranim imenom	Poruka o grešci	Prikazana poruka o grešci	Očekivana greška
Filtriranje filmova: Status "On digitalization"	Prikazani odgovarajući filmovi	Prikazani odgovarajući filmovi	Prošao
Filtriranje filmova: Status "Finished"	Prikazani odgovarajući filmovi	Prikazani odgovarajući filmovi	Prošao
Promjena uloge postojećeg korisnika	Uloga korisnika uspješno promijenjena	Uloga korisnika uspješno promijenjena	Prošao
Promjena uloge korisnika koji ne postoji	Poruka o grešci	Prikazana poruka o grešci	Očekivana greška

#### Napomena o očekivanim greškama

- Testovi koji nisu prošli ( **Očekivana greška**) završili su neuspješno jer su namjerno konstruirani kako bi testirali rubne uvjete i reakciju sustava na greške.
- Sustav je u ovim slučajevima ispravno prepoznao grešku, prikazao odgovarajuću poruku i spriječio neželjene radnje.

## Korištene tehnologije i alati

### Frontend

Kao osnovni programski jezik za razvoj klijentskog dijela aplikacije korišten je **JavaScript (ES6+)**, dok je za provjeru tipova podržan i **TypeScript**. Korišten je **React** (verzija 18.3.1), popularna JavaScript biblioteka za izgradnju interaktivnih korisničkih sučelja. React omogućuje stvaranje samostalnih komponenti koje se mogu ponovno koristiti i lako ažurirati, čime se poboljšava učinkovitost razvoja. Za stiliziranje je korišten **bootstrap** (verzija 5.3.3), koji omogućava korištenje gotovih stilova, a time i brzo postavljanje dizajna uz minimalno korištenje CSS-a. Za upravljanje rutama koristi se **React Router** (verzija 6.27.0). Za IDE je izabran **Visual Studio Code**.

### Backend

Osnovni programski jezik korišten na poslužiteljskoj strani jest **Java** (verzija 21.0.4 LTS). Ona je odličan izbor zbog svoje stabilnosti i pouzdanosti, a nudi i mnoštvo razvojnih okvira i alata. Od razvojnih okvira, korišten je **Spring boot** (verzija 3.5.5) koji omogućava brzi početak razvoja, automatsku konfiguraciju te još mnoštvo pogodnosti. Za bazu podataka odabrana je **PostgreSQL** (verzija 16.2) koja je poznata po svojoj fleksibilnosti i pouzdanosti. Za IDE je izabran **IntelliJ IDEA 2024.2.0.2 (Ultimate Edition)** koji nudi automatsko dopunjavanje teksta, povezivanje na bazu te još mnogo toga, a studentima je besplatno dostupan koristeći prijavu službenim mailom.

### Ostale informacije

Za verzioniranje koda korišten je **Git** (verzija 2.45) te **GitHub**. Ispitivanje je provedeno korištenjem **Selenium**-a za automatsko ispitivanje te **JUnit5**-a za ispitivanje komponenti. Aplikacija je postavljena na **Render**-u, za koji postoji besplatna i plaćena verzija, a pritom je korišten **Docker**.

## Vodič za pokretanje i postavljanje aplikacije ReStore-Films

---

### Preduvjeti

- Node.js i npm: Node.js v20.13.1, npm 10.5.2



- **React:** React 18.3.1
- **Vite:** Vite 4.3.9
- **Maven:** Apache Maven 4.0.0
- **Docker:** Docker 27.3.1
- **PostgreSQL klijent psql:** PostgreSQL 14.15
- **Git:** Git 2.47.0
- **IntelliJ IDEA**
- **Visual Studio Code**

## Kloniranje repozitorija

```
git clone https://github.com/petarmajic/ReStore-Films.git
```

---

## 1. Pokretanje frontenda

**Koraci za pokretanje lokalnog okruženja frontenda:**

### 1. Instalacija ovisnosti:

- Pozicionirajte se u direktorij IzvorniKod/frontend: `cd ReStore-Films/IzvorniKod/frontend`
- U direktoriju **frontend** ovisnosti se instaliraju naredbom:

```
npm install
```

### 2. Pokretanje lokalnog razvojnog poslužitelja:

Za pokretanje frontend aplikacije lokalno koristi se naredba:

```
npm run dev
```

### 3. Pristup frontend aplikaciji:

Nakon pokretanja, aplikacija je dostupna na URL-u:

```
http://localhost:5173
```

### 4. Globalni pristup frontend aplikaciji:

Frontend aplikacija postavljena je na Render platformi i dostupna globalno putem URL-a:

```
https://restore-films-frontend.onrender.com
```

---

## 2. Buildanje frontenda za Render

Frontend aplikacija je uspješno postavljena na Render, a proces uključuje:

1. **Izrada build verzije:**

Build verzija kreira se naredbom:

```
npm run build
```

2. **Objava na Renderu:**

- Build naredba:

```
npm run build
```

- Korišten je direktorij `dist` kao “publish directory”.  
Render automatski koristi ovaj direktorij za posluživanje statičkih datoteka.
- 

## 3. Pokretanje backenda

Koraci za lokalno pokretanje backend aplikacije:

1. **Dodavanje varijabli okruženja u IntelliJ IDEA:**

U IntelliJ IDEA, u konfiguraciji pokretanja (Edit Configurations) za BackendApplication potrebno je dodati varijable okruženja (Environment Variables):

```
APP_BACKEND_URL=http://localhost:8080
APP_FRONTEND_URL=http://localhost:5173
SPRING_DATASOURCE_PRIMARY_PASSWORD=qgo7HGZcadzx0s1jKVRas13i5ZUTrqQa
SPRING_DATASOURCE_PRIMARY_URL=jdbc:postgresql://dpg-csrht952ng1s7389ftqg-a.oregon-postgre
sslmode=require&charSet=UTF8
SPRING_DATASOURCE_PRIMARY_USERNAME=primarydb_e9xm_user
SPRING_DATASOURCE_SECONDARY_PASSWORD=tndoKaPQmIipjwHrkgAGLn0JgH97D4jK
SPRING_DATASOURCE_SECONDARY_URL=jdbc:postgresql://dpg-csrieht2ng1s7389lppg-a.oregon-postgre
sslmode=require&charSet=UTF8
SPRING_DATASOURCE_SECONDARY_USERNAME=secondarydb_user
```

2. **Pokretanje backend poslužitelja:**

Za pokretanje aplikacije lokalno u IntelliJ IDEA, koristi opciju “Run” i pokreni BackendApplication.

### 3. **Pristup backend aplikaciji:**

Backend aplikacija dostupna je na URL-u:

```
http://localhost:8080
```

### 4. **Globalni pristup backend aplikaciji:**

Backend aplikacija postavljena je na Render platformi i dostupna globalno putem URL-a:

```
https://restore-films-backend.onrender.com
```

---

## 4. **Buildanje Docker slike za Render**

### **Docker konfiguracija za backend:**

Backend aplikacija na Renderu postavljena je kao Docker aplikacija. Proces uključuje:

#### 1. **Izrada Docker slike:**

Docker slika kreirana je naredbom iz root direktorija:

```
docker build -t restore-films-app -f IzvorniKod/backend/Dockerfile .
```

#### 2. **Testiranje Docker kontejnera lokalno:**

Docker slika testirana je lokalno s pomoću:

```
docker run -p 8080:8080 restore-films-backend
```

#### 3. **Postavljanje na Render:**

Na Render platformi korišten je `Dockerfile`. Render je automatski prepoznao Docker konfiguraciju i uspješno postavio aplikaciju.

---

## 5. **Globalne baze podataka**

Baze podataka korištene su za pohranu podataka aplikacije i povezane su s backend aplikacijom na Render platformi.

Pristup globalnim bazama podataka:

## 1. macOS/Linux terminal:

Za pristupanje radnoj (primarnoj) bazi podataka na Renderu:

```
PGPASSWORD=qgo7HGZcadzx0s1jKVRas13i5ZUTrqQa psql -h dpg-csrht952ng1s7389ftqg-a.oregon-postgres-render.com
```

Za pristupanje arhivskoj (sekundarnoj) bazi podataka na Renderu:

```
PGPASSWORD=tndoKaPQmIipjwHrkgAGLn0JgH97D4jK psql -h dpg-csrieht2ng1s7389lppg-a.oregon-postgres-render.com
```

## 2. Windows terminal:

Za pristupanje radnoj (primarnoj) bazi podataka na Renderu:

```
SET PGCLIENTENCODING=UTF8
set PGPASSWORD=qgo7HGZcadzx0s1jKVRas13i5ZUTrqQa
psql -h dpg-csrht952ng1s7389ftqg-a.oregon-postgres.render.com -U primarydb_e9xm_user primarydb
```

Za pristupanje arhivskoj (sekundarnoj) bazi podataka na Renderu:

```
SET PGCLIENTENCODING=UTF8
set PGPASSWORD=tndoKaPQmIipjwHrkgAGLn0JgH97D4jK
psql -h dpg-csrieht2ng1s7389lppg-a.oregon-postgres.render.com -U secondarydb_user secondarydb
```

---

# Opis pristupa aplikaciji na javnom poslužitelju

## Pristup aplikaciji

Aplikacija je dostupna putem javnog linka na Renderu: Restore Films Frontend

### Koraci za pristup:

1. Otvorite navedeni URL u internetskom pregledniku.
2. Prijavite se putem OAuth2 autentifikacije. Autorizacija korisnika omogućuje siguran pristup aplikaciji.
3. Nakon prijave, sustav automatski dodjeljuje ulogu korisniku i postavlja je na **Djelatnik**.
4. **Administrator** sustava može promijeniti ulogu korisnika u **Djelatnik** ili **Voditelj** ovisno o privilegijama korisnika.

## Upute za korištenje aplikacije

### 1. Uloga: Djelatnik

Korak	Akcija	Opis
1	Otvorite aplikaciju	Pristupite aplikaciji putem linka.
2	Skenirajte bar kodove	Kliknite na opciju za otvaranje kamere i skeniranje bar kodova s filmskih traka.
3	Učitavanje podataka	Ako bar kod nije dostupan: <ul style="list-style-type: none"><li>- Kliknite gumb za učitavanje <b>XML datoteke</b>.</li><li>- Ako podaci nisu dostupni ni putem XML-a, omogućeno je ručno unositi podatke.</li></ul>
4	Pregledajte učitane filmske trake	Kliknite na gumb <b>List</b> : <ul style="list-style-type: none"><li>- <b>Scanned barcodes</b>: Prikazuje sve skenirane i učitane filmske trake.</li><li>- <b>Grouped barcodes</b>: Prikazuje filmske trake pripremljene za digitalizaciju kroz grupiranje.</li></ul>
5	Grupirajte filmske trake	Filmske trake automatski se grupiraju prema sljedećim pravilima: <ul style="list-style-type: none"><li>- Trake se sortiraju po trajanju (od najvećeg prema najmanjem).</li><li>- Grupiranje osigurava da svaka grupa traje maksimalno <b>45 minuta</b>.</li><li>- Duže trake automatski se dijele na manje segmente s oznakama ("1. dio", "2. dio").</li></ul>
6	Iznesite filmske trake iz skladišta i generirajte PDF za digitalizaciju	Kliknite na gumb <b>Digitization</b> za preuzimanje PDF dokumenta koji sadrži: <ul style="list-style-type: none"><li>- Evidenciju grupiranih filmskih traka koje treba iznijeti iz skladišta.</li><li>- Napomena: Potreban je <b>potpis</b> djelatnika koji preuzima kutije iz skladišta.</li></ul>
7	Unesite filmske trake u arhiv i generirajte PDF	Kliknite na opciju <b>Archive</b> i slijedite korake: <ul style="list-style-type: none"><li>- Kliknite na gumb <b>Load</b> za učitavanje PDF-a koji sadrži evidenciju grupa iznesenih iz skladišta.</li><li>- Kliknite na gumb <b>Archive</b> za generiranje novog PDF-a s potvrdom o unosu grupa u skladište.</li></ul>

Korak	Akcija	Opis
		- Napomena: Potreban je <b>potpis</b> djelatnika koji unosi kutije u skladište.

## 2. Uloga: Voditelj

Korak	Akcija	Opis
1	Otvorite aplikaciju	Pristupite aplikaciji putem linka.
2	Pregledajte sve skenirane filmske trake	Kliknite na gumb <b>Digitization</b> kako biste pregledali sve filmske trake koje su djelatnici unijeli u sustav.
3	Uredite podatke o filmskim trakama	U okviru <b>Movie list</b> :  - Odaberite filmsku traku koja još nije napustila skladište. - Uredite potrebne podatke i spremite promjene.
4	Pregledajte grupe filmskih traka i njihove statuse	U okviru <b>Group statistics</b> :  - Pregledajte sve grupe filmskih traka s prikazom njihovih statusa ( <b>Na digitalizaciji, Završeno</b> ). - Koristite funkciju filtriranja kako biste lakše pregledali grupe prema statusima.
5	Generirajte statistiku grupa	Kliknite na opciju za preuzimanje statistike u PDF formatu: - Statistika prikazuje broj grupa koje su trenutno na digitalizaciji i broj grupa koje su završene i vraćene u skladište.
6	Pregledajte podatke o djelatnicima	Kliknite na gumb <b>Employees</b> kako biste pregledali rad djelatnika u sustavu: - Prikazuje se popis djelatnika s detaljima o njihovom radu, uključujući: - Broj grupa koje je djelatnik iznio iz skladišta. - Broj grupa koje je djelatnik unio u skladište.
7	Generirajte statistiku rada djelatnika	Kliknite na opciju za preuzimanje PDF-a sa statistikom o radu djelatnika:

Korak	Akcija	Opis
		- PDF prikazuje detaljne podatke o doprinosu svakog djelatnika sustavu, što omogućuje lakše praćenje učinkovitosti i aktivnosti zaposlenika.

### 3. Uloga: Administrator

Korak	Akcija	Opis
1	Otvorite aplikaciju	Pristupite aplikaciji putem linka.
2	Pregledajte sve korisnike sustava	Kliknite na gumb <b>Users</b> kako biste otvorili popis svih korisnika registriranih u sustavu.
3	Promijenite ulogu korisnika	U okviru <b>Users</b> :  - Pored svakog korisnika prikazana je trenutna uloga. - Kliknite na <b>drop-down menu</b> pored uloge korisnika. - Odaberite novu ulogu (npr. djelatnik, voditelj, administrator) iz ponuđenog izbornika. - Nakon odabira, promjena će se automatski spremiti.
4	Izbrišite korisnika iz sustava	U okviru <b>Users</b> :  - Pored svakog korisnika nalazi se gumb <b>Delete</b> . - Kliknite na gumb <b>Delete</b> pored korisnika kojeg želite ukloniti iz sustava.

## Zaključak

Rad na ovom projektu bio je izazov za nas sedmero studenata inženjera. Iako smo se susreli s brojnim zadacima na fakultetu, kroz ovaj projekt stekli smo važno iskustvo u razvoju aplikacija od nule. Radili smo s PostgreSQL bazama podataka, razvojem backenda u Spring Bootu i frontenda u ReactJS-u što nam je omogućilo bolje razumijevanje kako komponente aplikacije međusobno komuniciraju i kako dizajnirati učinkovitu arhitekturu.

Osim tehničkog područja, važnu lekciju naučili smo i u pogledu vođenja projekta. Koordinacija sedmero ljudi, raspodjela vremena i definicija prioriteta bili su među najvećim izazovima. S vremenom smo naučili kako raspodijeliti zadatke, kako upravljati napretkom projekta i kako se nositi s rokovima. Također, proces učenja sintakse za Spring i React bio je izazovan, no omogućio nam je da uspješno ostvarimo ciljeve.

Dinamika rada bila je ključna za uspjeh projekta. Redovito smo se sastajali uživo u studentskom domu i na fakultetu, a kad nismo mogli biti zajedno, koristili smo Discord za koordinaciju. Ova kombinacija fizičkih i online sastanaka dala nam je fleksibilnost i pomogla da ostanemo povezani kao tim. Svaki sastanak bio je prilika da razgovaramo o napretku, postavimo ciljeve i razjasnimo nesigurnosti, što je ojačalo naš timski duh. Kroz suradnju, naučili smo kako bolje komunicirati, dijeliti ideje i podržavati jedni druge, što je ključno za uspješan timski rad.

S ponosom gledamo na sve što smo postigli. Iako smo se suočili s izazovima, svaki nas je naučio novim vještinama te smo izgradili solidnu osnovu za buduće projekte. Ovo iskustvo nas je pripremio za još veće i zahtjevnije izazove u budućnosti.

---

## Budući rad - funkcionalnosti koje nisu implementirane

Kategorija	Opis
Hijerarhijska organizacija djelatnika	Voditelji imaju vlastite mape s pripadajućim djelatnicima. Djelatnici su grupirani prema svom voditelju.
Ograničenje pristupa	Voditelji ne vide druge voditelje. Voditelji nemaju pristup djelatnicima drugih voditelja.
Prikaz odnosa	Vizualni prikaz hijerarhije.

## Dnevnik promjena dokumentacije

Datum	Opis promjene	Autor
27. listopada 2024.	Inicijalna stranica	Petar Majić



Datum	Opis promjene	Autor
27. listopada 2024.	Dodan template za wiki dokumentaciju	Petar Majić
28. listopada 2024.	Dodani dionici i aktori	Petar Majić
28. listopada 2024.	Ažurirani dionici i aktori	Petar Majić
28. listopada 2024.	Dodani obrasci uporabe i dijagrami	Petar Majić
28. listopada 2024.	Dodani sekvencijski dijagrami	Petar Majić
4. studenoga 2024.	Ažurirani dijagrami obrazaca i sekvencijski dijagrami, dodana arhivska baza podataka	Lucija Kukavica
4. studenoga 2024.	Dodani 1. i 2. iz wiki predloška	Petar Majić
4. studenoga 2024.	Ažurirana Home page	Lucija Kukavica
6. studenoga 2024.	Dodani dijagrami arhivske i radne baze podataka	Lucija Kukavica
6. studenoga 2024.	Ažurirani opisi obrazaca uporabe te dodani opisi sekvencijskih dijagrama	Lucija Kukavica
6. studenoga 2024.	Ažurirani sekvencijski dijagrami te odgovarajući opisi	Zvonimir Kilić
6. studenoga 2024.	Ažurirana 2. Analiza zahtjeva	Amalija Mustapić
6. studenoga 2024.	Ažurirani nefunkcionalni zahtjevi	Amalija Mustapić
6. studenoga 2024.	Dodani funkcionalni zahtjevi za svakog dionika	Amalija Mustapić
6. studenoga 2024.	Ažuriran naziv tima	Petar Majić
6. studenoga 2024.	Ažurirani prioriteti zahtjeva	Amalija Mustapić
7. studenoga 2024.	Opisan projektni zadatak	Amalija Mustapić
7. studenoga 2024.	Dodane dodatne točke u opis zadatka	Amalija Mustapić
8. studenoga 2024.	Ažurirana 5. Arhitektura i dizajn sustava	Lucija Kukavica
9. studenoga 2024.	Ažurirana 5. Arhitektura i dizajn sustava	Lucija Kukavica
9. studenoga 2024.	Dodani opisi tablica	Lucija Kukavica

Datum	Opis promjene	Autor
12. studenoga 2024.	Dodan sekvencijski dijagram za voditelja	Amalija Mustapić
12. studenoga 2024.	Dodan sekvencijski dijagram za voditelja te pripadajući opis	Lucija Kukavica
12. studenoga 2024.	Uređeno prema novom predlošku	Petar Majić
12. studenoga 2024.	Preraspodjela zahtjeva prema novom predlošku	Petar Majić
12. studenoga 2024.	Ažurirana 5. Arhitektura i dizajn sustava	Anastazija Čelar
12. studenoga 2024.	Dodan dijagram razreda radne baze podataka	Anastazija Čelar
13. studenoga 2024.	Ažurirani dijagrami baze podataka, opisi tablica te dodan uvodni dio	Lucija Kukavica
13. studenoga 2024.	Ažurirani dijagrami obrazaca uporabe	Lucija Kukavica
13. studenoga 2024.	Izbrisani komentari preuzeti iz predloška	Lucija Kukavica
13. studenoga 2024.	Za obrasce uporabe navedeno koje funkcionalne zahtjeve obuhvaćaju	Amalija Mustapić
13. studenoga 2024.	Dodan opis arhitekture sustava	Amalija Mustapić
15. studenoga 2024.	Ažurirana 9. stavka po predlošku	Petar Majić
15. studenoga 2024.	Popunjene točke prikaza aktivnosti grupe	Petar Majić
15. studenoga 2024.	Dodani linkovi za issue	Petar Majić
22. siječnja 2025.	Ažurirani opisi za sekvencijske dijagrame	Lucija Kukavica
22. siječnja 2025.	Ažurirani sekvencijski dijagrami	Lucija Kukavica
22. siječnja 2025.	Dodani opisi za dijagrame stanja, aktivnost i razreda	Lucija Kukavica
22. siječnja 2025.	Ispravak pravopisnih pogrešaka	Lucija Kukavica
22. siječnja 2025.	Modificirani opisi obrazaca uporabe	Lucija Kukavica
22. siječnja 2025.	Modificirani dijagrami obrazaca uporabe	Lucija Kukavica
22. siječnja 2025.	Modificirani nefunkcijski zahtjevi te aktori i njihovi funkcijski zahtjevi	Lucija Kukavica
22. siječnja 2025.	Ažuriran 4. Arhitektura i dizajn sustava	Anastazija Čelar

Datum	Opis promjene	Autor
22. siječnja 2025.	Popravljeni funkcijski zahtjevi	Lucija Kukavica
22. siječnja 2025.	Ažuriran opis zadatka	Lucija Kukavica
22. siječnja 2025.	Modificirani dijagram razreda i dijagrami baze podataka te opisi	Anastazija Čelar
23. siječnja 2025.	Ažurirana 2. Analiza zahtjeva	Anastazija Čelar
23. siječnja 2025.	Dodan dijagram razreda (Controllers)	Anastazija Čelar
23. siječnja 2025.	Modificirani dijagram razreda i dijagrami baze podataka	Anastazija Čelar
23. siječnja 2025.	Napravljen dijagram komponenata	Petar Majić
23. siječnja 2025.	Ažurirani prioriteti zahtjeva	Amalija Mustapić
23. siječnja 2025.	Napravljen dijagram razmjesta	Petar Majić
24. siječnja 2025.	Dodani dijagrami stanja i aktivnosti	Anastazija Čelar
24. siječnja 2025.	Ažurirana 3. Specifikacija zahtjeva sustava	Anastazija Čelar
24. siječnja 2025.	Ažurirana 4. Arhitektura i dizajn sustava	Anastazija Čelar
24. siječnja 2025.	Ažurirana 2. Analiza zahtjeva	Anastazija Čelar
24. siječnja 2025.	Promijenjeni opisi tablica baze podataka	Anastazija Čelar
24. siječnja 2025.	Dokumentirani testovi u Seleniumu	Petar Majić
24. siječnja 2025.	Dokumentirani testovi u JUnit-u	Amalija Mustapić
24. siječnja 2025.	Dokumentirane popis tehnologija za implementaciju aplikacije	Matej Šimek
24. siječnja 2025.	Dokumentiran zaključak i budući rad	Petar Majić
24. siječnja 2025.	Dokumentirane upute za puštanje u pogon	Petar Majić
24. siječnja 2025.	Dokumentiran vodič za korištenje	Petar Majić
24. siječnja 2025.	Ažuriran popis literature	Petar Majić
24. siječnja 2025.	Dokumentiran prikaz aktivnosti grupe	Petar Majić

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>

5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

## Dnevnik sastajanja

### 1. sastanak

- **Datum:** 16. listopada 2024.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* upoznavanje i dogovor zajedničke komunikacije > \* uspostavljena WhatsApp i Discord grupa svih članova > \* raspodjela članova na backend i frontend dio, rasprava o preferencijama svakog člana

### 2. sastanak

- **Datum:** 22. listopada 2024.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* napraviti prezentaciju. ideje izrade aplikacije > \* postaviti GIT repozitorij sa svim članovima i predviđenom organizacijom: <https://github.com/petarmajic/ReStore-Films/issues/18> > \* definirati i dokumentirati funkcijske i nefunkcijske zahtjeve aplikacije: <https://github.com/petarmajic/ReStore-Films/issues/19> > \* ažurirati dokumentaciju prema predlošku

### 3. sastanak

- **Datum:** 28. listopada 2024.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* postaviti testni commit na repozitorij: <https://github.com/petarmajic/ReStore-Films/issues/1> > \* napraviti dijagrame obrazaca uporabe i sekvencijske dijagrame: <https://github.com/petarmajic/ReStore-Films/issues/2> > \* napraviti inicijalizaciju backend dijela aplikacije: <https://github.com/petarmajic/ReStore-Films/issues/5> > \* ažurirati dokumentaciju prema predlošku

### 4. sastanak

- **Datum:** 05. studenoga 2024.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* dizajnirati radnu bazu podataka i pripremiti potrebne entitete, atribute i relacije: <https://github.com/petarmajic/ReStore-Films/issues/6> > \* napraviti frontend dio aplikacije za login i skeniranje

barkoda: <https://github.com/petarmajic/ReStore-Films/issues/20>  
> \* napraviti backend dio aplikacije za login i učitavanje metapodataka za skenirani barkod: <https://github.com/petarmajic/ReStore-Films/issues/21> > \* ažurirati dokumentaciju prema predlošku

#### 5. sastanak

- **Datum:** 12. studenoga 2024.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* napraviti responzivni dizajn aplikacije: <https://github.com/petarmajic/ReStore-Films/issues/22> > \* prebaciti implementaciju radne baze iz H2 u Postgres bazu i napraviti dijagrame razreda: <https://github.com/petarmajic/ReStore-Films/issues/23> > \* ažurirati dokumentaciju prema predlošku

#### 6. sastanak

- **Datum:** 19. prosinca 2024.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* Implementacija kontrolera za filmske trake (CRUD operacije) > \* Kreiranje repozitorija za filmske trake > \* Implementacija servisa za poslovnu logiku (grupiranje filmskih traka prema trajanju, dodavanje metapodataka) > \* Razrada sučelja za voditelja: pregled filmskih traka, filtriranje, i uređivanje > \* Razrada sučelja za administratora: pregled i mijenjanje uloge > \* Razrada sučelja za djelatnika: skeniranje bar kodova, unos XML datoteka, grupiranje filmskih traka > \* Dokumentacija: Ažurirati dokumentaciju prema implementiranim funkcionalnostima

#### 7. sastanak

- **Datum:** 28. prosinca 2024.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* Implementacija funkcionalnosti za grupiranje filmskih traka prema trajanju (s maksimalnim trajanjem od 45 minuta po grupi) > \* Implementacija arhiviranja filmskih traka > \* Završavanje sučelja za djelatnika, omogućiti grupiranje filmskih traka i generiranje PDF-a za digitalizaciju > \* Implementacija sučelja za voditelja, omogućiti pregled statistike rada djelatnika > \* Dokumentacija: Ažurirati dokumentaciju prema novim funkcionalnostima

#### 8. sastanak

- **Datum:** 05. siječnja 2025.

- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* Završavanje svih CRUD operacija za filmske trake, korisnike, i digitalizaciju > \* Završavanje sučelja za voditelja, omogućiti pregled statistike rada djelatnika > \* Završavanje sučelja za administratora, dodati brisanje korisnika sustava

#### 9. sastanak

- **Datum:** 12. siječnja 2025.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* Završavanje svih API-ja za sve funkcionalnosti > \* Završna optimizacija servisa > \* Završavanje svih sučelja

#### 10. sastanak

- **Datum:** 18. siječnja 2025.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* Implementacija svih testova za backend (unit testovi za kontrolere, servise i repozitorije) > \* Završavanje testova za frontend (E2E testovi u Seleniumu) > \* Dokumentacija: Završavanje dokumentacije, izrada dijagrama arhitekture i dizajna sustava, dijagrama komponenta i razmjesta

#### 11. sastanak

- **Datum:** 22. siječnja 2025.
- **Prisustvovali:** svi članovi tima
- **Teme sastanka:** > \* Posljednje provjere za sve API-je i sučelja i analiza testiranja > \* Dokumentacija: Završavanje svih dijagrama (razreda, arhitekture, razmjesta) > \* Dokumentacija: Završavanje korisničkih i tehničkih uputa za aplikaciju

## Tablica aktivnosti

Aktivnost	Petar	Lucija		Anastazija		Amalija
	Matijević	Matej Šimek	Kukavica	Češćić	Zvonimir Kilić	Mustapić
Upravljanje projektom	15h					
Opis projektnog zadatka			1h	1h		5h
Funkcionalni zahtjevi	2h	2h	2h	2h		6h
Opis pojedinih obrazaca	2h	2h	2h			2h

Aktivnost	Petar Ma- jić	Matej Šimek	Lucija Kukav- ica	Anastazija Če- lar	Zvonimir Kilić	Irean Vuk	Amalija Mustapić
Dijagram obrazaca			3h	3h	3h		
Sekvencijski dijagrami			3h	3h	3h		3h
Opis ostalih zahtjeva							4h
Arhitektura i dizajn sustava	1h	3h	10h	10h		2h	2h
Baza podataka			30h	30h			
Dijagram razreda			5h	15h			
Dijagram stanja			3h	3h			
Dijagram aktivnosti			3h	3h			
Dijagram komponenti	3h						
Korištene tehnologije i alati		2h					
Ispitivanje programskog rješenja	20h						30h
Dijagram razmještaja	3h						
Puštanje u pogon	25h						
Dnevnik sastajanja	1h						
Zaključak i budući rad	1h						
Popis literature	1h						
Izrada frontend dijela aplikacije u React okruženju					90	150	
Izrada backend dijela aplikacije u SpringBoot okruženju	15h	120h	40h	40h			
Izrada baze podataka			5h	5h			
Spajanje s bazom podataka	10h	2h					
Izrada prezentacije	3h						

## Dijagram pregleda promjena

## Ključni izazovi i rješenja

### Izazovi:

Tijekom razvoja projekta, suočili smo se s nekoliko izazova koji su utjecali na brzinu napretka. Prvi problem nastao je zbog kašnjenja u konfiguraciji dviju baza podataka – arhivske i radne. Osim toga, migracija s H2 baze na Post-



Figure 27: image



greSQL donijela je dodatne poteškoće koje smo morali prevladati. Drugi izazov bio je loše planiranje početka kodiranja, što je dovelo do kašnjenja u razvoju i spriječilo nas da odmah započnemo s ključnim funkcijama aplikacije. Treći izazov pojavio se s promjenom dijagrama razreda i dizajna baze podataka. Tijekom rada na projektu shvatili smo da neki dijelovi dizajna nisu bili dovoljno učinkoviti, što je zahtijevalo prilagodbe i unapređenja kako bi sustav bio optimalniji.

### **Rješenja:**

Kako bismo riješili ove izazove, morali smo se prilagoditi i implementirati dvije baze podataka u sustav. Također, prebacivanje s H2 baze na PostgreSQL zahtijevalo je dodatni napor, ali smo uspješno završili taj zadatak. Iako smo počeli s kašnjenjem, naporan rad svakog člana tima omogućio je nadoknadu izgubljenog vremena i nastavak prema planu. Kroz konstruktivnu raspravu i zajednički rad, prilagodili smo dizajn radne baze podataka i implementirali potrebne promjene.

### **Zaključno:**

Unatoč početnim izazovima, tim je pokazao iznimnu predanost. Kroz trud i zajednički rad, uspjeli smo završiti projekt na vrijeme. Naučili smo mnogo o timskom radu, prilagodbi na nove tehnologije, upravljanju projektima i kako se nositi s promjenama u dizajnu, što će nam biti od velike pomoći u budućim projektima.

## **Programsko inženjerstvo ak. god. 2024./2025.**

**Sveučilište u Zagrebu**

**Fakultet elektrotehnike i računarstva**

**ReStore-Films**

**Grupa: <TG08.1>**

**Tim:**

*Voditelj tima:* >- Petar Majić (petar.majic@fer.unizg.hr) - Full-stack programer, dokumentacija

*Backend programeri:* >- Lucija Kukavica (lucija.kukavica@fer.unizg.hr)  
>- Anastazija Čelar (anastazija.celar@fer.unizg.hr) >- Matej Šimek  
(matej.simek@fer.unizg.hr)  
*Frontend programeri:* >- Zvonimir Kilić (zvonimir.kilic@fer.unizg.hr)  
>- Amalija Mustapić (amalija.mustapic@fer.unizg.hr) >- Leon Vuk  
(leon.vuk@fer.unizg.hr)

**Nastavnik: Vlado Sruk** ReStore-Films