

# ДОДАТОК А

## Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Дата звіту 6/5/2025  
Дата редагування ---



Звіт не був оцінений

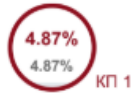
### Звіт подібності

#### метадані

Назва організації  
**Kharkiv National University of Radio Electronics**  
Заголовок  
**2025\_Б\_ПІ\_ПЗПІ-21-1\_Бойченко\_М\_Ю\_скорочений**  
Автор  
Науковий керівник / Експерт  
**Бойченко Матвій ЮрійовичСяген Кардаш**  
підрозділ  
**каф. ПІ**

#### Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

6489

Кількість слів

52876

Кількість символів

#### Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		24





#### Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

#### 10 найдовших фраз

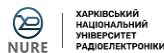
Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	100 1.54 %
2	Ігнатова В.А._ТУП-21а_робота 11/14/2024 National University "Lviv Politechnika" (NULP2)	60 0.92 %
3	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	47 0.72 %
4	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	38 0.59 %

5	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	19 0.29 %
6	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	13 0.20 %
7	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	13 0.20 %
8	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	13 0.20 %
9	<a href="https://er.nau.edu.ua/bitstreams/e6237ba0-8f9c-47e5-ad1e-4d30d84e4aaf/download">https://er.nau.edu.ua/bitstreams/e6237ba0-8f9c-47e5-ad1e-4d30d84e4aaf/download</a>	13 0.20 %
<b>з бази даних RefBooks (0.00 %)</b> 		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
<b>з домашньої бази даних (0.00 %)</b> 		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
<b>з програми обміну базами даних (0.92 %)</b> 		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Ігнатова В.А._ІТУП-21а_робота 11/14/2024 National University "Lviv Politechnika" (NULP2)	<b>60 (1) 0.92 %</b>
<b>з Інтернету (3.95 %)</b> 		
ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download">https://openarchive.nure.ua/bitstreams/5dce5cb3-8dce-4c7b-b35c-4522edfc8618/download</a>	<b>243 (7) 3.74 %</b>
2	<a href="https://er.nau.edu.ua/bitstreams/e6237ba0-8f9c-47e5-ad1e-4d30d84e4aaf/download">https://er.nau.edu.ua/bitstreams/e6237ba0-8f9c-47e5-ad1e-4d30d84e4aaf/download</a>	13 (1) 0.20 %

# ДОДАТОК В

## Слайди презентації



### Програмна система для організації зборів коштів

Бойченко М. Ю., ПЗП-21-1

Науковий керівник: ст. викл. Ляпога В. М.

19 червня 2025



## Мета роботи

- Об'єкт розробки – програмна система із серверною та мобільною частинами для організації збору коштів на соціальні, екологічні й медичні ініціативи.
- Мета розробки – створити масштабну та безпечну платформу, що забезпечує реєстрацію користувачів, ведення донорських профілів, автоматизовану розсилку сповіщень і адміністративний моніторинг активності.
- Метод реалізації – стек технологій .NET 8 (C#), Entity Framework Core, Identity, PostgreSQL, ASP.NET Core Web API, OAuth 2.0.



## Аналіз існуючих рішень та проблем

### Donate24

#### Переваги:

- Велика кількість функціоналу
- Розвинена система

#### Недоліки:

- Підозри шахрайства, недовіра користувачів
- Переускладнений інтерфейс
- Недостача прозорості



### Monobank Jar

#### Переваги:

- Зручний інтерфейс
- Можливість задонатити без авторизації

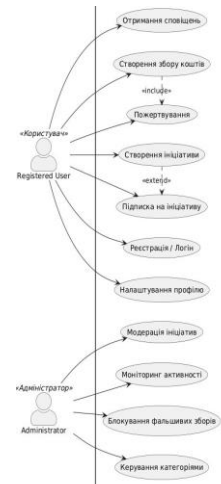
#### Недоліки:

- Обмежений перегляд деталей
- Відсутність належного контролю зборів



## Постановка задачі та опис системи

- Система проведення зборів, створення ініціатив як загальних маркерів специфіки зборів, проведення пожертв та проглядання результатів.
- Вимоги по функціоналу:
  - o Модуль користувачів (авторизація, редагування профілю, верифікація).
  - o Сповідання (тригер на створення зборів по підписці на збори, налаштування сповіщень, механізм підписки)
  - o Адмін-панель (перевірка зборів, редагування категорій та блокування ініціатив та зборів + моніторинг користувачів в системі).

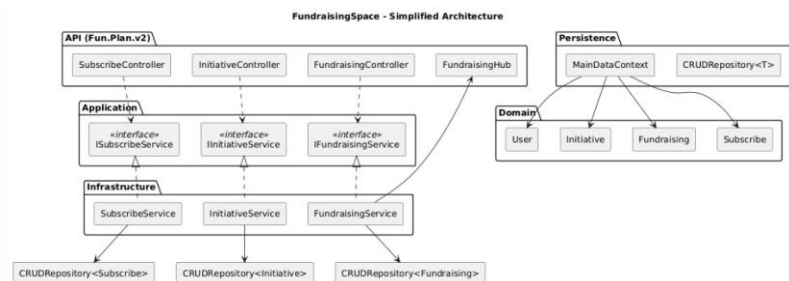
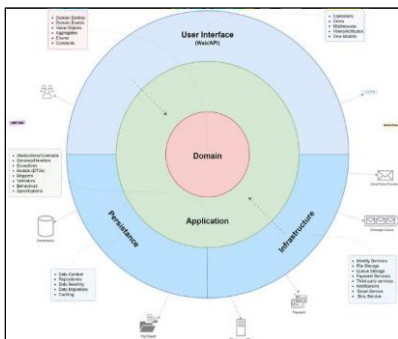


## Вибір технологій розробки

- Net 6 lts
- MSSQL, POSTGRES
- EFCore
- STRIPE
- Gauth 2.0



## Архітектура створеного програмного забезпечення



Clean-architecture – в нашій системі забезпечується Domain який відповідає за основні моделі, Infrastructure який є основним механізмом забезпечення логіки, Persistence це робота з даними, API ендпоінт додатку та Application компонентні моделі DTO та абстракції



## Файлова структура

У нашій системі реалізовано чітку Clean Architecture з поділом на проекти за ролями: Fun.Application містить бізнес-логіку у вигляді сервісів (DonateService, InitiativeService, TokenService, UserService), інтерфейсів для залежностей (IFun.Services, IRepository, IUnitOfWork) і DTO (ComponentModels), які служать для передачі даних між шарами; Fun.Domain описує доменні сутності у Fun.Models, що є чистими моделями без залежності від інфраструктури — це основа бізнес-логіки

```

> a C# TokenService.cs
> a C# TwoFactorService.cs
> a C# UserService.cs

```

```

> a C# SubscribeComponentModel.cs
> a C# UserActiveFilterModel.cs
> a C# UserAuthenticateModel.cs
> a C# UserProfileComponentModel.cs
> a C# UserRegisterModel.cs

```

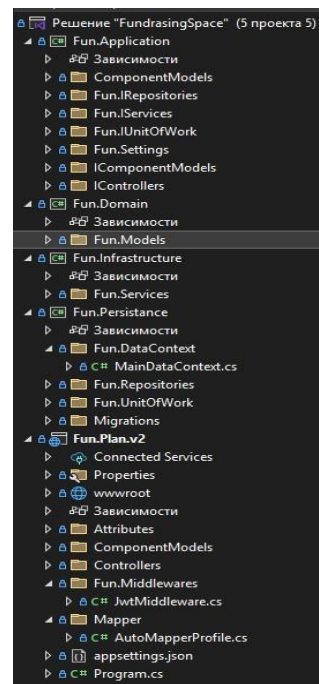
```

> a C# SubscribeController.cs
> a C# TwoFactorController.cs
> a C# UserController.cs

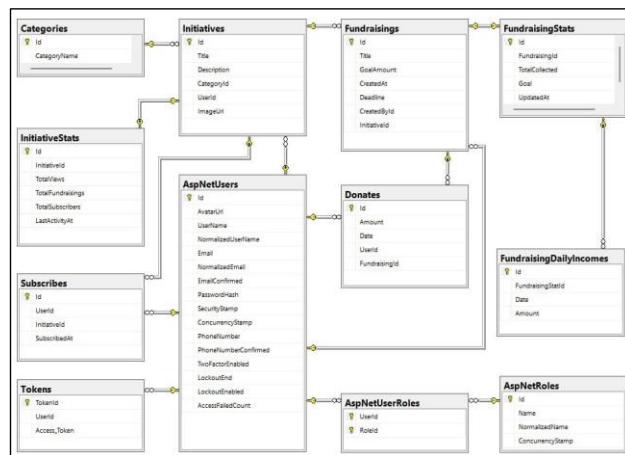
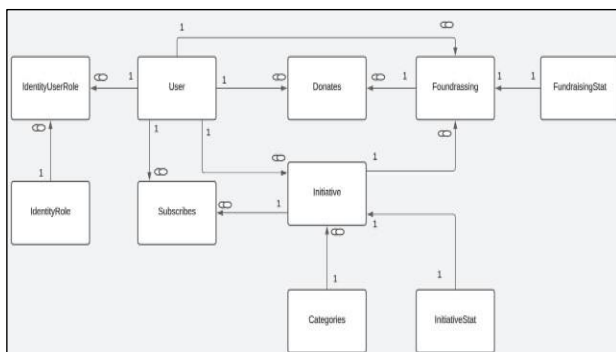
```



Певні частини структури системи для специфіки роботи з юзером



## Структура бази даних

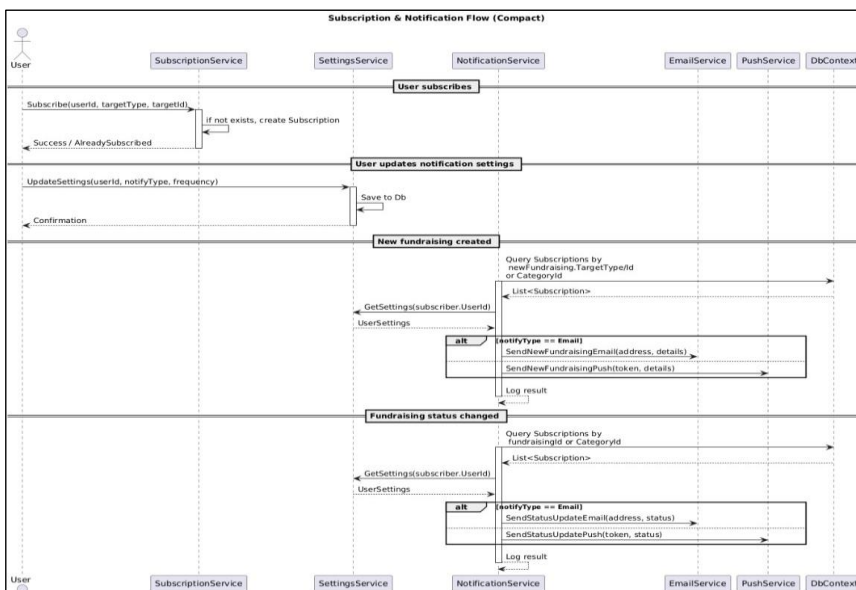


Загальні відносини між сутностями

Детальні сутності із полями та приховані таблиці з Identity

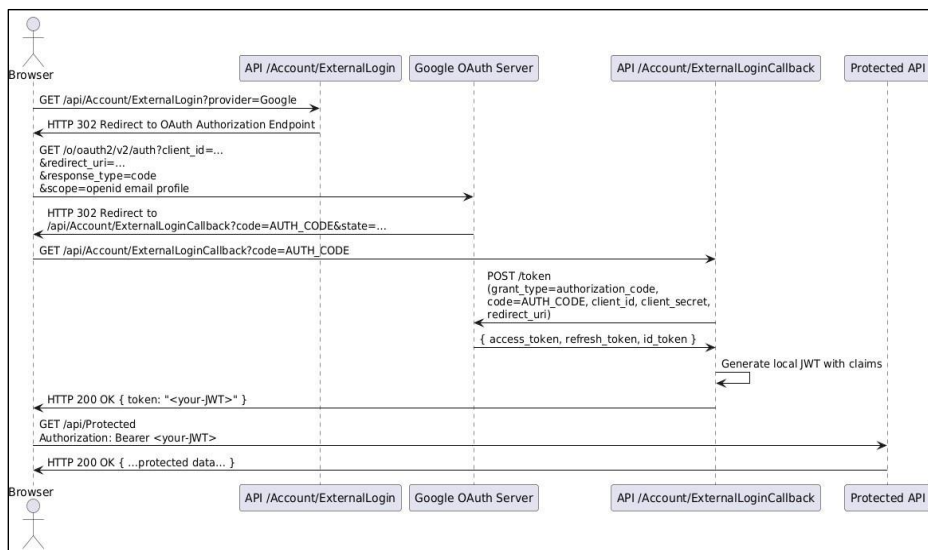
# Алгоритми та методи 1

Логіка підписок та сповіщень



# Алгоритми та методи 2

OAuth + JWT pipeline



# Алгоритми та методи 3 (приклад коду)

```

await _hubContext
    .Clients
    .Group($"initiative-{created.InitiativeId}")
    .SendAsync("NewFundraisingCreated", new
    {
        fundraisingId = created.Id,
        title = created.Title,
        goal = created.GoalAmount,
        deadline = created.Deadline
    });
var subject = $"Нова кампанія: {created.Title}";
var body = $"У ініціативі «{created.Title}» створено новий збір «{created.Title}» з метою {created.GoalAmount:C}.";
foreach (var email in toNotifyEmails.Distinct())
{
    await _emailSender.SendAsync(email, subject, body);
}
return created;

```

```

var donor = await _db.Users.FindAsync(userId);
if (donor != null)
{
    var fundraisingEntity = await _db.Fundraisings.FindAsync(fundraisingId);
    var subject = $"Дякуємо за донат на «{fundraisingEntity?.Title}»";
    var body = $"Привіт, {donor.UserName}!\n\n" +
        $"Ви знову підтримали збір «{fundraisingEntity?.Title}» сумою {amount:C}.\n" +
        $"Дякуємо за вашу допомогу!";
    await _emailSender.SendAsync(donor.Email, subject, body);
}
var fundraising = await _db.Fundraisings
    .Include(f => f.Initiative)
    .FirstOrDefault(f => f.Id == fundraisingId);
if (previousTotal < fundraising.GoalAmount
    && stat.TotalCollected >= fundraising.GoalAmount)
{
    var subs = await _db.Subscribes
        .Where(s => s.InitiativeId == fundraising.InitiativeId)
        .Select(s => s.User)
        .ToListAsync();
    foreach (var user in subs)
    {
        await _db.SaveChangesAsync();
    }
}

```

Надсилання сповіщень  
про створення збору  
через SignalR Hub

Сповіщення на пошту  
після створення  
донату на збір



## Тестування

Регістрація, email-верифікація, Google OAuth і профіль			
№	Опис випадку	Отримуваний результат	Висновок
1	POST /auth/register {email, pwd}	201 Created, статус Pending Verify, у Redis з'явився лист-шаблон	Пройдено
2	Перейти за /auth/verify?token=...	200, статус Verified, refresh+access JWT	Пройдено
3	POST /auth/login з валідними cred	200, повернено пари tokenів, role=User	Пройдено
4	GET /profile із JWT	200, donations=[], subscriptions=[]	Пройдено
5	PATCH /profile змінити avatar URL & bio	200, updatedAt оновився, avatar URL прийшов валідацію MIME	Пройдено
6	GET /auth/google/url → redirect, OAuth consent, callback /auth/google/callback	302 на returnUrl, видано JWT з claim LoginProvider=Google	Пройдено
7	POST /auth/login з неправильним pwd (5 разів)	Після 5-ї спроби – 423 Locked 10 хв, audit-entry LoginLock	Пройдено
8	Спроба POST /donations без email-верифікації (новий користувач)	403 EmailNotVerified	Пройдено
9	DELETE /auth/logout → GET /profile	401 Unauthorized	Пройдено
10	DELETE /auth/user із JWT Google-логіна	204, акаунт помічено IsSoftDeleted=true	Пройдено

Підписки, сповіщення та адмін-дешборд донорів			
№	Опис випадку	Отримуваний результат	Висновок
1	POST /subscriptions {type=Category, id=medical}	204, запис у Subscriptions	Пройдено
2	PATCH /settings/notifications {email=daily, push=instant}	200, налаштування збережено	Пройдено
3	Створюється новий збір категорії medical	У череді PushTasks — завдання → FCM надсилає push; SMTP — лист у digest-чергу	Пройдено
4	Відкрити push, перейти за deep-link	Мобільний клієнт відкриває деталі ініціативи ID	Пройдено
5	POST /donations {amount=75} → Stripe webhook succeeded	У Donations новий запис, totalSum(user)+75	Пройдено
6	GET /admin/donors?sort=totalSum&dir=desc&take=5 (JWT Admin)	JSON TOP-5, перший totalSum ≥ 75	Пройдено
7	GET /admin/donors?minCount=0&maxCount=0	Порожній масив (фільтр «без донатів»)	Пройдено
8	PATCH /admin/donors/{id}/tag {tag=VIP}	200, tag додано; повторний PATCH → 409 TagExists	Пройдено
9	POST /admin/broadcast/email {target=VIP, template=thanks}	Створено batch-job в Redis; у SMTP-інбоксі > 0 листів	Пройдено
10	Зашифрувати /admin/donors з токеном звичайного User	403 Forbidden	Пройдено

- тестування редагування користувача
- та авторизація
- перевірка валідації донату
- перевірка редіректу

- тестування сповіщень
- перевірка атрибутів та роботи ролей
- створення категорій
- тестування дашборду адміністратора по юзерам
- пуш нотифікації



## Майбутнє вдосконалення системи

- Розширення системи авторизації за допомогою Meta Login API, Apple Id та інших
- Впровадити чат-бот підтримки на базі AI для миттєвих відповідей на донорські запитання.
- Розгорнути багатомовний інтерфейс із підтримкою української, англійської та інших мов.
- Додати версію двофакторної аутентифікацію через SMS і TOTP.
- Інтеграція з CRM-системами та маркетинговими платформами через REST API вебхуки.



---

## Підсумки

У результаті розробки системи, а саме модулю донорів та адміністраторів – було створено програмну систему для організації збору коштів, яка передбачає повноцінний набір функцій для взаємодії з користувачами, донорами, адміністраторами та зовнішніми сервісами. Основну увагу під час роботи було приділено проєктуванню модулів реєстрації та авторизації користувачів (включаючи можливість інтеграції з Google OAuth 2.0), структури ведення історії донатів, формування профілів, логіки системи сповіщень — як email і push-повідомлень, а також модулю адміністрування для валідації ініціатив, аналізу зборів та активності юзерів.

