

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

**Paieškos proceso ir jos rezultatų pateikimo  
vartotojams panaudojamumas VUL Santaros  
klinikų puslapyje**

**The Usability of the Search Process and Presenting its Results  
to the User for VUH Santaros klinikos website**

Kursinis darbas

Atliko:	4 kurso 3 grupės studentas	
	Tomas Kiziela	(parašas)
Darbo vadovas:	doc. Kristina Lapin	(parašas)

Vilnius – 2020

## **TURINYS**

<b>ĮVADAS .....</b>	<b>3</b>
<b>1. SISTEMOS ARCHITEKTŪROS MODELIS .....</b>	<b>4</b>
<b>2. SISTEMĄ REALIZUOJANČIOS TECHNOLOGIJOS .....</b>	<b>5</b>
2.1. Technologijos duomenų bazės valdymui.....	5
2.2. Technologijos puslapių kūrimui .....	5
2.3. Technologijos komunikuojančio kodo vykdymui.....	5
<b>REZULTATAI IR IŠVADOS .....</b>	<b>6</b>
<b>ŠALTINIAI .....</b>	<b>7</b>

# Įvadas

Uždaviniai:

1. Pasirinkti sistemos architektūros modelį
2. Palyginti ir pasirinkti technologijas reikalingas puslapio kūrimui
3. Sukurti galutinio sprendimo prototipą

# 1. Sistemos architektūros modelis

Prieš kuriant internetinį tinklą reikia apgalvoti, kokia bus sistemos architektūra. Tai nulemia įgyvendinimo sudėtingumas, populiarūs sprendimai ir naudotojų poreikiai.

Vienas iš populiariausių ir paprasčiausių architektūros modelių yra Modelis-Vaizdas-Valdiklis (Model-View-Controller, toliau MVC)[Per18]. Šis modelis sudarytas iš trijų sluoksnių: duomenų sluoksnio, vaizdo sluoksnio ir valdiklių sluoksnio. Duomenų sluoksnis atsakingas už duomenis reikalingus programos veikimui, daugeliu atveju duomenų bazę, kurioje gali būti saugomi puslapių tekstai[Sta13]. Vaizdo sluoksnis pateikia vartotojui vaizdą, pavyzdžiui puslapį ir mygtukus. Valdiklių sluoksnis skirtas komunikacijai tarp vaizdo ir modelio sluoksnių, jis priima vartotojo įvestį ir pateikia rezultatus iš duomenų bazės. Visa tai leidžia atlikti tinklapiui reikalingas funkcijas kaip duomenų, saugojimas, puslapių rodymas, paieškų atlikimas, filtravimas ir žinučių ar komentarų siuntimas. Naudojant MVC modelį kodas atskiriamas pagal sluoksnius, tai leidžia izoliuoti komponentus ir dėl to kodą lengviau plėsti, kyla mažiau klaidų bei jas lengviau pataisyti. Šios savybės palaiko objektinio programavimo metodiką ir padeda turėti aiškiai suprantamą projekto struktūrą.

Vienas svarbus punktas yra, kad sistema būtų responsyvi ir galėtų aptarnauti didelį kiekį vartotojų, tačiau tai universalus poreikis ir ši architektūra tam netrukdo. Kadangi kuriamas tinklapis neturi išskirtinių bruožų, dėl kurių reiktų galvoti naujus sprendimus, galima naudoti MVC modelį[Sta09; Sta15]. Galiausiai, autorius jau turi patirties su šiuo modeliu, taigi nereikės mokytis šio modelio pagrindų.

## **2. Sistemą realizuojančios technologijos**

Norint įgyvendinti sistemą MVC modeliu, reikia pasirinkti kokios technologijos bus naudojamos kiekvienam sluoksniui. Šiuo atveju sluoksnių elementai būtų duomenų bazė ir jos valdymas (Model), puslapiai (View) ir komunikuojantis kodas (Controller).

### **2.1. Technologijos duomenų bazės valdymui**

MongoDB

### **2.2. Technologijos puslapių kūrimui**

Angular.js, React, Ruby on Rails

### **2.3. Technologijos komunikuojančio kodo vykdymui**

.NET, Java, Ruby

## **Rezultatai ir išvados**

## Šaltiniai

- [Per18] Per Christensson. Mvc. 2018. URL: <https://techterms.com/definition/mvc>.
- [Sta09] Stack Overflow. When not to use mvc in a web application? 2009. URL: <https://stackoverflow.com/questions/1531907/when-not-to-use-mvc-in-a-web-application>.
- [Sta13] Stack Overflow. Using a database to store and get html pages for website. 2013. URL: <https://stackoverflow.com/questions/19767510/using-a-database-to-store-and-get-html-pages-for-website>.
- [Sta15] Stack Exchange. Where we should not use mvc? 2015. URL: <https://softwareengineering.stackexchange.com/questions/280383/where-we-should-not-use-mvc>.