МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально науковий інститут комп'ютерних наук та управління проєктами

Кафедра управління проєктами

«Допущений до захисту»
Завідувач кафедри
_____ Чернов С.К.
«09» грудня 2024р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти **«магістр»**

на тему: Управління вимогами проєкту створення платформи для захисту персональних даних відповідно до GDPR (аналог One Trust)

Виконав: студент 6171мз групи
_____ Кривда В.І.
(підпис)
Керівник роботи: д.т.н., професор
_____ Чернов С.К.
(підпис)

Миколаїв – 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально науковий інститут комп'ютерних наук та управління проєктами

Кафедра Управління проєктами
Спеціальність 122 Комп'ютерні науки
Освітня програма «Управління проєктами»

«ЗАТВЕРДЖУЮ»
Гарант освітньої програми
_____Чернова Л.С.
(підпис)
«02» грудня 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТА
на здобуття ступеня вищої освіти **«магістр»**

Студентці ***Кривді Вікторії Ігорівні***

1. Тема роботи: «Управління вимогами проєкту створення платформи для захисту персональних даних відповідно до GDPR (аналог One Trust)»
Затверджені наказом ректора № 1227-уч. від «02» грудня 2024 року
2. Термін подання роботи: до 09.12.2024р.
3. Вихідні дані по роботі: фахова література, довідники з управління проєктами, матеріали конференцій, репозитарій університету
4. Перелік питань, що належать до розробки (найменування розділів)
Розділ 1. Теоретичні основи управління проєктами створення релізів до платформи для захисту персональних даних
Розділ 2. Розробка платформи для захисту персональних даних відповідно до GDPR
Розділ 3. Управління проєктом розробки платформи для захисту персональних даних відповідно до GDPR
Розділ 4. Охорона праці
Розділ 5. Охорона навколишнього середовища
5. Перелік презентаційних матеріалів виконаний в програмі Power Point

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 4. Охорона праці	Мозговий А.М., доцент		
Розділ 5. Охорона навколишнього середовища	Мозговий А.М., доцент		

7. Дата видачі завдання 16.09.2024р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вивчення літературних джерел з предмету дослідження	Вересень 2024	Виконано
2	Складання розгорнутого плану магістерської роботи та ознайомлення керівника з планом кваліфікаційної роботи	Вересень 2024	Виконано
3	Написання розділу 1	Вересень – жовтень 2024	Виконано
4	Написання розділу 2	Жовтень 2024	Виконано
5	Написання розділу 3	Листопад 2024	Виконано
6	Написання розділів з охорони праці та навколишнього середовища (4,5)	Листопад 2024	Виконано
7	Оформлення магістерської роботи	Грудень 2024	Виконано
8	Передача магістерської роботи рецензенту для рецензування	Грудень 2024	Виконано
9	Передача магістерської роботи науковому керівникові для написання відгуку	до 05.12.2024	Виконано
10	Попередній захист магістерської роботи	09.12.2024	Виконано
11	Захист магістерської роботи	23.12.2024	Виконано

Студент _____ Кривда В.І.
(підпис)

Керівник роботи _____ Чернов С.К.
(підпис)

АНОТАЦІЯ

Кривда В. І. Управління вимогами проєкту створення платформи для захисту персональних даних відповідно до GDPR (аналог One Trust). Кваліфікаційна робота зі спеціальності 122 «Комп'ютерні науки», освітньої програми «Управління проєктами». Миколаїв: Національний університет кораблебудування імені адмірала Макарова. 2024 рік. 92 с.

В роботі проведено аналіз особливостей проєкту створення платформи для захисту персональних даних відповідно до GDPR, виконано огляд засобів з управління вимогами IT-проєктів, розроблено моделі проєкту в MS Project, розроблено інформаційну систему управління вимогами проєкту платформи для захисту персональних даних відповідно до GDPR та проведено аналіз ефективності управління вимогами в проєкті.

Ключові слова: захист персональних даних, платформа для захисту персональних даних; управління вимогами IT-проєкту; регламент захисту даних GDPR; інтелектуальна система; машинне навчання, 122 «Комп'ютерні науки».

ABSTRACT

Viktoria Kryvda. Managing the requirements of a project to create a platform for the protection of personal data in accordance with the GDPR (analogous to One Trust). Qualification work on specialty 122 "Computer Science", educational program "Project Management". Mykolaiv: Admiral Makarov National University of Shipbuilding. 2024. 92 pag.

The paper analyzes the features of the project for creating a platform for personal data protection in accordance with the GDPR, reviews the tools for managing IT project requirements, develops project models in MS Project, develops an information system for managing the requirements of the project for the platform for personal data protection in accordance with the GDPR, and analyzes the effectiveness of requirements management in the project.

Keywords*:* personal data protection, personal data protection platform; IT project requirements management; GDPR data protection regulation; intelligent system; machine learning.

ЗМІСТ

ВСТУП

З розвитком діяльності організацій у різних сферах виникла потреба в розробці методів і підходів, які забезпечують ефективне управління їх процесами. Це призвело до об’єднання певних послідовностей дій, спрямованих на досягнення конкретної мети, у вигляді проєктів. Особливо актуальним це стало для високотехнологічних галузей, де велика частка роботи пов'язана з інноваціями. У таких випадках більшість завдань є унікальними, мають визначену тривалість, а також чіткі дати початку й завершення. Це принципово відрізняє їх від управління функціональними підрозділами, де виконуються стандартні завдання без чітких часових рамок.

У відповідь на ці виклики компанії та організації почали активно впроваджувати та розробляти основи проєктного підходу в управлінні. Спочатку методи управління не були чітко структуровані чи об’єднані в рамках єдиної теорії. Основна увага зосереджувалася на вирішенні завдань організації та оптимізації технологічних процесів управління людськими й матеріальними ресурсами. Водночас важливим для керівників різних рівнів було отримання формалізованих звітів про стан підрозділів організації. Прозорість звітності стала ключовою умовою для об’єктивного оцінювання результатів управлінських рішень і планування подальших кроків для ефективного управління. При цьому вкрай важливим є захист персональних даних.

Захист персональних даних за GDPR – це забезпечення їх конфіденційності, цілісності та доступності. Це включає застосування різноманітних технічних і організаційних заходів, зокрема шифрування, контроль доступу та анонімізацію.

Основними принципами є законність, прозорість, цільова обмеженість, мінімізація даних, точність, обмеження зберігання, цілісність і конфіденційність.

Права суб'єктів даних полягають у доступі до даних, їх виправлення, видалення, обмеження обробки, перенесення, заперечення проти обробки, захист від автоматизованого прийняття рішень.

Обов’язки компаній полягають у призначенні відповідального за захист даних (DPO), повідомлення про порушення (до 72 годин), забезпечення безпеки через технічні та організаційні заходи.

Об’єктом дослідження у цій роботі є процеси управління проєктом розробки платформи для захисту персональних даних відповідно до GDPR.

Предмет дослідження – моделі та технології управління платформи для захисту персональних даних відповідно до GDPR.

Мета роботи – створення поточного релізу платформи є забезпечення відповідності актуальним вимогам GDPR, впровадження нового функціоналу для підвищення безпеки та продуктивності, а також вдосконалення користувацького досвіду.

Основні завдання кваліфікаційної роботи:

- Провести аналіз особливостей проєкту створення платформи для захисту персональних даних відповідно до GDPR.
- Провести огляд засобів з управління вимогами IT-проєктів.
- Розробити моделі проєкту в MS Project.
- Розробити інформаційну систему управління вимогами проєкту платформи для захисту персональних даних відповідно до GDPR.
- Провести аналіз ефективності управління вимогами в проекті створення платформи для захисту персональних даних відповідно до GDPR.

Наукова новизна роботи: розроблено інформаційну систему управління вимогами проєкту платформи для захисту персональних даних відповідно до GDPR.

Практичне значення: розроблено моделі управління вимогами проєкту розробки платформи для захисту персональних даних, що базується на

використанні сучасних програмних засобів, зокрема інформаційної системи Microsoft Project.

Методи дослідження: у роботі застосовано системний аналіз, вивчення літературних джерел, формалізацію, узагальнення та моделювання.

Інструментальні засоби: для управління проєктом розробки платформи для захисту персональних даних відповідно до GDPR використано інформаційну систему Microsoft Project.

Структура та обсяг кваліфікаційної роботи магістра. Кваліфікаційна робота складається з анотації, вступу, п’ятьох розділів, висновків, списку використаних джерел з 30 найменувань – на 4 сторінках, 1 додатку на 6 сторінках. Повний обсяг роботи – 100 сторінок.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЄКТАМИ СТВОРЕННЯ РЕЛІЗІВ ДО ПЛАТФОРМИ ДЛЯ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ

1.1 Особливості проєкту створення платформи для захисту персональних даних відповідно до GDPR

Компанія «DataGuard Solutions» спеціалізується на розробці програмного забезпечення у сфері захисту даних. Це продуктова IT-компанія, орієнтована на створення сучасних і надійних платформ для захисту персональних даних, з урахуванням міжнародних стандартів. Компанія активно співпрацює з регуляторами і консультує організації щодо впровадження технологічних рішень, які відповідають нормативно-правовій базі [1].

Проєкт спрямований на організації та підприємства, які обробляють значні обсяги персональних даних громадян Європейського Союзу. Споживачами продукту будуть як великі корпорації, що працюють у сфері електронної комерції наприклад, міжнародні онлайн-ритейлери, такі як «GlobalStore EU», так і фінансові установи, які обслуговують клієнтів у різних країнах ЄС, наприклад, банки та платіжні системи. Крім того, платформою можуть користуватися невеликі технологічні компанії, які розробляють додатки для європейського ринку і прагнуть забезпечити дотримання вимог GDPR із мінімальними витратами на розробку власних рішень.

Загальний регламент захисту даних (GDPR) — це нормативний акт Європейського Союзу, спрямований на захист персональних даних та конфіденційності громадян. Абревіатура розшифровується як General Data Protection Regulation. Регламент встановлює правила щодо збору, обробки та зберігання персональних даних, щоб забезпечити права громадян на конфіденційність та контроль над своїми даними [2]. Під терміном «регламент» мається на увазі обов’язковий до виконання нормативний документ, який має однакову юридичну силу у всіх державах-членах ЄС. Він є загальним, оскільки

охоплює всі аспекти обробки персональних даних незалежно від галузі або організації, що їх обробляє. Регламент застосовується до будь-яких організацій, що працюють з даними громадян ЄС, незалежно від їх місцезнаходження. GDPR регулює захист даних, таких як ім’я, адреса, електронна пошта, IP-адреса, медична інформація та фінансові дані.

Під «захистом даних» розуміємо забезпечення конфіденційності, цілісності та доступності даних, що включає різні технічні та організаційні заходи, такі як шифрування, контроль доступу та анонімізація [3].

Стейкхолдерів проєкту можна поділити чотири групи: клієнти — організації, що використовують платформу для захисту персональних даних, регуляторні органи; консультанти з GDPR; а також технічна команда DataGuard Solutions. Клієнти очікують від продукту автоматизації процесів забезпечення відповідності GDPR, мінімізації ризиків штрафів і можливості оперативного реагування на запити користувачів. Регулятори прагнуть гарантувати, що платформа допомагає організаціям дотримуватись правил і забезпечує прозорість обробки даних. Консультанти з GDPR очікують від платформи адаптивності до змін у нормативній базі та зручності інтеграції з існуючими процесами. Технічна команда зосереджена на створенні ефективного, масштабованого та безпечного продукту, який задовольняє вимоги клієнтів і регуляторів.

Особливість проєкту полягає у високій динаміці змін у вимогах регуляторів і необхідності їх швидкої імплементації. Вимоги GDPR можуть змінюватися через нові загрози, сучасні наукові рішення або судові прецеденти. Тому платформа повинна бути адаптивною і модульною. Окрім того, проєкт реалізується у тісній співпраці з різними регуляторними органами, що вимагає чіткої координації та розуміння специфіки кожного національного регулятора в країнах ЄС.

Створення першої версії платформи значно відрізняється від її подальших оновлень. Початкова версія передбачає створення основної архітектури системи,

включаючи модулі для збирання згоди користувачів, обробки запитів на доступ до даних, видалення даних та звітності. Вона також має включати базові технічні засоби захисту, такі як шифрування даних, анонімізація та контроль доступу. Подальші релізи зосереджені на оновленнях відповідно до нових вимог GDPR, впровадженні інструментів для виявлення нових загроз та підвищенні продуктивності системи. Релізи також можуть включати поліпшення користувацького інтерфейсу, адаптацію до нових мов і регіонів та інтеграцію з іншими сервісами.

Функціонал платформи охоплює управління згодою користувачів на обробку даних, реалізацію права на доступ, видалення та передачу даних, а також створення автоматизованих звітів для регуляторів. Інтеграція з системами сповіщення про витоки даних забезпечує виконання вимог повідомлення про інциденти впродовж 72 годин.

Платформа підтримує шифрування та анонімізацію персональних даних, включає модулі для обробки запитів користувачів щодо їх прав, а також забезпечує моніторинг відповідності нормативним вимогам у режимі реального часу. Проблема в управлінні проєктом полягає у непередбачуваності змін у регуляторній базі та складнощах у забезпеченні відповідності новим вимогам у короткі строки. Існує ризик пропуску важливих змін через нерегулярність оновлень і залежність від численних джерел інформації, це вимагає від команди розробників не тільки високої технічної компетенції, але й постійної взаємодії з консультантами, галузевими експертами та регуляторами для своєчасної ідентифікації нових вимог і ризиків. IT-проєкт пов’язаний зі створенням оновлених релізів відповідно до оновлених вимог GDPR, впровадженням інструментів захисту від нових загроз та підвищенні продуктивності системи. Відмінністю цього проєкту буде не регулярне впровадження нових релізів, тому що критерієм ініціації проєкту зі створення нового релізу буде критична маса накопичених змін, а не час, як у класичних продуктових проєктах.

1.2 Особливості управління проєктами створення релізів до платформи для захисту персональних даних

Платформа для захисту персональних даних відповідно до GDPR є складною технічною та організаційною системою, що має поєднуватиме високий рівень безпеки, адаптивність до змін нормативної бази та інтеграцію з існуючими рішеннями споживачів. Вона забезпечує шифрування, анонімізацію, контроль доступу, автоматизовану обробку запитів на видалення або передачу даних, а також інструменти для виявлення і повідомлення про інциденти у визначені строки.

Платформа має бути сумісною із регуляторними вимогами різних країн, що додає складності через специфічність та нерівномірність змін у цих вимогах. Важливим викликом є необхідність адаптувати функціонал до нових кіберзагроз і технологій, а також враховувати судові рішення, які можуть впливати на тлумачення положень GDPR, що є поширеною практикою у європейських країнах.

Проєкти з вдосконалення такої платформи вирізняються нерегулярним характером релізів, які зініціюються накопиченням критичної маси змін, а не за чітким графіком. Це робить їх складними для управління, адже враховуються не тільки зміни у нормативній базі, але й нові технічні виклики та рекомендації експертів [4].

Ризиків додає й необхідність підтримувати високі стандарти якості, оскільки неврахування навіть незначних деталей може призвести до штрафів чи втрати репутації. Для зниження ризиків неврахування усіх вимог необхідно створити систему багаторівневого моніторингу. Ця система має включати підписку на офіційні оновлення від регуляторів, участь у галузевих конференціях, консультації з експертами, використання автоматизованих інструментів для відстеження змін, а також регулярні аудити відповідності платформи нормативним вимогам і аналіз безпеки. Постійне навчання команди, з акцентом на актуальні зміни, також є невід’ємною частиною процесу зниження

ризиків [5].

Для ефективного управління проєктами створення релізів найкраще використовувати сучасні інструменти управління IT-проєктами, такі як Jira для організації завдань і планування спринтів, Confluence для документування вимог і технічних специфікацій, Wrike для координації між командами, а також TeamGantt для візуалізації графіків за допомогою діаграм Ганта. Ці інструменти дозволяють краще організувати процес, підвищити прозорість виконання та забезпечити синхронізацію між учасниками проєкту [6].

Щодо методологій управління, до таких проєктів можуть бути застосовані Agile, Kanban, Waterfall, Hybrid та Critical Chain Project Management (CCPM). Agile, наприклад, забезпечить швидкість ітераційного розвитку, але його ефективність може бути обмежена нерегулярністю оновлень [7]. Kanban добре підходить для моніторингу поточного стану виконання завдань і зменшення затримок, але не забезпечує довгострокового планування. Waterfall пропонує чітку структуру, однак недостатньо гнучкий для швидкої адаптації до змін. Комбінація методологій, яка поєднує гнучкість Agile із структурованістю Waterfall, є більш універсальною і підходить для нашого випадку. CCPM корисний для оптимізації ресурсів і врахування залежностей між задачами, але вимагає значних зусиль в управлінні. У випадку проєктів із нерегулярними релізами найбільш оптимальною є гібридна методологія, яка дозволяє створювати структуровані релізи, враховуючи чіткі вимоги GDPR, та зберігати достатню гнучкість для адаптації до змін.

Застосування Agile-елементів, таких як Kanban-дошки для управління потоком завдань [8], у поєднанні із Waterfall-орієнтованими діаграмами Ганта для довгострокового планування забезпечить необхідний баланс між чіткістю, адаптивністю і якістю виконання проєктів.

1.3 Засоби з управління вимогами IT-проєктів

Вимоги в проєктах – це формальні або неформальні описання потреб, очікувань чи обмежень, які повинні бути враховані під час створення, впровадження або оновлення продукту чи системи [9]. Вони є основою для всіх етапів управління проєктом, визначаючи, яким повинен бути кінцевий результат, його функції, характеристики та обмеження. Вимоги в «звичайних» проєктах часто мають статичний характер, базуючись на сталих, чітко визначених очікуваннях замовника, наприклад, будівництво споруди чи розробка промислового продукту з фіксованими характеристиками. Натомість вимоги в IT-проєктах є значно динамічнішими та складнішими, адже враховують швидко змінювані технологічні середовища, специфіку кінцевого користувача та необхідність інтеграції з існуючими цифровими системами. Вони часто змінюються під час реалізації проєкту, що вимагає гнучкості в управлінні [10].

Управління вимогами — це систематизований підхід щодо виявлення, аналізу, документування, перевірки, затвердження, моніторингу та контролю змін вимог протягом усього життєвого циклу проєкту чи продукту [11]. Цей процес забезпечує розуміння усіма сторонами вимог однаково, і те, що вимоги відображають потреби зацікавлених сторін, а також залишаються актуальними та контрольованими упродовж виконання проєкту. Управління вимогами забезпечує успішне завершення проєктів, через зменшення ризику невідповідності кінцевого продукту очікуванням клієнтів або користувачів.

Управління вимогами в «звичайних» проєктах зазвичай орієнтоване на стійкість і передбачуваність. Вимоги часто формуються на ранніх етапах і залишаються фіксованими, що дозволяє ефективно планувати ресурси, час і бюджет. У той же час, управління вимогами в IT-проєктах характеризується високою динамікою змін, необхідністю частої перевірки актуальності вимог, їхньої пріоритизації та інтеграції з іншими системами. Наприклад, в IT-проєкті для розробки веб-додатка нові вимоги можуть з’явитися на кожному етапі, що викликає потребу в адаптивних методах управління.

Особливістю управління вимогами в «звичайних» IT-проєктах, у яких продукт створюється і передається замовнику, є чіткий життєвий цикл вимог: від початкового збору й аналізу до перевірки відповідності готового продукту. Завданням є досягнення максимального узгодження між замовником і командою розробників. Натомість у проєктах, як наш, де розробляються нерегулярні релізи платформи, ключовою складністю є необхідність постійного відстеження змін у середовищі, інтеграції нових вимог у вже існуючу архітектуру платформи та балансування між стабільністю і гнучкістю. Крім того, такі проєкти стикаються з ризиком, що накопичені зміни не будуть своєчасно враховані, що може вплинути на відповідність вимогам нормативної бази або очікуванням користувачів.

В практиці управління IT-проєктами застосовують різні підходи до управління вимогами, так в [12] автори описують особливості управління вимогами у стратегічних IT-проєктах, аналізують техніки та моделі, які застосовуються для оцінки ризиків і контролю життєвого циклу вимог.

В [13] досліджено використання моделей SysML та інструментів на базі великих мовних моделей (LLMs) для перевірки технічних специфікацій відповідно до вимог проєкту.

Дослідження фокусується на специфікаціях у сфері розумних мереж (smart grid), де вимоги формуються в природній мові та перевіряються за допомогою LLM і формальних методів, таких як SysML та обмеження OCL. Автори експериментально показали, що LLMs можуть визначати, які вимоги виконуються, а які — ні, з точністю до 94%. Вони порівняли LLM із традиційними формальними методами, запропонувавши використання інструкцій з покроковою логікою та покращених алгоритмів для точнішої перевірки.

Запропоновані підходи можуть бути корисними для перевірки відповідності вимог у нашому проєкті з нерегулярними релізами, особливо якщо специфікації вимог формуються в природній мові. Проте, LLM слід

використовувати як додатковий інструмент до формальних методів, оскільки повна заміна може бути недостатньо надійною для критичних оновлень, пов’язаних із безпекою та відповідністю нормативним вимогам.

Дослідження [14] описує моделі прогнозування обсягів роботи на основі оцінки вимог у контексті розробки програмного забезпечення.

Отже, можна зробити висновок, що більшість існуючих моделей та підходів до управління вимогами в IT-проєктах зосереджені на ручному збиранні, структуруванні та обробці вимог. Хоча ці підходи забезпечують певну ефективність у стабільних проєктах, вони мають значні обмеження, коли мова йде про динамічні проєкти зі змінними та нерегулярними релізами, як наш. Існуючі системи, як правило, покладаються на користувача для формування вимог і встановлення пріоритетів, що не тільки збільшує тривалість процесу, але й підвищує ризик пропуску критичних вимог або виникнення конфліктів між ними.

Висновки до розділу 1. В першому розділі проаналізовано предметну область управління вимогами IT-проєктів, розглянуто особливості та методи управління вимогами до різнорідних ситуацій.

Аналіз особливостей проєкту створення платформи для захисту персональних даних відповідно до GDPR виявив необхідність контролювати вимоги до виконання проєктних робіт для кожного розділу. Платформа потребує постійної адаптації до актуальних вимог безпеки даних, тому власник продукту має постійно відслідковувати зовнішні чинники що забезпечують чи регулюють рівень безпеки. При накопиченні критичної маси вимог необхідно оновлювати платформу та випускати її новий реліз. Однією з особливостей цього проєкту є те, що такі релізи є не регулярними за часом.

З урахуванням цього було розглянуто особливості даного проєкту та засобі зі збору та формування вимог до поточних релізів. Відсутність автоматизації та інтелектуального аналізу у розглянутих засобах управління вимогами означає,

що ці підходи не забезпечують проактивного виявлення та вирішення проблем, таких як суперечності в запитах або непослідовність у трактуванні вимог різними сторонами.

Для проєктів зі специфікою нерегулярних релізів, де зміни накопичуються з часом і потребують гнучкої інтеграції, використання традиційних моделей управління вимогами є недостатньо ефективним. Ручне управління стає значним бар'єром для оперативного реагування на зміни в нормативній базі, появу нових технічних загроз чи запити користувачів. Особливо це актуально в умовах багатокомпонентних платформ, які потребують інтеграції нових функцій без порушення стабільності існуючої архітектури.

Таким чином, необхідно створити інтелектуальну систему для збирання та управління вимогами. Така система буде використовувати сучасні технології штучного інтелекту та машинного навчання для автоматичного збирання вимог із різних джерел, аналізу їхньої пріоритетності, виявлення можливих конфліктів і пропонування оптимальних рішень. Це дозволить мінімізувати ручну роботу, забезпечити оперативність процесу управління та знизити ризики неврахування важливих змін. Інтелектуальна система стане основним інструментом для ефективного управління вимогами у проєкті, забезпечуючи високу якість релізів навіть у умовах динамічного середовища та нерегулярних оновлень. Результати кваліфікаційної роботи представлені у тезах доповідей Міжнародної науково-практичної конференції [30].

РОЗДІЛ 2. РОЗРОБКА ПЛАТФОРМИ ДЛЯ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ ВІДПОВІДНО ДО GDPR

2.1 Розробка продукту проєкту

Даний проєкт спрямований на створення інноваційної платформи для захисту персональних даних, яка забезпечує відповідність вимогам GDPR та інтеграцію з європейськими регуляторами. Початковий етап розробки платформи включав створення базової архітектури, реалізацію функцій для управління згодою користувачів, обробки запитів на доступ і видалення даних, забезпечення шифрування, анонімізації та контролю доступу. Платформа з самого початку була спроектована як масштабована та адаптивна система, що враховує змінні регуляторні вимоги і нові загрози.

Нині реалізується проєкт створення чергового релізу платформи, який включає оновлення функціоналу та вдосконалення технічної бази для підвищення її продуктивності та безпеки. У цьому контексті особливу увагу приділено впровадженню нових механізмів моніторингу кіберзагроз, адаптації до оновлених нормативних вимог та інтеграції нових технологічних рішень для автоматизації процесів. Проєкт передбачає виявлення і обробку накопичених вимог, аналіз їх пріоритетності та забезпечення повної сумісності оновлень із існуючими компонентами [15].

Тому, метою проєкту створення поточного релізу платформи є забезпечення відповідності актуальним вимогам GDPR, впровадження нового функціоналу для підвищення безпеки та продуктивності, а також вдосконалення користувацького досвіду.

Для досягнення мети були сформовані наступні завдання:

1. Збір та аналіз вимог з різних джерел, включаючи регуляторні оновлення, кіберзагрози та запити користувачів, із подальшим формуванням технічного завдання для релізу.

2. Перевірка відсутності конфліктів між новими вимогами та попереднім

функціоналом платформи для забезпечення стабільності системи.

3. Розробка нового функціоналу шляхом кодування згідно з технічним завданням.

4. Тестування нового функціоналу та інтеграція його до існуючої архітектури платформи для забезпечення коректної роботи.

5. Випуск релізу, включаючи розгортання оновленої платформи та її передачу користувачам.

Якщо проєкт випуску релізу вважати повторювальним, можна створити шаблон продукту проєкту, який забезпечує уніфікований підхід до реалізації нових функціональних оновлень платформи [16].

Шаблон продукту проєкту релізу:

Продуктом кожного релізу є оновлена версія платформи для захисту персональних даних. Основний функціонал:

– реалізація нового технічного функціоналу відповідно до змін у вимогах GDPR та нормативній базі;

– поліпшені засоби кіберзахисту, зокрема шифрування, контроль доступу та моніторинг потенційних загроз;

– оптимізація продуктивності системи, включаючи зменшення затримок при обробці запитів;

– вдосконалення користувацького інтерфейсу для підвищення зручності взаємодії;

– забезпечення інтеграції нових модулів із існуючою архітектурою платформи.

Структура продукту проєкту:

1. Основна архітектура: Оновлений бекенд і бази даних із впровадженим новим функціоналом.

2. Функціональні модулі: Модулі для управління вимогами GDPR, кібербезпеки, шифрування і анонімізації.

3. Інтерфейс користувача: Адаптивний інтерфейс, оптимізований для

нових функцій.

4. Система інтеграції: Механізми для інтеграції нового функціоналу в існуючу систему з мінімальними змінами.

5. Тестові сценарії: Автоматизовані і ручні тести для перевірки функціональності та відповідності вимогам.

Для розробки такої платформи найкраще підходять мови, що забезпечують високу продуктивність, безпеку та гнучкість у масштабуванні. Для Backend це Python або Java завдяки їх надійності, розвиненим бібліотекам для безпеки (наприклад, Flask/Django для Python, Spring для Java) та легкості інтеграції з іншими технологіями. Для Frontend — JavaScript із використанням фреймворків React або Angular для створення зручного інтерфейсу користувача. Бази даних — MongoDB для неструктурованих даних [17].

Критеріями успішного завершення проєкту будемо вважати:

– виконання всіх вимог технічного завдання без конфліктів із попереднім функціоналом;

– успішне завершення тестування, включаючи функціональні, регресійні, інтеграційні та навантажувальні тести;

– успішне розгортання оновленої версії платформи в продакшн-середовищі;

– позитивний відгук користувачів щодо нових функцій та оновленого інтерфейсу;

– повна відповідність нової версії платформі нормативним вимогам GDPR.

Після завершення проєкту слід провести аудит усіх етапів його реалізації за такими критеріями: відповідність функціоналу платформи вимогам, визначеним у технічному завданні; аналіз ефективності нових рішень (наприклад, зменшення часу обробки запитів); відсутність багів або інших критичних дефектів під час експлуатації; результати тестування підтверджують стабільність та інтегрованість оновлень; відсутність претензій з боку

регуляторних органів щодо відповідності платформи нормативним вимогам; позитивні відгуки від користувачів або замовників проєкту [18].

Таким чином, шаблон, критерії успіху та методи оцінки дозволяють стандартизувати процес розробки релізів і забезпечувати їх високу якість навіть у складних динамічних середовищах.

2.2 Розробка моделі проєкту в MS Project

Змістом ІТ-проєкту створення релізу платформи із захисту персональних даних можна вважати сукупність всіх робіт, які необхідно виконати для досягнення мети проєкту та розробки продукту, включаючи реалізацію нового функціоналу, вдосконалення існуючих компонентів і забезпечення відповідності оновленим вимогам. Зміст охоплює технічне завдання, збір вимог, кодування, тестування, інтеграцію та розгортання платформи, а також забезпечення її функціональної та нормативної відповідності. Він визначає межі проєкту, допомагаючи уникнути виконання зайвих робіт або пропуску критично важливих етапів [15].

Створення «шаблону змісту» для повторювальних релізів забезпечує систематичний підхід до виконання робіт і дозволяє скоротити час наступного планування. Такий шаблон включає основні етапи та задачі, які є типовими для кожного релізу: збір та аналіз вимог, перевірка сумісності нових функцій із поточним функціоналом, розробка та тестування програмного коду, інтеграція оновлень і випуск релізу. Шаблон може також передбачати стандартні підходи до перевірки якості, управління ризиками та оцінки продуктивності.

Унікальність проєкту, яка є його головною ознакою, не виключає можливості використання шаблону змісту. Шаблон слугує базовою структурою, яку можна адаптувати відповідно до специфічних вимог кожного релізу. Наприклад, якщо в рамках конкретного релізу необхідно впровадити нові функції захисту або адаптувати платформу до оновлених регуляторних вимог, ці унікальні аспекти додаються до стандартного змісту, створюючи збалансовану

комбінацію типових і специфічних завдань.

Для формування змісту релізу на основі зібраних вимог необхідно спочатку провести їх аналіз, класифікуючи за категоріями, такими як функціональні, технічні, регуляторні чи користувацькі. Потім створюється технічне завдання, яке визначає кінцевий функціонал, що має бути реалізований у межах релізу. Це завдання розбивається на окремі компоненти, що стають основою для створення WBS. Зміст релізу буде чітко задокументований, щоб уникнути непорозумінь між учасниками проєкту та забезпечити відповідність очікуванням усіх стейкхолдерів [17].

Результатом управління змістом є створення ієрархічної структури робіт, яка відображає всі основні етапи проєкту.

До укрупненої WBS проєкту створення релізу платформи додано зазначені вище фази:

1. Збір і аналіз вимог,
2. Перевірка сумісності функцій,
3. Розробка нового коду,
4. Тестування та інтеграція,
5. Випуск релізу.

Кожна з цих фаз ділиться на окремі задачі, які відповідають основним роботам, визначеним у змісті проєкту. Такий підхід забезпечує прозорість виконання робіт і дає змогу контролювати виконання завдань на всіх етапах проєкту.

Для шаблону, кожну фазу розбито на 3 роботи, але для кожного конкретного релізу ця кількість може змінюватися в залежності від завдань та обсягу робіт.

Шаблон WBS із деталізацією робіт для створення релізу подано в таблиці 2.1.

Таблиця 2.1

WBS проєкту створення релізу платформи із захисту персональних даних

№	Проєкт створення релізу платформи із захисту персональних даних
1	**1. Збір і аналіз вимог**
2	Виявлення вимог з регуляторних джерел та від стейкхолдерів.
3	Аналіз і класифікація зібраних вимог (функціональні, технічні, регуляторні).
4	Формування технічного завдання для релізу.
6	**2. Перевірка сумісності функцій**
7	Аналіз можливих конфліктів нових вимог із поточним функціоналом.
8	Розробка змін до архітектури платформи для інтеграції нових функцій.
9	Створення документації щодо взаємодії між новими і наявними модулями.
11	**3. Розробка нового коду**
12	Розробка програмного коду відповідно до технічного завдання.
13	Написання автотестів для основних функціональних компонентів.
14	Проведення внутрішньої перевірки якості коду (code review).
16	**4. Тестування та інтеграція**
17	Функціональне тестування нових компонентів системи.
18	Інтеграційне тестування для перевірки сумісності з існуючими модулями.
19	Створення та виконання сценаріїв регресійного тестування.
21	**5. Випуск релізу**
22	Розгортання оновленої платформи у тестовому середовищі.
23	Підготовка релізних нотаток та документації для користувачів.
24	Перехід платформи до продакшн-середовища та перевірка її стабільності.

Для визначення термінів планування та реалізації проєкту було визначено функціональний склад команди.

Формування та управління командою проєкту створення релізу платформи із захисту персональних даних є ключовим для успішного досягнення цілей проєкту. Команда складається з фахівців із різних сфер, кожен із яких виконує

унікальну роль, необхідну для реалізації проєкту. До команди проєкту уключено наступні ролі: маркетолог, проектний менеджер, розробник, тестувальник, юрист, аналітик.

Маркетолог відповідає за аналіз запитів користувачів, прогнозування ринкових потреб і забезпечення релевантності функціональних оновлень для кінцевих користувачів.

Проектний менеджер координує дії команди, забезпечує дотримання строків, вирішення конфліктів і контроль за виконанням задач відповідно до технічного завдання.

Розробник займається створенням програмного коду, інтеграцією нового функціоналу та забезпеченням технічної сумісності оновлень із поточними компонентами платформи.

Тестувальник виконує функціональне, інтеграційне та регресійне тестування, щоб гарантувати стабільність і якість релізу.

Юрист забезпечує відповідність платформи нормативним вимогам, включаючи оновлення відповідно до GDPR, а також оцінює ризики, пов’язані з юридичними аспектами.

Аналітик збирає вимоги з різних джерел, формує технічне завдання, аналізує ризики та оцінює вплив нових функцій на платформу.

Управління командою зосереджується на ефективній комунікації, визначенні чітких ролей і завдань, а також моніторингу прогресу [18].

Проектний менеджер створює плани, використовує для управління завданнями MS Project та забезпечує регулярну взаємодію між членами команди через щотижневі зустрічі або спринт-планування. Це дозволяє команді злагоджено працювати навіть у динамічному середовищі розробки релізу платформи.

Після визначення ролей та функціонального навантаження членів команди було розроблено план виконання проєктних робіт [19], який подано в таблиці 2.2.

Таблиця 2.2

План виконання проєктних робіт

№	Робота проєкту	Дні	Початок	Завершення
	Проєкт створення релізу платформи із захисту персональних даних	**48**	**03.12.2024**	**07.02.2025**
1	**Збір і аналіз вимог**	**9**	**03.12.2024**	**16.12.2024**
2	Виявлення вимог з регуляторних джерел та від стейкхолдерів.	4	03.12.2024	09.12.2024
3	Аналіз і класифікація зібраних вимог (функціональні, технічні, регуляторні).	2	09.12.2024	11.12.2024
4	Формування технічного завдання для релізу.	3	11.12.2024	16.12.2024
5	**Перевірка сумісності функцій**	**8**	**16.12.2024**	**26.12.2024**
6	Аналіз можливих конфліктів нових вимог із поточним функціоналом.	2	16.12.2024	18.12.2024
7	Розробка змін до архітектури платформи для інтеграції нових функцій.	4	18.12.2024	24.12.2024
8	Створення документації щодо взаємодії між новими і наявними модулями.	2	24.12.2024	26.12.2024
9	**Розробка нового коду**	**15**	**26.12.2024**	**16.01.2025**
10	Розробка програмного коду відповідно до технічного завдання.	10	26.12.2024	09.01.2025
11	Написання автотестів для основних функціональних компонентів.	5	09.01.2025	16.01.2025
12	Проведення внутрішньої перевірки якості коду (code review).	6	08.01.2025	16.01.2025
13	**Тестування та інтеграція**	**8**	**16.01.2025**	**28.01.2025**
14	Функціональне тестування нових	3	16.01.2025	21.01.2025

	компонентів системи.			
15	Інтеграційне тестування для перевірки сумісності з існуючими модулями.	2	21.01.2025	23.01.2025
16	Створення та виконання сценаріїв регресійного тестування.	3	23.01.2025	28.01.2025
17	**Випуск релізу**	**8**	**28.01.2025**	**07.02.2025**
18	Розгортання оновленої проект платформи у тестовому середовищі.	3	28.01.2025	31.01.2025
19	Підготовка релізних нотаток та документації для користувачів.	5	28.01.2025	04.02.2025
20	Перехід платформи до продакшн-середовища та перевірка її стабільності.	3	04.02.2025	07.02.2025

На основі плану було створено графік Ганта, який подано на рисунку 2.1.

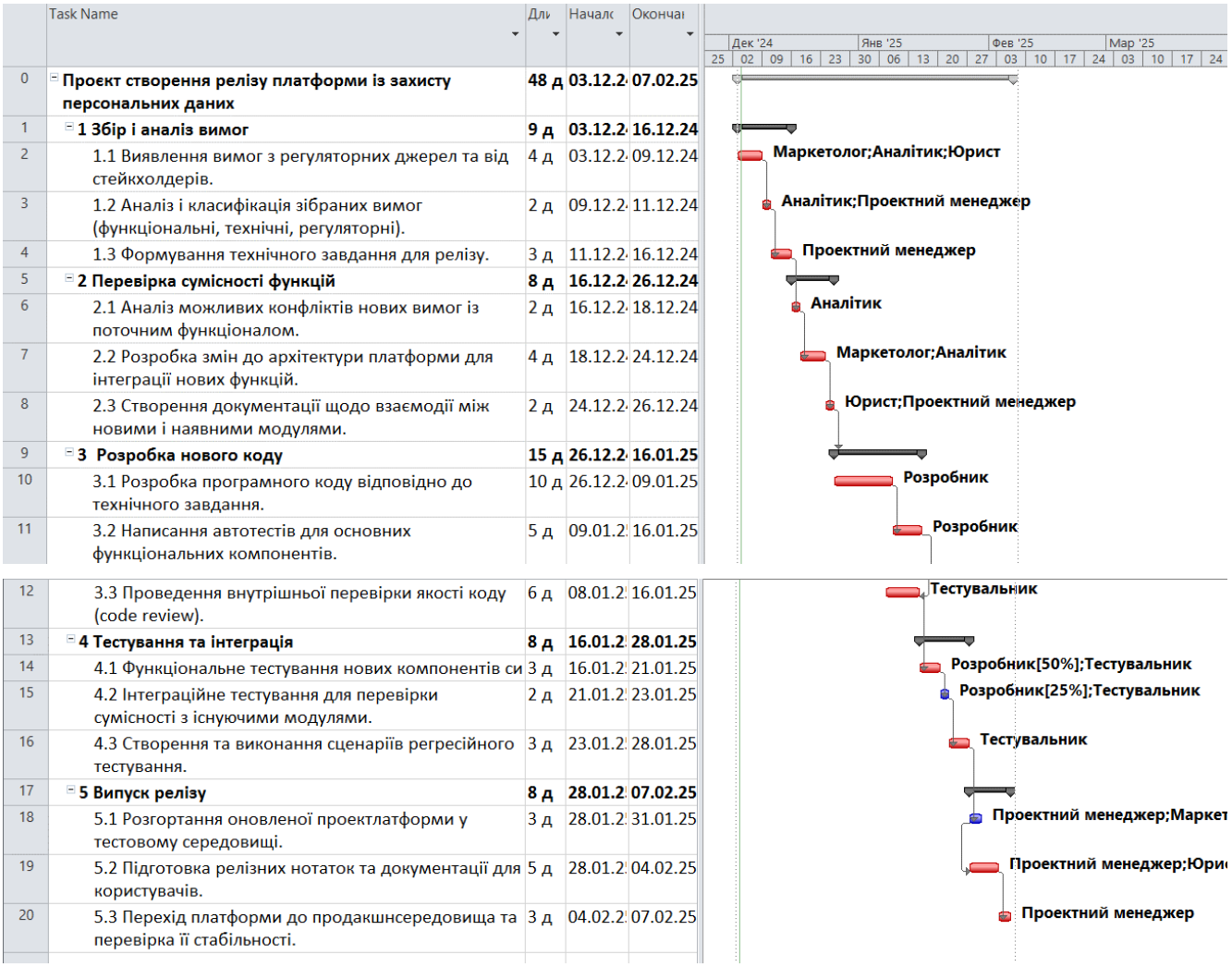
Після визначення послідовності робіт проєкту та ресурсів, що їх виконують була отримана тривалість релізу, яка склала приблизно 2 календарних місяці.

Розрахунок завантаження членів команди проєкту створення релізу платформи є важливим інструментом для забезпечення ефективного виконання завдань. Після призначення виконавців на кожну роботу аналізується обсяг робочого часу, який необхідний для виконання задач, у порівнянні з доступним ресурсом. Це дозволяє уникнути перевантаження або недовантаження окремих членів команди, що впливає на продуктивність та строки виконання проєкту.

Існує декілька методів визначення завантаженості ресурсів [18]. Метод лінійного планування передбачає створення графіку робіт із чіткими часовими рамками, метод аналізу поточного завантаження оцінює обсяг вже запланованих задач, а прогнозний підхід використовує історичні дані для передбачення можливих обсягів робіт.

Розрахунок завантаження ресурсів дозволяє уникнути ризиків, пов’язаних

із затримками або неефективним використанням часу, сприяє рівномірному розподілу задач у команді та створює умови для вчасного виконання всіх етапів проєкту.



	Task Name	Дл	Начал	Окончан
0	**Проєкт створення релізу платформи із захисту персональних даних**	**48 д**	**03.12.2**	**07.02.25**
1	**1 Збір і аналіз вимог**	**9 д**	**03.12.2**	**16.12.24**
2	1.1 Виявлення вимог з регуляторних джерел та від стейкхолдерів.	4 д	03.12.2	09.12.24
3	1.2 Аналіз і класифікація зібраних вимог (функціональні, технічні, регуляторні).	2 д	09.12.2	11.12.24
4	1.3 Формування технічного завдання для релізу.	3 д	11.12.2	16.12.24
5	**2 Перевірка сумісності функцій**	**8 д**	**16.12.2**	**26.12.24**
6	2.1 Аналіз можливих конфліктів нових вимог із поточним функціоналом.	2 д	16.12.2	18.12.24
7	2.2 Розробка змін до архітектури платформи для інтеграції нових функцій.	4 д	18.12.2	24.12.24
8	2.3 Створення документації щодо взаємодії між новими і наявними модулями.	2 д	24.12.2	26.12.24
9	**3 Розробка нового коду**	**15 д**	**26.12.2**	**16.01.25**
10	3.1 Розробка програмного коду відповідно до технічного завдання.	10 д	26.12.2	09.01.25
11	3.2 Написання автотестів для основних функціональних компонентів.	5 д	09.01.2	16.01.25
12	3.3 Проведення внутрішньої перевірки якості коду (code review).	6 д	08.01.2	16.01.25
13	**4 Тестування та інтеграція**	**8 д**	**16.01.2**	**28.01.25**
14	4.1 Функціональне тестування нових компонентів си	3 д	16.01.2	21.01.25
15	4.2 Інтеграційне тестування для перевірки сумісності з існуючими модулями.	2 д	21.01.2	23.01.25
16	4.3 Створення та виконання сценаріїв регресійного тестування.	3 д	23.01.2	28.01.25
17	**5 Випуск релізу**	**8 д**	**28.01.2**	**07.02.25**
18	5.1 Розгортання оновленої проектлатформи у тестовому середовищі.	3 д	28.01.2	31.01.25
19	5.2 Підготовка релізних нотаток та документації для користувачів.	5 д	28.01.2	04.02.25
20	5.3 Перехід платформи до продакшнсередовища та перевірка її стабільності.	3 д	04.02.2	07.02.25

Рисунок 2.1 – Графік Ганта проєкту

Розподіл ролей було виконано в MS Project, результат подано на рисунку 2.2 у вигляді ресурсного профілю проєкту. По осі OY зображено час роботи відповідного ресурсу у роботах проєкту, по осі OX представлено життєвий цикл проєкту (ЖЦП) з інтервалом в декаду (10 днів). Кожен ресурс марковано відповідним кольором, що відображено на рисунку 2.2.

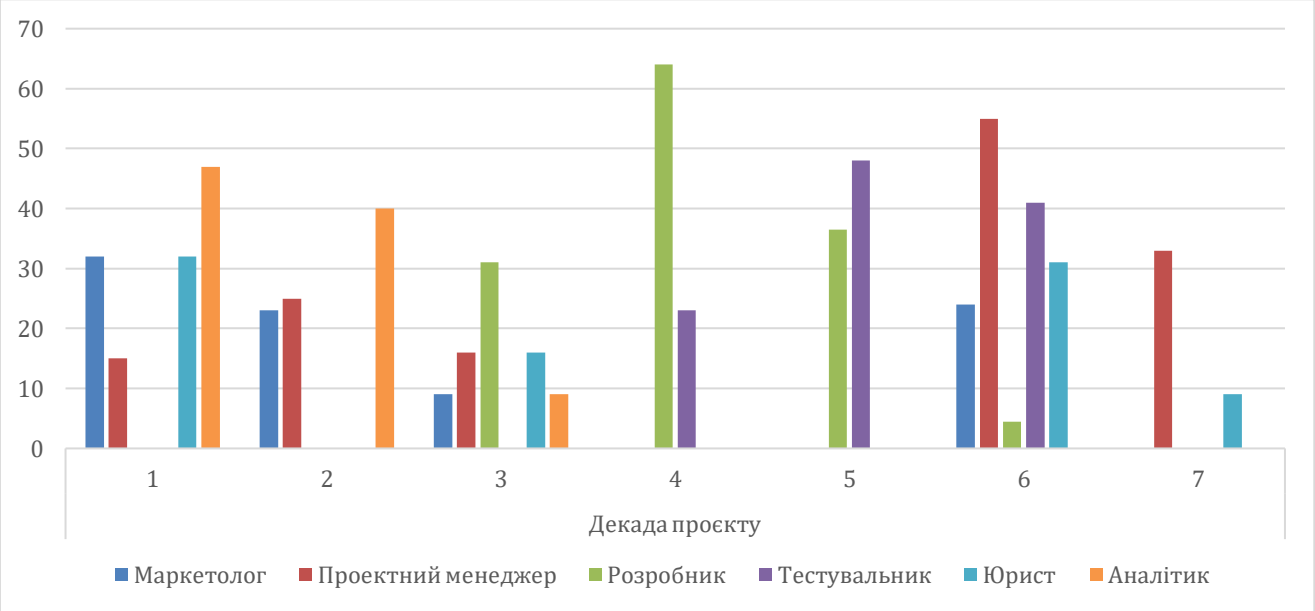


Рисунок 2.2 – Ресурсний профіль проєкту

За результатами аналізу графіка можна стверджувати, що усі ресурси працюють рівномірно, в межах своїх компетенцій та проєктних робіт. Аналітик задіяний в першу половину ЖЦП, для аналізу поточної інформації з метою формування ТЗ. Тестувальник навпаки, працює у другу половину проєкту, розробник коду програмного забезпечення — у середині проєкту, від формування ТЗ до передачі коду на тестування. Маркетолог, проєктний менеджер та юрист задіяні впродовж всього ЖЦП.

Наступним кроком було створено бюджет проєкту. Управління вартістю проєкту створення релізу платформи після призначення виконавців кожній роботі та визначення їх оплати праці є необхідним для забезпечення фінансової прозорості та контролю витрат.

Бюджет включає оплату праці членів команди, витрати на інфраструктуру, ліцензії програмного забезпечення, тестування та можливі непередбачені витрати. Визначення оплати праці залежить від тривалості виконання задач, складності робіт та погодинної ставки членів команди [20].

Для розрахунку бюджету застосовуються різні методи [21]. Метод за оцінкою витрат базується на прогнозах витрат для кожного етапу проєкту. Метод

на основі аналогів використовує дані з попередніх проєктів із подібними характеристиками. Метод експертної оцінки залучає фахівців для аналізу та визначення реалістичного бюджету.

В даному випадку був застосований метод аналогій, який спирався на дані з проєкту створення самої платформи.

Розрахунок бюджету необхідний для запобігання перевитратам, узгодження ресурсів із фінансовими можливостями та забезпечення своєчасного фінансування кожного етапу проєкту. Це сприяє успішній реалізації проєкту, оскільки дозволяє планувати та контролювати витрати відповідно до затвердженого бюджету.

Бюджет розраховано в середовищі MS Project, результати представлено в таблиці 2.3.

Таблиця 2.3

Бюджет проєкту створення релізу

№	Робота проєкту	Витрати, грн.
	Проєкт створення релізу платформи із захисту персональних даних	**146 600**
1	**Збір і аналіз вимог**	**38 760**
2	Виявлення вимог з регуляторних джерел та від стейкхолдерів.	21 560
3	Аналіз і класифікація зібраних вимог (функціональні, технічні, регуляторні).	9 400
4	Формування технічного завдання для релізу.	7 800
5	**Перевірка сумісності функцій**	**27 760**
6	Аналіз можливих конфліктів нових вимог із поточним функціоналом.	3 200
7	Розробка змін до архітектури платформи для інтеграції нових функцій.	12 160
8	Створення документації щодо взаємодії між новими і наявними модулями.	12 400
9	**Розробка нового коду**	**28 080**
10	Розробка програмного коду відповідно до технічного завдання.	15 200

11	Написання автотестів для основних функціональних компонентів.	7 600
12	Проведення внутрішньої перевірки якості коду (code review).	5 280
13	**Тестування та інтеграція**	**10 080**
14	Функціональне тестування нових компонентів системи.	4 920
15	Інтеграційне тестування для перевірки сумісності з існуючими модулями.	2 520
16	Створення та виконання сценаріїв регресійного тестування.	2 640
17	**Випуск релізу**	**41 920**
18	Розгортання оновленої проект платформи у тестовому середовищі.	12 120
19	Підготовка релізних нотаток та документації для користувачів.	22 000
20	Перехід платформи до продакшн-середовища та перевірка її стабільності.	7 800

Орієнтовна вартість випуску релізу впродовж 2 місяців склала 500 тис. грн.

2.3 Зовнішнє середовище ІТ-проєкту

Зовнішнє оточення ІТ-проєкту — це сукупність факторів, які знаходяться за межами самого проєкту, але впливають на його хід, успіх чи ефективність. До нього належать економічні, правові, технічні, соціальні та ринкові умови, які можуть як створювати можливості, так і становити ризики для проєкту [21].

Зовнішнє середовище може бути макро- або мікрорівневим. Макрорівень включає глобальні фактори, такі як економічна ситуація, законодавство та технологічні тренди. Мікрорівень охоплює взаємодії з клієнтами, партнерами, постачальниками та конкурентами. Дослідження зовнішнього середовища необхідне для визначення потенційних ризиків, таких як зміни регуляторних вимог або поява нових технологій, і пошуку можливостей, таких як нові ринки чи партнери. Отримані результати використовуються для коригування плану проєкту, щоб мінімізувати загрози й ефективно використати наявні ресурси.

Зовнішнім середовищем можна частково управляти через активну

взаємодію з регуляторами, партнерами та клієнтами, а також через адаптацію стратегії проєкту до змін. Для нашого проєкту зовнішнім середовищем є регуляторні органи ЄС, що встановлюють вимоги GDPR, технологічні тенденції у сфері кібербезпеки, конкуренти на ринку платформ захисту даних, а також потреби користувачів. Їхнє дослідження дозволить своєчасно адаптувати платформу до змін і забезпечити її конкурентоспроможність.

Ризики IT-проєкту можна визначити шляхом аналізу внутрішніх і зовнішніх факторів, які можуть вплинути на виконання задач проєкту [22].

Для цього використаєм інструменти аналізу зацікавлених сторін, оцінку потенційних загроз, попередній досвід команди та аналіз сценаріїв. Ризики поділяються на внутрішні, що залежать від ресурсів і процесів у команді, та зовнішні, такі як регуляторні зміни, технічні нововведення чи економічні обставини. Якщо не ідентифікувати всі наявні ризики, це може призвести до значних затримок, перевищення бюджету чи невідповідності результатів очікуванням стейкхолдерів.

Ризик IT-проєкту — це випадкова подія або умова, яка може негативно або позитивно вплинути на успіх проєкту, якщо вона відбудеться. Згідно з [23], ризики в IT включають технічні збої, невідповідність вимогам чи нестачу ресурсів. На відміну від ризиків у будівництві, які часто є фізичними та передбачуваними (наприклад, погодні умови чи матеріальні затримки), ризики в IT пов’язані з абстрактними аспектами, такими як складність технологій, швидкість змін або залежність від сторонніх постачальників.

За управління ризиками у нашому проєкті відповідатиме проектний менеджер, оскільки він контролює загальний хід проєкту, комунікацію між членами команди та координацію задач. Його основними обов’язками є ідентифікація ризиків на ранніх етапах, розробка плану реагування та моніторинг реалізації відповідних заходів. Це корелює з його функцією забезпечення дотримання строків і бюджету, адже правильне управління ризиками знижує ймовірність непередбачуваних проблем.

Ідентифікація ризиків є процесом визначення та документування можливих загроз, які можуть вплинути на виконання проєкту. Для нашого проєкту були визначені наступні 10 основних ризиків, які можуть вплинути на процес розробки релізу платформи. Результат подано у таблиці 2.4.

Таблиця 2.4

Аналіз ризиків проєкту створення релізу

Ризики	Імовірність (1-10)	Наслідки (1-10)	Рейтинг ризику (імовірність * наслідки)	Наслідки
Невідповідність вимог	9	9	81	Затримки в реалізації оновлень
Технічні збої	7	8	56	Збої в роботі платформи
Зміни в законодавстві	9	7	63	Штрафи та репутаційні втрати
Затримки у тестуванні	5	6	30	Продовження строків проєкту
Недостатнє фінансування	7	9	63	Неможливість завершення релізу
Конфлікти в команді	4	5	20	Низька продуктивність команди
Нестача технічних ресурсів	5	7	35	Неможливість реалізації функцій
Неправильна інтеграція	6	8	48	Проблеми з інтеграцією оновлень
Затримки у зборі вимог	8	9	72	Затримки в розробці
Недоліки безпеки	6	9	54	Ризики витоків даних

Для зниження наслідків і мінімізації ризиків у проєкті створення релізу

платформи із захисту персональних даних необхідно впровадити кілька заходів, зокрема для найбільш значущих ризиків, таких як невідповідність вимог, зміни в законодавстві та затримки у зборі вимог. Ці ризики мають найбільшу імовірність та значний вплив на проєкт, оскільки стосуються ключових етапів формування змісту релізу. Важливо зазначити, що ці ризики значною мірою зумовлені зовнішніми факторами, на які команда проєкту має обмежений вплив. Тому необхідно розробити стратегічний підхід до їхньої мінімізації.

Для ризику «зміни в законодавстві» доцільно створити інтелектуальну систему, яка буде автоматично відстежувати оновлення у нормативній базі та законодавчих актах. Така система повинна інтегрувати інформацію з офіційних джерел регуляторів, що дозволить оперативно отримувати зміни та адаптувати вимоги до них. Це значно скоротить час на збір інформації та знизить ризик затримок у зборі вимог. Для мінімізації ризику «невідповідність вимог» інтелектуальна система також може автоматизувати перевірку вимог на предмет їхньої узгодженості та відповідності як законодавству, так і бізнес-потребам.

Щодо ризику «затримки у зборі вимог», впровадження такої системи допоможе оптимізувати процес збору, оскільки інформація про нові вимоги буде надходити автоматично. Крім того, можна запланувати регулярні консультації з регуляторами та експертами, щоб завчасно отримувати інформацію про можливі зміни. Для усіх інших ризиків необхідно розробити плани реагування, наприклад, забезпечення резервів ресурсів, перевірка функціоналу на ранніх етапах тестування та впровадження стандартів для уникнення конфліктів у команді.

Таким чином, інтеграція інтелектуальної системи відстеження вимог не лише мінімізує вплив найбільш значущих ризиків, але й покращить загальну ефективність управління проєктом, забезпечуючи відповідність платформи нормативним вимогам у режимі реального часу.

Стейкхолдери IT-проєктів – це всі особи, групи або організації, які можуть впливати на проєкт або відчувати його вплив. Це можуть бути як внутрішні

учасники проєкту, так і зовнішні зацікавлені сторони, які мають свої інтереси та вимоги щодо кінцевого результату. Ефективне управління стейкхолдерами є важливою складовою успішного виконання проєкту [24].

Нашими стейкхолдерами є такі ключові групи:

- регулятори, які встановлюють вимоги щодо відповідності платформи GDPR;
- користувачі платформи, які очікують високої функціональності та захисту персональних даних;
- клієнти – організації, що впроваджують платформу;
- команда розробників, яка відповідає за технічну реалізацію;
- інвестори, які фінансують проєкт;
- консультанти з кібербезпеки, що оцінюють рівень безпеки платформи;
- партнери, які забезпечують технічну підтримку [25].

Вплив стейкхолдерів на IT-проєкт може бути як позитивним, так і негативним. Позитивний вплив проявляється через підтримку, фінансування, корисні відгуки або надання ресурсів. Негативний вплив може включати затримки через регуляторні вимоги, недостатнє фінансування, конфлікти між вимогами або недоліки у комунікації. Для визначення впливу стейкхолдерів необхідно проводити аналіз їхніх інтересів, ролей у проєкті та ступеня залученості. Наші стейкхолдери можуть впливати наступним чином: регулятори забезпечують відповідність законодавству, але можуть створювати затримки через зміни в нормативній базі; користувачі дають зворотний зв’язок, який допомагає удосконалювати платформу, однак можуть висувати суперечливі вимоги; клієнти визначають бізнес-цілі проєкту, але можуть затримувати процес через зміни пріоритетів; команда розробників є ключовим рушієм реалізації, однак можливі ризики через низьку продуктивність; консультанти допомагають у покращенні безпеки, але можуть запропонувати дорогі рішення [26]. Результат аналізу стейкхолдерів подано в таблиці 2.5.

Таблиця 2.5

Управління стейкхолдерами проєкту

Стейкхолдери	Вплив (позитив/негатив)	Як саме здійснюється вплив	Наслідки та дії
Регулятори	Негативний	Встановлюють жорсткі вимоги	Можливі затримки через адаптацію до змін
Користувачі	Позитивний	Дають зворотний зв’язок	Покращення платформи відповідно до потреб
Клієнти	Позитивний/негативний	Формують бізнес-цілі	Затримки через зміни пріоритетів
Розробники	Позитивний	Реалізують технічні рішення	Висока якість коду та функціоналу
Консультанти	Негативний	Рекомендації дорогих рішень	Підвищення витрат на розробку
Інвестори	Позитивний	Фінансують проєкт	Забезпечення ресурсів для виконання
Партнери	Позитивний	Забезпечують технічну підтримку	Стабільна інтеграція та оновлення

Висновки до розділу 2. У розділі було сформовано мету і задачі проєкту створення платформи для захисту персональних даних відповідно до GDPR, а саме до поточного релізу. Подані цілі для успішного завершення проєкту. Сформоване бачення продукту, припущення та обмеження щодо проєкту.

Розробка заходів з управління проєктом створення релізу для платформи із захисту персональних даних показала, що ефективна реалізація такого проєкту вимагає комплексного підходу до управління цілями, задачами, ресурсами та взаємодією з оточенням. Основна мета проєкту – забезпечити відповідність

платформи актуальним вимогам GDPR, підвищити її продуктивність та інтегрувати нові засоби захисту даних. Це досягається через виконання чітко визначених задач: від збору вимог і кодування до тестування, інтеграції та випуску релізу. Важливою умовою є адаптивність планування, оскільки зміни регуляторних вимог і нові загрози можуть виникати нерегулярно.

В результаті управління змістом проєкту сформовано шаблон змісту, який дозволить стандартизувати підхід до повторюваних релізів. В результаті управління строками проведено розподіл завдань на фази з конкретними дедлайнами. Всього в проекті виділено п’ять фаз. Відстеження прогресу запропоновано з використанням діаграми Ганта. Загальна тривалість випуску релізу становить орієнтовно 50 робочих днів.

Заходи з управління командою дозволило синхронізувати роботу усіх членів команди: маркетолога, проектного менеджера, розробника, тестувальника, аналітика та юриста. Важливо було забезпечити прозору комунікацію та рівномірний розподіл навантаження для усунення конфліктів та забезпечення успішного завершення проєкту. Бюджет проєкту враховує оплату праці команди, витрати на тестування, інфраструктуру та непередбачені витрати. Розрахована вартість проєкту склала орієнтовно 500 тис. грн.

Ідентифіковано зовнішнє оточення проєкту, яке уключає регуляторні органи, конкурентів та ринкові умови. Його дослідження допомогло адаптувати проєкт до змін і мінімізувати ризики. Виявлено ризики проєкту, їх ймовірність та наслідки, які можуть вплинути на проєкт. Особливу увагу було приділено ризикам, пов’язаним із вимогами: невідповідність вимог, зміни в законодавстві та затримки у зборі вимог. Вони є найбільш критичними, адже можуть порушити строки та бюджет проєкту.

Проаналізовано стейкхолдерів проєкту, зокрема регулятори, користувачі, клієнти, команда, інвестори та партнери, які здійснюють як позитивний, так і негативний вплив на проєкт. Управління їхніми інтересами та мінімізація конфліктів дозволяють зосередитись на досягненні цілей проєкту. Важливо було

виявляти очікування стейкхолдерів задля їх вираховування у плануванні. За результатами досліджень та аналізу, що проведено у другому розділі задля забезпечення успішного завершення проєкту було запропоновано створення інтелектуальної системи управління вимогами. Така система має автоматично збирати вимоги від регуляторів, користувачів та експертів з кібербезпеки, аналізувати їх на відповідність та відстежувати ризики, що пов’язані зі стейкхолдерами. Вона дозволить значно скоротити час на прийняття рішень, знизити ризики затримок і забезпечити високу якість кожного релізу платформи.

РОЗДІЛ 3 УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ ПЛАТФОРМИ ДЛЯ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ ВІДПОВІДНО ДО GDPR

3.1 Розробка моделі управління вимогами проєкту розробки платформи для захисту персональних даних

Розробка моделі інтелектуального управління вимогами для проєкту створення релізу платформи захисту персональних даних забезпечить автоматизацію процесів збору, аналізу та структурування вимог. Така модель дозволяє зменшити залежність від ручної праці, уникнути людських помилок і забезпечити своєчасність формування технічного завдання.

Інтелектуальна модель управління — це система, що використовує методи штучного інтелекту, машинного навчання та автоматизації для аналізу великих обсягів даних, прийняття рішень і адаптації до змін у середовищі проєкту [27].

Приклади таких моделей включають системи управління ризиками, які використовують алгоритми прогнозування, або автоматизовані платформи моніторингу відповідності нормативам у сфері фінансів.

Наша модель складається з п’яти основних блоків:

- збору вимог;
- аналізу та класифікації;
- перевірки на відповідність;
- пріоритизації;
- формування технічного завдання.

Кожен із цих блоків має специфічну функцію та взаємодіє з іншими для досягнення мети — створення чіткого й актуального переліку вимог для реалізації поточного релізу. Структурна схема моделі приведена на рисунку 3.1.

Блок збору вимог є початковим компонентом моделі. Його структура включає джерела даних, такі як регуляторні бази, зворотний зв’язок від користувачів та звіти про кіберзагрози. На вході до блоку надходить неструктурована інформація з різних джерел. Блок обробляє її за допомогою

парсингу, інтеграції через API та аналізу текстів із використанням алгоритмів обробки природної мови (NLP) [28].

На виході формується початковий перелік сирих вимог, готовий до подальшої обробки.

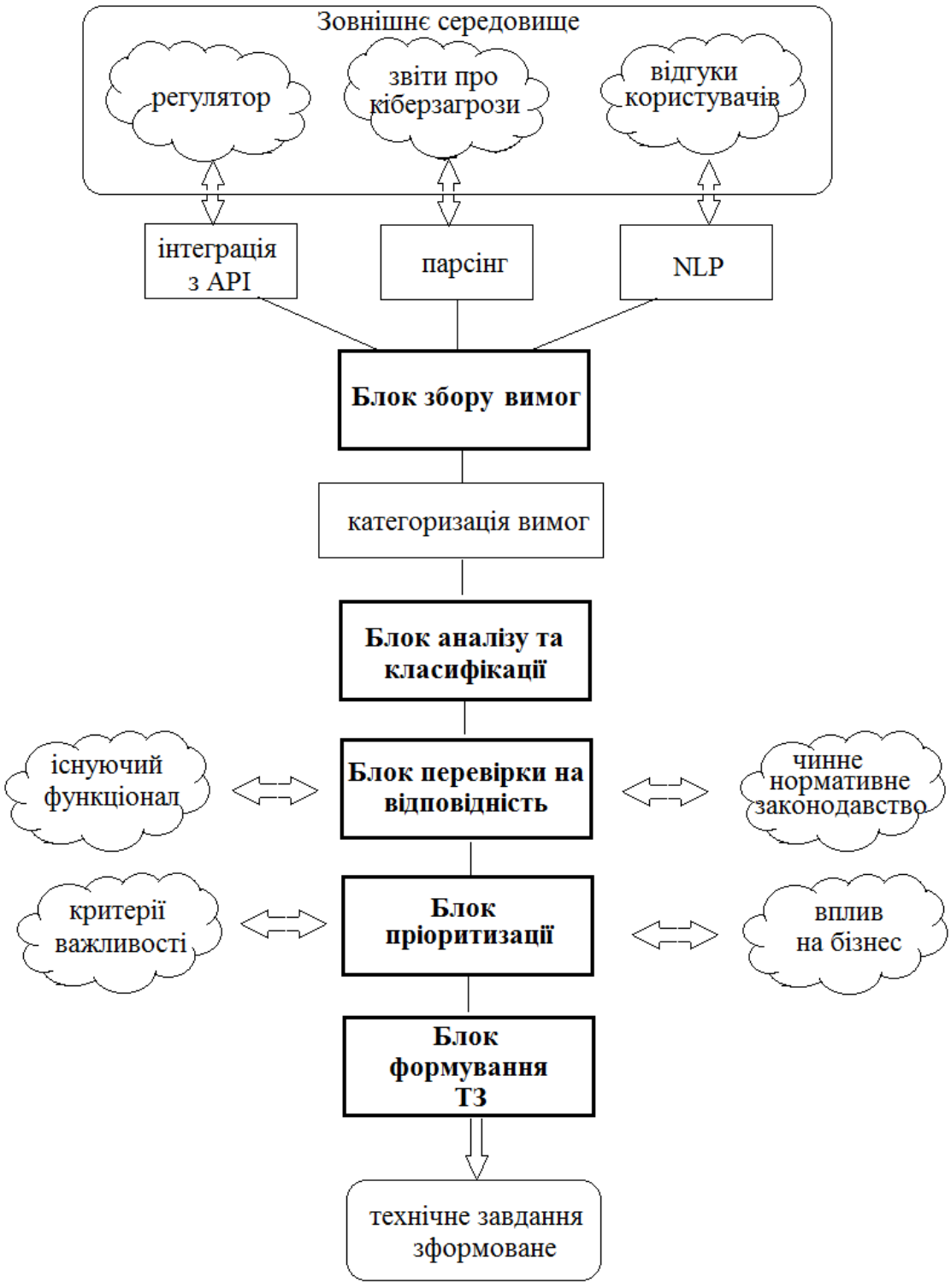


Рисунок 3.1 – Структурна схема моделі

Блок аналізу та класифікації відповідає за структурування зібраних вимог. На вході цей блок отримує список сирих вимог, який класифікується за категоріями: функціональні, технічні, регуляторні та користувацькі. Алгоритми машинного навчання забезпечують точність категоризації. На виході формується структурований перелік вимог, що дозволяє краще розуміти їхню природу та взаємозв’язок [29].

Блок перевірки на відповідність забезпечує узгодженість вимог із нормативними вимогами та технічною архітектурою платформи. Цей блок отримує на вході структуровані вимоги, які перевіряються за двома напрямами: відповідність регуляторним стандартам (наприклад, GDPR) та сумісність із поточним функціоналом платформи. Для цього використовуються порівняльний аналіз, бази нормативів і технічні специфікації. На виході залишається лише узгоджений і перевірений список вимог.

Блок пріоритизації визначає, які вимоги мають найвищу важливість для реалізації. Вхідні дані — це узгоджений список вимог. За допомогою критеріїв важливості, впливу на бізнес та складності реалізації, кожна вимога оцінюється та ранжується. У результаті формується список пріоритетних вимог, що відповідають найбільш актуальним потребам релізу.

Блок формування технічного завдання завершує роботу моделі. На вході він отримує пріоритетні вимоги та генерує детальне технічне завдання. Цей блок використовує шаблони для опису функціоналу, критеріїв прийняття та строків реалізації. На виході створюється готове технічне завдання, яке слугує основою для ініціації проєкту створення релізу.

Інтелектуальна природа моделі полягає в її здатності самостійно збирати та обробляти дані, навчатися на основі попередніх проєктів і забезпечувати точність результатів. Наприклад, завдяки інтеграції з регуляторними базами даних система автоматично реагує на зміни в нормативній базі, що мінімізує ризик затримок у зборі вимог. Крім того, модель враховує взаємодію зі стейкхолдерами, збираючи інформацію про їхні очікування та враховуючи її в

процесі формування вимог.

На виході цієї моделі формується актуальний та узгоджений перелік вимог для ініціації проєкту створення поточного релізу платформи. Цей перелік стає основою для планування, розробки, тестування та впровадження оновлень, забезпечуючи відповідність платформи регуляторним вимогам, очікуванням користувачів і потребам бізнесу. Модель не лише оптимізує процес управління вимогами, але й створює фундамент для адаптації до швидких змін у середовищі проєкту, що є критично важливим для успіху сучасних IT-проєктів.

3.2. Інформаційна система управління вимогами проєкту платформи для захисту персональних даних відповідно до GDPR

Опис параметрів системи та ключових метрик. Система управління вимогами для платформи захисту персональних даних відповідно до GDPR використовує дані, що вводяться менеджером проєкту, аналітиком та автоматично збираються з зовнішніх джерел. Такий підхід забезпечує повноту та актуальність вхідних параметрів для аналізу, що сприяє підвищенню якості управління вимогами та відповідності нормативним стандартам.

Для роботи системи потрібен збір певних параметрів, а саме:

Параметри, які вводяться менеджером проекту:

– Тип змін у системі: Вказує, які зміни були внесені до функціоналу платформи (нові функції, зміни в інтерфейсі, оновлення безпеки).

– Дата останнього оновлення: Фіксує дату останнього релізу або оновлення системи.

– Пріоритет змін: Визначає, наскільки критичними є нові вимоги (високий, середній, низький).

Параметри, які вводяться аналітиком:

– Дані про інциденти: Кількість та типи виявлених порушень безпеки або витоків даних, що потребують аналізу.

– Рівень ризику: Оцінка рівня загрози, пов’язаної з новими інцидентами (високий, середній, низький).

Параметри, які отримуються автоматично:

– Оновлення нормативних документів: Система автоматично відстежує зміни у GDPR та пов’язаних регулятивних актах.

– Звіти про інциденти: Автоматичний збір інформації про останні витоки даних, атаки або порушення безпеки з відкритих джерел (наприклад, баз даних інцидентів).

Основними метриками для оцінки ефективності управління вимогами є наступні метрики:

– Точність відповідності (Compliance Accuracy): Частка вимог платформи, що відповідають актуальним нормативним вимогам GDPR. Висока точність (понад 90%) свідчить про ефективність системи у відстеженні та впровадженні змін.

– Час реагування на зміни (Response Time): Час від моменту виявлення змін у регулятивних актах або інцидентів до моменту внесення відповідних коректив у вимоги. Чим коротший цей показник, тим швидше система адаптується до нових викликів.

– Кількість виявлених невідповідностей (Non-Compliance Count): Кількість випадків, коли платформа не відповідає актуальним вимогам. Зниження цього показника демонструє покращення управління вимогами.

– Рівень ризику порушень безпеки (Security Risk Level): Оцінка загального рівня ризику системи після внесення змін. Зниження рівня ризику свідчить про підвищення захищеності платформи.

– Частота оновлень вимог (Update Frequency): Кількість оновлень вимог за певний період часу. Показник допомагає оцінити, наскільки система оперативно адаптується до змін у середовищі.

Ці параметри забезпечують системі необхідну інформаційну базу для аналізу та адаптації вимог у реальному часі. Ключові метрики дозволяють

оцінити ефективність роботи системи та її здатність до своєчасного реагування на зміни у регуляторних вимогах і загрозах безпеці.

Визначення функціональних та нефункціональних вимог. У процесі розробки інтелектуальної системи управління вимогами для платформи захисту персональних даних важливо чітко визначити функціональні та нефункціональні вимоги до її роботи. Система повинна забезпечувати оперативне відстеження змін у регуляторних вимогах, аналіз безпекових інцидентів та автоматичне формування нових вимог для платформи.

На початковому етапі розробки було сформульовано набір основних функціональних вимог (FR), які визначають необхідні можливості системи для ефективного управління вимогами:

1. Введення інформації про зміни у регуляціях.

FR 1.1 Функція введення змін у нормативних актах:

FR 1.1.1 Система повинна дозволяти менеджеру проєкту вводити дані про зміни у нормативних актах, що стосуються захисту персональних даних.

FR 1.1.2 Введені дані повинні бути збережені у базі даних для подальшого аналізу та оновлення вимог системи.

2. Аналіз інцидентів безпеки

FR 2.1 Функція отримання інформації про інциденти:

FR 2.1.1 Система повинна аналізувати отриману інформацію і виявляти інциденти, що можуть вплинути на відповідність системи GDPR.

FR 2.1.2 Інформація про інциденти повинна бути збережена для подальшого аналізу та формування нових вимог.

3. Формування нових вимог на основі аналізу зовнішнього середовища

FR 3.1 Функція формування нових вимог:

FR 3.1.1 Система повинна надавати автоматичні рекомендації щодо нових або оновлених вимог до функціоналу на основі змін у нормативних актах та аналізі інцидентів.

FR 3.1.2 Рекомендації повинні бути доступні для перегляду менеджеру проєкту та аналітику через інтерфейс системи.

4. Аналіз відповідності вимогам GDPR

FR 4.1 Функція аналізу відповідності вимогам:

FR 4.1.1 Система повинна надавати аналітичні дані про відповідність поточних вимог новим нормативним вимогам GDPR.

FR 4.1.2 Система повинна оцінювати ризики невідповідності та рекомендувати зміни в функціоналі для забезпечення відповідності.

5. Оновлення вимог і функціоналу платформи

FR 5.1 Функція оновлення вимог:

FR 5.1.1 Система повинна автоматично вносити зміни в вимоги та функціонал платформи відповідно до нових нормативних вимог та інцидентів безпеки.

FR 5.1.2 Зміни повинні бути реалізовані без необхідності вручну оновлювати кожен компонент системи.

6. Автоматичне отримання даних про інциденти

FR 6.1 Функція інтеграції з зовнішнім джерелом даних про інциденти:

FR 6.1.1 Система повинна автоматично отримувати інформацію про витоки даних або інші інциденти безпеки з відкритих джерел, таких як API для виявлення витоків даних (Have I Been Pwned API).

FR 6.1.2 Отримана інформація повинна бути збережена та використана для подальшого аналізу і адаптації вимог.

Крім цього, було встановлено перелік критично важливих нефункціональних вимог (NFR), що забезпечують продуктивність, безпеку та зручність використання системи:

NFR 1. Система повинна бути доступною 99% часу, забезпечуючи високу надійність та відмовостійкість при високому навантаженні.

NFR 2. Усі дані, що вводяться або обробляються системою, повинні бути зашифровані з використанням сучасних методів криптографії.

NFR 3. Доступ до чутливої інформації має бути обмежений ролями користувачів, з можливістю керування правами доступу.

NFR 4. Час реакції на запит щодо зміни нормативних вимог або інцидентів не повинен перевищувати 1 хвилини при обробці даних до 1000 записів.

NFR 5. Система повинна підтримувати можливість масштабування для роботи з великою кількістю проектів (до 10,000 записів).

NFR 6. Інтерфейс системи повинен бути інтуїтивно зрозумілим, з можливістю адаптації до різних пристроїв (комп'ютери, планшети).

NFR 7. Інтерфейс повинен підтримувати зручний доступ до основних функцій через просту навігацію.

NFR 8. Система повинна вести детальний журнал всіх операцій, що включає введення даних, зміни вимог та інциденти, для подальшого аудиту.

Визначені функціональні та нефункціональні вимоги забезпечують здатність системи ефективно виконувати ключові завдання управління вимогами. Це дозволяє адаптувати платформу до змін у регуляторних нормах та мінімізувати ризики невідповідності. Описані вимоги також слугують основою для моделювання прецедентів та побудови діаграм взаємодії акторів із системою.

Моделювання прецедентів. Діаграма прецедентів відображає взаємодію акторів із функціоналом системи та описує основні сценарії її використання. Ця діаграма допомагає візуалізувати зв'язки між користувачами системи та її функціональними можливостями, що сприяє кращому розумінню вимог до системи. Прецеденти відповідають основним функціональним вимогам, що визначають ключові можливості системи.

Актори системи:

Менеджер проєкту (Project Manager) – Менеджер проєкту є головним користувачем системи. Його основні функції включають введення даних про зміни у нормативних актах, перегляд рекомендацій щодо нових вимог, управління процесом оновлення функціоналу платформи на основі змін у регуляціях та інцидентах.

Аналітик (Analyst) – виконує функцію контролю якості системи. Він аналізує дані про інциденти безпеки, оцінює відповідність вимогам GDPR та визначає ризики невідповідності. Аналітик використовує отримані дані для коригування вимог до системи.

Зовнішнє джерело інформації (Have I Been Pwned API) – стороннє джерело надає актуальні дані про інциденти безпеки (наприклад, витоки даних або атаки). Через API система автоматично отримує інформацію про інциденти, які можуть впливати на відповідність вимогам GDPR.

Прецеденти:

Введення інформації про зміни у регуляціях. Система дозволяє вводити дані про зміни в нормативних актах, що стосуються захисту персональних даних. Ці дані будуть використовуватися для подальшого аналізу та оновлення вимог платформи.

Аналіз інцидентів безпеки. Система виконує аналіз інформації про інциденти, таких як витоки даних або атаки, що можуть вплинути на відповідність системи вимогам GDPR.

Формування нових вимог на основі аналізу зовнішнього середовища. Система генерує нові або оновлені вимоги до функціоналу на основі змін у нормативних актах та аналізі інцидентів, таких як витоки даних або кібератаки.

Аналіз відповідності вимогам GDPR. Система аналізує поточні вимоги на відповідність новим нормативним стандартам GDPR та оцінює ризики невідповідності.

Оновлення вимог і функціоналу платформи. Система автоматично оновлює вимоги та функціонал платформи відповідно до нових нормативних вимог та виявлених інцидентів безпеки.

Автоматичне отримання даних про інциденти. Система автоматично отримує дані з зовнішніх джерел, таких як API для витоків даних або інцидентів безпеки, що впливають на відповідність системи GDPR (рисунок 3.1).

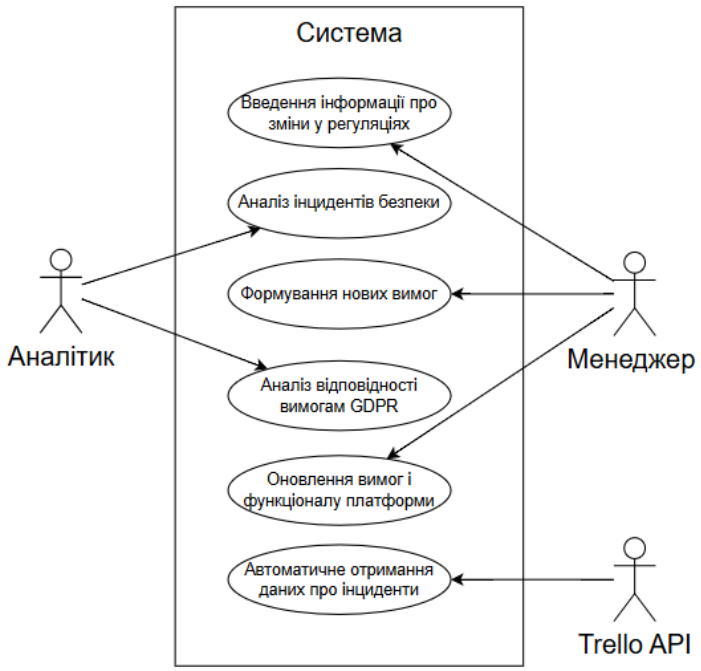


Рисунок 3.1 – Діаграма взаємодії системи з користувачами

Ця діаграма допомагає краще зрозуміти функціональні можливості системи та її взаємодію з користувачами.

Вибір інструментів для розробки. Вибір інструментів для розробки є важливим етапом у створенні інформаційної системи, оскільки саме вони визначають можливості реалізації функціональних вимог, якість кінцевого продукту та ефективність роботи команди розробників. Для проєкту з розробки системи управління вимогами відповідно до GDPR було враховано специфіку задач і вимоги до програмного забезпечення.

Серверна частина. Для реалізації серверної частини системи обрано Node.js, оскільки ця технологія забезпечує високу продуктивність і можливість обробки великої кількості одночасних з’єднань, що є критично важливим для нашої системи, оскільки вона повинна працювати з великою кількістю запитів на отримання та обробку даних у реальному часі. Node.js є ідеальним вибором для серверних додатків, які потребують швидкої обробки запитів і масштабованості.

Це також дозволяє використовувати JavaScript на сервері, що спрощує процес розробки і підтримки коду. Для роботи з HTTP-запитами використовуватиметься фреймворк Express.js, який є легким і швидким для створення веб-сервісів та API, що буде корисно для інтеграції з різними сторонніми джерелами даних, такими як Have I Been Pwned API.

API для отримання даних про інциденти. Для отримання інформації про витоки даних, кібератаки або інші інциденти безпеки буде використовуватись Have I Been Pwned API. Це один з найбільш надійних і популярних ресурсів для моніторингу витоків персональних даних, що дозволяє отримувати інформацію про витоки даних для аналізу впливу на персональні дані користувачів. API дозволяє перевіряти, чи були дані користувача зламані або викрадені в ході великих витоків, що має важливе значення для забезпечення відповідності стандартам GDPR.

Основні функції API:

– Перевірка адреси електронної пошти на наявність у базах даних витоків.

– Доступ до історії порушень для кращого аналізу інцидентів безпеки.

– Інтеграція з іншими сервісами для автоматичного отримання актуальних даних.

Клієнтська частина. Для створення адаптивного та інтерактивного інтерфейсу вибрано React.js. Це популярна бібліотека для створення інтерфейсів користувача, яка дозволяє ефективно працювати з компонентами та їх станом. React допоможе побудувати інтуїтивно зрозумілий інтерфейс для менеджера проєкту та аналітика, а також забезпечить швидку інтерактивність при роботі з даними, такими як оновлення нормативних вимог, інцидентів безпеки та формування звітів.

База даних. Для зберігання інформації про вимоги, інциденти, нормативні акти та зміни у системі використовуватиметься PostgreSQL, оскільки ця система керування базами даних (СКБД) забезпечує високу продуктивність при обробці

великих обсягів структурованих даних і підтримує складні запити. Вона ідеально підходить для збереження історичних даних, таких як інформація про попередні інциденти безпеки або зміни нормативних документів.

Безпека і авторизація. Для забезпечення безпеки доступу до функціоналу платформи буде використано JSON Web Tokens (JWT). Це стандарт для автентифікації, який дозволяє безпечно передавати дані між клієнтом і сервером. Кожен користувач (менеджер проєкту або аналітик) отримає унікальний токен для доступу до системи, що дозволить здійснювати контроль доступу та уникати несанкціонованого доступу.

Розгортання. Розгортання системи буде здійснюватися на платформі AWS (Amazon Web Services). Це одна з найбільш надійних і масштабованих хмарних платформ, яка забезпечує безпечне зберігання даних, високу доступність сервісів і простоту масштабування. AWS дозволяє легко розгортати веб-додатки, бази даних і сервіси машинного навчання, що відповідає вимогам проєкту щодо забезпечення безперервної роботи системи.

Вибір технологій обґрунтовано їх відповідністю вимогам проєкту, зокрема високою продуктивністю, масштабованістю та зручністю інтеграції з зовнішніми сервісами.

Уявлення архітектури ІС. Для ізоляції компонентів і розділення логіки основні серверні компоненти, включаючи backend і frontend для середовищ розробки та продакшену, базу даних PostgreSQL, а також інтерфейс для взаємодії з нею, було контейнеризовано за допомогою Docker Compose. Усі сервіси розділено на окремі Docker-контейнери. Docker Compose використовується для зручного керування контейнерами, їх налаштуванням та мережевим з’єднанням (рисунок 3.2).

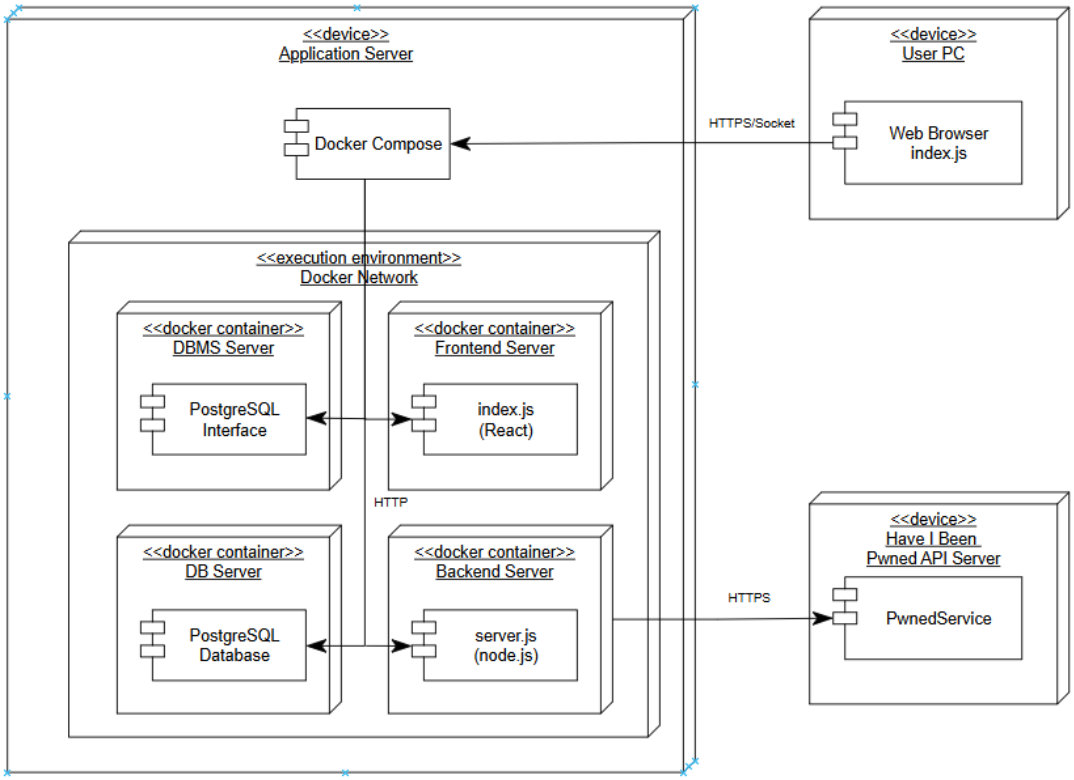


Рисунок 3.2 – Структурна схема архітектури ІС

Для ефективної взаємодії між компонентами використовуються налаштування для Docker Network, що дозволяє безпечну і швидку передачу даних між сервісами.

Для інтеграції з сервісом Have I Been Pwned створено окремий компонент — Have I Been Pwned API Server, який автоматично отримує актуальну інформацію про витоки даних і інциденти безпеки. Взаємодія з цим компонентом здійснюється через сервіс PwnedService, що використовує безпечний протокол HTTPS для передачі даних.

Ця архітектура забезпечує високу гнучкість і масштабованість системи, що дозволяє ефективно адаптувати її до змінюваних вимог і зростаючих навантажень. Використання контейнеризації за допомогою Docker Compose дозволяє ізолювати різні компоненти системи, спрощуючи їхнє масштабування та керування. Завдяки інтеграції з Have I Been Pwned API і забезпеченню

безпечного обміну даними через HTTPS, система гарантує високий рівень захисту даних. Це дозволяє не тільки зберігати актуальність і точність інформації, але й забезпечити відповідність вимогам безпеки, що є критично важливим для платформи, яка працює з персональними даними та повинна відповідати стандартам GDPR.

Зберігання інформації про інциденти безпеки, нормативні зміни, вимоги до функціоналу платформи та інші ключові дані, в системі передбачена наступна структура бази даних. Всі дані будуть зберігатися в реляційній базі даних PostgreSQL, яка надає можливість зберігати як поточні, так і історичні дані для аналізу та подальшого використання.

Таблиці бази даних (рисунок 3.3):

Projects (Проєкти). Ця таблиця зберігає загальну інформацію про проєкти, які перебувають в системі.

– ProjectID (PK): Унікальний ідентифікатор проєкту.

– Name: Назва проєкту.

– Description: Опис проєкту.

– StartDate: Дата початку проєкту.

– EndDate: Очікувана дата завершення проєкту.

Regulations (Нормативні акти). Таблиця для зберігання інформації про нормативні зміни, що стосуються захисту персональних даних, зокрема GDPR.

– RegulationID (PK): Унікальний ідентифікатор нормативного акту.

– RegulationTitle: Назва нормативного акту.

– Description: Опис нормативного акту.

– EffectiveDate: Дата вступу в силу нормативного акта.

– ProjectID (FK): Зв’язок з таблицею Projects, щоб визначити, для яких проєктів актуальні ці зміни.

Incidents (Інциденти безпеки). Таблиця для зберігання інформації про інциденти безпеки, зокрема витоки даних або кібератаки.

– IncidentID (PK): Унікальний ідентифікатор інциденту.

– IncidentType: Тип інциденту (наприклад, витік даних, кібератака).

– Description: Опис інциденту.

– IncidentDate: Дата виявлення інциденту.

– Severity: Серйозність інциденту (висока, середня, низька).

– RegulationID (FK): Зв’язок з таблицею Regulations для визначення, який нормативний акт був порушений через інцидент.

Requirements (Вимоги). Таблиця для зберігання вимог до функціоналу системи, зокрема, змін, що повинні бути внесені внаслідок нормативних змін або інцидентів.

– RequirementID (PK): Унікальний ідентифікатор вимоги.

– Description: Опис вимоги.

– Priority: Пріоритет вимоги (високий, середній, низький).

– CreatedDate: Дата створення вимоги.

– DueDate: Очікувана дата виконання вимоги.

– ProjectID (FK): Зв’язок з таблицею Projects, щоб визначити, до якого проєкту стосується ця вимога.

– RegulationID (FK): Зв’язок з таблицею Regulations для визначення, яка зміна в нормативних актах потребує цієї вимоги.

PwnedData (Дані про витоки). Таблиця для зберігання даних про витоки, що були виявлені через Have I Been Pwned API.

– PwnedID (PK): Унікальний ідентифікатор витоку.

– Email: Адреса електронної пошти, що була виявлена в результаті витоку.

– BreachDate: Дата витоку даних.

– BreachDescription: Опис витоку.

– IncidentID (FK): Зв’язок з таблицею Incidents, що визначає, який інцидент безпеки спричинив витік.

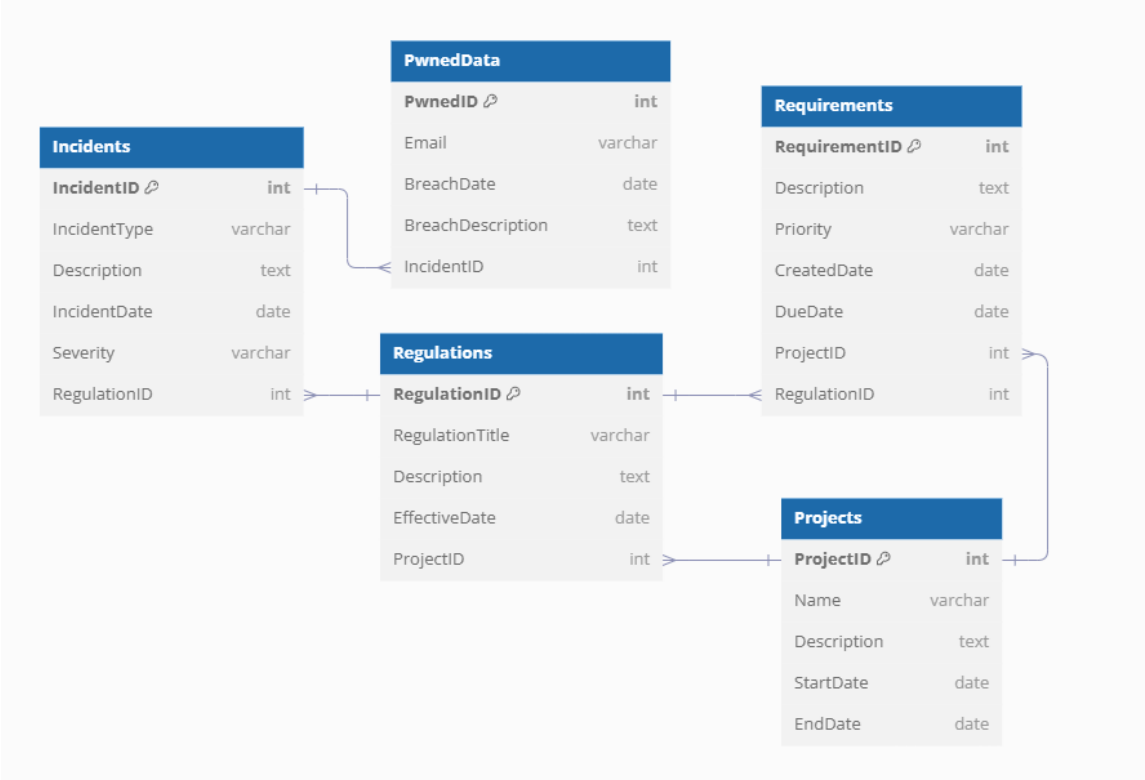


Рисунок 3.3 – Таблиці баз даних

Інтерфейс системи управління вимогами IT-проекту. Інтерфейс системи розроблений для зручного і інтуїтивно зрозумілого доступу до основних функцій. Кожна сторінка платформи має чітко визначену роль у системі та дозволяє користувачам швидко орієнтуватися, виконувати необхідні завдання і отримувати актуальну інформацію (рисунок 3.4).

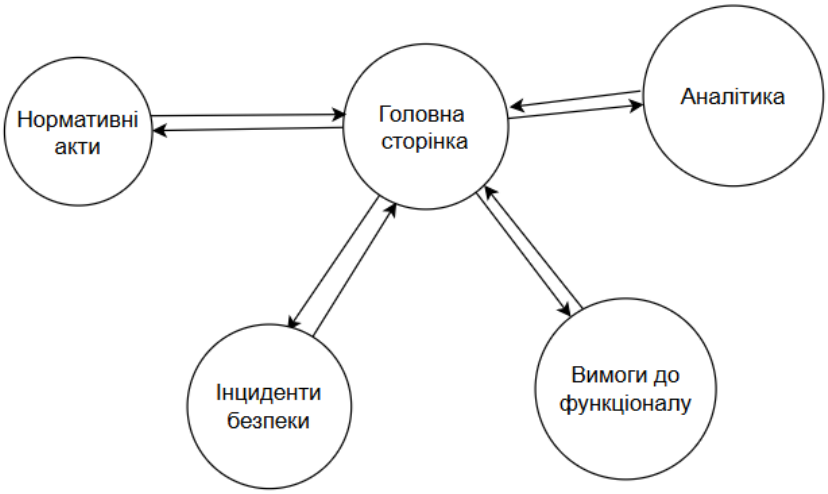


Рисунок 3.4 – Структурна схема інтерфейсу системи

Основні сторінки:

– *Головна сторінка.* Головна сторінка є вікном в проєкт і надає користувачу загальну інформацію про поточний стан. Тут можна побачити основну статистику по виконаним задачам, їхнім термінам, а також інформацію про інциденти безпеки. Це перша точка доступу до всіх функцій системи, де користувач може швидко перевірити загальний стан проєкту, його актуальні потреби та важливі оновлення.

– *Сторінка нормативних актів.* На цій сторінці користувач може переглядати існуючі зміни у нормативних актах, таких як GDPR, а також додавати нові нормативні документи, що впливають на захист персональних даних. Система автоматично оновлює вимоги та функціонал відповідно до змін у нормативному середовищі. Тут користувач може побачити деталі кожної зміни, її ефект на систему та план по виконанню.

– *Сторінка інцидентів безпеки.* Ця сторінка дозволяє користувачеві переглядати та керувати інформацією про інциденти безпеки, які вплинули або можуть вплинути на проєкт. Це включає витоки даних, кібератаки або інші подібні інциденти. Користувач може переглядати деталі кожного інциденту, його тип, серйозність та вплив на систему. Після аналізу інцидентів система може генерувати рекомендації щодо адаптації функціоналу або оновлення вимог.

– *Сторінка вимог до функціоналу.* Ця сторінка дає змогу переглядати і оновлювати вимоги, що базуються на змінах нормативних актів або інцидентах безпеки. Користувач може додавати нові вимоги, змінювати існуючі, а також встановлювати пріоритети для їх реалізації. Система автоматично оновлює ці вимоги в разі необхідності, що дозволяє завжди залишатися у відповідності до вимог GDPR та інших стандартів безпеки.

– *Сторінка аналітики.* Сторінка для перегляду аналітичних даних, що дозволяє оцінити ефективність роботи системи, точність прогнозів та відповідність нормативним вимогам. Тут користувачі можуть аналізувати

метрики, порівнювати заплановані і фактичні показники, а також отримувати зворотний зв’язок щодо ефективності змін, що були впроваджені. Ця сторінка надає візуалізацію даних, що дозволяє швидко оцінити стан проекту.

Всі сторінки взаємодіють між собою, забезпечуючи безперервну навігацію і доступ до всіх основних функцій платформи. Користувач може з будь-якої сторінки легко повернутися на головну, а також перейти до іншої сторінки для виконання конкретних завдань, таких як перегляд нормативних актів, аналіз інцидентів або управління вимогами. Ця діаграма дозволяє чітко уявити, як організована навігація між різними функціональними блоками системи, що спрощує її використання.

Аналіз ефективності ІС. Ефективність впровадження інформаційної системи управління вимогами для платформи захисту персональних даних відповідно до GDPR оцінювалась шляхом порівняння ключових метрик двох варіантів одного й того ж проєкту: версії без використання системи та версії з використанням MVP системи. Обидва варіанти виконувались в однакових умовах, що дозволяє здійснити об’єктивне порівняння ефективності роботи системи. Порівняння проводилось на основі таких параметрів, як обсяг задач, складність змін у нормативних актах, а також рівень інцидентів безпеки.

Впровадження системи дозволило автоматизувати процеси моніторингу змін нормативних актів, управління вимогами та аналізу інцидентів безпеки, що значно зменшило часові затрати на ці операції. Крім того, система надає можливість своєчасно виявляти інциденти та оновлювати вимоги відповідно до нових нормативних стандартів, що підвищує точність відповідності.

Початкові умови та методологія аналізу. Для аналізу ефективності системи були обрані п’ять ключових метрик: *точність відповідності (Compliance Accuracy)* — відсоток вимог, що відповідають актуальним нормативним актам; *час реагування на зміни (Response Time)* — час, який необхідно для внесення змін до вимог після виявлення нових нормативних актів чи інцидентів. Кількість виявлених невідповідностей (Non-Compliance Count) —

кількість вимог, що не відповідають актуальним нормативним стандартам; *рівень ризику порушень безпеки (Security Risk Level)* — оцінка ймовірності порушень безпеки в системи; *частота оновлень вимог (Update Frequency)* — кількість оновлень вимог на рік.

Аналіз проводився на основі двох варіантів проєкту:

– *Без використання системи* — у цьому варіанті всі процеси оновлення вимог, моніторингу інцидентів та зміни нормативних актів виконувались вручну.

– *Із використанням системи (MVP)* — у цьому варіанті використовувалась система, що здійснювала моніторинг змін, виявлення інцидентів та автоматичне оновлення вимог.

Результати порівняння. Без використання системи:

– Точність відповідності (Compliance Accuracy): 67%. З-за відсутності автоматизованих інструментів для оновлення вимог точність відповідності нормативним актам була на рівні 67%.

– Час реагування на зміни (Response Time): 3-5 днів. Час, необхідний для внесення змін у систему після виявлення змін у нормативних актах чи інцидентів, становив кілька днів.

– Кількість виявлених невідповідностей (Non-Compliance Count): 25%. Через людський фактор і відсутність централізованої системи для моніторингу змін значна частина вимог не відповідала актуальним нормативним вимогам.

– Рівень ризику порушень безпеки (Security Risk Level): 70%. Відсутність автоматичного моніторингу інцидентів та змін у нормативних актах призводила до високого рівня ризику безпеки.

– Частота оновлень вимог (Update Frequency): 5-7 разів на рік. Оновлення вимог відбувались вручну, що обмежувало їхню частоту.

Результати порівняння. З використанням системи (MVP):

– Точність відповідності (Compliance Accuracy): 90%. Завдяки автоматизації процесів моніторингу нормативних актів та інцидентів, точність відповідності зросла до 90%.

– Час реагування на зміни (Response Time): 1-2 години. Автоматичне отримання даних про зміни та інциденти дозволило знизити час реагування до кількох годин.

– Кількість виявлених невідповідностей (Non-Compliance Count): 5%. Автоматичне оновлення вимог і регулярний моніторинг змін зменшили кількість невідповідностей до мінімуму.

– Рівень ризику порушень безпеки (Security Risk Level): 30%. Завдяки інтеграції з Have I Been Pwned API та автоматичному моніторингу інцидентів, ризик порушень безпеки значно знизився.

– Частота оновлень вимог (Update Frequency): 15-20 разів на рік. Система автоматично здійснює оновлення вимог на основі отриманих даних про зміни та інциденти.

В таблиці 3.1 наведено порівняльний аналіз ефективності інформаційної системи без використання системи та з використанням системи (MVP).

Таблиця 3.1

Порівняльний аналіз

Метрика	Без використання системи	З використанням системи (MVP)
Точність відповідності	67%	90%
Час реагування на зміни	3-5 днів	1-2 години
Кількість виявлених невідповідностей	25%	5%
Рівень ризику порушень безпеки	70%	30%
Частота оновлень вимог	5-7 разів на рік	15-20 разів на рік

Впровадження системи також дозволило значно збільшити частоту оновлень вимог, що гарантує актуальність та відповідність платформи до змін у нормативному середовищі та інцидентам безпеки.

Результат впровадження системи є надзвичайно позитивним і доводить, що автоматизація процесів значно підвищує ефективність управління вимогами та забезпечує кращу безпеку платформи.

3.3 Аналіз ефективності управління вимогами в проекті створення платформи для захисту персональних даних відповідно до GDPR

Забезпечення відповідності Загальному регламенту захисту даних (GDPR) є важливим викликом для сучасних компаній, які працюють із персональними даними. Зростаюча кількість регуляторних норм та їх постійні зміни ускладнюють управління вимогами, створюючи ризики невідповідності, що може призвести до штрафів, репутаційних втрат та зниження ефективності бізнесу. Особливістю проєкту розробки платформи для захисту персональних даних є необхідність швидкої адаптації до нових стандартів безпеки та змін у законодавстві. Традиційні методи управління вимогами часто є недостатньо гнучкими, що спричиняє значні затримки у розробці та додаткові витрати.

Рішенням цієї проблеми є розробка інтелектуальної системи управління вимогами, яка дозволяє автоматично відстежувати зміни в регуляторних нормах, аналізувати їхній вплив на проєктну документацію та забезпечувати відповідність вимог проєкту новим стандартам. Це мінімізує ризики, скорочує час на впровадження змін та оптимізує роботу команди.

Технічна архітектура системи інтелектуального аналізу даних (ІАД). Система ІАД побудована за блочною архітектурою, що забезпечує гнучкість, масштабованість і можливість інтеграції з існуючими інструментами управління проєктами. Основними компонентами системи є:

1. ***Блок збору вимог в регуляторних нормах.*** Цей компонент інтегрується з офіційними джерелами регуляторної інформації (наприклад, порталами GDPR,

національними регуляторами) і здійснює регулярний моніторинг змін. Дані, зібрані модулем, структуровані для подальшого аналізу.

2. Блок аналізу та перевірки на відповідність. Цей компонент аналізує зміни, визначені модулем відстеження, та порівнює їх із поточними вимогами проєкту. У разі виявлення невідповідності Блок формує оновлену вимогу та інтегрує її у проєктну документацію. Він також визначає вплив змін на хід розробки.

3. Блок пріоритизації. Цей компонент забезпечує інтеграцію системи з інструментами управління проєктами, такими як Jira або Confluence. Він дозволяє автоматично оновлювати беклог, додавати нові завдання в спринти та сповіщати відповідальних осіб про зміни.

4. Блок формування ТЗ

Цей компонент генерує звіти про зміни в регуляторних нормах, їхній вплив на проєкт і статус адаптації. Сповіщення надсилаються зацікавленим сторонам для забезпечення прозорості процесу.

Взаємодія компонентів

1. Блок відстеження змін збирає дані про нові регуляторні норми.
2. Дані передаються до модуля аналізу, де порівнюються із поточними вимогами проєкту.
3. У разі виявлення невідповідностей Блок аналізу генерує оновлені вимоги, які додаються до беклогу за допомогою модуля інтеграції.
4. Кінцеві користувачі отримують звіти та сповіщення про статус адаптації через Блок звітності.

Така архітектура забезпечує ефективне управління вимогами, гнучкість у роботі з регуляторними змінами та мінімізацію ризиків, пов’язаних із невідповідністю GDPR. Система автоматизує рутинні процеси, зменшуючи витрати на впровадження змін і підвищуючи швидкість реакції на нові стандарти.

Опис алгоритму адаптації вимог. Алгоритм адаптації вимог у системі

побудований так, щоб автоматично виявляти зміни в регуляторних нормах, аналізувати їхній вплив на поточні вимоги проекту та інтегрувати оновлення у процес розробки. Алгоритм складається з кількох етапів:

5. Моніторинг змін у регуляторних нормах. Система регулярно перевіряє офіційні джерела, такі як портал GDPR, національні регулятори або сторонні бази даних змін у законодавстві.

Вхідні дані: текстові файли, JSON або XML-файли, що містять оновлення.

Методи обробки*:* парсинг текстів та вилучення релевантної інформації (нових вимог або змін у існуючих).

Технології: API для доступу до зовнішніх джерел, бібліотеки для NLP (Natural Language Processing), такі як SpaCy або NLTK.

Код для вилучення нових регуляторних норм:

```
import requests
from bs4 import BeautifulSoup
# Завантаження даних з порталу GDPR
url = «https://gdpr-info.eu»
response = requests.get(url)
soup = BeautifulSoup(response.text, ‘html.parser’)
# Витягнення тексту статей
articles = soup.find_all(‘article’)
for article in articles:
   print(article.get_text())
```

Аналіз впливу змін. Виявлені зміни порівнюються з існуючими вимогами проекту. Система аналізує, які вимоги потребують оновлення або додаткових уточнень.

- ***Порівняння:*** виконується на основі ключових слів, концептів або прямих збігів у тексті.
- ***Результат:*** список нових або змінених вимог, які потрібно адаптувати.

- **Методи**: TF-IDF для текстового порівняння, порівняння за векторними представленнями документів (Word2Vec, BERT).

Приклад алгоритму порівняння вимог:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
# Існуючі вимоги
existing_requirements = [
    «Data storage must comply with EU regulations.»,
    «Users can request data deletion at any time.»
]
# Нові регуляторні зміни
new_requirements = [
    «Data storage must specify geolocation of the server.»,
    «Users have the right to access their data within 30 days.»
]
# TF-IDF аналіз
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(existing_requirements + new_requirements)
similarity_matrix = cosine_similarity(tfidf_matrix)
print(«Схожість між вимогами:»)
print(similarity_matrix)
```

Генерація оновлених вимог. На основі результатів аналізу система формує нові вимоги та додає їх до проектної документації.

Шаблон формування: кожна вимога документується за стандартною структурою (опис, мета, джерело, пріоритет).

Пріоритезація: змінені вимоги автоматично отримують статус високого пріоритету.

Приклад формування вимоги:

```
{
  «requirement_id»: «REQ-101»,
  «description»: «Data storage must specify geolocation of the server.»,
  «source»: «GDPR Article 32»,
  «priority»: «High»,
  «status»: «Pending»
}
```

5. Інтеграція вимог у беклог. Система автоматично додає оновлені вимоги до інструментів управління проектами, таких як Jira або Trello.

Процес: створюються нові завдання з прив'язкою до відповідальних осіб.

Статус завдання: відображається в беклозі як окремий елемент із пріоритетом впровадження.

Приклад інтеграції через Jira API:

```
import requests
url = «https://your-jira-instance/rest/api/2/issue»
headers = {
   «Content-Type»: «application/json»,
   «Authorization»: «Basic <your_api_token>»
}
data = {
   «fields»: {
      «project»: {
         «key»: «GDPR»
      },
      «summary»: «Update data storage requirements»,
      «description»: «Data storage must specify geolocation of the server.»,
      «issuetype»: {
         «name»: «Task»
```

```
        },
        «priority»: {
            «name»: «High»
        }
    }
}
response = requests.post(url, json=data, headers=headers)
print(response.status_code, response.json())
```

Моніторинг виконання. Система відстежує статус виконання нових вимог, сповіщаючи відповідальних осіб про прогрес. Звіти про виконання автоматично генеруються для менеджерів проекту.

Результати роботи алгоритму. Алгоритм забезпечує автоматичне оновлення вимог відповідно до нових регуляторних змін, скорочує час на адаптацію та мінімізує ризики невідповідності. Його інтеграція в процес розробки підвищує ефективність і точність управління вимогами.

Приклад роботи системи

Сценарій:

Компанія впроваджує платформу для захисту персональних даних. Система управління вимогами отримує оновлення в регуляторних нормах GDPR, яке вимагає додаткових заходів безпеки щодо зберігання персональних даних на серверах, розташованих у межах Європейського Союзу (наприклад, геолокація серверів).

Вхідні дані:

1. Поточна вимога:
   - «Персональні дані повинні зберігатися на захищених серверах.»
2. Нова вимога:
   - «Персональні дані повинні зберігатися на серверах із вказівкою геолокації в межах ЄС.»

Процес роботи системи:

3. Моніторинг: Система виявляє нову норму GDPR через офіційний API регуляторних змін.

4. Аналіз: Використовуючи алгоритм порівняння тексту, система ідентифікує, що існуюча вимога потребує оновлення.

5. Генерація: Система формує оновлену вимогу:

6. Ідентифікатор: REQ-202.

7. Опис: «Додати вимогу до документації: персональні дані повинні зберігатися на серверах із зазначенням геолокації в межах ЄС.»

8. Інтеграція:

9. Система додає нове завдання у беклог розробки через Jira API.

10. Завдання автоматично позначається як високопріоритетне з терміном виконання в 10 днів.

Моніторинг виконання: Прогрес впровадження змін відображається в системі у вигляді звіту.

Результат:

Нову вимогу додано до проектної документації. Розробники успішно реалізували її в межах визначеного терміну.

Результати і переваги.

Результати:

1. *Ефективність адаптації:*
   - Система скоротила час впровадження нової вимоги на 40%, автоматично аналізуючи регуляторні зміни та інтегруючи їх у процес розробки.
2. *Точність:*
   - Уникнуто ризиків невідповідності завдяки автоматичному порівнянню вимог із новими стандартами.
3. *Зменшення витрат:*
   - Знижено витрати на ручний аналіз змін завдяки автоматизації процесу.

Переваги:

1. *Автоматизація:* Система виконує рутинні завдання, такі як моніторинг, аналіз і оновлення вимог, без втручання людини.

2. *Гнучкість:* Легко адаптується до змін у регуляторних стандартах, незалежно від частоти чи обсягу оновлень.

3. *Інтеграція:* Вбудовується в існуючі інструменти управління проектами (Jira, Confluence), що спрощує робочий процес.

4. *Прозорість:* Регулярно генерує звіти про зміни та їхній статус, забезпечуючи зручну комунікацію з усіма зацікавленими сторонами.

Таблиця 3.2

Візуалізація результатів

Етап	**Час без системи (години)**	**Час із системою (години)**	**Скорочення часу (%)**
Моніторинг змін	10	1	90%
Аналіз змін	8	2	75%
Генерація вимог	6	1	83%
Інтеграція	5	2	60%

Графік, що показує скорочення часу на етапах процесу:

```
import matplotlib.pyplot as plt

stages = [‘Моніторинг’, ‘Аналіз’, ‘Генерація’, ‘Інтеграція’]
time_without_system = [10, 8, 6, 5]
time_with_system = [1, 2, 1, 2]

plt.figure(figsize=(8, 5))
plt.bar(stages, time_without_system, alpha=0.7, label=’Час без системи’,
```

```
color=’blue’)
plt.bar(stages, time_with_system, alpha=0.7, label=’Час із системою’, color=’orange’, width=0.4, align=’edge’)
plt.ylabel(‘Час (години)’)
plt.title(‘Скорочення часу на етапах процесу’)
plt.legend()
plt.grid(axis=’y’, linestyle=’–‘, alpha=0.7)
plt.tight_layout()
plt.show()
```

Висновок до розділу 3. Впровадження такої системи дозволяє значно підвищити ефективність управління вимогами, забезпечуючи відповідність проекту актуальним регуляторним нормам та формувати актуальне технічне завдання до нового релізу платформи.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. Поняття та зміст охорони праці

Охорона праці має два основні значення. У широкому розумінні охорона праці охоплює всі гарантії трудового законодавства, що захищають права працівників. Наприклад, це норми, які забороняють необґрунтоване звільнення (ст. 40, 41 КЗпП України), та гарантії права працівників на оскарження незаконних дій роботодавця в комісіях по трудових спорах або судах.

Охорона праці — це система заходів, спрямованих на створення безпечних і нормальних умов праці під час виконання робочих завдань. Це включає технічні, санітарно-гігієнічні та організаційні заходи.

Згідно із Законом України «Про охорону праці» (ст. 1), це сукупність правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та профілактичних заходів, спрямованих на збереження здоров’я, життя і працездатності працівників.

До охорони праці належать:

- Полегшені умови для жінок, неповнолітніх і працівників зі зниженою працездатністю.
- Виробнича санітарія: мінімізація впливу шкідливих факторів.
- Виробнича безпека: запобігання небезпечним виробничим факторам.
- Пожежна безпека: заходи для запобігання пожежам і зменшення їхніх наслідків.

Галузь «Охорона праці» пов’язана з багатьма науками, зокрема медициною, токсикологією, фізіологією, механікою, правознавством, а також ергономікою, яка вивчає взаємодію людей і техніки.

Правові та організаційні засади охорони праці створюють основу для соціального захисту працівників та впровадження санітарно-гігієнічних і технічних рішень у виробництві.

4.2. Принципи правової охорони праці

Стаття 2 Закону України «Про охорону праці» встановлює, що дія його поширюється на всіх юридичних та фізичних осіб, які відповідно до законодавства використовують найману працю, та на всіх працюючих.

Стаття 4 цього закону визначає, що засади державної політики в галузі охорони праці базуються на 10 основних принципах. У теорії права під правовими принципами розуміються керівні ідеї, які виражають суть, основні властивості та загальну спрямованість розвитку правових норм в межах всієї системи права або її окремих галузей чи інститутів.

Державна політика в галузі охорони праці базується на принципах:

- пріоритету життя і здоров'я працівників, повної відповідальності роботодавця за створення належних, безпечних і здорових умов праці;
- підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництв, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці;
- комплексного розв'язання завдань охорони праці на основі загальнодержавної, галузевих, регіональних програм з цього питання та з урахуванням інших напрямів економічної і соціальної політики, досягнень в галузі науки і техніки та охорони довкілля;
- соціального захисту працівників, повного відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;
- встановлення єдиних вимог з охорони праці для всіх підприємств та суб'єктів підприємницької діяльності незалежно від форм власності та видів діяльності;
- адаптації трудових процесів до можливостей працівника з урахуванням його здоров'я та психологічного стану;
- використання економічних методів управління охороною праці, участі держави у фінансуванні заходів щодо охорони праці, залучення добровільних

внесків та інших надходжень на ці цілі, отримання яких не суперечить законодавству;

- інформування населення, проведення навчання, професійної підготовки і підвищення кваліфікації працівників з питань охорони праці;

- забезпечення координації діяльності органів державної влади, установ, організацій, об'єднань громадян, що розв'язують проблеми охорони здоров'я, гігієни та безпеки праці, а також співробітництва і проведення консультацій між роботодавцями та працівниками (їх представниками), між усіма соціальними групами під час прийняття рішень з охорони праці на місцевому та державному рівнях;

- використання світового досвіду організації роботи щодо поліпшення умов і підвищення безпеки праці на основі міжнародного співробітництва [18].

4.3. Особливості організації охорони праці в ІТ-компанії

Організація охорони праці в ІТ-компанії має свої особливості, зумовлені специфікою діяльності, що пов'язана переважно з інтелектуальною працею, роботою за комп'ютерами та у високотехнологічному середовищі. Основні аспекти:

1. Психофізіологічні особливості роботи

– Розумове навантаження: робота вимагає тривалої концентрації, аналізу даних та прийняття рішень.

– Стресові фактори: строки виконання проєктів, великий обсяг завдань, нерегулярний графік роботи.

2. Ергономічні умови праці

– Організація робочого місця з урахуванням ергономіки (зручні крісла, регульовані столи, якісне освітлення).

– Забезпечення обладнанням, яке мінімізує навантаження на очі (монітори з низьким рівнем випромінювання, регульованою яскравістю).

– Введення правил регулярних перерв для профілактики комп’ютерного зорового синдрому та порушень постави.

3. Санітарно-гігієнічні умови

– Контроль за мікрокліматом офісу (температура, вологість, вентиляція).

– Забезпечення чистоти та регулярного прибирання робочих приміщень.

4. Профілактика професійних захворювань

– Превентивні заходи щодо захворювань опорно-рухового апарату, серцево-судинної системи та очей.

– Організація медичних оглядів, консультацій лікарів, забезпечення страхування здоров’я працівників.

5. Організація гнучкого графіка

– Надання можливості працювати віддалено або за гнучким графіком для зниження стресу та підтримки балансу між роботою і особистим життям.

6. Психологічна підтримка

– Проведення тренінгів з управління стресом і роботою в команді.

– Наявність корпоративного психолога або служби підтримки.

7. Навчання з охорони праці

– Проведення інструктажів з безпеки праці та пожежної безпеки.

– Ознайомлення з вимогами щодо безпечного користування офісною технікою та електрообладнанням.

8. Безпека в офісі

– Забезпечення належного стану електромереж, серверного обладнання та системи вентиляції.

– Розміщення засобів пожежогасіння та евакуаційних схем.

9. Кібербезпека

– Забезпечення безпеки інформації як важливого активу IT-компанії.

– Проведення навчання з правил безпечної роботи з даними.

10. Соціальні аспекти

– Облаштування зон відпочинку для релаксації працівників.

– Організація спортивних активностей, курсів йоги або масажу для підтримки здоров’я.

Особливості організації охорони праці в ІТ-компанії спрямовані на створення комфортного, безпечного та мотивуючого робочого середовища для забезпечення високої продуктивності та збереження здоров’я співробітників.

Кожне підприємство, з яким співпрацює компанія, висуває власні вимоги щодо охорони праці. Усі співробітники проходять своєчасне навчання та перевірку знань із цих питань, а фахівці, які працюють безпосередньо на виробничих об'єктах, обов'язково використовують засоби індивідуального захисту.

Важливою складовою ефективної системи охорони праці в ІТ-компанії є відповідність роботи співробітників вимогам і критеріям замовників. Наприклад, якщо на підприємстві обов’язковим є використання засобів захисту або спецодягу, ці правила поширюються й на працівників компанії. Дотримання таких норм формує превентивну культуру безпеки праці в організації.

Розумний підхід до організації охорони праці створює у працівників відчуття стабільності, захищеності прав та уваги з боку керівництва. Налагоджена система безпеки також сприяє зниженню плинності кадрів, що позитивно впливає на ефективність і стабільність роботи підприємства [19].

Правові засади забезпечення безпеки ІТ спеціаліста під час роботи можуть регулюватися чинними законодавчими актами, в залежності від форми їх співпраці з роботодавцем (замовником). У разі коли формою співпраці є трудовий договір або контракт – така співпраця підпадає під дію Закону України про охорону праці від 14.10.92 № 2695-XII. За цим законом роботодавець зобов’язаний створити для працівника умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці, у тому числі розробляти інструкції з охорони праці та проводити відповідні інструктажі з працівниками.

Якщо трудовим договором передбачена дистанційна робота – роботодавець несе відповідальність тільки у межах використання таким працівником обладнання та засобів, рекомендованих або наданих роботодавцем. Іншою, більш розповсюдженою, формою співпраці є надання ІТ спеціалістами послуг за угодами цивільно-правового характеру. За таким договором замовник доручає спеціалісту на свій страх і ризик виконати певну роботу у визначені терміни, при цьому контроль за процесом роботи та безпекою спеціаліста, з боку замовника, законодавством не передбачений [19].

Новою, гібридною, формою співпраці в ІТ сфері є гіг-контракт. Це особлива форма цивільно-правового договору у Дія City — правовому та податковому просторі для ІТ-компаній в Україні. Співвідносини у межах такого контракту регулюються Законом України Про стимулювання розвитку цифрової економіки в Україні від 15.07.2021 №1667-IX, затвердженими з урахуванням його положень внутрішніми документами резидента Дія Сіті та укладеними з гіг-спеціалістами гіг- контрактами». За умови сплати Єдиного соціального внеску спеціаліст є застрахованою особою від нещасного випадку на виробництві та професійного захворювання, за загальнообов’язковим державним соціальним страхуванням від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності.

Розслідування (спеціальні розслідування) нещасних випадків, що трапляються з застрахованими особами, проводяться на підставі Порядку розслідування та обліку нещасних випадків, професійних захворювань та аварій на виробництві, затвердженого Постановою Кабінету Міністрів України від 17 квітня 2019 р. № 337. Якщо комісією з розслідування (спеціального розслідування) такий випадок буде визнано страховим, тобто пов’язаним з виробництвом на потерпілого буде складено акт за формою Н-1/П. На підставі такого акту працівник, а у разі його смерті члени його родини, набуває права відшкодування шкоди, заподіяної працівникові внаслідок ушкодження його здоров’я або у разі смерті працівника. Такі виплати здійснюється Фондом

соціального страхування України. Як підсумок, хочемо зазначити, що ІТ-сфера не відноситься до галузей з високим рівнем виробничого травматизму та професійної захворюваності. Це можна пояснити тим, що здебільшого у роботі не використовуються потенційно небезпечні машини, механізми або обладнання.

Висновок до розділу 4. Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров’я і працездатності людини у процесі трудової діяльності. Розглянуті правові та організаційні основи охорони праці, які гарантують соціальний захист працівників. Кожне підприємство, з яким співпрацює компанія, має власні вимоги щодо охорони праці. Усі співробітники регулярно проходять навчання та перевірку знань з питань безпеки праці, а фахівці, що працюють безпосередньо на виробництвах, обов'язково використовують засоби індивідуального захисту. Однією з ключових складових функціонування системи охорони праці в ІТ-компанії є відповідність роботи працівників вимогам і критеріям замовників. [19].

РОЗДІЛ 5. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

5.1 Основні поняття охорони навколишнього середовища та вплив ІТ індустрії на навколишнє середовище

Навколишнє середовище є однією з найважливіших категорій сучасної науки і практики, оскільки воно охоплює всі природні і соціально-економічні умови, в яких існують живі організми, зокрема людина. Це комплекс природних, культурних, економічних і технологічних факторів, які впливають на стан здоров'я, розвиток суспільства і функціонування екосистем.

У широкому сенсі навколишнє середовище включає всі елементи природи, такі як атмосфера, водні ресурси, ґрунти, флора і фауна, а також людську діяльність, яка змінює ці природні компоненти. Важливими аспектами є взаємодія між людьми та їхнім середовищем, а також ефекти цих взаємодій на довкілля, таких як забруднення, зміна клімату, втрата біорізноманіття та виснаження природних ресурсів.

Дослідження навколишнього середовища є важливою складовою екології, охорони природи та сталого розвитку, і націлене на збереження та раціональне використання природних ресурсів для майбутніх поколінь.

Навколишнє середовище є об'єктом правового регулювання суспільних відносин з приводу природи як на національному, так і на міжнародному законодавчому рівні.

Охороною навколишнього середовища визнається система заходів, які спрямовані на забезпечення сприятливих і безпечних умов середовища проживання і життєдіяльності людини.

Охорона навколишнього природного середовища є новою формою взаємодії людини з природою.

Ця форма взаємодії з'явилася в сучасних умовах і являє собою систему державних і громадських заходів, які мають на меті:

гармонійний розвиток і взаємодію суспільства і природи;

збереження існуючих екологічних спільнот і природних ресурсів.

Найважливішими факторами навколишнього середовища є:

– Атмосфера — газова оболонка Землі, що включає кисень, азот, вуглекислий газ та інші гази, необхідні для дихання живих організмів. Атмосфера також відіграє ключову роль у регулюванні клімату та температури на планеті.

– Водні ресурси — включають океани, моря, річки, озера, підземні води та водяні ресурси. Вода є основою життя на Землі, і її чистота має вирішальне значення для здоров'я людей, тварин та рослин.

– Ґрунти — верхній шар земної кори, що підтримує рослинний світ і є джерелом поживних речовин для сільськогосподарських культур. Ґрунти забезпечують стабільність екосистем, а їхня деградація може призвести до ерозії та зниження родючості.

– Біорізноманіття — різноманітність живих організмів, включаючи рослини, тварин, мікроорганізми, а також екосистеми, в яких вони існують. Біорізноманіття забезпечує стабільність екосистем і підтримує природні цикли, такі як кругообіг води та поживних речовин.

– Екосистеми — це спільноти живих організмів та їхнього середовища, які взаємодіють між собою, забезпечуючи стабільність навколишнього середовища. Екосистеми включають лісові, водні, степові та інші природні комплекси, кожен з яких виконує важливу роль у підтримці балансу навколишнього середовища.

– Клімат — сукупність погодних умов, які визначають температуру, вологість, атмосферний тиск і інші метеорологічні фактори в даній місцевості. Клімат має великий вплив на життя людей, сільське господарство, здоров'я та екосистеми.

– Мінеральні ресурси — природні ресурси, які включають корисні копалини, такі як металеві та неметалеві мінерали, які використовуються в промисловості та будівництві. Надмірна експлуатація цих ресурсів може призвести до їхнього виснаження та екологічних проблем.

– Соціально-економічні фактори — включають людську діяльність, культуру, технології та економічні умови, які визначають, як люди взаємодіють з навколишнім середовищем. Вони можуть впливати на рівень забруднення, використання ресурсів і на збереження чи деградацію природних умов.

Ці фактори взаємодіють між собою, і їхній стан визначає здоров'я планети та якість життя людей.

Охорона навколишнього середовища полягає в збереженні та відновленні природних ресурсів для того, щоб попередити негативний вплив людини на природу та здоров’я людей.

Проблема охорони навколишнього середовища з точки зору ІТ стає все більш актуальною через зростаючий вплив технологій на довкілля. Інформаційні технології, попри свою користь, можуть мати значний негативний ефект на навколишнє середовище. Основні проблеми, пов’язані з ІТ та охороною навколишнього середовища, включають:

– Енергоспоживання і викиди CO_2: Великі дата-центри та сервери, що використовуються для обробки, зберігання і передачі даних, споживають величезні обсяги електроенергії, що часто походить від викопних джерел. Це призводить до високих викидів парникових газів, що сприяють зміні клімату.

– Електронні відходи: Погіршення якості та швидкий розвиток технологій сприяють накопиченню електронних відходів (e-waste), таких як старі комп'ютери, мобільні телефони, телевізори та інші електронні пристрої. Відсутність ефективних систем переробки цих відходів може призвести до забруднення ґрунтів і водних ресурсів токсичними матеріалами, такими як ртуть, свинець та кадмій.

– Неекологічне виробництво ІТ-продукції: Виробництво комп'ютерної техніки та електроніки часто вимагає значних природних ресурсів, таких як метали, мінерали та енергія. Деякі з цих матеріалів добуваються з порушенням екологічних стандартів, що призводить до деградації навколишнього середовища.

– Ризики з точки зору «чорного» енергоспоживання: Технології, що використовуються для майнінгу криптовалют (наприклад, біткоїн), мають високі енергетичні вимоги, що також додає навантаження на енергетичні ресурси і збільшує викиди CO_2.

– Використання хмарних технологій: Хоча хмарні обчислення дозволяють знижувати споживання енергії на рівні окремих організацій, глобальні дата-центри, що обслуговують хмарні сервіси, можуть призвести до значних витрат енергії, якщо не забезпечується їхня енергоефективність.

– Проблема шумового забруднення: Багато IT-компаній використовують потужні комп’ютери та сервери, що можуть створювати значний рівень шуму, що також має негативний вплив на навколишнє середовище та здоров'я людей.

– Рішення та ініціативи для зменшення екологічного впливу IT:

– Енергоефективність: Перехід на відновлювальні джерела енергії для живлення дата-центрів та оптимізація енергоспоживання серверних і обчислювальних потужностей може значно знизити викиди CO_2.

– Переробка електронних відходів: Важливо розвивати програми з переробки та утилізації електронних відходів, щоб зменшити негативний вплив на довкілля. Компанії можуть відповідально ставитися до процесу утилізації, використовуючи «зелені» технології.

– Зниження викидів від криптовалютного майнінгу: Використання більш енергоефективних алгоритмів майнінгу та перехід на екологічно чисті джерела енергії можуть допомогти зменшити негативний вплив криптовалют на навколишнє середовище.

– Розвиток «зелених» технологій: Впровадження «зелених» IT-програмних рішень, зокрема, систем енергоефективного управління енергоспоживанням і «розумних» мереж, здатних оптимізувати використання ресурсів, може значно зменшити вплив на природу.

– Зміна підходів до розробки та використання продуктів: Індустрія може впроваджувати принципи сталого розвитку, орієнтуючись на економічну

ефективність при збереженні екологічної стабільності, зокрема, через більш довговічні та ремонтопридатні продукти.

– Використання хмарних технологій: Хмарні сервіси можуть бути набагато ефективнішими в плані використання ресурсів у порівнянні з традиційними серверними системами, якщо вони оптимізовані для енергетичної ефективності та екологічного підходу.

Таким чином, IT-сфера має великий потенціал для поліпшення стану навколишнього середовища шляхом впровадження інновацій, технологій та практик, орієнтованих на сталий розвиток і екологічність.

Під охороною навколишнього середовища прийнято розуміти сукупність міжнародних, державних і регіональних заходів щодо захисту навколишнього середовища, закріплених в правових актах, інструкціях і стандартах, і доводять вимоги до кожного конкретного забруднювача [20].

5.2. Права та обов’язки громадян та органів державної влади щодо охорони навколишнього природного середовища

Права та обов'язки громадян та органів державної влади щодо охорони навколишнього природного середовища регулюються низкою законодавчих актів та норм, спрямованих на збереження екології, запобігання забрудненню та підтримання сталого розвитку. Згідно з українським законодавством, зокрема, Законом України "Про охорону навколишнього природного середовища" та іншими екологічними нормативами, визначаються права та обов'язки як громадян, так і органів державної влади.

Права громадян у сфері охорони навколишнього середовища:

– Право на здорове навколишнє середовище: Кожен громадянин має право на сприятливі умови для життя та здоров'я, які включають чисте повітря, воду та інші природні ресурси. Це право забезпечується через доступ до екологічної інформації та право на участь у процесах, що стосуються охорони природи.

– Право на екологічну інформацію: Громадяни мають право на отримання інформації про стан навколишнього середовища, про виконання екологічних норм та стандартів органами влади і підприємствами.

– Право на участь у громадських обговореннях: Громадяни мають право брати участь у процесах екологічної експертизи та громадських слуханнях щодо проектів, які можуть мати вплив на навколишнє середовище.

– Право на подачу скарг та позовів: Громадяни можуть подавати скарги на порушення екологічних прав до органів державної влади, органів місцевого самоврядування, а також звертатися до суду для захисту своїх екологічних прав.

– Право на відшкодування шкоди: Якщо порушення екологічних норм завдає шкоди здоров'ю або майну, громадяни мають право на відшкодування збитків від винних осіб чи організацій.

Обов'язки громадян щодо охорони навколишнього середовища:

– Збереження природних ресурсів: Громадяни повинні дотримуватись екологічних стандартів, не забруднювати природне середовище, економно ставитися до використання природних ресурсів.

– Дотримання екологічних вимог: Особи зобов'язані виконувати вимоги екологічного законодавства, не порушувати правила використання земель, водних та лісових ресурсів, не здійснювати несанкціоноване викидання відходів.

– Участь у збереженні природної спадщини: Громадяни повинні брати участь у заходах по охороні природи, таких як посадка дерев, утримання чистоти у громадських місцях, участь у волонтерських екологічних ініціативах.

– Інформування та співпраця: Громадяни мають обов'язок інформувати відповідні органи про порушення екологічних норм та активно співпрацювати з органами, що здійснюють охорону навколишнього середовища.

Права органів державної влади у сфері охорони навколишнього середовища:

– Регулювання та контроль: Державні органи мають право розробляти та затверджувати нормативно-правові акти в сфері екології, контролювати дотримання екологічних стандартів та заходів щодо захисту природи.

– Проведення екологічної експертизи: Органи влади мають право проводити екологічну експертизу проектів, що можуть мати значний вплив на навколишнє середовище, та приймати рішення про їх дозволи або заборону.

– Збір та обробка екологічної інформації: Вони мають право збирати, аналізувати та розповсюджувати інформацію про стан навколишнього середовища серед громадян та організацій.

– Штрафи та санкції: Органи влади мають право накладати штрафи та застосовувати інші санкції до підприємств, установ або осіб, які порушують екологічні норми.

Обов'язки органів державної влади щодо охорони навколишнього середовища:

– Забезпечення екологічного законодавства: Органи державної влади зобов'язані забезпечити виконання екологічних законів та норм, створювати умови для сталого розвитку, а також запобігати забрудненню навколишнього середовища.

– Захист природних ресурсів: Влада має обов'язок охороняти природні ресурси, забезпечувати їх раціональне використання та збереження для майбутніх поколінь.

– Освіта та просвіта: Державні органи зобов'язані інформувати громадян про важливість охорони навколишнього середовища, проводити освітні кампанії та екологічні заходи для підвищення обізнаності населення.

– Забезпечення фінансування екологічних програм: Влада зобов'язана забезпечити фінансування для реалізації екологічних програм, таких як охорона заповідників, відновлення екосистем, боротьба з забрудненням та змінами клімату.

Отже, правова система України передбачає як права, так і обов'язки для громадян та органів державної влади, спрямовані на охорону навколишнього природного середовища. Активна участь громадян та ефективне управління з боку держави є основними чинниками збереження довкілля та сталого розвитку країни [21-26].

ВИСНОВКИ

В магістерській роботі проведено аналіз особливостей проєкту створення платформи для захисту персональних даних відповідно до GDPR, огляд засобів з управління вимогами IT-проєктів, розроблено моделі проєкту в MS Project, інформаційну систему управління вимогами проєкту платформи для захисту персональних даних відповідно до GDPR, а також проведено аналіз ефективності управління вимогами в проекті створення платформи для захисту персональних даних відповідно до GDPR.

Аналіз предметної області управління вимогами IT-проєктів вказав на необхідність створити інтелектуальну систему для збирання та управління вимогами, що буде використовувати сучасні технології штучного інтелекту та машинного навчання для автоматичного збирання вимог із різних джерел, аналізу їхньої пріоритетності, виявлення можливих конфліктів і пропонування оптимальних рішень. Це дозволить мінімізувати ручну роботу, забезпечити оперативність процесу управління та знизити ризики неврахування важливих змін. Інтелектуальна система стане основним інструментом для ефективного управління вимогами у проєкті, забезпечуючи високу якість релізів навіть у умовах динамічного середовища та нерегулярних оновлень

Впровадження розробленої системи дозволяє значно підвищити ефективність управління вимогами, забезпечуючи безперервну відповідність нормативним стандартам у проєкті шляхом автоматичного моніторингу та адаптації вимог, мінімізуючи ризики і підвищуючи ефективність управління. Цей підхід надає компанії конкурентну перевагу, оскільки дозволить швидко реагувати на зміни в регуляторному середовищі та дозволить позбутися порушення без необхідності ручного втручання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Федишин І.Б. Менеджмент проектної діяльності підприємств промисловості: прогресивні міжнародні стандарти. Матеріали VIII-ої Всеукраїнської науково-практичної конференції пам'яті почесного професора ТНТУ, академіка НАН України М.Г. Чумаченка «Інновації: аспекти управління, виробництва, сфери обслуговування» ТНТУ імені Івана Пулюя, (Тернопіль, 28 березня 2019 року) С.80-81
2. Бабаєв В.М. Управління проектами: Навчальний посібник для студентів спеціальності «Управління проектами» / Бабаєв В.М. – Харків: ХНАМГ, 2006. – 244 с.
3. Петренко О.І. Управління проектами: навчальний посібник / О.І. Петренко, О.І. Горбенко. К: СІК ГРУП Україна, 2019. – 280 с.
4. Віта Галушка Теоретико-методичні засади управління проєктами. DOI https://doi.org/10.32849/2663-5313/2020.7.72
5. Дудник О. О. Удосконалення процесів управління портфелем проєктів в Іткомпаніях // Економічна кібернетика: теорія, практика та напрямки розвитку. 2017. С. 52-54.
6. Дудник О. О. Аналіз процесів управління портфелем проєктів в Іткомпаніях // X Міжнародна науково-практична Інтернет-конференція "Сучасні проблеми моделювання соціально-економічних систем". 2018.
7. Дудник О. О. Процес вибору виконавців ІТ-проєктів у портфоліо проєктів // Економічна кібернетика: теорія, практика та напрямки розвитку. 2018. С. 50- 52.
8. Дудник О. О. Аналіз оптимальної структури бази знань для нечітких експертних систем // XI Міжнародна науково-практична Інтернетконференція "Сучасні проблеми моделювання соціально-економічних систем". 2019.

9. Дудник О. О. Переваги використання нечіткої логіки в процесі управління ІТ-проєктами // XIII Міжнародна науково-практична Інтернет-конференція “Сучасні проблеми моделювання соціально-економічних систем”. 2021.
10. Дудник О. О. Економіка та апарат нечітких експертних систем: тренди розвитку та застосування // Економічна кібернетика: теорія, практика та напрямки розвитку”. 2023. С. 69-73.
11. ІТ-експорт у 2020. За даними НБУ щодо платіжного балансу і зовнішньої торгівлі // BRDO [Електронний ресурс]. URL: https://brdo.com.ua/wpcontent/uploads/2021/02/IT-eksport-u-2020-2.pdf (дата звернення: 21.08.2023).
12. Соколовська З. М., Дудник О. О. Експертні системи: теорія та прикладне застосування: [монографія]. Одеса: Екологія, 2022. 408 с.
13. Aggarwal A., Thakur G. S. M. Techniques of Performance Appraisal – A Review // International Journal of Engineering and Advanced Technology. 2013.
14. Ambler S. W. The Disciplined Agile (DA) Framework [Електронний ресурс]. URL: http://www.disciplinedagiledelivery.com/introduction-to-dad (дата звернення: 22.08.2023). 11.Andrienko-Bentz O. Export-oriented segment of Ukraine’s IT services market: Status quo and prospects // PwC Ukraine [Електронний ресурс]. URL: https://www.slideshare.net/RuslanSivoplyas/exportoriented-segment-of-ukrainesit-services-market2016 (дата звернення: 21.08.2023).
15. Anghel A. Outsourcing Company vs. Product Company // SlideShare [Електронний ресурс]. URL: https://www.slideshare.net/anghelalexandra/ outsourcing-company-vs-product-company (дата звернення: 22.08.2023).
16. Arnuphaptrairong T. Top ten lists of software project risks: Evidence from the literature survey 2011.P. 1–6.

17. Атаманчук П.С., Мендерецький В.В., Панчук О.П., Білий Р.М. Охорона праці в галузі: навч. Посібник. Київ,«Центр учбової літератури», 2017 р.332с.
18. Яремко З.М., Тимошук С.В., Третяк О.І. Ковтун Р.М. Охорона праці: навч. Львів : Видавничий центр ЛНУ імені Івана Франка, 2010. 374 с.
19. Пожарова О.В. Охорона праці : навчальний посібник. Одеса. 2022. 86 с.
20. Джигирей В.С. Екологія та охорона навколишнього природного середовища. — К., 2006.
21. Закон України "Про охорону навколишнього природного середовища". – Закон України від 25.06.1991 № 1264-XII.
22. Закон України "Про екологічну експертизу". – Закон України від 9 лютого 1995 року № 45/95-ВР.
23. Закон України "Про екологічну інформацію". – Закон України від 30 червня 1999 року № 2558-XIV.
24. "Охорона навколишнього природного середовища" / М. С. Бенжик. – Львів: Вид-во Львівської політехніки, 2015.
25. "Загальні засади екологічного права України" / А. І. Івахненко. – Київ, 2013.
26. Екологічне право України: підручник / В. П. Гудима. – Київ: Юрінком Інтер, 2014.
27. Singh, Dheeraj and Chandra, Shalini, «Mitigating Uncertainty and Enhancing Trust in AI: Harmonizing Human-Like, System-Like Features with Innovative Organizational Culture» (2024). *ACIS 2024 Proceedings*. 22. https://aisel.aisnet.org/acis2024/22
28. Лемешко, І., & Козакова, Н. (2023). РОЗРОБКА ЧАТ-БОТУ ДЛЯ АВТОМАТИЧНОГО ПІДБОРУ ВАКАНСІЙ З ВИКОРИСТАННЯМ АЛГОРИТМІВ ПАРСИНГУ І ОБРОБКИ ПРИРОДНОЇ МОВИ (NLP). Collection of scientific papers «SCIENTIA», (June 16, 2023; Athens, Greece), 126-129.

29. Бродкевич, В. М., & Ремесло, В. Я. (2018). Алгоритми машинного навчання (МН) та глибокого навчання (ГН) і їх використання в прикладних додатках. Міжнародний науковий журнал Інтернаука, (11 (1)), 56-60.
30. Кривда, В., Чернов, С. (2024). Проєкт створення платформи для захисту персональних даних відповідно до GDPR. V Всеукраїнська науково-практичній інтернет-конференція *«Інформаційні технології: моделі, алгоритми, системи».*, https://itconf.nuos.edu.ua/2024/wp-content/uploads/sites/7/publications/kryvda-_nr.pdf

ДОДАТОК А

Фрагмент коду

```
// Встановлення залежностей
const express = require('express');
const bodyParser = require('body-parser');
const { Pool } = require('pg');
const axios = require('axios');
// Ініціалізація Express
const app = express();
app.use(bodyParser.json());
// Налаштування підключення до PostgreSQL
const pool = new Pool({
  user: 'yourUser',
  host: 'localhost',
  database: 'yourDatabase',
  password: 'yourPassword',
  port: 5432,
});
// Маршрут для отримання всіх нормативних актів
app.get('/api/regulations', async (req, res) => {
  try {
    const result = await pool.query('SELECT * FROM regulations');
    res.status(200).json(result.rows);
  } catch (err) {
    console.error(err);
```

```
    res.status(500).send('Server error');
  }
});
// Маршрут для додавання нового нормативного акта
app.post('/api/regulations', async (req, res) => {
  const { regulationTitle, description, effectiveDate, projectId } = req.body;
  try {
    await pool.query(
      'INSERT INTO regulations (regulation_title, description, effective_date, project_id) VALUES ($1, $2, $3, $4)',
      [regulationTitle, description, effectiveDate, projectId]
    );
    res.status(201).send('Regulation added successfully');
  } catch (err) {
    console.error(err);
    res.status(500).send('Server error');
  }
});

// Маршрут для отримання всіх інцидентів безпеки
app.get('/api/incidents', async (req, res) => {
  try {
    const result = await pool.query('SELECT * FROM incidents');
    res.status(200).json(result.rows);
  } catch (err) {
```

```
    console.error(err);
    res.status(500).send('Server error');
  }
});
// Маршрут для додавання нового інциденту безпеки
app.post('/api/incidents', async (req, res) => {
  const { incidentType, description, incidentDate, severity, regulationId } = req.body;
  try {
    await pool.query(
      'INSERT INTO incidents (incident_type, description, incident_date, severity, regulation_id) VALUES ($1, $2, $3, $4, $5)',
      [incidentType, description, incidentDate, severity, regulationId]
    );
    res.status(201).send('Incident added successfully');
  } catch (err) {
    console.error(err);
    res.status(500).send('Server error');
  }
});
// Маршрут для отримання всіх вимог
app.get('/api/requirements', async (req, res) => {
  try {
    const result = await pool.query('SELECT * FROM requirements');
    res.status(200).json(result.rows);
  } catch (err) {
```

```
    console.error(err);
    res.status(500).send('Server error');
  }
});
// Маршрут для додавання нової вимоги
app.post('/api/requirements', async (req, res) => {
  const { description, priority, createdDate, dueDate, projectId, regulationId } =
req.body;
  try {
    await pool.query(
      'INSERT INTO requirements (description, priority, created_date, due_date,
project_id, regulation_id) VALUES ($1, $2, $3, $4, $5, $6)',
      [description, priority, createdDate, dueDate, projectId, regulationId]
    );
    res.status(201).send('Requirement added successfully');
  } catch (err) {
    console.error(err);
    res.status(500).send('Server error');
  }
});
// Маршрут для інтеграції з API Have I Been Pwned для отримання інцидентів
app.get('/api/pwned', async (req, res) => {
  const email = req.query.email;
  try {
    const response = await
axios.get(`https://haveibeenpwned.com/api/v3/breachedaccount/${email}`, {
```

```
      headers: {
        'User-Agent': 'Node.js',
        'hibp-api-key': 'your-api-key', // Використовуйте свій API ключ
      },
    });
    res.status(200).json(response.data);
  } catch (err) {
    if (err.response && err.response.status === 404) {
      res.status(404).send('No breaches found');
    } else {
      console.error(err);
      res.status(500).send('Server error');
    }
  }
});

// Підключення до сервера
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```