**АНОТАЦІЯ**

Метою бакалаврської кваліфікаційної роботи була розроблення програмної системи (веб-платформи) для фінансової підтримки контентмейкерів.

Програмна система розроблена за принципами клієнт-серверної архітектури. Серверна частина була розроблена на платформі ASP.NET Core з використанням таких бібліотек та фреймворків як Entity Framework, ASP.NET Identity та ASP.NET SignalR. Клієнтська частина була розроблена за допомогою наступних бібліотек: Razor Pages, Blazor, Bootstrap, JQuery.

Розроблена програмна система є кросплатформеною веб-платформою. Вона володіє зручним та мінімалістичним інтерфейсом, містить систему ролей, в якій неавторизовані користувачі мають доступ тільки до функціоналу для здійснення пожертв контентмейкеру, та перегляду його (контентмейкера) менійної інформації та посилань на соціяльні мережі. Користувачі які авторизовані як контентмейкер мають змогу створювати спеціальні віджети для подальшого використовування на прямих трансляціях, створювати сторінку-візитівку та отримати кошти з пожертв.

До програмної системи створено специфікацію вимог. Проектування та розробка здійснювалась згідно вимог описаних у специфікації та спираючись аналізу предметної області.

Робота містить реалізацію програмної системи та пояснювальну записку. Записка складається з \_\_\_ розділів та \_\_\_ додатків. Розмір записки без додатків містить \_\_ сторінку.

**ABSTRACT**

The purpose of the bachelor's thesis was to create a software system (web platform) for financial support of content makers.

The software system is developed on the principles of client-server architecture. The server part was developed on the ASP.NET Core platform using libraries and frameworks such as Entity Framework, ASP.NET Identity and ASP.NET SignalR. The client part was developed using the following libraries: Razor Pages, Blazor, Bootstrap, JQuery.

The developed software system is a cross-platform web platform. It has a user-friendly and minimalist interface, contains a system of roles in which unauthorized users have access only to the functionality for making donations to the content maker, and view his (content maker) menu information and links to social networks. Users who are authorized as content makers have the opportunity to create special widgets for further use on live broadcasts, create a business card page and receive funds from donations.

A specification of requirements has been created for the software system. Design and development was carried out according to the requirements described in the specification and based on the analysis of the subject area.

The work includes the implementation of the software system and an explanatory note. The note consists of \_\_\_ sections and \_\_\_ appendices. The size of the note without attachments contains \_\_ page.

**ЗМІСТ**

[**РОЗДІЛ 1 7**](#_Toc103499357)

[**СТВОРЕННЯ НА МОНЕТИЗАЦІЯ ВЕБ КОНТЕНТУ 7**](#_Toc103499358)

[**1.1. Соціальні мережі та платформи для контентмейкерів 7**](#_Toc103499359)

[**1.2. Прямі трансляції 10**](#_Toc103499360)

[**1.3. Монетизація контенту 12**](#_Toc103499361)

[**1.4. Висновки до розділу 1 15**](#_Toc103499362)

[**РОЗДІЛ 2 17**](#_Toc103499363)

[**ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ 17**](#_Toc103499364)

[**2.1. Загальна постановка завдання 17**](#_Toc103499365)

[**2.2. Специфікація вимог до програмного продукту 17**](#_Toc103499366)

[**2.2.1.1. Призначення та мета 17**](#_Toc103499367)

[**2.2.1.2. Продукти-аналоги 17**](#_Toc103499368)

[**2.2.2. Загальний опис 18**](#_Toc103499369)

[**2.2.3. Характеристики системи 19**](#_Toc103499370)

[**2.2.3. Вимоги зовнішніх інтерфейсів 26**](#_Toc103499371)

[**2.2.5. Інші нефункціональні вимоги 27**](#_Toc103499372)

[**2.2.5.3. Атрибути якості програмного продукту 27**](#_Toc103499373)

[**2.3. Вибір засобів для виконання завдання 28**](#_Toc103499374)

[**2.4. Висновок до розділу 2 29**](#_Toc103499375)

[**РОЗДІЛ 3 30**](#_Toc103499376)

[**АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ СИСТЕМИ ПОШУКУ ГЕОЛОКАЦІЇ ТА ІНФОРМАЦІЇ ПРО АВТОМАТИЧНІ ДЕФІБРИЛЯТОРИ 30**](#_Toc103499377)

[**3.1 . Проектування поведінки розроблюваної системи 30**](#_Toc103499378)

[**3.2. Загальна архітектура системи 31**](#_Toc103499379)

[**3.3. Проектування бази даних 32**](#_Toc103499380)

[**3.4. Проектування серверної частини 34**](#_Toc103499381)

[**3.5. Проектування клієнтської частини 34**](#_Toc103499382)

[**3.6. Висновок до розділу 3 34**](#_Toc103499383)

[**РОЗДІЛ 4 35**](#_Toc103499384)

[**РОЗДІЛ 5 36**](#_Toc103499385)

[**ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОБОТИ 36**](#_Toc103499386)

[**5.1. Економічна характеристика проектного рішення 36**](#_Toc103499387)

[**ВИСНОВКИ 37**](#_Toc103499388)

[**СПИСОК ЛІТЕРАТУРИ 38**](#_Toc103499389)

[**ДОДАТОК А. Діаграма прецедентів 40**](#_Toc103499390)

[**ДОДАТОК Б. Діаграма бази даних 41**](#_Toc103499391)

**ВСТУП**

Сучасна людина переживає фантастичні часи. Людство зараз живе під час Інформаційної ери. Інформаційна ера (англійською Digital Age, Computer Age, Information Age) або ще відома як Ера комп'ютерів, інформаційна епоха або електронна епоха – це концепція про сучасний світ та сучасні віки, для яких характерні широкі можливості вільно передавати чи приймати інформацію і мати миттєвий доступ до всесвітніх знань, що в попередні епохи розвитку людства зробити було б важко або неможливо. [1]

Людство розвивається дуже стрімко і технології розвиваються та еволюціонують швидше ніж самі люди. З кожним днем нової інформації стає ще більше і більше, а отримати до неї доступ стає все легше. Щодня через простори інтернету проходить мільйони терабайт інформації і людина потрапляє у інформаційний безлад. Людина має нестримну жагу до нової інформації та новин, вона стає емоційно залежною від неї.

З розвитком інформаційних технологій дуже сильно змінився стиль людського життя. Так само сильно і змінився спосіб проведення дозвілля. Людина дедалі більше надає перевагу перегляду соціальних мереж, телебачення, фільмів, серіалів, інтернет спілкуванню чи просто віртуальному світові над світом реальним та живим спілкуванням. Тому способи споживання інформації дуже стрімко розвиваються та мають шалену конкуренцію. Щороку створюються нові соціальні мережі, стрімко розвиваються старі. За користувача ведеться пекельна війна.

Одним із домінуючих способів отримування інформації є візуальний. Для людини він є найпростішим для сприйняття. Тому зараз шаленої популярності набирають платформи для перегляду фото-відео контенту, накшталт Instagram, YouTube, Tik-Tok, Twitch, Netflix і ще дуже і дуже багато інших. Усі ці платформи мають шалені заробітки від продажі реклами, підписок чи цифрових товарів.

Нова епоха породила і нові професії. Професія «блогер» чи «контентмейкер» стає надзвичайно популярною та навіть престижною. Для цієї професії існує безліч різних інструментів для створення та поширення контенту.

Одним із інструментів сучасних контентмейкерів та лідерів думок є прямі трансляції, або як їх прийнято називати «стріми». Зараз вийти у прямий ефір може кожен бажаючий. Найвідомішими платформами для проведення прямих трансляціє є Facebook, Twitch та YouTube. Вони показують дуже високі статистичні показники. Старший віце-президент Twitch Даміан Бернс, на своїй сторінці у Twitter, поділився статистикою платформи минулого року. За його даними, у 2020-му сервісом щодня користувалися понад 26,5 млн користувачів. Як зазначено у статистиці, продовж 2020 року користувачі спостерігали за прямими трансляціями на Twitch більше трильйона хвилин. Окрім цього, Бернс підмітив, що упродовж цього періоду на сервіс щомісяця приходило понад 6 млн нових контентмйкерів, а середня глядацька аудиторія Twitch в будь-який момент часу склала понад 2 млн людей.

Жодна праця не буває безоплатною, тому віртуальний контент по різному монетизується у мережі. Контентмейкери отримають гроші від платформ, рекламодавців, партнерських програм, спонсорів та напряму від фанатів. Як і фанати та прихильники певних релігій дають пожертвування на свою церкву, так і фанати певного блогера можуть дати пожертвування йому. Цей вид пожертвування має назву – донат.

**РОЗДІЛ 1**

**СТВОРЕННЯ НА МОНЕТИЗАЦІЯ ВЕБ КОНТЕНТУ**

* 1. **Соціальні мережі та платформи для контентмейкерів**

Сьогодні існують сотні соціальних мереж які щодня заохочують тисячі нових користувачів доєднатися до них. Соціальні мережі поступово доповнюють та частково заміняють реальне життя. Уже існують сервіси спілкування, знайомств та побачень, пошуку роботи, отримання новин, обговорення всіляких питань та тем, перегляду фільмів та серіалів, пошуку усього та усе. Соціальні мережі витісняють та замінюють привичні засоби отримання інформації та розваг, такі як телевізор, газети, книги, радіо, живі вебінари та форуми. Зараз усе переходить у цифровий світ.

Компанія Cloudflare, що створює інтернет-сервіси, спільно з сайтом Alexa.com, та збирає статистику про інтернет-сайтів, випустила рейтинг найпопулярніших сайтів за 2021 рік

Топ-10 — найпопулярніші домени соціальних мереж (кінець) 2021 року за даними Cloudflare:[2]

1. TikTok.com
2. Facebook.com
3. YouTube.com
4. Twitter.com
5. Instagram.com
6. Snapchat.com
7. Reddit.com
8. Pinterest.com
9. LinkedIn.com
10. Quora.com

Щомісяця ці соціальні мережі мають мільярди активних користувачів. За даними Statista.com [3], аудиторія Facebook налічує 2.910 млрд активних користувачів, YouTube 2.562 млрд, Instagram 1.263 млрд.

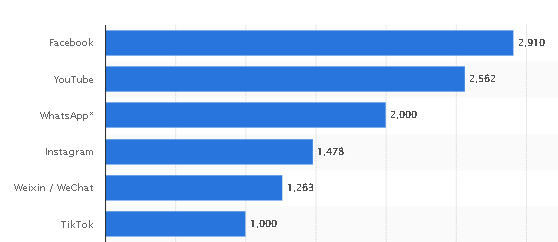


Рис. 1.1. Найпопулярніші соціальні мережі в усьому світі станом на січень 2022 року, ранжовані за кількістю активних користувачів щомісяця

Як можна побачити зі списку, найпопулярнішими є соціальні мережі, які пропонують у переважному відео-контент. TikTok взагалі став феноменом. Ця соціальна мережа дозволяє ділитися людям з усього світу короткими відео (зараз існує можливість створювати відео до 4хв). Також платформа дозволяє редагувати відео, накладати на нього ефекти та музику, спілкуватися з іншими користувачами. Окрім цих інструментів вона також дає можливостей вести прямі трансляції. По даних аналітиків, кількість активних користувачів у 2021р становила більше 1 мільярду.

Інтернет не є тихим місцем, як і люди на Землі (особливо під час пандемії) не тихі чи пасивні, а активні та реактивні. Хоча на вершині рейтингу доменів, здається, не спостерігається різких злетів і падінь протягом року (TikTok і YouTube були винятком), можна побачити, як така подія, як переворот у М’янмі та наступне вірусне відео, могли принести YouTube займає перше місце в нашому рейтингу. Ми також бачили, як вплинула електронна комерція протягом року, як веб-сайти, пов’язані з космосом, мали великий (онлайн) рік із важливими подіями, і як Netflix піднявся під час Різдва.[2]

Темою моєї бакалаврської кваліфікаційної роботи є платформа для підтримки творців які орієнтуються на відео-контент та проведення прямих трансляцій. Тому надалі буде розглядатися саме призначені для цього платформи.

Я виділив дві соціальні мережі, або як їх можна назвати «відео-хостинги», які на мій погляд є найзручнішими та найпопулярнішими для публікації відео матеріалів, це – YouTube та Twitch.

YouTube — веб-платформа для обміну відео в Інтернеті та соціальних мереж, яка функціонує з 2005 року та набула надзвичайної світової популярності. Її творцями є колишні працівники PayPal Стів Чен, Чад Херлі та Джавед Карім. Штаб-квартира знаходиться в Сан-Бруно, Каліфорнія. 9 жовтня 2006 року компанія Google придбала YouTube за 1,65 мільярда доларів.

YouTube існує уже більше 15 років, але з часом платформа ставала лише кращою, швидшою та міцнішою. Загалом сервіс отримує 14,3 мільярда відвідувань на місяць: це більше, ніж у Facebook, Вікіпедії, Amazon та Instagram. Якщо говорити про стать користувачів YouTube, то серед них віком від 18 років 46,1% ідентифікують себе як жінки, а 53,9% – як чоловіки. Як повідомляє Statista, деякі користувачі відвідують сервіс навіть кілька разів на день. Тим часом 92% американських користувачів стверджували, що користуються платформою щотижня, а 98% кажуть, що мають доступ до сайту щомісяця.

Маючи щомісяця понад 14 мільярдів відвідувань, YouTube є одним із найбільш відвідуваних сайтів відразу після своєї материнської компанії Google. Пошуковий термін «YouTube» є також третім за популярністю у Google. Щохвилини на сервісі «YouTube» транслюється понад 694 000 годин відео. Це більше, ніж у найвідомішої стрімінгової платформи Netflix, де користувачі транслюють 452 000 годин відео.

За статистичними даними, приблизно 81% користувачів Інтернету бодай раз користувалися YouTube. Зі стрімким розвитком смартфонів, значно збільшилась кількість користувачів, що заходять на сайт через мобільний телефон, їхня частка складає 22% від усіх відвідувачів сервісу. Користувачі операційної системи Android по всьому світу користуються додатком YouTube на своїх телефонах у середньому 23,7 години щомісяця.

Twitch — це популярний американський сервіс для потокового відео в прямому ефірі. Нею керує дочірня компанія Amazon. Сервіс зосереджується на прямій трансляції відеоігор («летсплеї»), включаючи трансляції кіберспортивних змагань, музичні трансляції, трансляції творчого контенту та потоків «в реальному житті». Twitch був представлений у червні 2011 року як додатковий продукт платформи Justin.tv. Вміст на сайті можна переглядати як у прямому ефірі, так і через відео за запитом. Amazon швидко помітив потенціал Twitch і у 2014 році він купив компанію майже за мільярд доларів. Це дозволило отримати їм близько 40 000 000 підписок.

Сервіс Twitch отримав великий наплив нових користувачів саме під час пандемії COVID-19, коли майже уся планета закрилася на карантин. Після активної фази карантину активність сервісу не переставала збільшуватися. Тільки за перші три місяці 2021 року користувачі Twitch переглянули 326 мільярдів хвилин контенту, що є майже половиною від 660 мільярдів за цілий 2019 рік. Тільки в першому кварталі 2021 року це понад 5 мільярдів годин перегляду на Twitch. У 2021 році щомісяця нараховується 9,7 мільйона стрімерів. У 2020 році в середньому за місяць користувачі переглядали Twitch 93 мільярди хвилин. Статистика Twitch за 2020 рік показує, що 140 мільйонів глядачів переглядали контент на платформі щомісяця.

Це звучить дивно для платформи про ігри, але статистика Twitch показує, що це найпопулярніша тема взагалі. «Just Chatting» — раніше тег відеоблогу категорії IRL — призначений для спілкування стрімерів зі своїми глядачами. Хто б міг подумати, що платформа в кінцевому підсумку охопить свої коріння як мережу «життя»? Цей жанр допоміг популяризувати «lifestyle streaming» Багато ігрових стрімерів також використовують цю тему.

Значну частку популярних стрімерів складають саме особи жіночої статі з особливо вираженою фізичною красою. На подібних прямих трансляціях відеогра зазвичай відкладається на задній план, а основний акцент переходить на глибокий виріз декольте, миле личко чи на місце для сидіння на кріслі. Проте досі існують вірні сини та доньки світу геймерів, яким нецікаві ці всі плотські угоди.

* 1. **Прямі трансляції**

Так як створення нових відеороликів може займати дуже багато часу та потребувати значних зусиль, серед блогерів стало популярним проводити прямі трансляції, або як їх називають у інтернеті стріми, а блогера, який їх проводить – стрімером.

Пряма трансляція — це потокове мультимедіа, яке одночасно записується та транслюється в режимі реального часу через Інтернет. Його часто називають просто стрімом. Пряма трансляція – це ефективний та ефективний спосіб досягнення будь-яких цілей і завдань, які ви хочете досягти. Пряму трансляцію можна використовувати для просування та оголошення ваших нових продуктів і послуг, оголошення новин великої компанії, навчання онлайн-класу тощо. Пряма трансляція дає брендам можливість охопити мільйони людей по всьому світу та взаємодіяти з ними.

Стрімер— це особа, яка транслює себе в режимі онлайн за допомогою прямої трансляції або попередньо записаного відео. Жанри включають гру у відеоігри, навчальні посібники або соло-чати.

Одним із типів стрімерів є летсплеєри. Вони набули популярності ще з самого започаткування прямих трансляцій, завдяки цьому вони і до цих пір збирають найбільшу аудиторію та фан-базу в Інтернеті. Летсплей (англ “Lets-play”) – один з жанрів розважального контенту, який включає проходження комп’ютерних ігор, ігрові марафони, гайди. Для багатьох стрімінг був хобі, але з ростом аудиторії та появою рекламодавців прибутки дозволили заробляти та займатися улюбленою справою у ролі роботи. Щодня з'являється все більше новачків, які пробують зацікавити аудиторію своєю особистістю, а не тільки вмінням грати ігри. Найвідоміші стримери відеоігор — PewDiePie та Ninja, яким вдається щороку заробляють мільйони доларів лише за допомогою стрімінгу не включаючи пожертви глядачів.

Доступність сервісу породило багато різного контенту у тому числі і негативного. Тому у лютому 2018 року Twitch оновив свою політику щодо прийнятного вмісту, щоб наказати, щоб будь-який вміст, який він вважає ненависним, призупинявся на своїй платформі. Користувачам Twitch тепер заборонено транслювати будь-які ігри, які мають рейтинг «Тільки для дорослих» (AO) у Сполучених Штатах Радою з оцінки розважального програмного забезпечення (ESRB), незалежно від її рейтингу в будь-якому іншому географічному регіоні, а також будь-яку гру, яка містить «вміст відверто сексуального характеру» або «безпідставне насильство», або вміст, який порушує умови використання сторонніх служб.

Для запису відео та проведення прямих трансляцій також існує багато додаткових інструментів. Проте найбільшої набрала OBS (Open Broadcaster Software) - безкоштовне програмне забезпечення з відкритим кодом для запису та створення трансляцій. [4] Програма написана на C/C++ і створена за допомогою Qt, OBS Studio забезпечує захоплення в реальному часі, композицію сцени, запис, кодування та трансляцію через протокол обміну повідомленнями в реальному часі (RTMP). Він може транслювати відео на будь-яке місце призначення з підтримкою RTMP, включаючи YouTube, Twitch, Instagram та Facebook. Хтось або платить вам за ваш контент безпосередньо, або вони платять за доступ до вашої аудиторії.

* 1. **Монетизація контенту**

Усі сучасні платформи та соціальні мережі дають можливість користувачам заробляти на власному контенті. Це зветься монетизацією. Монетизація контенту – це спосіб використання вмісту, щоб, коли користувачі споживали його, ви заробляли гроші. Вам можуть заплатити як самі користувачі, так і третя сторона, яка продає продукти через ваш вміст. Почати експериментувати з монетизацією вмісту легко. Наприклад, такі платформи, як YouTube і Facebook, уже мають вбудовані сервіси, які дають вам змогу користуватися рекламою, коли ваша зацікавленість досягає певного порогу.

Коли контентмейкер безпосередньо продаєте свій контент, він заробляєте гроші безпосередньо від користувачів. Наприклад, коли контент знаходиться за платним доступом або структурований в онлайн-курс, його можна вважати як власний продукт. Хоча насправді ваша сторінка у будь якій соціальній мережі не належить блогеру, а є власністю самої мережі.

Коли контентмейкер продає свою аудиторію, то він працює з третьою стороною, яка часто зветься спонсором, щоб задовольнити їхні маркетингові потреби. Блогери часто продають рекламу, як це відбувається у інших засобах масової інформації.

Платформи монетизації вмісту, такі як Facebook і Youtube, діють як третя сторона між вами та рекламодавцями. Блогер публікує свій вміст на платформі, а платформа платить вам за показ реклами. (Платформа, в свою чергу, оплачується рекламодавцями.) Якщо контентмейкер переважно публікуєте свій контент у соціальній мережі, це найпростіший і прямий спосіб почати його монетизувати.

Ще одним із способу заробітку є продакт-плейсмент. У певному сенсі він є старомодною формою маркетингу. Що стосується контенту, він працює так само, як і з фільмами: блогер показуєте продукт спонсора у своїй роботі, і вони платять йому фіксовану ставку, що є хорошим плюсом. Продакт-плейсмент – це досить простий спосіб почати отримувати прибуток за контент і заробляти гроші.

Існує також афілійований (партнерський) маркетинг. У партнерському маркетингу контентмейкер заохочує своїх користувачів купувати продукти чи послуги. Однак, на відміну від продакт-плейсменту, кожна покупка відстежується, і ви стягуєте комісію на основі того, яка частина йог аудиторії бере участь. Є кілька способів зробити це. Одним з них є використання реферального коду чи промокоду.

Для монетції бази шанувальників, блогери створють нові продукти, щоб продавати людям, які люблять їхню роботу. Одяг, кружки, декоративні подушки та інший мерчендайз — все це досить прості варіанти. Популярним способом заробітку від аудиторії є пожертвування або донати.

Простіше кажучи, донат – це різновид подарунка, який робить юридична чи фізична особа (особи), як показує практика, на благодійні цілі та потреби або на користь певної справи. Донат набуває різної форми, включаючи фінансову підтримку, послуги, певні товари (одяг, іграшки, техніка, електронні пристрої, продукти харчування, транспортні засоби тощо). Предмети гуманітарної допомоги, допомоги розвитку, екстреної допомоги, для медичних потреб (кров або органи для трансплантації) також можуть бути складовою пожертвування. Чудовим прикладом є пожертвування, які здійснює український народ та увесь світ для боротьби з гидким та бридким злом яке успішно дохне на кордонах з сусідньою державою болотом, за допомогою чудових подарунків наших західних партнерів. [5]

Величезні можливості відкриваються для некомерційних організацій, які можуть організувати збір коштів у соціальних мережах. З огляду на величезну кількість користувачів інтернету та соціальним мережами, їх варто використовувати, щоб збільшити кількість своєї аудиторії та у результаті і базу донатерів.

Користувачі, яким подобається контент їхнього улюбленого блогера підтримують його фінансово, жертвуючи йому кошти. Окрім прояву любові чи поваги до блогера, фанати привертають до себе увагу улюбленого творця.

Для заохочення глядачів прямих трансляцій жертвувати кошти стрімеру було придумано багато заохочувальних схем. Одна з ним – це можливість засвітити свій псевдонім серед великої аудиторії інтернет користувачів та надіслати якусь звістку улюбленому блогеру. Після того, як фанат надсилає певну грошову суму, у прямому ефірі з’являється повідомлення, яке можу супроводжуватися звуком та анімаційною картинкою, яке містить ім’я користувача та його повідомлення.

Донати стали невід’ємним джерелом заробітку для контентмейкерів. Надзвичайної популярності набирає система Patreon, яка за допомогою спеціальної підписки дозволяє підтримати улюбленого контентмейкера та отримати доступ до ексклюзивного контенту. Patreon— це американська платформа для членства, яка надає бізнес-інструменти для розробників контенту для запуску служби підписки. Він допомагає творцям і художникам отримувати щомісячний дохід, надаючи винагороди та пільги для своєї аудиторії. Patreon стягує комісію в розмірі від 9 до 12 відсотків місячного доходу творців на додаток до плати за обробку платежів.[6]

Окрім Patreon також існує ще одна схожа платформа – OnlyFans. Користувачі OnlyFans за спеціальною підпискою також отримають доступ до ексклюзивного контенту дещо іншого характеру. Здебільшого авторами цього контенту є особи жіночої статі, різного віку, які наділені красивими очима.

OnlyFans — це служба підписки на інтернет-контент, що базується в Лондоні, Великобританія. Творці контенту можуть заробляти гроші від користувачів, які підписалися на їхній контент — «шанувальників». Це дозволяє творцям контенту щомісячно отримувати фінансування безпосередньо від своїх шанувальників, а також одноразові поради та функцію оплати за перегляд (PPV). Веб-сайт має два мільйони творців контенту та 130 мільйонів користувачів.

Заради нових донатів стрімери часто готові піти навіть на нерозумні та небезпечні поступки. Великої популярності набирають прямі трансляції, на яких за пожертвування певної суми можна заставити блогера зробити якусь неприємну річ, як наприклад випити гидку речовину, велику кількість алкоголю, отримати стусанів від інших учасників стріму, отримати розряд током, показати якусь частину тіла, зробити якусь фізичну вправу, потрясти задницею чи грудьми. Коротко кажучи, контентмейкер перетворюється на іграшку чи клоуна для глядачів. На платформах з можливістю публікувати «дорослий» контент, за пожертвування блогер може здійснювати певні вульгарні дії, які у суспільстві можуть набувати аморального характеру. Цікавою можливістю є можливість донатом керувати спеціальним девайсом стрімера. Усі ці дії, це тільки вершина айсбергу фантазії блогера, часто вони роблять жахливіші речі задля грошей.

* 1. **Висновки до розділу 1**

В цьому розділі описано феномен соціальних мереж, їхню тепер уже невід’ємну роль у житті сучасної людини. Середньостатистичний житель планети Земля вже не може уявити своє життя без інтернету. Багато людей використовують соціальні мережі для розваг та проведення вільного часу. У мережі для цього з’явилося безліч сервісів та платформ. Це породило нову професію – блогер або контентмейер.

У багатьох блогерів їхнє хобі, створювати контент перетворилося у справжню та повноцінну роботу. Найпопулярнішим напрямком у блогерстві є створення фото чи відео-контенту. Найпопулярнішими платформами є TikTok, Facebook, YouTube, Twitter та Instagram. Існує багато способів заробітку грошей зі свого контенту, тобто монетизації. Найвідомішими є: продає реклами, спонсорство, пожертвування, продаж мерчендайзу.

**РОЗДІЛ 2**

**ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ**

* 1. **Загальна постановка завдання**

Завдання полягає в створенні системи, яка дозволить монетизувати прямий ефір шляхом приймання пожертв від аудиторії. Для звичайного користувача система дасть змогу взаємодіяти з контентмейкером та фінансово його підтримати. У свою чергу контентмейкерам система має дати змогу створювати інструменти для відображення пожертв, візуалізації статистики та певних показників (список з останніх донатерів, список з найкращих за певний час донатерів тощо) на прямій трансляції. Для цього було вирішено розробити веб-застосунок зі зручним та інтуїтивно зрозумілим інтерфейсом та можливістю створювати спеціальні віджети, які є веб-сторінкою, що відображають потрібну інформацію про пожертвування у режимі реального часу.

Вхідними даними для виконання даного завдання є специфікація вимог та список обраних інструментів та рішень.

Вихідними даними є розроблене програмне забезпечення яке реалізує поставлену мету.

* 1. **Специфікація вимог до програмного продукту**
     1. **Вступ**
        1. **Призначення та мета**

Основними призначеннями даної системи є монетизація прямих ефірів шляхом пожертвувань від аудиторії. За допомогою такої системи неавторизовані користувачі зможуть пожертвувати кошти автору прямої трансляції. В даній системі буде реалізована авторизація, яка дозволить створювати користувачів-творців (стрімерів) , які зможуть створювати візуальні елементи для прямої трансляції, які демонструють взаємодію з аудиторію та їхню, фінансову підтримку. Отримувачам пожертв система може дати можливість виводити сповіщення про надходження пожертв, статистики пожертв та прогресу збору коштів у прямому ефірі, даючи веб-посилання на спеціальні «віджети», які можна додати до програми для проведення прямих трансляцій.

* + - 1. **Продукти-аналоги**

Враховуючи те, що прямі трансляції стали дуже популярними та приносять колосальні прибутки, систем, які дозволяють монетизувати контент саме для українських творців, на мій погляд, мало і конкуренції між ними немає. Тому, багато українських блогерів користуються іноземними сервісами. Одним із колись найпопулярніших був рашистський сервіс Donation Alerts (<https://www.donationalerts.com/>), але зараз із зрозумілих причин цей сервіс заблокований на території України та його використання є недоцільним. Популярним сервісом є каліфорнійський streamlabs (https://streamlabs.com/) , що окрім можливості монетизації прямої трансляції, дає спеціальну модифікацію сервісу OBS для проведення прямих трансляцій. Проте сервіс має велику комісію та погано взаємодіє з українською валютою.

Для використання в Україні існують декілька вітчизняних сервісів. Найвідомішим є Дяка (Diaka) ( <https://diaka.ua/amortisation>) – сервіс є дуже простим та має стандартний набір можливостей подібних сервісів. Комісія зі зборів варіюється взалежності від суми (чим більша сума, тим менша комісія). Ще одним українським сервісом є Donatello (<https://donatello.to/>) – він є майже аналогічним, але дає дещо більше можливостей. За даними самого сайту на квітень 2022 року кількість користувачів становить 6176 зареєстрованих. Проте сервіс використовує підсистему Foundy, тому комісія є досить великою, яка включає комісію самого сервісу та Foundy (3,75% + 5%).

* + 1. **Загальний опис**

**2.2.2.1. Характеристики продукту**

Основними можливостями системи є:

* Реєстрація та авторизація
* Надсилання пожертвувань (донатів) творцю
* Створення віджетів для відображення повідомлень (донатів)
* Створення віджетів для відображення статистики
* Створення віджетів для відображення «збору коштів»
* Створення власних віджетів
* Створення сторінки-візитівки автора
* Вивід коштів з системи
* Відображення віджетів у прямому ефірі
* Відображення повідомлень (донатів) у прямому ефірі
* Відображення статистики у прямому ефірі
* Відображення «збору коштів» у прямому ефірі

**2.2.2.2. Класи користувачів**

Система матиме кілька два класс користувачів, що будуть володіти різними привілеями і для яких буде доступний різний функціонал.

* Неавторизований користувач – йому не потрібно проходити процедуру авторизації. З цими правами з системою будуть працювати представники цільового сегмента користувачів – люди, які хочуть фінансово підтримати контентмейкера або переглянути його профіль щоб отримати покликання на соціальні мережі блогера.
* Контентмейкер (блогер, автор) – для входу в систему повинен пройти авторизацію. Має можливість заповнити інформацію про себе, додати опис, додати покликання на соціальні мережі. Має можливість створювати спеціальні віджети, та використовувати їх для прямих трансляцій чи інших власних потреб. Також він має можливість вивести накопичені у системі кошти.

**2.2.2.3. Середовище функціонування**

Для функціонування серверної частини системи потрібні:

● Платформа з встановленим ASP.NET Core та всіма необхідними пакетами, які описані в файлі project.assets.json.

● Сервер бази даних – SQL Server, на якому створені та заповнені необхідні колекції даних.

Для коректної роботи клієнтської частини, потрібен будь-який із сучасних веб-браузерів.

* + 1. **Характеристики системи**
       1. **Створення віджета «Сповіщення»**

2.2.3.1.1. Опис і пріоритет

Користувач з правами контентмейкера має можливість створити та налаштувати віджет «Сповіщення» («Notification»), який у режимі реального часу відображає повідомлення-пожертвування від інших користувачів (фанатів).

Пріоритет: Високий.

2.2.3.1.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Сповіщення» («Notification»), та натиснути «Створити новий». Після цього відкриється форма для створення віджета, яка містить конфігураційні поля та має можливість обрати файли (анімацію та звук), які супроводитимуть сповіщення про донат.

2.2.3.1.3. Функціональні вимоги

1. Відображення доступних змінних у формуванні шаблону тексту-заголовку.
2. Можливість попереднього перегляду та прослуховування доданих файлів.
3. Валідація усіх полів .
4. Можливість додати декілька файлів.
5. Можливість зміни та видалення доданих файлів.
6. Генерація посилання на веб-сторінку відображення віджету.

**2.2.3.2. Створення віджета «Статистика»**

2.2.3.2.1. Опис і пріоритет

Користувач з правами контентмейкера має можливість створити та налаштувати віджет «Статистика» («Statistic»), який у режимі реального часу відображає обрані користувачем статистичні дані у обраному ним вигляді та форматі.

2.2.3.2.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Статистика» («Statistic»), та натиснути «Створити новий». Після цього відкриється форма для створення віджета, яка містить конфігураційні поля та має можливість налаштувати стиль відображення віджету.

2.2.3.2.3. Функціональні вимоги

1. Відображення доступних змінних у формуванні шаблону тексту-заголовку.
2. Можливість вибору типу віджету («Топ» (рейтинг жертодавців), «Останній донатер», «Зібрана сума»).
3. Валідація усіх полів .
4. Можливість вибору способу відображення віджету («Список», «Бігучий рядок», «Слайдер»)..
5. Можливість вибору напрямку руху відображення віджету.
6. Можливість вибору часових рамок (часового періоду). за які будуть обиратися статистичні дані для віджету.
7. Можливість установити швидкість анімації (прокручування).
8. Генерація посилання на веб-сторінку відображення віджету.

**2.2.3.3. Створення віджета «Збір коштів»**

2.2.3.3.1. Опис і пріоритет

Користувач з правами контентмейкера має можливість створити та налаштувати віджет «Збір коштів» («Foundraising»), який у режимі реального часу відображає обрані користувачем статистичні дані у обраному ним вигляді та форматі.

Пріоритет: Високий.

2.2.3.3.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Збір коштів» («Foundraising»), та натиснути «Створити новий». Після цього відкриється форма для створення віджета, яка містить конфігураційні поля та має можливість налаштувати стиль відображення віджету.

2.2.3.3.3. Функціональні вимоги

1. Відображення доступних змінних у формуванні шаблону тексту-заголовку.
2. Можливість увімкнути та вимкнути віджет.
3. Валідація усіх полів .
4. Можливість змінити початкову, кінцеву та зібрану суми.
5. Можливість задати розмір віджету, ширину рамок та заокруглення.
6. Можливість зміни кольори відображення елементів віджету.
7. Генерація посилання на веб-сторінку відображення віджету.

**2.2.3.4. Відображення та робота віджета «Повідомлення»**

2.2.3.4.1. Опис і пріоритет

Користувач який має доступ до посилання на віджет, має можливість відвідати та використовувати веб-сторінку яка відображає роботу віджета у власних потребах. Веб сторінка віджета у режимі реального часу реагує на надходження пожертв контентмейкеру та відображає їх. Веб-сторінка показує заголовок та текст повідомлення, супроводжуючи його появу налаштованим звуком та картинкою. Взалежності від налаштувань віджету, повідомлення може бути зачитане синтезом мови.

Пріоритет: Високий.

2.2.3.4.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Повідомлення» («Notification»), обрати у виведеному списку усіх створених віджетів бажаний та натистути на кнопку «Відобразити». Після цього відкриється веб сторінка з відображенням роботи віджету .

2.2.3.4.3. Функціональні вимоги

1. Відображення заголовку повідомлення відповідно до заданого шаблону.
2. Відображення картинки.
3. Відтворення звуку повідомлення.
4. Відтворення звуків повідомлення та відображення картинок з заданого набору у вибраній послідовності (по черзі чи випадково).
5. Синтез мовлення тексту повідомлення.
6. Відображення повідомлень у режимі реального часу.
7. Можливість установити швидкість анімації (прокручування).

**2.2.3.5. Відображення та робота віджета «Статистика»**

2.2.3.5.1. Опис і пріоритет

Користувач який має доступ до посилання на віджет, має можливість відвідати та використовувати веб-сторінку яка відображає роботу віджета у власних потребах. Веб сторінка віджета оновлюєтья декільки раз за хвилину, щоб показувати актуальні дані. Веб-сторінка показує статистичні дані у заданому виді та форматі.

Пріоритет: Високий.

2.2.3.5.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Статистика» («Statistic»), обрати у виведеному списку усіх створених віджетів бажаний та натистути на кнопку «Відобразити». Після цього відкриється веб сторінка з відображенням роботи віджету .

2.2.3.5.3. Функціональні вимоги

1. Відображення заголовку рядка-елемента відповідно до заданого шаблону.
2. Оновлення (актуалізація) даних.
3. Відображення статистичних даних у вигляді списку.
4. Відображення статистичних даних у вигляді бігучого рядку.
5. Відображення статистичних даних у вигляді слайдеру.
6. Відтворення анімації (прокручування) за заданою швидкістю.
7. Відображення вибраних статистичних даних («Топ» (рейтинг жертодавців), «Останній донатер», «Зібрана сума»).
8. Відображення руху віджету у вибраному напрямку.
9. Відображення даних віджету відповідно до обраних часових рамок (часового періоду).

**2.2.3.6. Відображення та робота віджета «Збір коштів»**

2.2.3.6.1. Опис і пріоритет

Користувач який має доступ до посилання на віджет, має можливість відвідати та використовувати веб-сторінку яка відображає роботу віджета у власних потребах. Веб сторінка віджета оновлюєтья декільки раз за хвилину, щоб показувати актуальні дані. Веб-сторінка показує статистичні дані у заданому виді та форматі.

Пріоритет: Високий.

2.2.3.6.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Збір коштів» («Fundraising»), обрати у виведеному списку усіх створених віджетів бажаний та натистути на кнопку «Відобразити». Після цього відкриється веб сторінка з відображенням роботи віджету .

2.2.3.6.3. Функціональні вимоги

1. Відображення заголовку відповідно до заданого шаблону.
2. Оновлення (актуалізація) даних.
3. Відображення віджету відповідно до заданих параметрів стилю (розміри та кольори елементів).
4. Відображення зібраної суми у відсотках.
5. Можливість вимкнення роботи віджету (нові пожертвування перестають враховуватися).

**2.2.3.7. Керування профілем**

2.2.3.7.1. Опис і пріоритет

Авторизований користувач (контентмейкер) має можливість керувати власним профілем. Він має можливість змінити пароль, змінити адрес електронної пошти, завантажити власні персональні дані та видалити їх з системи.

Пріоритет: Високий.

2.2.3.7.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Керування профілем» («Manage Accaunt»). Далі на лівій боковій панелі з’явиться список чотирьох доступних розділів: «Профіль», «Електронна пошта», «Пароль», «Персональні дані».

2.2.3.7.3. Функціональні вимоги

1. Можливість змінити адресу електронної пошти.
2. Можливість змінити пароль.
3. Можливість змінити персональні дані.
4. Можливість завантажити усі персональні дані.
5. Можливість видалити усі персональні дані

**2.2.3.8. Надсилання пожертв**

2.2.3.8.1. Опис і пріоритет

Неавторизований користувач має можливість надіслати контентмейкеру пожертву (донат).

Пріоритет: Високий.

2.2.3.8.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією користувач повиненмати посилання на профіль блогера, або знати його нікнейм. Далі він переходить за цим посиланням, заповнює потрібні поля (ім’я донатера, сума, повідомлення). Далі він тисне кнопку «Надіслати» і створює платіж.

2.2.3.8.3. Функціональні вимоги

1. Можливість обрати запропоновану суму.
2. Можливість вводу власної суми.
3. Можливість вводу повідомлення.
4. Можливість перейти на профіль (сторінку-візитівку) контентмейкера.

**2.2.3.9. Сторінка-візитівка**

2.2.3.9.1. Опис і пріоритет

Авторизований користувач (контентмейкер) має можливість налатувати свою власну сторінку-візитівку, яка міститиме коротку інформацію про нього та посилання на соціальні мережі. Будь який користувач може переглянути сторінку-візитівку контентмейкера та має змогу перейти на сторінку для пожертв.

Пріоритет: Середній.

2.2.3.9.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Візитівка». Після цього відкриється форма для заповнення потрібних даних та посилань, вибору картинок для профілю.

2.2.3.9.3. Функціональні вимоги

1. Можливість змінити тло профілю.
2. Можливість додати посилання на соціальні мережі.
3. Можливість додати коротку автобіографію.
4. Можливість змінити фото профілю
   * 1. **Вимоги зовнішніх інтерфейсів**
        1. **Користувацькій інтерфейси**

Інтерфейс системи (веб версія) складається з:

* «Домашня» сторінка, яка містить опис сервісу та його можливостей;
* Розділ «Панель користувача» який містить форми для створення/редагування/видалення віджетів, списки створених віджетів та отриманих повідомлень ;
* Розділ «Керування акаунтом» який містить форму для зміни особистих даних користувача;
* Бічне меню в якому містяться список підрозділів та кнопки переходу між розділами;

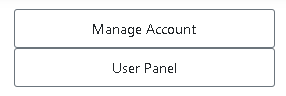


Рис. 2.1. Кнопки переходу між розділами

* На правому-верхньому куті карти розташовані дві кнопки, вигляд яких залежить від того, чи користувач авторизований (неавторизований): перехід до панелі користувача (вхід у систему); вихід з системи (реєстрація)



Рис. 2.2. Кнопка для переходу до панелі користувача (зліва) та виходу з системи (справа)



Рис. 2.3. Кнопка для входу в систему(зліва) та реєстрації (справа)

**2.2.5. Інші нефункціональні вимоги**

* + - 1. **Вимоги безпеки**

В проекті наявна система авторизації, яка дозволяє авторизованим користувачам додавати або змінювати віджети, налаштовувати персональну сторінку та спосіб отримання коштів. В базі даних, у записах про користувачів системи паролі зберігаються в зашифрованому вигляді, тому, якщо у разі скомпрометації бази даних, зловмисники не зможуть здійснити вхід в систему, так як паролі до облікових засобів, які вони отримають, будуть зашифровані.

* + - 1. **Вимоги продуктивності**

Кожен запит серверної частини має виконуватися асинхронно та максимально оптимізовано для найшвидшого відклику до сторони клієнта. Сповіщення про надходження пожертвування повинне відображатися у режимі реального часу з мінімальною затримкою.

* + - 1. **Атрибути якості програмного продукту**

Веб-аплікація повинна бути повністю адаптивною, тобто користувачі як з мобільних пристроїв повинні також мати змогу зручно і швидко користуватися системою. Повинна бути забезпечена валідація введених користувачем даних в будь-які наявні в системі поля, а також передбачені всі виняткові ситуації. У разі хибності введених даних система повинна вивести повідомлення про помилку. Для того, щоб забезпечити супровід даного продукту, та перенесення ресурсів системи на іншу платформу, повинні бути розроблені супутні інструкції.

* 1. **Вибір засобів для виконання завдання**

Для розробки даної системи було використано клієнт-серверну архітектуру, тому вона відповідно складається з клієнтської та серверної частини. Для розробки серверної частини буде використано платформу Microsoft .NET Core.

Ключовим фреймворком на сервері є ASP.NET Core – це кросплатформенний веб-фреймворк для проектів .NET Core, який надає широкий набір функцій для веб-аплікацій. Фреймворк об’єднує у єдину модель раніше окремі технології ASP.NET MVC та ASP.NET Web API. ASP.NET Core включає у себе такі корисні компоненти як Entity Framework Core, Identity Core, MVC Core та Razor Core. Для побудови клієнтської частини було використано вище згадану компоненту Razor Core, яка дозволяє будувати веб сторінки з використанням як html+css+javascript так і C# коду. Вона взаємодіє з архітектурною компонентою MVC Core яка забезпечує зв’язок між клієнстькою та серверною частинами. Для збереження даних буде використовуватись не реляційну базу даних – SQL Server. Для взаємодією з нею, на серверній частині буде використовуватися Entity Framework Core - структура об'єктно-реляційного відображення (ORM) яка використовується для взаємодії із базою даних.

Для створення під-системи авторизації та реєстрації та логіки користувачів була використана Identity Core, що дає готовий, широкий, зручний та гнучкий для змін функціонал. Для стилізації компонент інтерфейсу буде використовуватись фреймворк - Bootstrap, який є найпростішим інструментом у випадку якщо клієнт використовує Razor Core. Для реалізації відображення сповіщень у реальному часі було використано SignalR – це open source бібліотека програмного забезпечення для Microsoft ASP.NET, яка дозволяє серверній частині надсилати асинхронні сповіщення до веб-програм на стороні клієнта. Бібліотека також включає в себе компоненти JavaScript на стороні сервера і клієнта. SignalR використовує переваги WebSocket, HTML5 API, який забезпечує двонаправлений зв'язок між браузером і сервером.

* 1. **Висновок до розділу 2**

За допомогою результавів аналізу було створено специфікацію вимог до програмного забезпечення, в якій було описано основні сервіси системи та функціональні вимоги до них. Було визначено призначення та мету програмного продукту. Визначено та проаналізовано класи користувачів даної системи. Також були обрані програмні засоби для розробки даної веб-аплікації.

**РОЗДІЛ 3**

**АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ СИСТЕМИ ПОШУКУ ГЕОЛОКАЦІЇ ТА ІНФОРМАЦІЇ ПРО АВТОМАТИЧНІ ДЕФІБРИЛЯТОРИ**

* 1. **. Проектування поведінки розроблюваної системи**

В попередньому розділі було визначено два класи користувачів, які будуть наявні в системі, та функціональні вимоги які повинні бути реалізовані в проекті для задоволення потреб всіх цих класів користувачів. В даному розділі описано шляхи їхньої взаємодії з системою та доступний їм функціонал.

*Неавторизований користувач* увійшовши в систему попаде на головну (домашню) сторінку платформі, де він зможу ознайомитися з її можливостями та правилами використовування. Крім цього, він зможе авторизуватися у системі або зареєструватися. Також він має доступ по сторінки-профілю контентмейкера. Для цього йому потрібно мати на неї посилання. На цій сторінці він має змогу ознайомитися з короткою біографією блогера, перейти на соціальні мережі блогера а також здійснити пожертвування. Для пожертвування він має заповнити поле сума і по бажанню вказати свій псевдонім та вміст повідомлення. Також неавторизований користувач має доступ до перегляду роботи віджетів, але тільки у випадку, якщо він має на нього покликання. Проте це покликання не є у публічному доступі, а доступне тільки блогеру (власнику віджета), і блогеру варто зберігати його у таємниці, так як к логікою програми передбачений анонімний доступ до сторінки віджета, для його відобораження у програмах для проведення прямих трансляцій.

*Авторизованому користувачу або контентмейкеру (блогеру)* після авторизації до системи доступний весь функціонал, що й неавторизованому користувачу, а також доступ до панелі користувача, яка містить наступні можливості:

* керування персональними даними (зміна паролю, електронної почти, номеру картки);
* перегляд статистики;
* керування віджетами «Сповіщення» («Notification»), «Збір коштів» («Foundraising») та «Статистика» («Statistic») – створення, редагування, видалення, отримання покликання на сторінку роботи віджета;
* налаштування сторінки профілю (сторінки візитівки) – додавання фото профілю, тла профілю, додавання посилання на соціальні мережі;
* перегляд отриманих сповіщень;

В Додатку A зображена діаграма прецедентів, на якій розміщені актори, які є класами користувачів, а також функціонал для кожного з наявних в системі користувачів

**3.2. Загальна архітектура системи**

Під час розроблення будь-якої програмної системи необхідно недостатньо тільки чітко визначити її вимоги і функціональність, важливо є розроблення архітектури проекту. Нехтування даним етапом, або здійснення значущої помилки в проектуванні, може спричинити на етапі розроблення проблеми які будуть нести за собою фінансові та часові втрати.

Найкращим варіантом для розроблення даної системи, буде побудувати архітектуру, базовану на клієнт-серверній взаємодії та використаннями принципів MVC. Уся ця архітектурна буде реалізована за допомогою MVC Core. На стороні клієнта працюватиме веб-застосунок з використанням бібліотеки Microsoft .NET Core, за допомогою якого ми будемо представляти дані клієнту і опрацьовуватимемо його команди. Для авторизації, реєстрації нових користувачів, надсилання пожертв, роботи з даними, та інших операцій, які потребують виконання запитів до бази даних буде створена за допомогою компоненти MVC Core – контролери, які повертатимуть як результат відображення (сторінку) або REST відповідь, що буде серверною частиною системи.

Для реалізації логіки серверних ролей, акаунтів, авторизавції та реєстрації була обрана вбудована компонента Identity Core, яка автоматично створює потрібні модулі та всю логіку.

Для роботи з базою даних буде використано бібліотеку Entity Framework, яка дозволить проводити операції з базою даних за допомогою протоколу SSL. Базою даних для системи служитиме SQL Server.

Логічне представлення вище описаної архітектури зручно зобразити за допомогою діаграми розгортання, яка зображена на рис. 3.1.

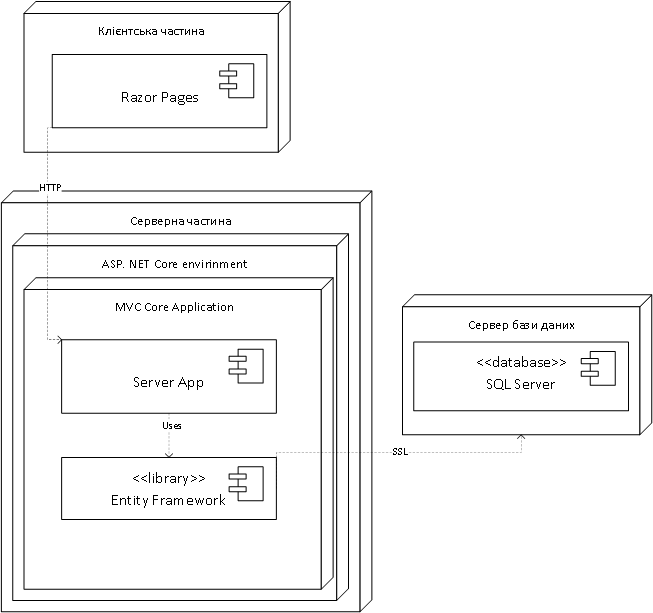


Рис. 3.1. Діаграма розгортання

**3.3. Проектування бази даних**

Для формування нашої бази даних було обрано SQL Server – одну з найпопулярніших баз даних. Дана СУБД підтримує транзакції та дозволяє створити структури даних, які б дозволяла б нам зберігати в певні поля тільки дані певного типу. Дана СУБД добре взаємодіє з ASP.NET та є чудовим вибором для даної технології.

Для створення структури бази даних було обрано так званий підхід “Code First”. Так як структура програми не є складно і не потребує складних рішень, то цей підхід є оптимальним. Цей підхід є одним із можливостей Entity Framework та Visual Studio та дозволяє створювати структуру БД за допомогою структури програми. Тобто, спершу створюються класи програми та налаштовується їх зв’язок між собою. Потім, за допомогою scaffold процедури, середовище Visual Studio створює потрібні таблиці та зв’язки. Для зміни у структуру БД використовуються міграції.

У базі даних присутні таблиці які мають префікс “AspNet”. Ці таблиці створені автоматично за допомогою ASP Identity.

В Додатку Б зображена діаграма бази даних.

У базі існують основні таблиці, та таблиці, які з’явилися у наслідок нормалізації (це таблиці типів/категорій). Дані авторизованого користувача зберігаються у таблиці “AspNetUsers”, яка має зв’язок один до одного з таблицею “UserProfile” – таблиця є собою сторінку профіля користувача. “UserProfile” має два файли для фотографій, тому вона має два зв’язки один до багатьох з таблицею “UploadData” – таблиця містить дані про завантажений до системи файл. Донат-сповіщення зберігаються у таблиці “DonateMsg” яка має зв’язок один до багатьох з таблицею “AspNetUsers”.

Для кожного віджета існує окрема таблиця. Всього їх є три: “FundraisingWidget”, “StatisticWidget” та “NotificationWidget”. Кожен віджет має стиль тексту, тому кожна з цих трьох таблиць має зв’язок один до багатьох з таблицею “TextStyle”, яка містить такі поля як: колір тексту, розмір шрифта, сім’ю шрифта, розмір простору між буквами та словами, прапорці «жирний», «косий», «підкреслений». Кожна з таблиць віджетів має зв’язок один до багатьох з “AspNetUsers”, так як користувач є власником і керівником віджетів та може мати багато їх у власності . Кожна таблиця віджетів має наступні спільні поля: назва, URL-посилання на сторінку відображення віджету, ідентифікатор контентмейкера. “FundraisingWidget” – це таблиця, яка зберігає дані віджета «Збір коштів», зберігає дані про початкову, кінцеву та зібрану суму грошей та кольори стилю відображення віджета. “StatisticWidget” – це таблиця, яка зберігає дані віджета «Статистика». Цей віджет має чотири категоріальних поля, тому таблиця має чотири зв’язки один до багатьох з наступними таблицями: “StatWidgetDirectionType”, “StatWidget”, “StatWidgetDisplayModeType”, “StatWidgetTimeIntervalType”. “NotificationWidget” – це таблиця, яка зберігає дані віджета «Сповіщення». Ця таблиця має зв’язок один до багатьох з таблицею “MsgWidgetContent”, яка зберігає набір зображення + звук для віджету. Так як ці зображення та звук є файлами, то вони зберігаються у вище згаданій таблиці “UploadData”, до якої “MsgWidgetContent” має зв’язок один до багатьох.

**3.4. Проектування серверної частини**

В загальному описі архітектури системи було визначено, що нашому програмному застосунку знадобиться серверна частина, яка надаватиме нам REST API та сторінки-відображення для запитів зі сторони клієнта.

Як було згадано вище, система буде реалізована за допомогою ASP.NET MVC Core, тому серверна частина буде складатися з ASP контролерів. Основу контролерів можна згенерувати за допомогою scaffold процедури і вносити зміни у згенерований код. В основі усіх контролерів лежать декілька основних функцій: створити (Create), редагувати (Edit), видалити (Delete), деталі (Details), список (Index). Проте кожна із цих функцій, окрім Index та Details мають два перевизначених формати для GET та POST запитів. GET запит повертає сторінку-відображення (форму) дії, а POST вже опрацьовує цю дію та вносить зміни до бази даних.

Серверна також частина містить SignalR «Хаб» (Hub), за допомогою якого сповіщення про донат надсилаютсья у режимі реального часу.

Маршрутизація реалізована також за допомогою MVC Core. Левова частка усіх кінцевих точок мають формат “{ControlerName}/{ActionName}”, за рідкісним виключенням для деяких функцій: “Donate/{nickname}” – містить кінцеву точку сторінки профіля контентмейкера з можливістю зробити пожертву, та маршрутів підсистеми Identity.

Підсумовуючи, на серверній частині слід описати декілька кінцевих точок (API endpoints), які будуть обробляти клієнтські запити. Список основних шляхів:

* /Identity/ – містить кінцеві точки для підсистеми акаунів, яка включає авторизацію, реєстрацію, кервування акаунтом;
* /Panel – містить кінцеві точки панелі користувача;
* /MsgWidgets – містить групу кінцевих точок для роботи з віджетами «Сповіщення», зокрема для читання, створення, редагування, відображення та видалення;
* /FundraisingWidgets – містить групу кінцевих точок для роботи з віджетами «Збір коштів», зокрема для читання, створення, редагування, відображення та видалення;
* /StatisticWidgets – містить групу кінцевих точок для роботи з віджетами «Статистика», зокрема для читання, створення, редагування, відображення та видалення;
* /UserProfiles/ містить кінцеві точки для роботи з сторінкою-профілем контентмейкера;
* /{ControlerName}/{ActionName} – кінцеві точки які містать функції контролерів;
* /donates – група кінцевих точок для хабу SignalR.

Діаграма послідовності (дивись Додаток В). дає змогу вдало продемонструвати роботу серверної частини системи, візуалізацію запитів, та їх подальше опрацювання.

**3.5. Проектування клієнтської частини**

Хоча робота програми MVC керується головним чином контролерами, але безпосередньо користувачеві програма доступна у вигляді сторінка-вигляд, яке формує зовнішній вигляд програми. В ASP.NET MVC сторінка-вигляд - це файли з розширенням cshtml, які містять код інтерфейсу користувача в основному на мові html. Тому для реалізації клієнтської частини буде використовуватися компонента Razor Core, яка є частиною MVC Core. Razor Core дозволяє використовувати код C# разом з html, що дозволяє швидко та зручно будувати потрібні веб-сторінки, які використовуватимуть модель даних, яка передаватиметься з серверної частини. Технологія Razor дозволяє створювати partial view (часткові сторінки), що дозволяє використовувати їх у декількох інших сторінках, це своєрідна модульність.

При виклик методу View контролер не виробляє рендеринг уявлення і не генерує розмітку html. Контролер лише готує дані та вибирає, яке уявлення треба повернути як об'єкт ViewResult. Потім вже об'єкт ViewResult звертається до двигуна представлення для рендерингу представлення у вихідний результат.

Клієнтська частина буде складатися з чотирьох типів веб сторінок: Razor Page (.cshtml+.cshtml.cs), Razor View(.cshtml), Razor Layout View (.cshtml) та Razor Component (.razor). Клієнтська частина є складовою єдиного проектного рішення (Visual Studio Project). Клієнтські сторінки розділені на дві папки: \Views\ (містить основні сторінки системи, тобто контролерів) та \Area\Identity\Pages \(містить сторінки для під-системи ідентифікації).

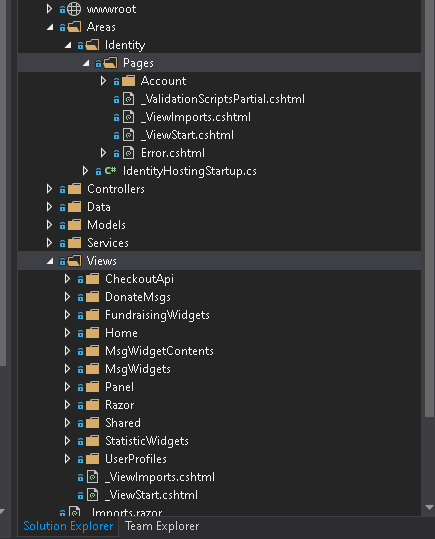


Рис.3.5.1. Клієнтська частина проекту

У папці \Views\ знаходяться підпапки, яки мають назви контролерів і які містять сторінки контролерів. Також, у папці \Views\ знаходиться папка “Shared”, яка містить у собі макети (Razor Layout), які є основою інших сторінок.

У папці \Area\Identity\Pages міститься під-система ідентифікації. Вона включає у себе як і клієнтське представлення (.cshtml) так і серверну бізнес логіку (.cshtml.cs), яка під час компіляції розділяється. Дані сторінки реалізовані за допомогою Razor Page, яка використовує Code Behind.

Для зручної та швидкої роботи було створено декілька часткових сторінок та компонент:

* \_FundraisingWidgetForm.cshtml – форма для віджету «Збір коштів»;
* \_MsgWidgetGrid.cshtml – форма для віджету «Сповіщення»;
* \_StatWidgetForm.cshtml – форма для віджету «Статистика»;
* \_PanelNav.cshtml – навігаціна панель;
* DisplayName.razor – компонента для власного відображення назви поля;

Для надання сторінка стилю я використав бібліотеку Bootstrap, яка дозволяє швидко та просто заставити сайт виглядати сучасно, мінімалістично та красиво.

**3.6. Висновок до розділу 3**

В даному розділі було описано та визначено загальну архітектуру системи у вигляді клієнта та сервера та використовує MVC патерн. Для візуального відображення системи було створено діаграму розгортання. Для демонстрації доступних сценаріїв взаємодії та використання системи для двох наявних класів користувачів та наглядного відображення було створено діаграму прецедентів. Також було визначено структуру та зв’язки таблиць бази даних. Для відображення деталей таблиць (полів, їхніх типів даних, ключів, зв’язків) було сформовано діаграму бази даних. Було описано роботу серверної частини та її зв’язок з клієнтською частиною. Так як використано підхід MVC, то обидві частини між собою тісно пов’язані. Було описано наявні кінцеві точки, на які клієнт буде відправляти запити, і сценарії для опрацювання кожного з цих запитів. Для демонстрації роботи кінцевих точок та їх сценаріїв опрацювання було стаоврено діаграму послідовності.

**РОЗДІЛ 4**

**РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ**

**4.1. Налаштування та реалізація серверної частини**

Основним інструментом для створення серверної частини програмної системи є ASP.Net MVC Core. З його допомогою було створено контролери та функції, що слугують кінцевими точками, до яких буде звертатися клієнтська сторона.

Для налаштування серверної сторони використовується файл Startup.cs, який містить потрібні функції.

Частина коду з файлу Startup.cs яка відповідає за налаштування сервісів:

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<ApplicationDbContext>(options =>

options.UseSqlServer(

Configuration.GetConnectionString("DefaultConnection")));

services.AddDefaultIdentity<ApplicationUser>(options =>

{

options.User.RequireUniqueEmail = true;

options.SignIn.RequireConfirmedAccount = true;

}).

AddEntityFrameworkStores<ApplicationDbContext>();

services.AddControllersWithViews().AddControllersAsServices();

services.AddRazorPages();

services.AddServerSideBlazor();

services.AddSignalR().AddNewtonsoftJsonProtocol(opt => {

opt.PayloadSerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;

}); ;

services.AddScoped<IRefreshModelService, RefreshModelService>();

services.AddScoped<DonateHub>();

services.AddHttpContextAccessor();

}

Частина коду з файлу Startup.cs яка відповідає за налаштування серверної частини узагальному:

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

app.UseDatabaseErrorPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

endpoints.MapRazorPages();

endpoints.MapBlazorHub();

endpoints.MapHub<DonateHub>("/donates");

});

var cultureInfo = new CultureInfo("en-US");

CultureInfo.DefaultThreadCurrentCulture = cultureInfo;

CultureInfo.DefaultThreadCurrentUICulture = cultureInfo;

}

Для безпечного та гнучкого під’єдняння до бази даних, налаштування хостингу та деяких інших налаштувань створено файл appsettings.json який містить певні параметри та їх значення (ConnectionStrings, DefaultConnection, AllowedHosts тощо).

Для взаємодії з базою даних використовується Entity Framework Core. Основним його елементом є контекст бази даних (ApplicationDbContext) який містить інформацію про структуру БД.

**4.2. Опис роботи системи та програмних рішень на клієнтській частині**

Платформа для фінансової підтримки контентмейкерів, повністю адаптивна, та може використовуватися через будь-який сучасний браузер, за наявності інтернету на пристрої користувача. Інтерфесй користувача даної системи було розроблено за принципами евристики Якоба Нільсена, а отже він простий та зручний у використанні.

**РОЗДІЛ 5**

**ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОБОТИ**

**5.1. Економічна характеристика проектного рішення**

# ВИСНОВКИ

**СПИСОК ЛІТЕРАТУРИ**

1. Інформаційна ера Вікіпедія – відкрита енциклопедія [Електронний ресурс]. – Режим доступу : <https://uk.wikipedia.org/wiki/Інформаційна_ера>
2. In 2021, the Internet went for TikTok, space and beyond [Електронний ресурс] – Режим доступу:

<https://blog.cloudflare.com/popular-domains-year-in-review-2021/>

1. Most popular social networks worldwide as of January 2022, ranked by number of monthly active users [Електронний ресурс]. – Режим доступу :

<https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>

1. OBS Open Broadcaster Software [Електронний ресурс]. – Режим доступу :

<https://obsproject.com/uk>

1. Донат – що це таке в соціальних мережах, відеоіграх та на стрімах. [Електронний ресурс]. – Режим доступу :

<https://termin.in.ua/donat/>

1. Patreon, Вікіпедія – відкрита енциклопедія [Електронний ресурс]. – Режим доступу :

<https://en.wikipedia.org/wiki/Patreon>

1. Документація по ASP.NET [Електронний ресурс]. – Режим доступу :

<https://docs.microsoft.com/uk-ua/aspnet/core/?view=aspnetcore-6.0>

1. Документація по Entity Framework [Електронний ресурс]. – Режим доступу :

<https://docs.microsoft.com/uk-ua/ef/>

1. Entity Framework Tutorial [Електронний ресурс]. – Режим доступу :

<https://www.entityframeworktutorial.net/>

1. Документація то ASP.NET Razor Pages [Електронний ресурс]. – Режим доступу :

<https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/razor-pages/>

# ДОДАТОК А. Діаграма прецедентів

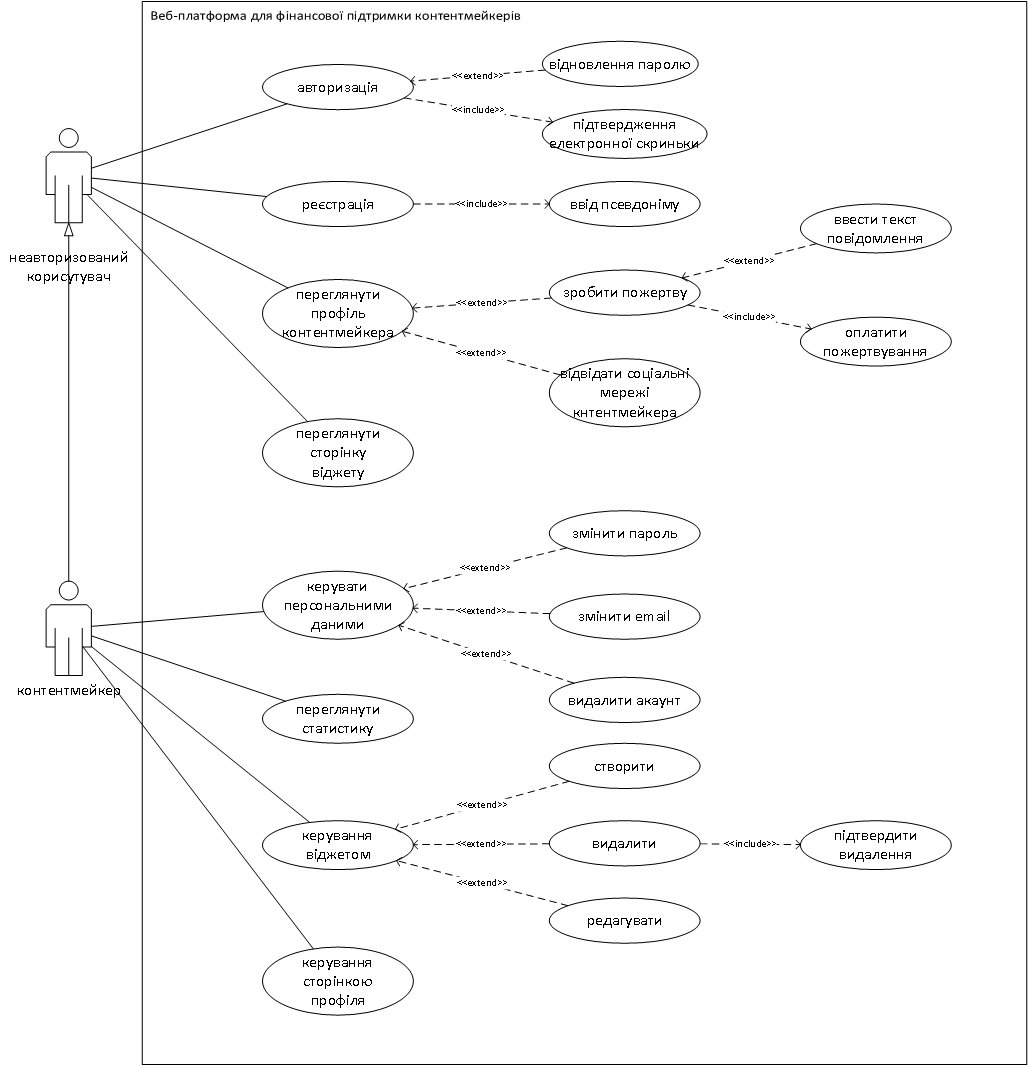


Рис. А.1. Діаграма прецедентів

# ДОДАТОК Б. Діаграма бази даних

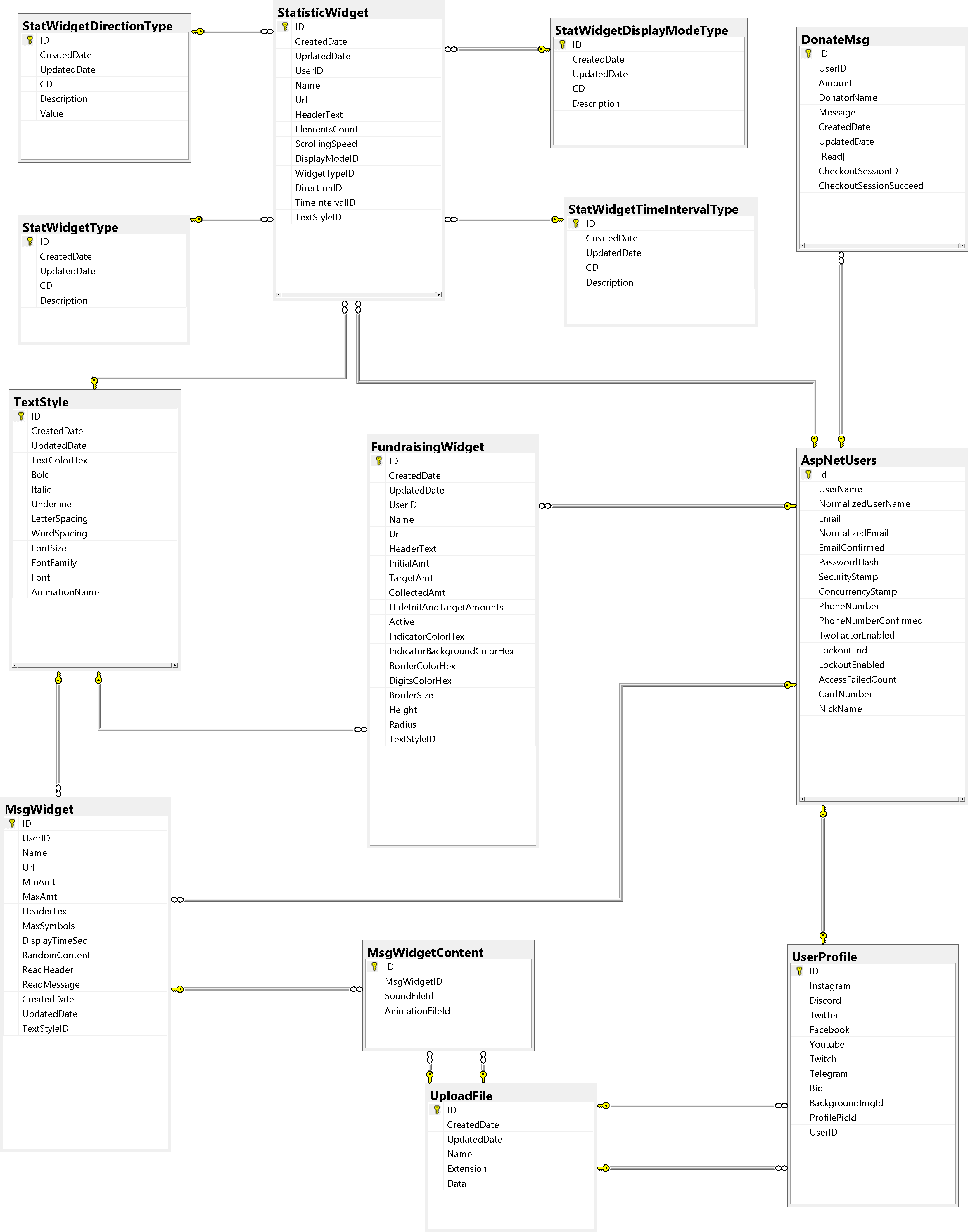


Рис. Б.1. Діаграма бази даних

# ДОДАТОК В. Діаграма послідовності

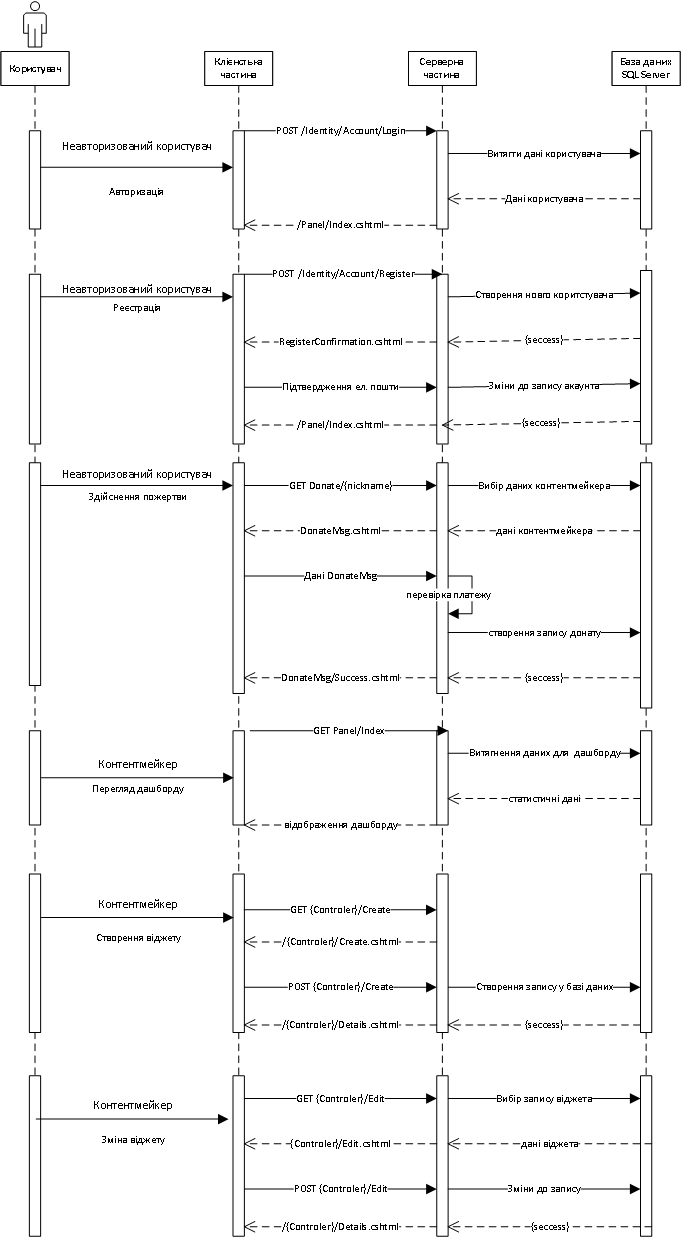


Рис. В.1. Діаграма послідонсті