

# VILNIAUS UNIVERSITETAS

## MATEMATIKOS IR INFORMATIKOS

### FAKULTETAS

#### **Programų sistemų kūrimo**

#### **Videotekos techninė ataskaita**

**Darbą atliko:**

**Simonas Vingis**

**Povilas Stašys**

**Aurimas Arlauskas**

**Tautvydas Mačiulis**

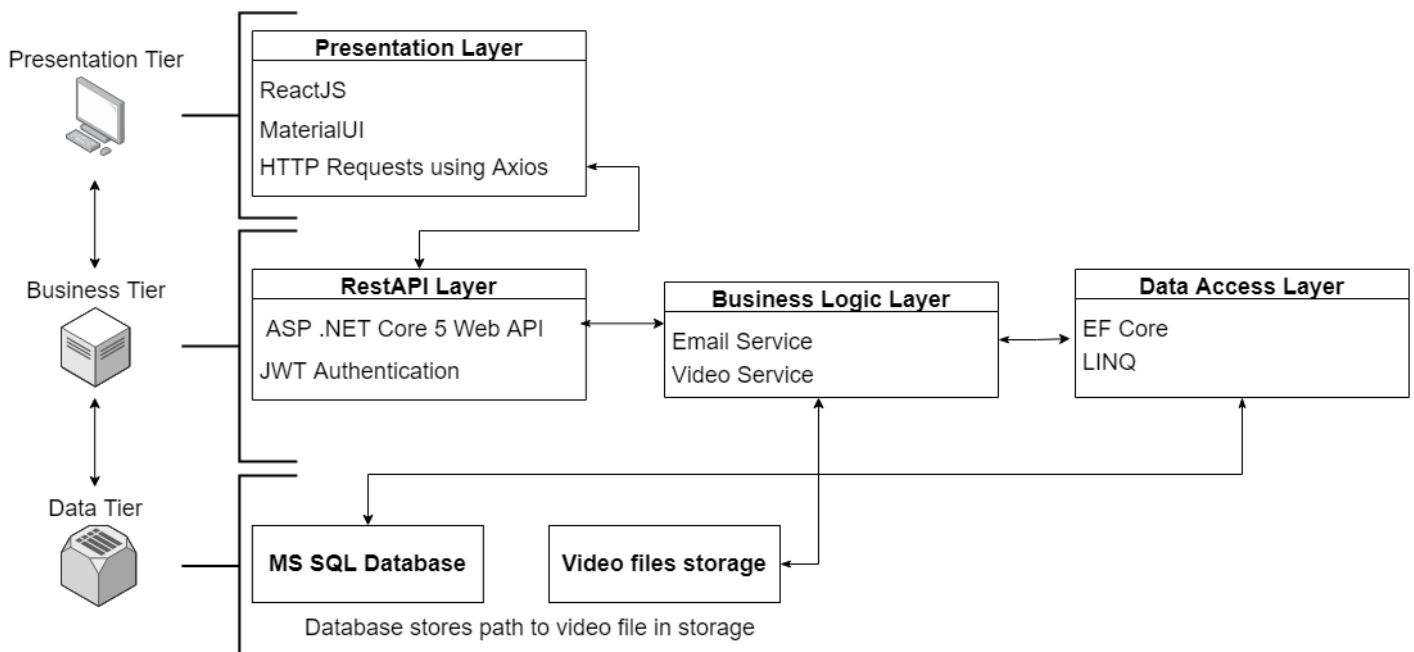
**Julius Rasimas**

# Sistemos struktūra

Projekto išeities tekstai: <https://github.com/kibartas/PSK>

Sistema suskirstyta į 3-Tier architektūrą:

- **Data Tier**
  - MS SQL duomenų bazė
  - Video failų saugykla
- **Business Tier**
  - Data Access Layer (projekte: [backend/DataAccess](#)) - skirtas bendravimui su duomenų baze:
    - ORM - .NET Core Entity Framework
    - Data Mapper – LINQ
  - Business Logic Layer (projekte: [backend/BusinessLogic](#)) - skirtas verslo logikos ir duomenų apdorojimui:
    - El. pašto laiškų siuntimo servisas
    - Vaizdo įrašų apdorojimo servisas
  - RestAPI Layer (projekte: [backend/RestAPI](#)) - skirtas paruoštų, apdorotų duomenų “on request” atidavimui vartotojo sąsajos sluoksniui:
    - ASP .NET Core 5 Web servisas
    - JWT autentikavimas
- **Presentation Tier**
  - Presentation Layer (projekte: [frontend](#)) - skirtas jau paruoštų duomenų išsitraukimui per RestAPI ir jų atvaizdavimui:
    - Interaktyvi vartotojo sąsaja su ReactJS
    - MaterialUI
    - HTTP užklausos su Axios



## Įgyvendinti reikalavimai

Concurrency	
<b>Failas (backend):</b> <a href="#">backend/RestAPI/Controllers/UsersController.cs</a> <b>Failas (frontend):</b> <a href="#">frontend/src/containers/LoginPage/LoginPage.js</a>	Naudojama naršyklės lokali saugykla (LocalStorage), kuriame saugomas naudotojo JWT Authorization token. Vartotojas gali dirbti atsidaręs kelis skirtukus.
<b>Eilutė (backend):</b> 108	
<b>Eilutė (frontend):</b> 25	
Security	
<b>Failas:</b> <a href="#">backend/DataAccess/Repositories/Users/UsersRepository.cs</a>	Entity Framework pasirūpina apsauga nuo SQL Injection atakų. Užklauso yra daromos naudojantis LINQ, kuris nenaudoja tekstų manipuliacijos. Tai užtikrina, kad užklausa nebus manipuliuojama ir gražinamas norimas ir teisingas rezultatas.
<b>Eilutė:</b> 27	
Data Access	
<b>Failas:</b> <a href="#">backend/DataAccess/Repositories/Videos/VideosRepository.cs</a>	Vykdamos nesudėtingos operacijos su duomenų baze, todėl naudojam ORM (Entity Framework) ir Data Mapper (LINQ).
<b>Eilutė:</b> 35	
Data consistency	
<b>Failas (backend):</b> <a href="#">backend/RestAPI/Controllers/VideosController.cs</a> <b>Failas (frontend):</b> <a href="#">frontend/src/components/PlayerPage.js</a>	Vartotojui bandant keisti tą patį resursą keliuose naršyklės languose jam pranešama, kad informacija neseniai pakito ir paklausiama ar vartotojas nori užrašyti savo informaciją “ant viršaus”, ar palikti duomenų bazėje esantį resursą. To pavyzdys projekte būtų video pavadinimo keitimas keliuose languose PlayerPage informacinėje šoninėje juostoje.
<b>Eilutė (backend):</b> 376	
<b>Eilutės (frontend):</b> 114 ir 283	
Memory management	
<b>Failas (AddScoped):</b> <a href="#">backend/BusinessLogic/Registry/ServiceCollectionExtensions.cs</a> <b>Failas (AddSingleton):</b> <a href="#">backend/RestAPI/Startup.cs</a>	Tinkamai parenkami komponentų gyvavimo ciklai. AddScoped sukuria komponentą, kuris gyvuoja iki užklauso pabaigos. Antrame faile naudojamas AddSingleton, kuris gyvuoja iki programos pabaigos.
<b>Eilutė (AddScoped):</b> 14 <b>Eilutė (AddSingleton):</b> 143	
Reactive programming	
<b>Failai:</b> <a href="#">frontend/src/api/VideoAPI.js</a> <a href="#">frontend/src/containers/PlayerPage/PlayerPage.js</a>	HTTP užklauso asinchroniškai siunčiamos naudojant Axios biblioteką, kuri yra Promise-based. Dėl to puslapis išlieka responsive, net ir ilgų operacijų metu. Vienas iš reaktyvaus programavimo pavyzdžių projekte būtų video atsisiuntimas. Jei vartotojas paspaudžia atsisiuntimo mygtuką, tai kol back-end'as paruoš video failą atsisiuntimui bei kol vyks pats atsisiuntimas, vartotojas vis tiek gali laisvai naviguoti į kitus puslapius bei naudotis kitais funkcionalumais.
<b>Eilutės:</b> VideoAPI.js: 52 PlayerPage.js: 131	
Cross-cutting functionality	
<b>Failas:</b> <a href="#">backend/RestAPI/Startup.cs</a>	Visos HTTP užklauso ir jų rezultatai žurnalizuojami faile. HTTP užklauso duomenų fiksavimui naudojami ASP .NET Core Middleware. Žurnalizuojami Exceptions, kurie nėra apdorojami programiškai.
<b>Eilutė:</b> 107	

## Extensibility

**Failas:**

[backend/BusinessLogic/Registry/ServiceCollecti  
onExtensions.cs](#)

**Eilutė:** 14

Projekte naudojamas Strategy Pattern. Pridėjus naują implementaciją, reikia pakeisti naudojamą implementaciją vienoje vietoje.