

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Управління проектами»

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему:

«Дослідження моделей управління проектом з розробки інформаційної системи для роботи з сервісами App Store та Google Play Market»

**Студентки 2-го курсу групи УПз-21**

**Науковий керівник:**

к.т.н., доцент

*(науковий ступінь, вчене звання)*

Єлизавети КЛЕСМАН

Олександр ТІМІНСЬКИЙ

*(ім'я, прізвище)*

*(ім'я, прізвище)*

\_\_\_\_\_

*(підпис студента)*

\_\_\_\_\_

*(дата)*

\_\_\_\_\_

*(підпис)*

**Попередній захист:**

\_\_\_\_\_

*(Висновок: «До захисту в Екзаменаційній комісії»)*

Завідувач кафедри

технологій управління, проф. \_\_\_\_\_

Віктор МОРОЗОВ \_\_\_\_\_

*(підпис)*

*(ім'я, прізвище)*

*(дата)*

**Київ-2024**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

Кафедра технологій управління

Освітній рівень Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Управління проєктами

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

професор Віктор МОРОЗОВ

“ \_\_\_\_\_ ” \_\_\_\_\_ 2024 року

**ЗАВДАННЯ  
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студентка: Клесман Єлизавета Вадимівна

Група: УПз-21

**1. Тема кваліфікаційної роботи**

«Дослідження моделей управління проєктом з розробки інформаційної системи для роботи з сервісами App Store та Google Play Market»

Затверджена протоколом від “28” червня 2024 р. № 13.

**2. Строк подання студентом готової роботи - “16” грудня 2024 р.**

**3. Цільова установка та вихідні дані до роботи:** аналіз методів, способів і засобів управління проєктами задля реалізації обраного проєкту; досягнення поставлених цілей і отримання запланованих результатів у межах виділеного часу та бюджету.

**4. Зміст роботи:**

Аналіз сучасних моделей управління проєктами, обґрунтування методології, розробка концепції проєкту, економічної моделі, організаційної структури, календарного плану, WBS, управління ресурсами, ризиками, фінансами та використання інструментів управління проєктами.

**5. Перелік графічного матеріалу (слайдів):**

Мета роботи, актуальність дослідження, обґрунтування вибору методології, концепція проєкту, модель організаційної структури команди проєкту, фінансова модель проєкту, підсумки фінансового планування, модель WBS проєкту, діаграма Ганта, концептуальна модель бази даних, логічна модель бази даних, матриця ризиків, висновки.

## 6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	План виконання роботи
1.	Вибір теми кваліфікаційної роботи	25.06.24
2.	Затвердження теми кваліфікаційної роботи та призначення наукового керівника	30.08.24
3.	Формування переліку нормативних матеріалів, літератури з проблематики кваліфікаційної роботи	09.09.24
4.	Розробка плану кваліфікаційної роботи і його погодження з науковим керівником	25.09.24
5.	Написання I розділу кваліфікаційної роботи	06.10.24
6.	Написання II розділу кваліфікаційної роботи	20.10.24
7.	Написання III розділу кваліфікаційної роботи	03.11.24
8.	Написання IV розділу кваліфікаційної роботи	17.11.24
9.	Підготовка висновків і пропозицій	24.11.24
10.	Оформлення кваліфікаційної роботи	01.12.24
11.	Передача кваліфікаційної роботи рецензенту для рецензування, перевірка на антиплагіат	11.12.24
12.	Передача кваліфікаційної роботи науковому керівникові	14.12.24
13.	Попередній захист кваліфікаційної роботи	16.12.24
14.	Захист кваліфікаційної роботи	23.12.24

Дата видачі завдання “01” вересня 2024 р.

Керівник роботи

к.т.н., доцент, Олександр ТІМІНСЬКИЙ  
(посада, ім'я, прізвище)

\_\_\_\_\_  
(підпис)

Завдання прийняла до виконання студентка групи УПз-21

Єлизавета КЛЕСМАН  
(ім'я, прізвище)

\_\_\_\_\_  
(підпис)

## ЗМІСТ

<b>ЗМІСТ</b>	4
<b>АНОТАЦІЯ</b>	6
<b>ТАБЛИЦЯ СКОРОЧЕНЬ ТА ПОЯСНЕНЬ</b>	7
<b>ВСТУП</b>	8
<b>РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ РІШЕНЬ В ГАЛУЗІ ОБ'ЄКТУ УПРАВЛІННЯ</b>	13
1.1. Аналіз методів оцінки впливів зовнішніх чинників на ІТ проєкт	13
1.2. Формулювання проблемної області	15
1.3. Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем	18
1.4. Характеристика індустрії управління відгуками та конкурентності в ній	20
1.4.1. Аналіз розміру ринку та тенденцій	20
1.5. Обґрунтування інвестиційної привабливості проєкту	25
1.5.1. Побудова графіку витрат і доходів	27
1.6. Мета дослідження	29
1.7. Аналіз та вибір методології управління проєктом	31
1.7.1. Традиційні методології управління проєктами	31
1.7.2. Гнучка методологія Agile	34
1.7.3. Гібридні методології управління проєктами	39
1.7.4. Вибір методології для досліджуваного проєкту	43
1.8. Обґрунтування вибору методології управління досліджуваним проєктом	43
1.9. Стислий опис проєкту	44
<b>РОЗДІЛ 2. ОПИС КОНЦЕПЦІЇ ПРОЄКТУ</b>	47
2.1. Формалізація ідеї проєкту	47
2.1.1. Актуальність проєкту, проблематика сфери	47
2.1.2. Комунікаційна стратегія	49
2.1.3. Опис модулів інформаційної системи, що є продуктом проєкту	53

2.1.4. Вимоги до продукту проєкту	54
2.2. Команда проєкту	55
2.3. Бюджет проєкту	62
2.3.1. Огляд витрат проєкту	62
2.3.2. Витрати на трудові ресурси	64
2.3.3. Матеріальні та технічні витрати	66
2.3.4. Непередбачені витрати та резерв бюджету	68
2.3.5. Підсумковий бюджет проєкту	69
2.3. Архітектура інформаційної системи продукту проєкту	71
2.4. Структура бази даних продукту проєкту	73
2.4.1. Концептуальна модель бази даних	73
2.4.2. Логічна модель бази даних	76
2.4.3. Зв'язки між сутностями у базі даних	78
2.4.4. Технічні особливості бази даних	78
2.4.5. Оптимізація продуктивності бази даних	79
<b>РОЗДІЛ 3. УПРАВЛІННЯ ЗМІСТОМ ТА ПРОДУКТОМ ПРОЄКТУ.</b>	
<b>ІНФОРМАЦІЙНІ РЕСУРСИ В УПРАВЛІННІ ПРОЄКТОМ</b>	81
3.1. Управління змістом проєкту. WBS і роботи проєкту	81
3.2. Вигляд фінального продукту	90
3.3. Управління часом проєкту. Діаграма Ганта	97
<b>РОЗДІЛ 4. УПРАВЛІННЯ РИЗИКАМИ ПРОЄКТУ</b>	104
4.1. Ідентифікація ризиків на етапі планування та реалізації	104
4.2. Оцінка ризиків. Використання матриці ймовірності та впливу ризиків.	107
4.3. Протиризикові заходи	112
<b>ВИСНОВКИ</b>	<b>115</b>
<b>ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ</b>	<b>117</b>
<b>ДОДАТОК А</b>	<b>126</b>

## АНОТАЦІЯ

кваліфікаційної магістерської роботи на тему:

### **«Дослідження моделей управління проектом з розробки інформаційної системи для роботи з сервісами App Store та Google Play Market»**

Студентка: Клесман Єлизавета Вадимівна

Науковий керівник: Тімінський Олександр Георгійович.

Рік захисту – 2024.

*Темою даної роботи* було обрано «Дослідження моделей управління проектом з розробки інформаційної системи для роботи з сервісами App Store та Google Play Market», предметною областю якої є управління проектами у сфері розробки інформаційних систем для автоматизації обробки відгуків користувачів мобільних додатків.

*Метою даної роботи* є аналіз, адаптація та впровадження сучасних моделей управління проектами, які забезпечують ефективну організацію процесів створення інформаційних систем.

На основі сформульованої мети поставлені наступні *задачі дослідження*: аналіз існуючих підходів до управління проектами, вибір і обґрунтування методології управління; формалізація концепції проекту; розробка архітектури інформаційної системи; планування ресурсів, термінів і задач; ідентифікація ризиків та розробка протиризикових заходів.

*Мета проекту* - розробка системи для автоматизації обробки відгуків з платформ App Store і Google Play Market, яка спрощує моніторинг, аналіз і управління відгуками користувачів.

*Об'єктом дослідження* є процес розробки та впровадження інформаційних систем в індустрії мобільних додатків, спрямовані на автоматизацію роботи з відгуками користувачів у платформах App Store та Google Play Market.

*Предметом дослідження* є процес управління проектом створення інформаційної системи для збору, аналізу та управління відгуками користувачів, включаючи планування, контроль виконання, управління ризиками та забезпечення якості розробки.

*Практична цінність* роботи полягає у формуванні рекомендацій щодо вибору методологій управління проектами, розробці концептуальних моделей бази даних, ідентифікації та оцінці ризиків, а також у розробці плану управління ними. Ці напрацювання можуть бути застосовані як для внутрішніх проектів ІТ компаній, так і для зовнішніх ІТ-проектів.

У кваліфікаційній роботі було досліджено сучасні моделі управління проектами, обґрунтовано вибір фреймворку Kanban, сформовано концепцію інформаційної системи, розроблено модель бази даних, проведено аналіз ризиків і запропоновано заходи щодо їх мінімізації.

Робота містить 129 сторінок з одним додатком, 18 рисунків та 12 таблиць.

## ТАБЛИЦЯ СКОРОЧЕНЬ ТА ПОЯСНЕНЬ

Скорочення	Значення	Пояснення
API	Application Programming Interface	Інтерфейс програмування додатків; набір функцій і протоколів для створення програмного забезпечення.
App Store	App Store	Онлайн-магазин додатків для пристроїв Apple.
Google Play	Google Play Market	Онлайн-магазин додатків для пристроїв Android.
WBS	Work Breakdown Structure	Ієрархічна структура робіт; інструмент для управління проектами.
PMBOK	Project Management Body of Knowledge	Посібник з управління проектами, що містить стандарти, методи та процеси.
DB	Database	База даних; система організації, зберігання та обробки даних.
HDD	Hard Disk Drive	Жорсткий диск; пристрій для зберігання даних.
GDPR	General Data Protection Regulation	Загальний регламент захисту даних, що регулює обробку персональних даних у ЄС.
Front-end	-	Частина програмного забезпечення, що відповідає за взаємодію з користувачем.
Back-end	-	Серверна частина програмного забезпечення; обробляє дані та бізнес-логіку.
Data layer	-	Рівень даних у програмному забезпеченні, відповідальний за взаємодію з базою даних.
QA	Quality Assurance	Забезпечення якості; процес тестування продукту для виявлення помилок.
UI	User Interface	Інтерфейс користувача; зовнішній вигляд програмного забезпечення для взаємодії з користувачем.
UX	User Experience	Користувацький досвід; загальне враження користувача від використання системи.
OBS	Organizational Breakdown Structure	Ієрархічна структура організації; показує розподіл обов'язків у проекті.
IDE	Integrated Development Environment	Інтегроване середовище розробки; програма для написання, тестування та налагодження коду.
OAuth 2.0	Open Authorization	Протокол авторизації, що дозволяє надавати доступ до ресурсів без передачі пароля.
NoSQL	Not Only SQL	Тип баз даних, що не використовує реляційну модель; підходить для великих обсягів даних.
SSL/TLS	Secure Sockets Layer / Transport Layer Security	Протоколи для захищеної передачі даних через мережу.
AES	Advanced Encryption Standard	Стандарт шифрування для захисту даних.

## ВСТУП

Сучасний розвиток галузі ІТ в Україні вимагає динамічного впровадження нових технологій, адаптації до світових стандартів та швидкої реакції на зміни у ринкових умовах. Україна поступово зміцнює свої позиції як важливий гравець у глобальній ІТ-індустрії, демонструючи значний потенціал у сферах розробки програмного забезпечення, мобільних додатків, хмарних сервісів та інформаційних систем. Ринок мобільних додатків особливо стрімко зростає, що спричинено збільшенням кількості користувачів смартфонів та популярністю цифрових послуг. Це підштовхує компанії до розробки високоякісних продуктів, які відповідають вимогам користувачів та забезпечують конкурентні переваги.

Проєкт створення системи управління відгуками для роботи з сервісами App Store та Google Play Market є актуальним у контексті сучасних викликів. Взаємодія з відгуками користувачів є критично важливим аспектом для підтримки репутації мобільних додатків, аналізу потреб клієнтів та підвищення лояльності аудиторії. В умовах зростаючої конкуренції в ІТ-індустрії ефективна система управління відгуками дозволяє не лише оперативно реагувати на проблеми користувачів, а й виявляти нові можливості для вдосконалення продуктів.

З огляду на специфіку обраного проєкту, розробка такої системи є важливим кроком до автоматизації процесів, що забезпечує зниження операційних витрат, покращення комунікації з користувачами та збільшення продуктивності команди підтримки. Крім того, створення централізованої інформаційної системи дозволяє інтегрувати дані з різних платформ, що сприяє більш точному аналізу та прийняттю управлінських рішень. Таким чином, даний проєкт має значний потенціал як для внутрішнього вдосконалення процесів компанії-розробника, так і для подальшого виходу на зовнішній ринок у вигляді готового програмного рішення.

*Метою даної роботи є дослідження моделей управління проектами, їх адаптація до потреб конкретного проекту з розробки інформаційної системи для роботи з сервісами App Store та Google Play Market. У роботі зосереджено увагу на пошуку найбільш ефективних підходів до управління, які забезпечують досягнення поставлених цілей у межах обмежених ресурсів та часу.*

Особливий акцент зроблено на виборі, аналізі та впровадженні сучасних методологій управління проектами, що враховують специфіку ІТ-індустрії та потреби внутрішнього проекту компанії. Завдання роботи включають аналіз традиційних, гнучких та гібридних моделей, а також дослідження інструментів і підходів до планування, управління ризиками, розподілу ресурсів та комунікації у команді.

Мета кваліфікаційної роботи також полягає у розробці комплексного підходу до управління проектом, що базується на сучасних принципах та інструментах, які підвищують ефективність реалізації проектів і забезпечують досягнення їх цілей у динамічному середовищі ІТ-ринку.

На основі сформульованої мети поставлені наступні *задачі дослідження*:

- Провести аналіз сучасних моделей управління проектами, включаючи традиційні, гнучкі та гібридні методології, для визначення найбільш ефективного підходу до управління проектом у сфері ІТ.
- Обґрунтувати вибір фреймворку Kanban як основи для управління проектом, з урахуванням специфіки реалізації проекту створення інформаційної системи для роботи з сервісами App Store та Google Play Market.
- Розробити концепцію інформаційної системи, яка включає структуру команди, визначення ключових модулів системи, архітектури бази даних, а також план її реалізації.
- Оцінити фінансові аспекти проекту, включаючи формування бюджету, розрахунок витрат на трудові ресурси, технічне та програмне забезпечення, а також прогнозування потенційних доходів.

- Продемонструвати практичне застосування обраної методології на прикладі управління реальним проектом, зокрема створення діаграми Ганта, використання інструментів управління задачами та планування.
- Розробити модель управління ризиками, яка охоплює ідентифікацію, оцінку ризиків, створення матриці ймовірності та впливу, а також протиризикові заходи для забезпечення успішної реалізації проекту.

*Мета* проекту, що досліджується в кваліфікаційній роботі - розробка інформаційної системи для роботи з відгуками користувачів у мобільних застосунках, що публікуються в App Store та Google Play Market. Система забезпечуватиме автоматизацію процесів збору, аналізу та управління коментарями, що дозволить компанії-розробнику мобільних ігор покращити взаємодію з користувачами, швидше реагувати на їхні потреби та підвищити якість підтримки клієнтів.

Розробка інформаційної системи включає створення функціональних модулів для аналізу відгуків, автоматизації відповідей і формування звітності, що реалізується на основі сучасних технологій баз даних і програмного забезпечення. Система також інтегруватиметься з платформами App Store і Google Play Market, забезпечуючи доступ до коментарів з різних джерел в одному місці.

У результаті проекту команда підтримки компанії отримає ефективний інструмент для роботи з відгуками, який знижує трудові витрати на обробку інформації, покращує користувацький досвід і підтримує позитивну репутацію продуктів на ринку.

*Об'єктом дослідження* є процес розробки та впровадження інформаційних систем в індустрії мобільних додатків, спрямовані на автоматизацію роботи з відгуками користувачів у платформах App Store та Google Play Market.

*Предметом дослідження* є процеси управління проектом створення інформаційної системи для збору, аналізу та управління відгуками користувачів,

включаючи планування, контроль виконання, управління ризиками та забезпечення якості розробки.

Для досягнення поставленої мети та вирішення завдань у кваліфікаційній роботі магістра застосовано комплекс методів дослідження, які забезпечили всебічне вивчення об'єкта і предмета дослідження, а також ефективність реалізації проєкту. Зокрема:

- Аналіз літературних та інформаційних джерел - використовувався для вивчення сучасних моделей управління проєктами, включаючи традиційні, гнучкі та гібридні методології. Цей метод дозволяє обґрунтувати вибір фреймворку Kanban як основного підходу до управління досліджуваним проєктом.

- Метод порівняльного аналізу - застосовувався для оцінки існуючих рішень в індустрії управління відгуками, аналізу конкурентів, визначення тенденцій та інвестиційної привабливості проєкту.

- Методи системного аналізу - використовувалися для побудови концептуальної та логічної моделі бази даних, визначення архітектури інформаційної системи та її модулів, що забезпечує структурність і системність запропонованих рішень.

- Метод моделювання - застосовано для створення ієрархічної структури робіт (WBS), діаграми Ганта, матриці ймовірності та впливу ризиків, що дозволило візуалізувати процес управління проєктом і здійснити детальне планування.

- Методи оцінки ризиків - використовувалися для ідентифікації, аналізу та оцінки ризиків, що можуть виникнути на різних етапах реалізації проєкту, із застосуванням матриці ймовірності та впливу.

- Метод спостереження - використовувався для оцінки ефективності використання обраних інструментів управління проєктом, таких як Jira, і процесів роботи команди.

*Практична цінність кваліфікаційної роботи* полягає у наступному:

- Аналіз та адаптація сучасних моделей управління проєктами. У роботі досліджено та адаптовано ефективні підходи до управління проєктами у сфері розробки інформаційних систем. Це дозволяє сформулювати рекомендації щодо вибору методологій управління для подібних проєктів, що можуть бути використані як у рамках компанії, так і для інших організацій.

- Розробка концепції інформаційної системи. У роботі розроблено концептуальну і логічну моделі бази даних, архітектуру системи та ключові модулі. Ці напрацювання можуть бути використані як приклад для впровадження аналогічних систем у різних сферах бізнесу, де необхідно автоматизувати обробку зворотного зв'язку.

- Практичні інструменти для управління проєктами. Запропоновано рішення щодо використання сучасних інструментів управління, таких як Jira, що демонструють переваги централізованого контролю за виконанням задач, оптимізації комунікацій у команді та моніторингу ризиків.

- Підвищення кваліфікації у галузі управління проєктами. В процесі роботи проведено аналіз ризиків, сформовано матрицю ймовірності та впливу, розроблено протиризикові заходи. Ці аспекти мають значну практичну цінність для розуміння принципів ризик-менеджменту в реальних проєктах.

- Універсальність отриманих результатів. Розроблені методичні підходи до управління проєктом, планування ресурсів, оцінки ризиків і створення бази даних можуть бути використані для подібних проєктів у різних галузях ІТ, що розширює спектр практичного застосування результатів роботи.

## РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ РІШЕНЬ В ГАЛУЗІ ОБ'ЄКТУ УПРАВЛІННЯ

### 1.1. Аналіз методів оцінки впливів зовнішніх чинників на ІТ проєкт

Успішне управління ІТ проєктом неможливе без урахування впливу зовнішніх чинників. До таких чинників належать ринкові умови, економічна ситуація, правові обмеження, технологічні зміни, екологічні ризики, політична стабільність та соціальні тренди. Для оцінки цих факторів використовують різні методи та інструменти, кожен з яких має свої переваги та недоліки [1], [2].

Методи оцінки зовнішніх чинників:

1. PESTLE-аналіз (Political, Economic, Social, Technological, Legal, Environmental): цей метод дозволяє оцінити політичні, економічні, соціальні, технологічні, правові та екологічні фактори, які можуть впливати на ІТ проєкт [3], [4]. Він підходить для оцінки макроекономічних та стратегічних ризиків.

- Політичні фактори: зміни в регуляціях або політичній ситуації можуть вплинути на доступ до ринку або роботу в певних регіонах.
- Економічні фактори: коливання валютних курсів, рівень інфляції чи зміна податкового законодавства можуть змінювати витрати на проєкт.
- Соціальні фактори: зміни у вподобаннях користувачів або демографічні тренди можуть впливати на попит.
- Технологічні фактори: новітні технології можуть вимагати адаптації продукту або створення нових інноваційних рішень.
- Правові фактори: вимоги до захисту персональних даних, як-от GDPR, можуть обмежувати методи збору й обробки даних користувачів.
- Екологічні фактори: вони можуть впливати на експлуатаційні витрати та репутацію компанії.

2. SWOT-аналіз: SWOT аналіз використовується для виявлення сильних і слабких сторін проєкту, а також можливостей та загроз, які виникають

під впливом зовнішніх факторів. Він допомагає формувати стратегії для мінімізації ризиків та максимізації переваг [5].

- Сильні сторони: аналіз потенційних переваг проєкту, що можуть бути посилені в зовнішньому середовищі.
- Слабкі сторони: виявлення слабких місць, які можуть бути посилені під впливом зовнішніх чинників.
- Можливості: визначення зовнішніх можливостей для розвитку проєкту.
- Загрози: оцінка можливих ризиків з боку зовнішнього середовища.

3. Аналіз зацікавлених сторін (Stakeholder Analysis): метод допомагає виявити вплив різних зацікавлених сторін на проєкт та їхні очікування. Важливо враховувати інтереси не тільки клієнтів та користувачів, але й регуляторів, інвесторів, партнерів, постачальників і конкурентів [6], [7].

4. Аналіз сценаріїв (Scenario Analysis): цей підхід дозволяє передбачити різні варіанти розвитку подій та підготувати кілька можливих стратегій дій [8]. Наприклад, можна оцінити вплив різних економічних криз або змін в технологічній інфраструктурі на проєкт.

5. Моніторинг конкурентного середовища: вивчення дій конкурентів допомагає зрозуміти зовнішні ризики і можливості. Регулярний аналіз продуктів і стратегій конкурентів дозволяє визначити тенденції на ринку та адаптувати проєкт під нові вимоги [9].

Зовнішні чинники можуть як позитивно, так і негативно впливати на ІТ проєкт. Використання різних методів оцінки, таких як PESTLE, SWOT, аналіз зацікавлених сторін та інших, допомагає отримати комплексну картину зовнішнього середовища. Це дозволяє підготувати проєкт до потенційних змін та знизити ризики невдачі.

## 1.2 Формулювання проблемної області

У сучасному світі мобільні додатки є ключовими інструментами взаємодії бізнесу та користувачів. Магазини App Store і Google Play Market стають основними платформами для розповсюдження додатків, а відгуки користувачів у цих магазинах відіграють вирішальну роль у формуванні репутації додатка. У рамках даного проєкту передбачається вирішення кількох важливих проблем, пов'язаних із ефективним управлінням зворотним зв'язком, репутацією додатків та аналітикою користувацьких відгуків [10], [11], [12].

### 1. Необхідність централізованого управління відгуками.

Мобільні додатки отримують відгуки від користувачів у магазинах App Store та Google Play. Ці відгуки мають значний вплив на прийняття рішень новими користувачами щодо завантаження додатків. Проте керувати ними часто буває складно через розпорошеність коментарів на різних платформах, що ускладнює їх аналіз і оперативне реагування. Відсутність централізованої системи управління ускладнює обробку великої кількості відгуків, що надходять щоденно, що призводить до упущення важливої інформації про проблеми з додатком або нові побажання користувачів.

### 2. Погіршення репутації через несвоєчасну реакцію на відгуки.

Несвоєчасна або неадекватна реакція на відгуки, особливо негативні, може значно вплинути на репутацію додатка. Негативні коментарі залишені без відповіді можуть призвести до зниження рейтингу додатка та втрати потенційних користувачів. З іншого боку, своєчасна відповідь може покращити репутацію, підвищити лояльність користувачів та виправити негативне враження. Отже, одна з ключових проблем - це відсутність механізмів швидкої і автоматизованої обробки відгуків. Приклади цих проблем представлені на рис. 1.1 та 1.2.

## Feedback metrics






APP	AVG. RATING	TOTAL REVIEWS	REPLY EFFECT	NO REPLY EFFECT	AVG. REPLY TIME
 Game iOS	4.25	53	0	0	7d 20h 5m
 Game Android	4.43 	745	+0.562 	-0.333	4d 15h 29m 

Рис. 1.1. Приклад порівняння метрик зворотного зв'язку для однієї гри на двох платформах

## Feedback metrics

APP	AVG. RATING	TOTAL REVIEWS	REPLY EFFECT	NO REPLY EFFECT	AVG. REPLY TIME
 Game 1	4.43 	745	+0.562 	-0.333	4d 15h 29m 
 Game 2	4.17	382	0	-1.333	12d 10h 20m
 Game 3	3.86	5.3k	+0.091	+0.25	15d 8h 45m

Рис. 1.2. Приклад порівняння метрик зворотного зв'язку для трьох ігор на одній платформі

### 3. Відсутність ефективних інструментів аналізу відгуків.

Існуючі системи збору відгуків здебільшого обмежуються простим отриманням даних, але не забезпечують інструментів для поглибленого аналізу зворотного зв'язку. Відсутність аналітичних механізмів, таких як автоматична класифікація відгуків за тональністю (позитивні, негативні, нейтральні) чи виявлення ключових тем, значно ускладнює виявлення критичних проблем або загальних тенденцій у взаємодії користувачів з додатком. Це уповільнює процес прийняття рішень щодо покращення додатка та усунення недоліків.

### 4. Обмежені можливості для автоматизації взаємодії з користувачами.

Команди підтримки часто стикаються з повторюваними питаннями та проблемами, які можна вирішити за допомогою шаблонів відповідей або автоматизованих механізмів обробки типових запитів. Проте магазини App Store

та Google Play Market мають обмежений функціонал роботи з відгуками. Зокрема, ці платформи не надають можливості налаштувати автоматизовані відповіді на поширені запитання або створити персоналізовані сценарії взаємодії з користувачами. Це призводить до того, що команди підтримки змушені обробляти більшість звернень вручну, що займає багато часу і ресурсів.

### 5. Відсутність централізованого управління репутацією.

Важливо не лише реагувати на кожен окремий відгук, а й керувати загальною репутацією додатка. Це вимагає наявності інструментів для моніторингу загального рейтингу, аналізу динаміки змін та своєчасної корекції стратегій взаємодії з користувачами. Відсутність такого інструментарію призводить до того, що негативні відгуки можуть накопичуватися, що погіршує загальну репутацію додатка та ускладнює процес залучення нових користувачів.

Приклад візуалізації даних по відгукам представлено на рис. 1.3.



Рис. 1.3. Приклад візуалізації даних по відгукам для гри, опублікованої в Google Play Market.

### **1.3 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем**

У сучасній практиці управління ІТ-проектами існує безліч інструментів і технологій, що допомагають вирішувати проблеми, які були виявлені на попередньому етапі аналізу. Особливу роль у цьому відіграють методи машинного навчання та природньо-мовного оброблення (NLP), а також інтеграція з різними платформами для централізованого збору даних і автоматизації процесів роботи з відгуками.

Вже сьогодні більшість компаній, що активно працюють з мобільними додатками, використовують методи машинного навчання для обробки великої кількості даних, що генеруються користувачами. Особливо важливу роль тут відіграють алгоритми природньо-мовного оброблення, які дозволяють автоматично аналізувати текстові відгуки. Ці алгоритми використовують методи семантичного аналізу для класифікації відгуків за різними критеріями, такими як тональність (позитивна, негативна, нейтральна), частота згадувань певних проблем чи функцій, а також за темами, які користувачі обговорюють найчастіше [13].

Наприклад, системи аналізу відгуків, як-от MonkeyLearn та Lexalytics, дозволяють налаштовувати спеціальні категорії для класифікації текстів за різними параметрами. Ці системи автоматично виявляють тенденції, що можуть сигналізувати про критичні проблеми у роботі додатку або про незадоволення користувачів певними функціями продукту. Така класифікація є ключовою для своєчасного реагування на негативні відгуки, що впливає на репутацію продукту [14], [15].

Окрім того, за допомогою глибинного навчання можна вдосконалювати системи рекомендацій щодо реакцій на відгуки. Наприклад, нейромережі, як-от GPT-3, можуть бути інтегровані для створення персоналізованих відповідей на основі аналізу конкретного контексту відгуку. Це дозволяє автоматизувати

процеси, що раніше потребували значних людських ресурсів, та значно підвищує ефективність роботи команд підтримки.

Однією з основних проблем, з якими стикаються компанії, що працюють на ринку мобільних додатків, є необхідність об'єднання відгуків з різних платформ в одну систему. Інтеграція відгуків з платформ App Store, Google Play Market та соціальних мереж дозволяє централізовано управляти взаємодією з користувачами, що значно спрощує процес аналізу даних і прийняття рішень.

AppFollow, одна з провідних платформ для управління відгуками, пропонує багатоканальні рішення для роботи з відгуками та повідомленнями з різних джерел. Компанії можуть централізовано обробляти всі відгуки, незалежно від того, з якого джерела вони надходять, та використовувати єдині алгоритми аналізу. Це дозволяє компаніям отримувати повну картину взаємодії з користувачами та оперативно реагувати на проблеми [16].

Ще один приклад успішної інтеграції багатоканальних рішень — це платформа Zendesk, яка пропонує інструменти для управління взаємодією з користувачами через різні канали комунікації, зокрема електронну пошту, соціальні мережі та чати. Компанії, які використовують такі рішення, можуть значно знизити час реакції на відгуки користувачів, що безпосередньо впливає на задоволеність клієнтів [17].

Ще одним важливим аспектом є автоматизація відповіді на типові запити через використання шаблонів. Це дозволяє значно пришвидшити процес надання відповіді, стандартизуючи якість обслуговування. Такі системи, як Freshdesk та Salesforce, дозволяють зберігати шаблони для типових запитань, що часто задаються користувачами, і використовувати їх для автоматичного реагування.

Наприклад, система Salesforce Service Cloud дозволяє створювати складні сценарії автоматичних відповідей, які залежать від змісту питання. Це особливо важливо для компаній, що отримують велику кількість повторюваних запитів і прагнуть скоротити час надання підтримки.

Крім автоматизації взаємодії з користувачами, важливим є інтеграція з платформами для аналітики, що дозволяють збирати дані про динаміку відгуків. Такі інструменти, як Power BI та Google Data Studio, можуть бути використані для побудови звітів на основі відгуків користувачів з різних джерел. Вони дозволяють візуалізувати дані щодо тематики відгуків, тональності та швидкості реакції команди підтримки [18].

Наприклад, компанія Spotify використовує інструменти аналітики для того, щоб моніторити відгуки про нові функції свого додатку. Це дозволяє їм своєчасно виявляти недоліки в роботі нових релізів та реагувати на них ще до того, як вони призведуть до масштабних негативних наслідків.

#### **1.4. Характеристика індустрії управління відгуками та конкурентності в ній**

У сучасному світі цифрових технологій управління відгуками стало важливим елементом для бізнесів різних галузей. Відгуки користувачів можуть значно впливати на репутацію компанії, формування іміджу продукту та навіть на рівень продажів. Тому сервіси, які надають можливості для збору, аналізу та управління відгуками, стали невід'ємною частиною стратегії багатьох компаній.

Індустрія сервісів для управління відгуками зростає разом із збільшенням популярності онлайн-торгівлі та соціальних медіа. За даними Statista, у 2023 році близько 80% споживачів приймають рішення про покупку на основі відгуків інших користувачів [19]. Це свідчить про важливість управління репутацією і взаємодії з клієнтами для підприємств. Сервіси управління відгуками забезпечують компаніям інструменти для збору та аналізу відгуків, що дозволяє їм оперативно реагувати на потреби споживачів та поліпшувати свої послуги.

##### **1.4.1. Аналіз розміру ринку та тенденцій**

За даними 2023 року від дослідницької компанії kbvresearch (глобальна компанія з дослідження ринку та консалтингу), очікується, що глобальний ринок програмного забезпечення для управління відгуками досягне 17,3 мільярда

доларів США до 2028 року, при цьому середньорічний темп зростання (CAGR) складе 11,7% упродовж прогнозованого періоду [20]. Графік представлено на рис. 1.4.

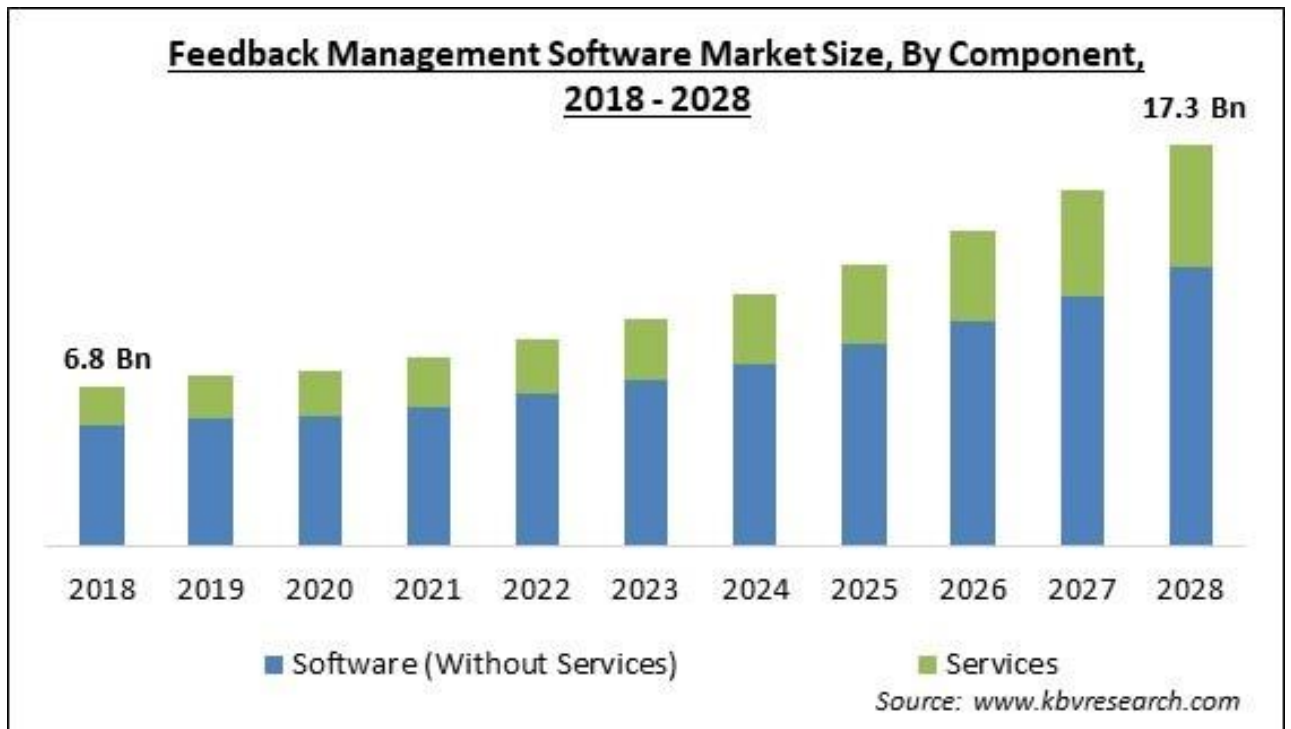


Рис. 1.4. Розмір ринку програмного забезпечення для управління відгуками 2018-2023 рр. та прогноз на 2024-2028 рр.

Згідно з прогнозами, розмір ринку програмного забезпечення для керування відгуками та зворотним зв'язком у 2024 році становив 10,11 мільярда доларів США. Очікується, що до 2031 року цей показник зросте до 28,04 мільярда доларів США, при цьому середньорічний темп зростання (CAGR) у період з 2024 по 2031 рік складе 13,60% [21].

Графік прогнозного зростання ринку програмного забезпечення для управління відгуками та зворотним зв'язком представлено на рис. 1.5.

## Feedback And Reviews Management Software Market

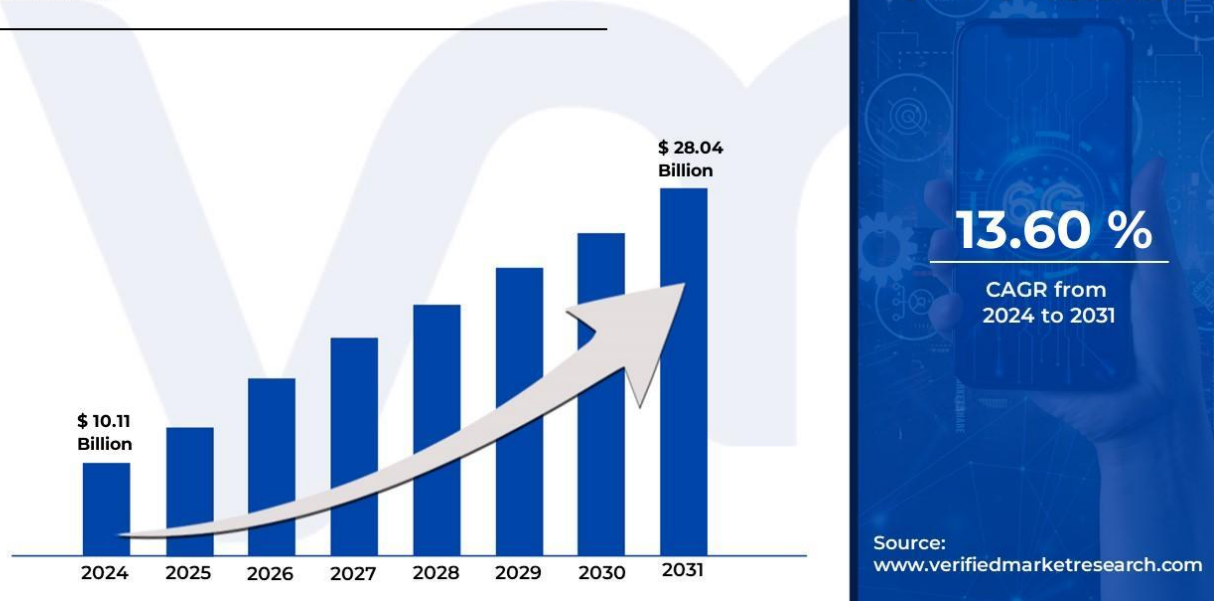


Рис. 1.5. Прогнозне зростання ринку програмного забезпечення для управління відгуками та зворотним зв'язком (2024-2031 роки)

У світі існує безліч сервісів, які пропонують рішення для збору відгуків, включаючи як платні, так і безкоштовні варіанти. Ці сервіси можуть охоплювати різні аспекти управління відгуками, такі як автоматизований збір, аналітика, моніторинг згадок в соціальних мережах та інтеграція з CRM-системами.

Сервіси для управління відгуками можна класифікувати на кілька основних категорій:

- Системи для збору відгуків: ці сервіси автоматизують процес збору відгуків від користувачів, надаючи їм можливість залишати коментарі на сайті компанії, у соціальних мережах або через електронну пошту. Приклади таких сервісів: SurveyMonkey, Typeform, Google Forms.
- Платформи для моніторингу відгуків: ці сервіси дозволяють компаніям відстежувати відгуки на різних платформах, включаючи соціальні медіа, форуми та сайти з відгуками. Серед відомих платформ - Mention, Brand24 та Hootsuite, які забезпечують аналітичні інструменти для оцінки репутації бренду.

- Інструменти для аналітики відгуків: ці сервіси аналізують зібрані дані про відгуки, виявляючи ключові тренди, настрої та проблеми, з якими стикаються споживачі. Зокрема, це можуть бути платформи, такі як Qualtrics та Reputation.com, які надають глибокий аналіз даних для виявлення патернів у відгуках.

- CRM-системи з функцією управління відгуками: багато CRM-систем, такі як Salesforce та Zoho CRM, інтегрують функції управління відгуками, що дозволяє бізнесам мати єдину платформу для взаємодії з клієнтами та управління їхніми запитамі [22].

Управління відгуками стає критично важливим елементом для успіху бізнесу з кількох причин:

- Формування репутації: відгуки відіграють ключову роль у формуванні репутації бренду. Позитивні відгуки можуть залучити нових клієнтів, тоді як негативні - відштовхнути їх. Сервіси для управління відгуками допомагають компаніям активно взаємодіяти з клієнтами, вирішувати їхні проблеми та створювати позитивний імідж.

- Покращення продуктів і послуг: аналіз відгуків дозволяє компаніям виявляти недоліки в своїх продуктах чи послугах та вживати заходів для їх поліпшення. Зворотний зв'язок від клієнтів може стати основою для розвитку нових функцій, продуктів або навіть зміни стратегії бізнесу.

- Забезпечення лояльності клієнтів: активна робота з відгуками показує клієнтам, що їхня думка важлива для компанії. Це, в свою чергу, сприяє підвищенню рівня лояльності та утриманню клієнтів.

- Підвищення конкурентоспроможності: компанії, які ефективно управляють відгуками, можуть отримати конкурентні переваги. Вміння реагувати на запити та побажання клієнтів створює позитивний імідж і допомагає виділитися серед конкурентів.

На сьогоднішній день індустрія сервісів для управління відгуками стикається з рядом нових викликів та трендів:

- Автоматизація процесів: багато компаній впроваджують автоматизовані системи збору та обробки відгуків. Це дозволяє скоротити час на обробку інформації та зменшити людський фактор у процесі.

- Використання штучного інтелекту: інструменти штучного інтелекту допомагають у аналізі великих обсягів даних, виявляючи патерни та тренди, що дозволяє компаніям швидко реагувати на зміни в споживчих вподобаннях.

- Зростання значення соціальних мереж: соціальні медіа стають все більш важливими платформами для збору відгуків. Клієнти залишають коментарі про продукти не лише на сайтах, але й у соціальних мережах, що вимагає від компаній активного моніторингу та реагування.

- Підвищення вимог до прозорості: сьогодні споживачі очікують від компаній прозорості в їхній діяльності. Управління відгуками має включати не лише позитивні, але й негативні відгуки, що свідчить про готовність компанії приймати критику та працювати над поліпшенням.

Управління відгуками стало важливим аспектом бізнес-стратегії у сучасному світі. Сервери, що надають можливості для роботи з відгуками, допомагають компаніям ефективно взаємодіяти з клієнтами, покращувати свої продукти та послуги, а також сформувати позитивний імідж. Оскільки конкуренція на ринку зростає, організації, які зможуть швидко адаптуватися до змін у споживчих вподобаннях і активно працювати з відгуками, зможуть досягти успіху в своїй діяльності.

У таблиці 1.1. представлено порівняння декількох компаній, які надають сервіси для управління відгуками. Ця таблиця надає аналіз основних компаній, що спеціалізуються на управлінні відгуками. Вона демонструє варіативність функцій та цінових моделей, що допомагає компаніям обирати оптимальні рішення відповідно до своїх потреб і бюджету.

**Аналіз конкурентів**

Компанія	Ключові функції	Ціна на місяць	Переваги	Недоліки
AppFollow	Збір відгуків, аналіз тональності, відповіді	\$200 - \$1000	Потужна аналітика, інтеграція з багатьма CRM-системами	Висока ціна для малого бізнесу
Zendesk	Управління взаємодією з клієнтами, шаблони відповідей	\$49 - \$215	Проста інтеграція з іншими інструментами	Обмежені функції аналізу відгуків
Reputation.com	Збір відгуків, аналітика на основі AI	\$300 - \$1000+	Потужна аналітика, функції звітності	Висока ціна, складне налаштування
SurveyMonkey	Збір відгуків, створення анкет	\$25 - \$99	Простота використання	Обмежені можливості автоматизації
Trustpilot	Відстеження відгуків, рейтинги	\$199 - \$499	Висока видимість у пошукових системах	Менше можливостей для індивідуальної інтеграції

**1.5. Обґрунтування інвестиційної привабливості проєкту**

Проєкт зі створення власної системи управління відгуками для внутрішніх потреб компанії дозволить значно оптимізувати витрати на обслуговування користувачів, підвищити якість комунікації та забезпечити гнучкість у розвитку продукту. Основна мета проєкту - створити спеціалізоване рішення, яке відповідає специфічним потребам компанії, що розробляє та підтримує три мобільні ігри для iOS та Android.

Використання сторонніх сервісів для управління відгуками обійдеться компанії приблизно в \$200-\$1000 на місяць, що складає від \$2,400 до \$12,000 на рік. Однак ці сервіси мають обмежені можливості для налаштування, а їх функціональність може не відповідати всім вимогам компанії. Зокрема,

адаптація інструментів, таких як аналіз відгуків за специфічними категоріями або інтеграція з внутрішньою CRM, є складною та часто обмеженою.

Розробка власної системи надає компанії можливість повного контролю над функціоналом і безпекою, а також дозволяє налаштувати інструменти відповідно до конкретних потреб. Хоча початкові витрати на створення продукту будуть вищими, ніж витрати на сторонній сервіс, у довгостроковій перспективі власна система забезпечить економію та дозволить адаптувати її для майбутнього виведення на ринок.

Після успішного впровадження система зможе вийти на зовнішній ринок у форматі SaaS-рішення, націленого на інші компанії зі схожими потребами. Це дозволить перетворити систему з витратного на прибутковий інструмент, залучаючи нові джерела доходу.

Ринок програмного забезпечення для управління відгуками продовжує активно зростати. Попит на такі продукти постійно збільшується. Враховуючи, що вартість ліцензії або підписки на подібні продукти для середніх компаній коливається від \$200 до \$1,000 на місяць.

Залежно від наданих функцій та адаптованості під потреби клієнтів, власний продукт може бути вигіднішою альтернативою для компаній, яким потрібне гнучке та економічне рішення. Завдяки можливості пропонувати унікальні налаштування, компанія має потенціал швидкого залучення клієнтів і стабільного зростання доходу.

З огляду на природне розширення клієнтської бази (наприклад, через рекомендації задоволених користувачів), збільшення інтересу до продукту, а також активний ринок, реалістично очікувати місячного приросту доходів на 5%. Крім того, у разі підвищення якості функцій та впровадження нових інструментів до системи (аналітика, інтеграція з іншими платформами) вартість підписки може зростати, що також сприятиме збільшенню доходів.

На основі цих розрахунків і прогнозів доходів побудуємо графік, що показує, коли проєкт почне приносити прибуток.

### **1.5.1. Побудова графіку витрат і доходів**

*Дата старту проєкту:* 8 січня 2025 року.

*Початкові витрати на розробку:* залучення розробників, дизайнерів, QA інженера і бізнес-аналітика з урахуванням оптимізації трудових ресурсів.

*Початок доходів:* після внутрішнього впровадження, приблизно на 8 місяць проєкту, можливий вихід системи на зовнішній ринок. Для швидкого виходу на ринок і залучення перших клієнтів, пропонується встановити початкову ціну підписки на рівні \$300 на місяць за базовий пакет з можливістю підвищення вартості при додаткових функціях або більшому обсязі даних. Ця ціна є достатньо конкурентоспроможною, щоб залучити клієнтів, які шукають якісне рішення за помірну вартість, і водночас дозволяє окупити витрати та почати отримувати дохід.

Зробимо приблизний план продажу:

- 1-7 місяці: залучення 0 клієнтів;
- 8-й місяць: залучення 2 клієнтів;
- 9-й місяць: додатково 3 клієнти (разом 5 клієнтів);
- 10-й місяць: додатково 4 клієнти (разом 9 клієнтів);
- 11-й місяць: додатково 5 клієнтів (разом 14 клієнтів);
- 12-й місяць: додатково 6 клієнтів (разом 20 клієнтів);
- 13-й місяць і надалі щомісяця: залучення 3-4 нових клієнтів на місяць.

Для побудови графіку, використаємо функції Excel. Внесемо дані у таблицю 1.2.

Таблиця 1.2.

## План продажу

Місяці	Витрати	План продажу	Клієнтів разом	Дохід	Підписка в \$
1	10876	0	0	0	300
2	10876	0	0	0	
3	10876	0	0	0	
4	10876	0	0	0	
5	6862	0	0	0	
6	6862	0	0	0	
7	6378	0	0	0	
8	6378	2	2	600	
9	6378	3	5	1500	
10	6378	4	9	2700	
11	6378	5	14	4200	
12	6378	6	20	6000	
13	6378	4	24	7200	
14	6378	4	28	8400	
15	6378	4	32	9600	
16	6378	4	36	10800	
17	6378	4	40	12000	
18	6378	4	44	13200	
19	6378	4	48	14400	
20	6378	4	52	15600	
21	6378	4	56	16800	
22	6378	4	60	18000	
23	6378	4	64	19200	
24	6378	4	68	20400	

За наведеними даними (табл. 1.2) побудуємо графік для співставлення доходів та витрат і визначення точки беззбитковості (рис. 1.5).

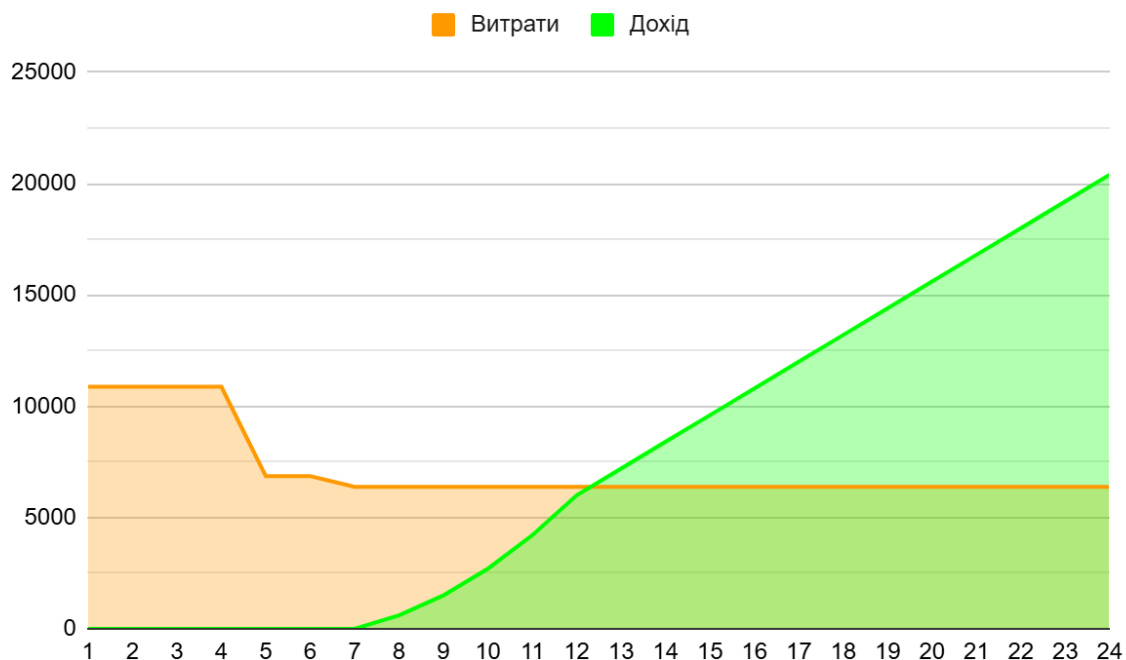


Рис. 1.6. Графік витрат та доходів

Згідно таблиці 1.2. та графіку на Рис. 1.6, проєкт почне приносити прибуток починаючи з 13 місяця після початку роботи.

### 1.6. Мета дослідження

Основною метою цього дослідження є розробка та аналіз моделей управління проєктом, які можуть бути використані для створення інформаційної системи, призначеної для взаємодії з сервісами App Store та Google Play Market. Дослідження зосереджується на ідентифікації, адаптації та впровадженні ефективних моделей управління, що відповідають специфічним вимогам розробки інформаційних систем у сфері мобільних додатків.

Однією з ключових задач є вивчення існуючих моделей управління проєктами, таких як традиційні (водоспадна модель) та гнучкі (Agile, Scrum, Kanban) підходи, і визначення, які з них найбільш підходять для проєктів, що передбачають розробку складних інформаційних систем. Це дослідження також має на меті адаптувати ці моделі до умов розробки системи для збору, аналізу та управління відгуками користувачів у магазинах додатків [1].

Конкретні завдання, що включаються в мету дослідження:

1. *Аналіз вимог та специфікацій*: дослідження починається з вивчення особливостей і вимог, які пред'являються до систем, що працюють із платформами App Store та Google Play. Це включає функції збору відгуків, аналізу тональності коментарів, інтеграції з іншими аналітичними інструментами та забезпечення швидкої відповіді на користувацькі запити.

2. *Вивчення існуючих моделей управління проєктами*: необхідно провести порівняльний аналіз моделей управління проєктами, які використовуються в ІТ-сфері, зокрема тих, що вже довели свою ефективність у розробці інформаційних систем. Серед таких моделей можуть бути Agile, Scrum, Kanban, а також гібридні підходи, які поєднують елементи різних методологій.

3. *Оцінка можливостей інтеграції системи з зовнішніми сервісами*: окремою задачею є аналіз можливостей інтеграції системи з CRM-системами, аналітичними платформами, а також із системами, що забезпечують управління взаємодією з клієнтами. Це дозволить забезпечити ефективний обмін даними та отримання зворотного зв'язку від користувачів додатків.

4. *Розробка концептуальної моделі управління проєктом*: на основі отриманих даних дослідження має на меті створити концептуальну модель управління проєктом для впровадження інформаційної системи "AppReviewPro". Це включає розробку ієрархічної структури робіт, планування ресурсів, розподіл ролей у команді та визначення етапів проєкту.

5. *Розробка моделей контролю якості та ризиків*: для успішного завершення проєкту необхідно передбачити механізми управління якістю та ризиками. Це включає оцінку потенційних загроз, таких як затримки у виконанні завдань, недоліки в системі безпеки даних або проблеми із сумісністю з платформами App Store та Google Play.

6. *Оцінка ефективності розробленої моделі*: на завершальному етапі дослідження планується оцінка ефективності запропонованої моделі управління проєктом через серію тестувань і аналізу результатів впровадження системи. Це

включає вивчення того, наскільки розроблена система сприяє підвищенню якості зворотного зв'язку з користувачами та покращенню продуктів компанії.

Очікувані результати:

- розробка науково обґрунтованої моделі управління проектом, яка дозволить впровадити автоматизовану систему для збору та аналізу відгуків з мобільних платформ.
- удосконалення взаємодії між розробниками та кінцевими користувачами, що дозволить швидше реагувати на відгуки та покращувати якість мобільних додатків.
- покращення управління ризиками в процесі розробки складних ІТ-проектів.

## **1.7. Аналіз та вибір методології управління проектом**

Управління проектом є комплексним процесом, що вимагає застосування певних методологій для досягнення цілей у встановлені строки, з урахуванням бюджету та якості. Методології управління проектами забезпечують системний підхід до планування, виконання та контролю проектних процесів, з метою ефективного досягнення бажаних результатів. Кожна методологія має свої особливості, і вибір відповідного підходу залежить від характеру проекту, його масштабів, вимог клієнтів та умов ринку.

### **1.7.1 Традиційні методології управління проектами**

Традиційні методології управління проектами базуються на послідовному виконанні етапів проекту та їх ретельному плануванні на початковій стадії. Вони найкраще підходять для проектів з чітко визначеними вимогами, де передбачається мінімум змін у ході реалізації. Серед традиційних методологій найбільш відомі Каскадна модель (Waterfall) та PMBOK (Project Management Body of Knowledge).

### Каскадна модель життєвого (Waterfall)

Каскадна модель, відома також як "водоспадна", є однією з моделей життєвого циклу, яка притаманна найстарішим та найвідомішим методологіям управління проєктами. Вона базується на лінійній послідовності етапів, кожен з яких повинен бути повністю завершений перед переходом до наступного. Далі розглянемо основні етапи в каскадній моделі [23], [24].

*Аналіз вимог.* На цьому етапі визначаються всі вимоги до проєкту. Замовник надає чіткий перелік того, що потрібно від системи або продукту, а команда проєкту фіксує ці вимоги у вигляді технічної документації.

*Проектування.* Після визначення вимог починається процес проектування системи або продукту. Цей етап включає розробку архітектури, вибір технологій та планування інтеграції різних компонентів.

*Реалізація (розробка).* Після завершення проектування команда переходить до безпосередньої розробки системи. Кожен модуль або компонент створюється відповідно до вимог та специфікацій, розроблених на попередніх етапах.

*Тестування.* Після завершення розробки проводиться ретельне тестування системи, щоб переконатися в її відповідності вимогам і відсутності помилок. Усі проблеми, виявлені на цьому етапі, повинні бути усунені перед впровадженням системи.

*Впровадження.* Після успішного тестування продукт впроваджується і стає доступним для кінцевих користувачів. На цьому етапі проєкт вважається завершеним.

*Експлуатація та обслуговування.* Після впровадження продукт потребує підтримки, що включає виправлення помилок і оновлення функціональності.

Переваги каскадної моделі:

- чітка структура і послідовність дій.

- легко управляти великими проектами, оскільки кожен етап має чітко визначені задачі.

- добре підходить для проектів з фіксованими вимогами, які не змінюються під час реалізації.

Недоліки каскадної моделі:

- недостатня гнучкість - якщо виявляється помилка на пізніх етапах, повернення до попередніх стадій може бути складним і дорогим.

- відсутність можливості оперативного реагування на зміни в вимогах, що ускладнює роботу з проектами, де необхідні регулярні коригування.

### PMBOK (Project Management Body of Knowledge)

PMBOK є стандартом, розробленим Project Management Institute (PMI), і використовується як загальний підхід до управління проектами у всьому світі. Ця методологія ґрунтується на п'яти основних групах процесів, кожна з яких включає певний набір дій, необхідних для ефективного управління проектом [25].

*Ініціація.* У ході цього процесу визначається проєкт, його цілі та задачі. Цей етап включає оцінку доцільності проєкту, а також визначення зацікавлених сторін і ключових критеріїв успіху.

*Планування.* На цьому етапі розробляється детальний план проєкту, що включає графіки, бюджети, ресурси та вимоги. У PMBOK на особливу увагу заслуговує визначення обсягу робіт, створення розкладу та планування управління ризиками.

*Виконання.* Після завершення планування команда переходить до реалізації плану, що включає розробку продукту або виконання робіт згідно з вимогами проєкту. У PMBOK підкреслюється важливість управління командою, комунікацій та якістю виконання завдань.

*Моніторинг і контроль.* Цей етап передбачає постійне відстеження прогресу проєкту, щоб переконатися, що він відповідає початковому плану.

Моніторинг дозволяє вчасно виявляти відхилення і коригувати плани у разі потреби.

*Завершення.* На завершальному етапі проєкт офіційно закривається, включаючи передання продукту або результатів проєкту замовнику, а також аналіз уроків, які можуть бути використані в майбутніх проєктах.

Переваги РМВОК:

- структурований і формалізований підхід, що дозволяє управляти складними проєктами.
- використання універсальних процесів і кращих практик, що застосовуються в багатьох галузях.
- детальний моніторинг і контроль процесів дозволяє мінімізувати ризики.

Недоліки РМВОК:

- потребує значної кількості документації та ресурсів для реалізації.
- процес може бути занадто складним і витратним для невеликих проєктів або команд.

Традиційні методології, такі як Каскадна модель і РМВОК, добре підходять для проєктів, де вимоги стабільні та зрозумілі з самого початку. Вони забезпечують чітку структуру, послідовність і контроль процесів, що особливо корисно для великих проєктів, які потребують ретельного планування і мінімізації ризиків. Однак, ці методології можуть виявитися недостатньо гнучкими для проєктів, де вимоги часто змінюються або де потрібна швидка адаптація до нових умов.

### **1.7.2 Гнучка методологія Agile**

Гнучка методологія управління проєктами Agile стала однією з найбільш популярних парадигм в управлінні проєктами, особливо в ІТ-сфері, де проєкти часто характеризуються мінливими вимогами, високою динамікою та

необхідністю швидкої адаптації до нових умов. Agile-методології ставлять у центр уваги постійну взаємодію між командами, замовниками та кінцевими користувачами, забезпечуючи гнучкість у плануванні і швидке реагування на зміни. Одними з найвідоміших і найчастіше використовуваних Agile-методів є Scrum та Kanban [26].

### Основні принципи Agile

Agile підходи базуються на Agile Manifesto, який був сформульований у 2001 році і включає наступні цінності та принципи [27].

*Люди та взаємодія важливіші за процеси та інструменти.* Agile підходи роблять акцент на комунікації між членами команди, співпраці з клієнтами та гнучкому плануванні, а не на строгих процесах і жорстких інструментах.

*Робочий продукт важливіший за документацію.* Пріоритет надається створенню працюючого продукту, навіть якщо це вимагає меншого обсягу формальної документації.

*Співпраця з клієнтом важливіша за контракти.* Agile стимулює тісну взаємодію з замовником на всіх етапах проєкту, щоб забезпечити відповідність результату вимогам та очікуванням клієнта.

*Готовність до змін важливіша за дотримання плану.* Agile дозволяє швидко реагувати на зміни у вимогах навіть на пізніх етапах розробки, забезпечуючи адаптивність і гнучкість процесів.

### Scrum

Scrum - це один з найпопулярніших фреймворків у межах Agile. Він орієнтований на короткі цикли розробки, відомі як спринти, кожен з яких зазвичай триває від 1 до 4 тижнів. Основна ідея Scrum полягає у тому, щоб розділити великий проєкт на серію невеликих, чітко визначених етапів, що дозволяє частіше отримувати зворотний зв'язок від замовника і відповідно адаптувати продукт до його вимог [28], [29].

Далі розглянемо основні елементи Scrum.

#### 1. Ролі в Scrum:

1.1. *Product Owner (Власник продукту)*: представляє інтереси замовника, формулює вимоги до продукту і відповідає за формування та пріоритизацію беклогу продукту.

1.2. *Scrum Master*: відповідає за дотримання Scrum-процесу в команді. Його роль полягає у тому, щоб забезпечити ефективну роботу команди, усувати перешкоди і допомагати дотримуватися принципів Agile.

1.3. *Development Team (Команда розробки)*: самоорганізована група фахівців, яка виконує роботу, пов'язану з розробкою продукту. Команда розробки відповідає за створення і постачання інкрементів продукту в кінці кожного спринту.

## 2. Процеси в Scrum:

2.1. *Спринти*: кожен спринт - це короткий цикл розробки, який починається з планування і завершується демонстрацією робочого інкременту продукту. Тривалість спринту визначається на початку проєкту і залишається сталою протягом усього його життєвого циклу.

2.2. *Щоденні стендапи (Daily Scrum)*: щоденні короткі зустрічі команди (зазвичай до 15 хвилин), на яких обговорюється прогрес, визначаються перешкоди та плануються подальші дії.

2.3. *Sprint Review (Огляд спринту)*: після завершення спринту проводиться огляд виконаної роботи. Команда демонструє продукт власнику продукту та іншим зацікавленим особам, отримуючи зворотний зв'язок.

2.4. *Sprint Retrospective (Ретроспектива спринту)*: цей етап дозволяє команді аналізувати процеси, що відбулися під час спринту, та визначати, що можна покращити в наступних циклах.

## 3. Scrum артефакти:

3.1. *Product Backlog*: це перелік усіх вимог і функцій, які потрібно реалізувати у проекті. Product Owner визначає пріоритетність елементів беклогу на основі потреб клієнта.

3.2. *Sprint Backlog*: це перелік завдань, які команда розробки планує виконати під час поточного спринту. Беклог спринту формується на основі Product Backlog під час планування спринту.

3.3. *Інкремент*: це робоча версія продукту, яка повинна бути завершена в кінці кожного спринту. Інкремент включає всі виконані функції, що були заплановані на спринт, і повинні бути готові до випуску. Серед переваг методології Scrum можна виділити наступні:

- Гнучкість: Scrum дозволяє швидко реагувати на зміни у вимогах проекту та потребах клієнтів.
- Прозорість процесів: щоденні стендапи та спринтові огляди забезпечують високу комунікацію та прозорість всіх процесів для зацікавлених сторін.
- Висока залученість замовника: замовник постійно отримує робочі версії продукту, що дозволяє йому давати оперативний зворотний зв'язок.

Недоліки Scrum:

- Складнощі у великих командах: Scrum може бути важко масштабувати для великих команд або складних проектів з багатьма взаємозалежними компонентами.
- Високі вимоги до самоорганізації команди: успіх Scrum залежить від рівня самоорганізації команди, що може бути викликом для менш досвідчених команд.

### Kanban

Kanban - це ще один популярний фреймворк методології Agile, який фокусується на візуалізації робочого процесу та безперервному вдосконаленні [30]. Kanban передбачає, що всі завдання проекту відображаються на спеціальній

дошці, що дозволяє команді та зацікавленим сторонам легко стежити за прогресом [31].

Далі розглянемо основні елементи Kanban.

*Kanban-дошка.* Основний інструмент методології, який відображає всі завдання команди у вигляді карток на дошці. Дошка зазвичай поділяється на кілька колонок, які представляють різні стадії виконання завдання, наприклад: "Очікує", "У процесі", "Завершено".

*Картки.* Кожне завдання відображається у вигляді картки на дошці. Картки містять інформацію про завдання, його відповідальну особу, дедлайни та іншу важливу інформацію.

*Обмеження за кількістю завдань.* Kanban обмежує кількість завдань, які можуть одночасно перебувати на кожному етапі процесу (наприклад, не більше 5 завдань у колонці "У процесі"). Це допомагає уникати перевантаження команди і дозволяє зосередитися на найважливіших завданнях.

*Потік завдань.* Kanban забезпечує безперервний потік завдань. Як тільки завдання завершено, його місце може зайняти нове, що дозволяє гнучко реагувати на змінні пріоритети.

Серед переваг фреймворка Kanban можна виділити наступні:

- Простота і прозорість: Kanban-дошка надає чітке уявлення про статус проєкту і дозволяє легко відслідковувати прогрес.
- Гнучкість у плануванні: на відміну від Scrum, де є чіткі спринти, Kanban дозволяє безперервно додавати нові завдання, коли попередні завершуються.
- Зменшення навантаження на команду: обмеження на кількість завдань, що виконуються одночасно, допомагає уникати перевантаження і забезпечує якісніше виконання завдань.

Також зазначимо недоліки фреймворка Kanban:

- Відсутність чітких етапів: через відсутність чітких спринтів або дедлайнів команди можуть втрачати темп або відкладати роботу на пізніше.

- Відсутність структури: Kanban не забезпечує структури для великих проєктів, що може призвести до хаосу, якщо проєкт має багато залежностей.

Хоча обидва фреймворки є частиною Agile-підходу, вони мають деякі відмінності. Scrum орієнтований на роботу в рамках фіксованих спринтів і передбачає регулярні зустрічі та огляди, тоді як Kanban є більш безперервним і гнучким підходом, що фокусується на візуалізації процесу та обмеженні завдань у роботі.

### **1.7.3. Гібридні методології управління проєктами**

Гібридні методології управління проєктами поєднують елементи як традиційних, так і гнучких підходів. Вони створюють баланс між структурованим, послідовним управлінням проєктами, яке характерне для традиційних методологій (наприклад, каскадна модель або РМВОК), та гнучкістю і швидкістю реагування на зміни, що характерні для Agile-підходів. Гібридний формат особливо підходить для проєктів зі складною структурою, де частина процесів потребує чітко регламентованого виконання, а інша частина вимагає адаптивного підходу для швидкої реакції на змінні вимоги замовників або ринкові умови.

Гібридний підхід передбачає адаптацію та налаштування проєктної методології залежно від специфіки проєкту, його етапів або вимог. У рамках такого підходу команда може використовувати класичні етапи проєктного планування та контролю на початку або для визначення критичних точок проєкту, але водночас інтегрувати гнучкі методи для оперативного виконання окремих частин проєкту. Це дозволяє забезпечити структуру там, де вона необхідна, і гнучкість там, де це можливо [32].

Серед переваг гібридних методологій можна виділити наступні:

- Гнучкість і контроль: гібридні методології дозволяють використовувати гнучкі підходи для тих частин проєкту, де зміни відбуваються

найбільш активно, зберігаючи водночас контроль за допомогою традиційних підходів у більш критичних етапах.

- Адаптація до специфіки проєкту: гібридний формат дозволяє компаніям обирати та поєднувати елементи різних методологій, щоб забезпечити найбільш ефективне виконання проєкту відповідно до його специфіки.

- Збалансований підхід до управління ризиками: структуровані етапи планування та моніторингу, що характерні для традиційних методологій, допомагають контролювати ключові ризики проєкту. Водночас гнучкі етапи сприяють швидкому реагуванню на нові виклики та ризики.

Одними з найбільш розповсюджених гібридних підходів є Scrumban та Lean. Ці методології поєднують у собі переваги як гнучких, так і традиційних підходів, дозволяючи адаптувати процеси управління проєктами залежно від вимог проєкту.

### Scrumban

Scrumban — це гібридний фреймворк, який поєднує елементи Scrum та Kanban. Він є особливо корисним для проєктів, де потрібен гнучкий підхід до планування завдань, але водночас важливо підтримувати постійний потік роботи, що характерний для Kanban [33].

Розглянемо основні елементи Scrumban.

*Планування за принципами Scrum.* У Scrumban використовується система спринтів, запозичена зі Scrum, але їх тривалість є більш гнучкою і може змінюватися залежно від потреб проєкту. Це дозволяє зберігати чіткий план і водночас швидко адаптувати його до нових вимог.

*Візуалізація процесів, як у Kanban.* Завдання в Scrumban відображаються на Kanban-дошці, що забезпечує наочність поточного прогресу проєкту. Команда постійно бачить, які завдання перебувають на стадії виконання, що допомагає ефективно керувати робочим процесом.

*Гнучке управління завданнями.* Scrumban не має строгого обмеження на кількість завдань у спринтах, як це прийнято у класичному Scrum. Натомість завдання можуть бути виконані безперервно, як у Kanban, з постійним моніторингом робочого навантаження.

Переваги Scrumban:

- Гнучкість планування: команда може адаптувати тривалість спринтів і структуру роботи відповідно до вимог проєкту.
- Ефективна візуалізація: Використання Kanban-дошки дозволяє легко відстежувати прогрес проєкту і забезпечує прозорість процесів.
- Безперервний потік роботи: Scrumban підтримує постійне виконання завдань без зупинок між спринтами, що дозволяє швидко реагувати на нові вимоги або зміни.

Недоліки Scrumban:

- Відсутність строгих рамок: гнучкість у Scrumban може стати проблемою для команд, які звикли до чітко структурованих процесів.
- Складність у масштабуванні: хоча Scrumban підходить для малих та середніх проєктів, для великих проєктів його може бути важче масштабувати через відсутність жорсткої структури.

### Lean-IT

Lean-IT - це фреймворк, який фокусується на максимізації цінності для клієнта та мінімізації втрат. Цей підхід походить з виробничих процесів, але з часом він став широко застосовуваним у розробці програмного забезпечення та управлінні проєктами [34].

Далі виділимо основні принципи Lean.

*Усунення зайвих витрат.* Lean орієнтований на усунення всього, що не додає цінності кінцевому продукту. Це означає зменшення неефективних етапів, зайвих витрат і часу на виконання завдань.

*Максимізація цінності.* Основна увага приділяється створенню максимального значення для клієнта. У Lean-проєкті кожне завдання розглядається з точки зору того, як воно додає цінність до кінцевого продукту.

*Постійне вдосконалення.* Lean передбачає, що кожен процес можна постійно покращувати. Команди аналізують свої дії, щоб знайти можливості для оптимізації роботи та підвищення продуктивності.

Серед основних етапів Lean можна виділити наступні.

*Ідентифікація цінності.* Проєкт починається з визначення того, що саме є найбільш важливим для клієнта, і яким чином можна забезпечити цю цінність.

*Створення потоку.* Процес організовується таким чином, щоб забезпечити плавний потік завдань, уникаючи затримок і блокувань. Використовуються інструменти для візуалізації, як-от Kanban-дошки, що допомагають бачити "вузькі місця" в процесі.

*Витягування завдань.* Lean забезпечує, що нові завдання додаються в роботу лише тоді, коли попередні вже завершені. Це дозволяє уникнути перевантаження і зменшує втрати через неефективний розподіл робочого часу.

*Постійне вдосконалення (Kaizen).* Команди Lean постійно аналізують процеси, шукають можливості для покращення та вдосконалення роботи, щоб зробити її більш ефективною.

Переваги Lean:

- Оптимізація процесів: Lean допомагає виявити і усунути будь-які непотрібні етапи, що підвищує продуктивність і скорочує час виконання завдань.
- Фокус на цінності: орієнтація на потреби клієнта гарантує, що команда завжди працює над тим, що дійсно важливо для досягнення кінцевого результату.
- Гнучкість і адаптивність: Lean дозволяє швидко адаптуватися до нових вимог або змін у проєкті, водночас забезпечуючи оптимізацію процесів.

Недоліки Lean:

- Складність у впровадженні: для успішного впровадження Lean необхідна сильна організаційна культура, орієнтована на постійне вдосконалення.

- Непридатність для проєктів з високою невизначеністю: Lean може бути менш ефективним для проєктів з високим рівнем невизначеності або проєктів, де вимоги постійно змінюються.

Scrumban підходить для проєктів з динамічними вимогами, де потрібно поєднувати короткі цикли розробки зі швидким реагуванням на зміни. Lean більше орієнтований на оптимізацію робочих процесів та мінімізацію втрат, що є критично важливим для довгострокових проєктів із складними процесами. Гнучкість у Scrumban вища завдяки поєднанню Scrum і Kanban, тоді як структурованість і ефективність — це головний акцент Lean.

#### **1.7.4. Вибір методології для досліджуваного проєкту**

Для даного проєкту обраний фреймворк Kanban, оскільки він надає можливість гнучко керувати процесами, забезпечуючи безперервний потік завдань та оперативне реагування на зміни. Kanban дозволяє візуалізувати робочий процес за допомогою Kanban-дошки, що допоможе контролювати прогрес і балансувати навантаження команди, уникаючи перевантажень. Цей підхід є оптимальним для проєктів, де вимоги можуть змінюватися в процесі роботи, та забезпечує ефективне управління у динамічних умовах.

### **1.8. Обґрунтування вибору методології управління досліджуваним проєктом**

Вибір моделі управління для проєкту розробки є критично важливим рішенням, яке безпосередньо впливає на успіх проєкту, зокрема на його виконання в строк, відповідність вимогам та ефективне використання ресурсів. Для проєкту зі створення системи управління відгуками, який характеризується високим рівнем динаміки та необхідністю швидкої адаптації до нових вимог,

було обрано гнучку методологію управління ІТ проєктами Agile, зокрема фреймворк Kanban.

Проєкт, пов'язаний із розробкою системи управління відгуками, передбачає необхідність швидкого реагування на змінні вимоги клієнтів або користувачів. Kanban, завдяки безперервному потоку завдань, дозволяє команді працювати у гнучкому режимі, без необхідності дотримання жорстко регламентованих спринтів, як це відбувається у Scrum. Завдяки цьому команда зможе додавати нові завдання або коригувати пріоритети в будь-який момент проєкту, що особливо важливо в умовах, де зміни відбуваються часто [35].

Використання Kanban-дошки надає чітке уявлення про те, на якому етапі знаходиться кожне завдання. Це дозволяє краще контролювати процес виконання робіт, своєчасно виявляти затримки та "вузькі місця", а також коригувати навантаження на команду. Для проєктів з розробки програмного забезпечення, де робочий процес може бути нерівномірним, така візуалізація є корисною для забезпечення прозорості й ефективного управління.

Kanban передбачає регулярний аналіз та вдосконалення процесів, що сприяє підвищенню ефективності роботи. Важливо, що модель дозволяє командам фокусуватися на якості кожного етапу розробки, зменшуючи кількість незавершених завдань і підвищуючи продуктивність.

Однією з ключових особливостей Kanban є обмеження кількості завдань, що можуть виконуватися одночасно. Це запобігає перевантаженню команди, дозволяючи їй зосередитися на найважливіших завданнях і забезпечуючи їхнє якісне виконання. Це також допомагає уникнути багатозадачності та сприяє більш ефективному використанню ресурсів [36].

### **1.9. Стислий опис проєкту**

Проєкт зі створення системи управління відгуками користувачів спрямований на розробку внутрішнього рішення для компанії, яка спеціалізується на розробці трьох мобільних ігор для платформ iOS та Android.

Основна мета проєкту - впровадження автоматизованої системи, яка дозволить ефективно збирати, аналізувати та обробляти відгуки користувачів з різних джерел, таких як App Store, Google Play та інші платформи.

#### Основні задачі проєкту проєкту

*Автоматизація збору відгуків.* Система буде інтегрована з основними платформами для мобільних додатків, що дозволить автоматично збирати відгуки користувачів у реальному часі.

*Аналіз відгуків за допомогою штучного інтелекту.* Для ефективного управління відгуками будуть використовуватися алгоритми машинного навчання, які дозволять класифікувати відгуки за тональністю (позитивні, негативні, нейтральні) та виявляти ключові проблеми, що згадуються користувачами.

*Інтеграція з існуючими інструментами компанії.* Система буде інтегрована з CRM та іншими аналітичними інструментами, щоб забезпечити централізоване управління комунікаціями з користувачами та отримувати комплексні звіти про стан взаємодії.

*Покращення зворотного зв'язку з користувачами.* Система забезпечить можливість швидкої та автоматизованої відповіді на типові запити користувачів, що значно знизить навантаження на команду підтримки.

#### Очікувані результати проєкту

*Підвищення рівня лояльності користувачів.* Швидка реакція на відгуки користувачів сприятиме зменшенню негативного досвіду і збільшенню задоволеності продуктами компанії.

*Покращення якості продукту.* Аналіз відгуків дозволить ідентифікувати основні проблеми та побажання користувачів, що стане основою для покращення ігор.

*Можливість подальшої комерціалізації.* Після впровадження системи в межах компанії планується її розвиток у вигляді комерційного продукту, який можна буде запропонувати іншим компаніям для управління відгуками.

Таким чином, цей проєкт націлений на вирішення важливих завдань, пов'язаних із якісним управлінням відгуками, що сприятиме покращенню взаємодії з користувачами та розвитку продуктів компанії.

## **РОЗДІЛ 2. ОПИС КОНЦЕПЦІЇ ПРОЄКТУ**

### **2.1. Формалізація ідеї проєкту**

Проєктування інформаційних систем для взаємодії з платформами App Store і Google Play Market є актуальним у зв'язку зі швидким розвитком мобільних додатків та зростанням важливості зворотного зв'язку від користувачів. Успішна робота з відгуками дозволяє підвищити якість продуктів, а також покращити взаємодію між розробниками та користувачами. Саме тому розробка систем, що забезпечують ефективний аналіз і відповідь на коментарі, стає важливим інструментом для забезпечення репутації мобільних додатків.

Проєкт, представлений у цьому дослідженні, спрямований на створення інформаційної системи, що автоматизує процес обробки відгуків користувачів з основних магазинів додатків. Розробка цієї системи є важливим етапом для покращення роботи команд підтримки, а також оптимізації взаємодії з кінцевими користувачами додатків.

#### **2.1.1. Актуальність проєкту, проблематика сфери**

*Назва проєкту:* розробка системи для управління коментарями до мобільних додатків у магазинах App Store та Google Play Market.

*Предметна область проєкту:* проєкт належить до сфери інформаційних систем та технологій, що використовуються для моніторингу та управління користувацькими відгуками у мобільних додатках. Основним завданням є автоматизація процесу збору, аналізу та обробки коментарів, що надходять з платформ App Store та Google Play.

*Проблематика:* основною проблемою, з якою стикаються розробники додатків, є значна кількість відгуків користувачів, які потребують швидкої реакції. Несвоєчасна відповідь на коментарі, особливо негативні, може призвести до втрати користувачів та зниження рейтингу додатків. Відсутність

централізованої системи для управління відгуками призводить до ускладнень в аналізі даних, що у свою чергу впливає на якість продукту.

*Аспекти аналізу:*

- Автоматизація збору даних з різних платформ (App Store та Google Play Market);
- Аналіз тональності відгуків (позитивні, негативні, нейтральні);
- Швидкість обробки відгуків та надання відповідей на коментарі;
- Оцінка впливу відгуків на репутацію додатків.

*Актуальність* проєкту обумовлена зростаючою роллю відгуків у формуванні репутації мобільних додатків та впливом цих відгуків на рішення користувачів щодо завантаження або видалення програм. У сучасних умовах якісна та швидка взаємодія з користувачами є ключовим фактором для утримання високих позицій додатків у рейтингах.

*Метою проєкту* є створення ефективної системи для автоматизації збору, аналізу та управління відгуками користувачів, що дозволить підвищити якість обслуговування та покращити репутацію мобільних додатків.

*Цілі проєкту:*

- Зібрати відгуки з магазинів додатків App Store та Google Play у режимі реального часу;
- Автоматизувати процес обробки відгуків та відповідей на них;
- Підвищити задоволеність користувачів за рахунок швидкої реакції на їхні коментарі;
- Оптимізувати процеси управління репутацією мобільних додатків.

*Результати проєкту:*

- Розробка інтерфейсу для збору відгуків з платформ App Store та Google Play Market;
- Створення алгоритмів для автоматичного аналізу тональності коментарів;

- Впровадження системи автоматизації відповідей на типові запити користувачів;
- Розробка механізмів моніторингу репутації додатків та оцінки динаміки змін відгуків.

*Продукт проєкту:* це система "AppReviewPro", яка включає модулі збору та аналізу коментарів, автоматизацію відповідей, а також функції моніторингу та оцінки репутації додатків.

*Вимоги до продукту проєкту:* система повинна бути здатною обробляти великі обсяги даних у реальному часі, забезпечувати високу швидкість реагування на коментарі та підтримувати зручний інтерфейс для користувачів різного рівня технічної підготовки. Важливою вимогою є також захист даних користувачів та відповідність нормам безпеки.

### **2.1.2. Комунікаційна стратегія**

*Цільовою аудиторією* цього проєкту є компанії, що розробляють мобільні додатки для платформ iOS та Android і прагнуть підвищити якість зворотного зв'язку з користувачами. Це можуть бути середні та великі розробники ігор або додатків, які мають необхідність автоматизувати процес збору та аналізу відгуків. Основними користувачами системи є:

- Розробники додатків;
- Команди підтримки;
- Аналітики, що аналізують зворотний зв'язок з користувачами.

*Комунікаційна стратегія всередині проєкту.* Внутрішня комунікація в команді проєкту є ключовим елементом для його успішного виконання, адже від ефективності обміну інформацією між учасниками залежить координація дій та виконання завдань. Основна мета комунікаційної стратегії полягає в забезпеченні безперебійного потоку інформації між розробниками, тестувальниками, менеджерами та іншими учасниками проєкту, щоб усі були в курсі поточного статусу роботи, можливих проблем і шляхів їх вирішення.

Регулярні зустрічі команд на яких обговорюються поточні результати, виявлені проблеми та способи їх подолання. Для покращення координації також використовуються цифрові платформи, такі як Slack, Trello чи Notion, які забезпечують обмін інформацією в реальному часі.

Важливим аспектом комунікаційної стратегії є чітке визначення ролей і відповідальності кожного члена команди. Це дозволяє уникнути непорозумінь та дублювання зусиль, забезпечуючи при цьому зосередженість кожного учасника на своїх завданнях. Прозорість у прийнятті рішень також має велике значення: через спільну платформу всі учасники можуть стежити за ходом виконання завдань і планів, що допомагає уникати затримок і сприяє досягненню цілей проєкту вчасно.

*Комунікаційна стратегія* назовні орієнтована на взаємодію з користувачами та партнерами, які мають зацікавленість у функціонуванні системи. Основна мета цієї стратегії полягає в забезпеченні ефективного зворотного зв'язку з користувачами та прозорості у співпраці з партнерами.

Перш за все, важливим напрямком є зворотний зв'язок від користувачів. Завдяки системі AppReviewPro користувачі можуть оперативно отримувати відповіді на свої запитання, що підвищує їхню задоволеність продуктами і сприяє поліпшенню взаємодії між користувачами та розробниками. Такий зворотний зв'язок допомагає швидко реагувати на проблеми та покращувати якість продукту.

Важливо забезпечити прозорість і оперативність у взаємодії з партнерами, які надають додаткові сервіси або технічну підтримку. Це включає своєчасне інформування про зміни, обговорення нових функцій та покращення, а також швидке вирішення технічних питань.

Окрім цього, стратегія включає надання інформаційної підтримки. Це означає створення документації, інструкцій та інших допоміжних матеріалів для використання системи як для клієнтів, так і для сторонніх організацій.

Згідно цих даних, можна сформуванати таблицю контекстного впливу первинних зацікавлених сторін (табл. 2.1).

Таблиця 2.1.

**Вплив первинних зацікавлених сторін**

Зацікавлена сторона проекту	Інтереси зацікавленої сторони в проєкті	Оцінка впливу
Компанія-розробник	Автоматизація збору та обробки відгуків користувачів	Високий
Команда підтримки	Зниження навантаження на підтримку через автоматизацію	Середній
Маркетингова команда	Отримання аналітики для покращення маркетингових стратегій	Середній
Аналітики	Отримання структурованих даних для аналізу відгуків	Високий

Основний вплив на хід проєкту мають компанія-розробник та аналітичні команди, оскільки їхні інтереси безпосередньо пов'язані з успішною реалізацією автоматизованої системи. Команда підтримки та маркетингова команда також зацікавлені в проєкті, оскільки він сприяє оптимізації їхніх робочих процесів.

Аналогічно можна сформуванати таблицю контекстного впливу вторинних зацікавлених сторін.

Таблиця 2.2.

**Вплив вторинних зацікавлених сторін**

Зацікавлена сторона проекту	Інтереси зацікавленої сторони в проєкті	Оцінка впливу
Кінцеві користувачі	Швидке вирішення проблем через покращену взаємодію з розробниками	Середній
Власники платформ App Store та Google Play	Покращення якості додатків на їхніх платформах	Високий
Постачальники технологій та сервісів	Інтеграція їхніх продуктів в автоматизовану систему	Низький

Серед вторинних зацікавлених сторін найбільший вплив мають кінцеві користувачі, оскільки саме вони є основним джерелом зворотного зв'язку.

Власники платформ App Store та Google Play також зацікавлені в успіху проєкту через підвищення якості додатків, що публікуються на їхніх платформах.

Цю інформацію можна поєднати у матриці впливів зацікавлених сторін (Рис. 2.1).

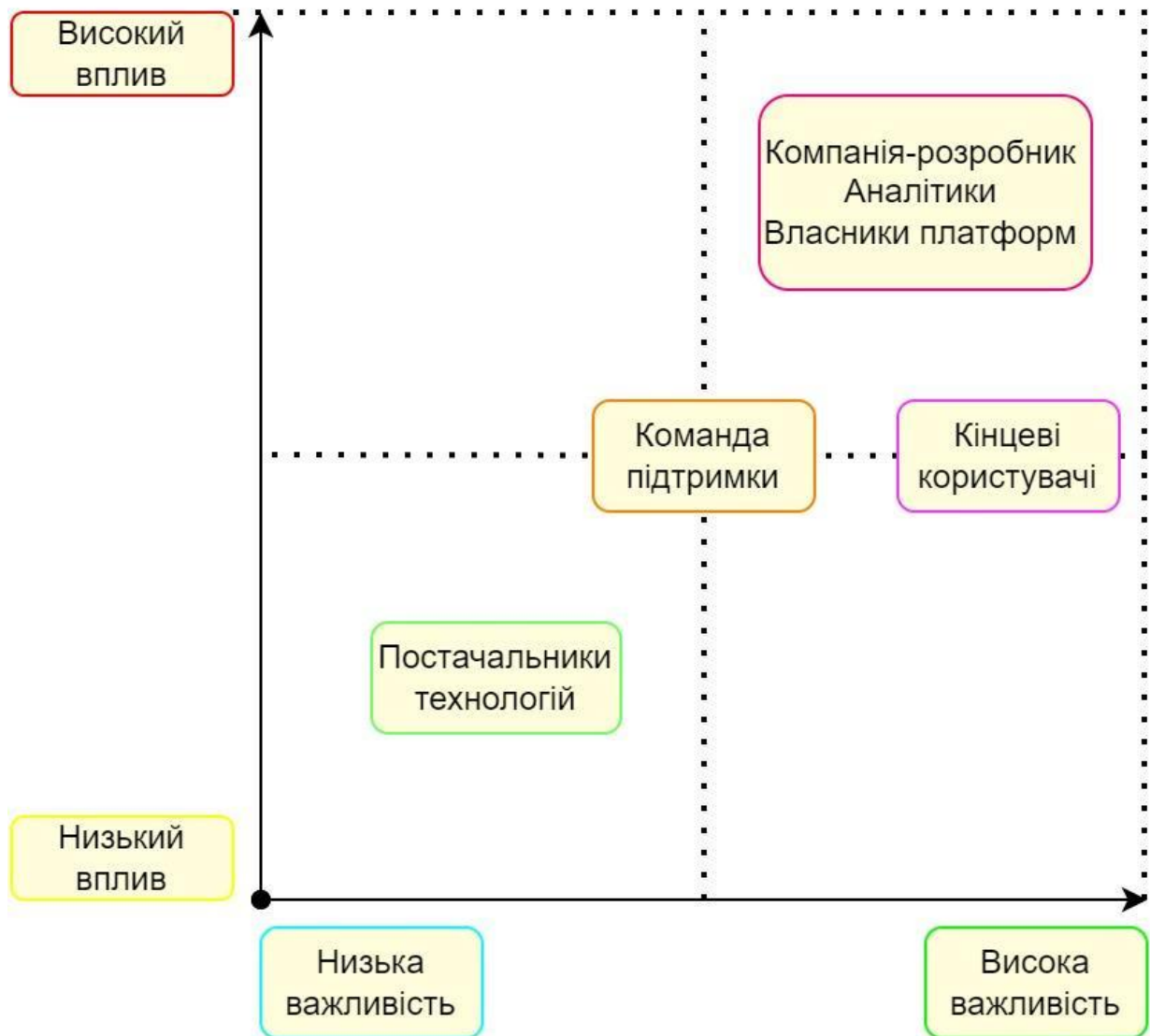


Рис. 2.1. Матриця зацікавлених сторін

Матриця показує, що найбільший вплив на проєкт мають компанія-розробник, аналітики та кінцеві користувачі, оскільки їхні інтереси найбільше пов'язані з успішною реалізацією системи [37], [38]. Власники платформ App Store і Google Play також важливі через їхню роль у забезпеченні стабільної взаємодії з додатками. Команда підтримки має середній вплив, оскільки вона

зацікавлена у зменшенні навантаження, але не є безпосереднім керівником процесу розробки.

### **2.1.3. Опис модулів інформаційної системи, що є продуктом проєкту**

Продукти проєкту є інформаційною системою, призначеною для автоматизації роботи з відгуками користувачів мобільних додатків у магазинах App Store та Google Play Market. Основним продуктом проєкту є система AppReviewPro, яка дозволяє компаніям, що розробляють мобільні додатки, ефективно збирати, аналізувати та обробляти відгуки користувачів. Система включає такі основні компоненти:

1. *Модуль збору відгуків:* цей компонент автоматично збирає відгуки користувачів із магазинів App Store і Google Play у режимі реального часу. Він дозволяє інтегрувати відгуки в єдину систему для зручного управління.

2. *Модуль аналізу відгуків:* цей компонент використовує алгоритми машинного навчання для класифікації відгуків за тональністю (позитивні, негативні, нейтральні) і виділення ключових проблем, які згадуються користувачами.

3. *Інтерфейс для комунікації з користувачами:* цей інтерфейс дозволяє командам підтримки та розробникам швидко відповідати на відгуки, використовуючи шаблони відповідей для типових питань або створюючи персоналізовані відповіді.

4. *Аналітичний модуль:* забезпечує візуалізацію даних та аналітику, що дозволяє компаніям відстежувати зміни у зворотному зв'язку з користувачами, оцінювати динаміку зміни відгуків і загальну репутацію продуктів.

5. *Модуль інтеграції з CRM та іншими інструментами:* система може бути інтегрована з існуючими інструментами компанії, такими як CRM-системи або аналітичні платформи, що дозволяє централізувати роботу з даними та підвищити ефективність обробки відгуків.

#### 2.1.4. Вимоги до продукту проєкту

Для успішного впровадження та ефективної роботи системи AppReviewPro встановлені як функціональні, так і нефункціональні вимоги до її компонентів.

##### *Функціональні вимоги:*

- Збір відгуків: система повинна мати можливість автоматично збирати відгуки з платформ App Store та Google Play у режимі реального часу.
- Аналіз відгуків: алгоритми машинного навчання повинні класифікувати відгуки за тональністю та виділяти основні проблеми, що згадуються користувачами.
- Швидка відповідь: система повинна підтримувати можливість швидкої відповіді на відгуки користувачів, включаючи шаблонні відповіді на типові питання.
- Інтеграція: система має бути інтегрована з CRM та іншими інструментами для забезпечення централізованого управління даними.

##### *Нефункціональні вимоги:*

- Масштабованість: система повинна бути здатною обробляти великі обсяги даних, забезпечуючи безперебійну роботу незалежно від кількості користувацьких відгуків.
- Продуктивність: час відповіді системи на запити не повинен перевищувати визначені ліміти для забезпечення високої продуктивності.
- Безпека: система повинна відповідати вимогам захисту персональних даних, особливо тих, що збираються з платформ App Store та Google Play. Важливо також забезпечити надійне шифрування зберігання даних та захист від несанкціонованого доступу.
- Надійність: система повинна бути доступною 24/7 та мати механізми для відновлення після збоїв, щоб забезпечити безперервну роботу та збереження даних.

Ці вимоги дозволяють забезпечити стабільну роботу системи, її інтеграцію в існуючі процеси компанії та зручне використання для кінцевих користувачів.

## 2.2. Команда проєкту

У цьому проєкті використовується залучення внутрішніх трудових ресурсів компанії для ефективного виконання задач. Кожна роль у команді відповідає за певні етапи розробки та впровадження системи управління відгуками. Це рішення дозволяє зменшити витрати на пошук та адаптацію нових працівників, використовуючи досвід і знання фахівців, які вже знайомі з потребами компанії та її продуктами [39], [40], [41], [42].

Далі розглянемо більш детально кожного члена команди, їх обов'язки та залученість у цей проєкт.

### 1. *Project Manager* та його обов'язки:

- Планування проєкту: розробляє детальний план реалізації проєкту, включаючи визначення основних етапів, задач та розподіл ролей між членами команди.
- Координація команди: забезпечує скоординовану роботу всіх учасників, проводить регулярні зустрічі для обговорення прогресу, вирішення проблем та коригування плану за потреби.
- Контроль термінів та бюджету: слідкує за дотриманням бюджету і термінів виконання кожного етапу. Забезпечує ефективне використання ресурсів.
- Комунікація зі стейкхолдерами: відповідає за взаємодію з керівництвом і стейкхолдерами, інформує їх про хід проєкту, досягнення та виклики.
- Ризик-менеджмент: визначає потенційні ризики для проєкту, розробляє стратегії для їх мінімізації, проводить оцінку впливу ризиків на проєкт.

Вимоги до кандидата:

- Вища освіта у сфері управління проєктами, інформаційних технологій або суміжних галузях.
- Досвід роботи на аналогічній посаді не менше 5 років.
- Глибокі знання методологій управління проєктами (Agile, Kanban).
- Вміння ефективно управляти командами та координація робочих процесів.
- Досвід у проведенні ризик-менеджменту та управлінні бюджетом проєкту.

Залученість Project Manager триває протягом всього проєкту, починаючи з першого дня і до завершення етапу впровадження.

2. Business Analyst та його обов'язки:

- Збір вимог від стейкхолдерів: спілкується зі стейкхолдерами, щоб зрозуміти їх потреби та очікування від системи. Створює список функціональних та нефункціональних вимог до продукту.
- Аналіз конкурентів: досліджує функціонал подібних систем для визначення можливостей та вдосконалень, які можна інтегрувати у продукт.
- Визначення специфікацій: на основі зібраних даних розробляє специфікації продукту, формує технічні вимоги до функціональних блоків системи.
- Вивчення інтеграційних можливостей: досліджує API App Store та Google Play Market, щоб забезпечити можливість зручної інтеграції системи з платформами.

Вимоги до кандидата:

- Вища освіта у сфері бізнес-аналізу, економіки або інформаційних технологій.
- Мінімум 3 роки досвіду роботи бізнес-аналітиком.

- Досвід збору та аналізу вимог, створення специфікацій.
- Знання методів аналізу конкурентів і ринку.
- Знання основ інтеграції та API.

Його активна залученість починається з перших днів і триває до етапу планування, а потім зменшується після розробки специфікацій.

### 3. UI/UX Designer та його обов'язки:

- Розробка дизайну інтерфейсу: створює макети інтерфейсу користувача, орієнтуючись на зручність, доступність та простоту використання. Забезпечує привабливий зовнішній вигляд і логічну структуру інтерфейсу.
- Прототипування: створює інтерактивні прототипи, які дозволяють команді та стейкхолдерам оцінити інтерфейс на ранніх стадіях проекту.
- Оптимізація користувацького досвіду: проводить аналіз користувацького досвіду (UX) та вносить зміни в інтерфейс для покращення взаємодії.
- Співпраця з розробниками: тісно співпрацює з Front-end і Back-end розробниками для інтеграції дизайну в код, пояснює ідеї і логіку інтерфейсу.

Вимоги до кандидата:

- Вища освіта у сфері дизайну, комп'ютерних наук або суміжних галузях.
- Досвід роботи в UI/UX дизайні не менше 2 років.
- Вміння працювати з інструментами створення макетів та прототипів (Figma, Adobe XD).
- Розуміння принципів зручності та доступності інтерфейсів.
- Досвід співпраці з розробниками для інтеграції дизайну.

Його залученість найактивніша під час розробки інтерфейсу, але дизайнер може бути залучений на пізніших етапах для внесення змін або оптимізації користувацького досвіду.

#### 4. Front-end Developers (дві особи) та їх обов'язки:

- Розробка інтерфейсу користувача: пишуть код для реалізації користувацького інтерфейсу, забезпечуючи доступ до функціональності системи для користувачів.
- Взаємодія з API: інтегрують інтерфейс з серверною частиною через API для відображення даних з бази відгуків та забезпечення зручної взаємодії з системою.
- Забезпечення адаптивності: розробляють інтерфейс з урахуванням адаптивності, щоб система працювала коректно на різних пристроях і розмірах екранів.
- Оптимізація продуктивності: оптимізують код інтерфейсу для швидкого завантаження сторінок, плавної анімації та мінімізації ресурсів, що використовуються.

Вимоги до кандидатів:

- Вища освіта у сфері комп'ютерних наук або програмування.
- Досвід роботи у Front-end розробці не менше 2-3 років.
- Знання HTML, CSS, JavaScript та сучасних фреймворків (React, Angular).
- Досвід створення адаптивних та оптимізованих веб-додатків.
- Розуміння роботи з API для інтеграції серверних даних.

Залученість Front-end Developers починається з проєктування інтерфейсу і триває до завершення написання коду, що включає близько 40 днів активної участі.

#### 5. Back-end Developer та його обов'язки:

- Проектування архітектури системи: розробляє архітектуру серверної частини системи, враховуючи функціональні блоки та їх взаємодію, щоб забезпечити стабільність і масштабованість системи.
- Створення бази даних: розробляє структуру бази даних для зберігання відгуків, даних користувачів та налаштувань системи. Забезпечує швидкий і зручний доступ до даних.
- Розробка API: створює API, через яке інтерфейс користувача може взаємодіяти з базою даних, передавати дані та отримувати результати.
- Забезпечення безпеки: реалізує механізми захисту даних користувачів та відгуків, включаючи контроль доступу та шифрування інформації.
- Оптимізація коду: постійно вдосконалює серверну частину, щоб забезпечити швидку роботу системи та ефективне використання ресурсів.

Вимоги до кандидата:

- Вища освіта у сфері інформаційних технологій або комп'ютерних наук.
- Досвід роботи у розробці серверних рішень від 3 років.
- Знання мов програмування (Java, Python, Node.js).
- Досвід роботи з базами даних (SQL, NoSQL).
- Навички забезпечення безпеки даних та оптимізації серверної продуктивності.

Залученість Back-end Developer починається на етапі проектування архітектури та триває до написання коду і тестування, що охоплює близько 70 днів.

6. QA Engineer та його обов'язки:

- Планування тестування: розробляє стратегії тестування системи, готує тест-плани і перелік тестових сценаріїв для перевірки функціональності.
- Функціональне тестування: проводить тестування кожної функції системи для забезпечення її коректної роботи згідно з вимогами.
- Навантажувальне тестування: перевіряє роботу системи під різними навантаженнями, щоб оцінити її стабільність і витривалість.
- Виявлення та фіксація помилок: Фіксує помилки, що виникають під час тестування, передає їх розробникам та повторно тестує після виправлення.
- Розробка тестових звітів: створює звіти з результатами тестування, що містять аналіз виконаних перевірок та виявлених проблем, пропонує рекомендації для оптимізації системи.

Вимоги до кандидата:

- Вища освіта у сфері інформаційних технологій або суміжних дисциплінах.
- Мінімум 2-3 роки досвіду у тестуванні програмного забезпечення.
- Знання методів функціонального, навантажувального та регресійного тестування.
- Досвід створення тест-планів та тестових сценаріїв.
- Навички документування помилок та складання звітів з тестування.

QA Engineer активно залучений на етапах тестування та оптимізації системи, що охоплює близько 34 днів його участі.

Організаційна структура проєкту (OBS, Organizational Breakdown Structure) — це схема, яка відображає взаємозв'язки між членами команди та їхні ролі в проєкті [43]. OBS допомагає структурувати команду таким чином, щоб

визначити, хто за що відповідає, та як різні учасники співпрацюють між собою. Це дозволяє чітко розподілити обов'язки та сприяє ефективному управлінню проектом, що знижує ризик непорозумінь і дублювання завдань.

На рисунку 2.2 показана OBS для проекту, де зображені основні ролі учасників та їхні взаємозв'язки.

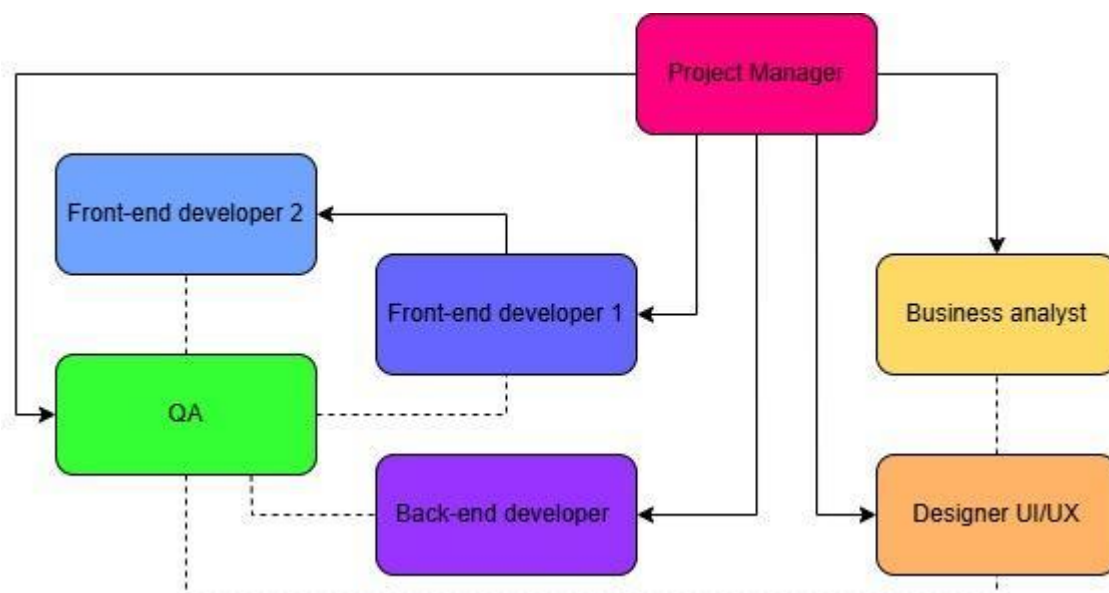


Рис. 2.2. OBS проекту

На наведеній організаційній структурі проекту (OBS) показані ролі ключових учасників команди та їхні взаємозв'язки. Це включає як функціональні ланцюги комунікації між ролями, так і співпрацю між окремими учасниками для досягнення ефективного виконання завдань. Розглянемо ролі кожного учасника та їхні взаємодії.

Project Manager (Проектний менеджер) — відображений у верхній частині структури, має основну відповідальність за управління проектом, координацію роботи всієї команди та досягнення цілей проекту. Він працює з усіма іншими членами команди для планування, моніторингу та контролю за виконанням завдань.

Business Analyst (Бізнес-аналітик) — взаємодіє з Project Manager, надаючи аналітичну підтримку, що включає збір вимог від стейкхолдерів і перетворення

їх у технічні завдання. Business Analyst також співпрацює з UI/UX дизайнером для забезпечення відповідності розробленого інтерфейсу бізнес-вимогам.

UI/UX Designer (UI/UX дизайнер) — відповідає за дизайн і користувацький інтерфейс проєкту. Співпрацює з Business Analyst для уточнення вимог та забезпечення зручності використання інтерфейсу, а також підтримує комунікацію з Front-end розробниками для передачі дизайн-макетів і впровадження їх у продукт.

Front-end Developers 1 і 2 (Фронт-енд розробники) — обидва розробники займаються реалізацією клієнтської частини проєкту. Вони працюють над впровадженням дизайну, наданого UI/UX дизайнером, та підтримують комунікацію з QA (контролем якості) для тестування готових функцій. Front-end Developers також взаємодіють з Project Manager для отримання завдань та звітування про прогрес.

Back-end Developer (Бек-енд розробник) — відповідає за серверну частину проєкту, включаючи логіку, базу даних і API, що використовуються для зв'язку з клієнтською частиною. Він співпрацює з Front-end розробниками для інтеграції функцій та забезпечення їхньої працездатності. Також активно комунікує з QA для тестування backend-функціоналу.

QA (Quality Assurance, Контроль якості) — відповідає за тестування всіх компонентів проєкту, щоб забезпечити їхню стабільність і відповідність вимогам. QA взаємодіє з усіма розробниками — як Front-end, так і Back-end — для тестування функцій, а також з Project Manager для звітування про проблеми та помилки.

## **2.3. Бюджет проєкту**

### **2.3.1. Огляд витрат проєкту**

Для реалізації проєкту зі створення системи управління відгуками передбачені кілька основних категорій витрат: трудові ресурси, витрати на обладнання, програмне забезпечення, зокрема серверні витрати для зберігання

відгуків та коментарів, а також непередбачені витрати [44]. Кожна з цих категорій є критичною для забезпечення якості проєкту, його своєчасного виконання та успішного впровадження.

Основні витрати припадають на оплату праці команди, що включає Project Manager, Front-end та Back-end розробників, UI/UX дизайнера, QA інженера та бізнес-аналітика. Кожен член команди має певну ставку та рівень зайнятості на проєкті, що формує загальну суму щомісячних витрат на трудові ресурси.

Ця частина витрат є однією з найбільших, оскільки залучення висококваліфікованих спеціалістів є необхідним для забезпечення належного виконання задач та підтримки якості проєкту на всіх етапах розробки.

Обладнання для проєкту включає сервери та простір для зберігання даних. Для забезпечення надійного зберігання всіх коментарів та відгуків користувачів передбачено використання жорсткого диска (HDD) із пропускнуою здатністю 300 MB/s і вартістю \$30/TB. Цей обсяг місця на диску є достатнім для зберігання всіх коментарів і відгуків для десяти мобільних додатків протягом усього терміну їх роботи. Низька вартість зберігання даних дозволяє підтримувати економічну ефективність проєкту в довгостроковій перспективі.

Для виконання задач розробки, тестування та підтримки проєкту необхідне програмне забезпечення, включаючи інструменти для розробки та тестування, ліцензії на IDE, бази даних та інші інструменти, необхідні для роботи над проєктом. Витрати на програмне забезпечення можуть коливатися залежно від конкретних інструментів, які будуть використовуватися, але вони є відносно стабільними і додають необхідну функціональність для кожного етапу проєкту.

Контроль бюджету є критичним аспектом управління проєктом, оскільки витрати повинні відповідати виділеним фінансовим ресурсам, щоб уникнути перевищення кошторису та забезпечити завершення проєкту в межах бюджету. Недотримання запланованих витрат може призвести до дефіциту фінансування на завершальних етапах, що вплине на якість та строки реалізації проєкту. Тому

правильний розподіл витрат та регулярний моніторинг бюджету допомагають зменшити фінансові ризики та забезпечити стабільність фінансування на всіх етапах розробки.

### **2.3.2. Витрати на трудові ресурси**

Розрахунок витрат на трудові ресурси передбачає залучення фахівців на основних позиціях, включаючи Project Manager, Business Analyst, Front-end Developer, Back-end Developer, UI/UX Designer та QA Engineer. Витрати визначаються виходячи з погодинної ставки для кожної ролі та її зайнятості протягом певного етапу проєкту. Нижче надається детальний розподіл, обчислення витрат по місяцях та таблиця витрат на кожного фахівця. Витрати на кожного фахівця можна перевірити за середніми зарплатами на сайтах Work.ua або SalaryExpert [45], [46].

На основі даних про середню заробітну плату для кожної спеціалізації в Україні, орієнтовні погодинні ставки виглядають наступним чином:

- Project Manager: \$12-\$18 на годину, для досвідчених спеціалістів, згідно з середньою зарплатою PM в Україні
- Front-end Developer: \$10-\$16 на годину для початкового рівня, залежно від досвіду, згідно з середньою зарплатою розробників в Україні
- Back-end Developer: \$11-\$18 на годину за даними того ж ресурсу, що відображає поточну ринкову ситуацію.
- UI/UX Designer: \$8-\$13 на годину, з урахуванням часткової зайнятості та середніх зарплат в Україні.
- QA Engineer: \$6-\$10 на годину для часткової зайнятості, згідно з актуальними ставками для тестувальників.
- Business Analyst: \$9-\$14 на годину.

Кожен член команди бере участь у проєкті на різних етапах, і їхня зайнятість варіюється відповідно до потреб.

Project Manager: залучений повністю протягом усього проєкту, працює на повній зайнятості 100%, що складає від \$1920 до \$2880 на місяць.

Front-end Developers: один працює на повній зайнятості, інший — на 50% упродовж 40 днів, тобто з 1 по 4 місяці. Це складає від \$1600 до \$3200.

Back-end Developer: 100% зайнятість на основних етапах розробки, що тривають від 1 до 6 місяця, з середньою місячною оплатою \$1840-\$2880.

UI/UX Designer: залучений частково, на 50% від загального часу проєкту, приблизно \$640-\$1040 щомісяця з 1 по 4 місяці.

QA Engineer: на 50% з 3 по 6 місяці на тестуванні та оптимізації, що дає витрати \$960-\$1600.

Business Analyst: часткова зайнятість (25%), залучений переважно на етапах дослідження і планування. Витрати складають \$360-\$560 на місяць.

Зведемо ці дані у таблицю 2.3.

*Таблиця 2.3*

**Таблиця витрат на трудові ресурси для перших 6 місяців роботи над проєктом**

Місяць	Project Manager	Business Analyst	Front-end Developer 1 (100%)	Front-end Developer 2 (50%)	Back-end Developer	UI/UX Designer (50%)	QA Engineer (50%)
1	\$1920-\$2880	\$360-\$560	\$1600-\$2560	\$800-\$1280	\$1840-\$2880	\$640-\$1040	-
2	\$1920-\$2880	\$360-\$560	\$1600-\$2560	\$800-\$1280	\$1840-\$2880	\$640-\$1040	-
3	\$1920-\$2880	-	\$1600-\$2560	\$800-\$1280	\$1840-\$2880	\$640-\$1040	\$960-\$1600
4	\$1920-\$2880	-	\$1600-\$2560	\$800-\$1280	\$1840-\$2880	\$640-\$1040	\$960-\$1600
5	\$1920-\$2880	-	-	-	\$1840-\$2880	-	\$960-\$1600
6	\$1920-\$2880	-	-	-	\$1840-\$2880	-	\$960-\$1600

Для етапів з 7 по 24 місяці витрати на трудові ресурси значно зменшуються, оскільки проєкт переходить у фазу підтримки та мінімального розвитку. На цьому етапі задіяні лише ключові фахівці, і їхня зайнятість обмежується частковим залученням. Наведемо розподіл витрат для цих місяців.

З огляду на зменшену кількість задач, часткова зайнятість залучених спеціалістів виглядає наступним чином.

- Project Manager: Зайнятість зменшується до 25% для контролю та координації поточної підтримки. Витрати складають \$480-\$720 на місяць.
- Front-end Developer: Залучення скорочується до 25% для можливого внесення змін та доопрацювань інтерфейсу. Місячні витрати: \$400-\$640.
- Back-end Developer: Зайнятість також знижується до 25%, що забезпечує підтримку серверної частини. Місячні витрати: \$460-\$720.
- UI/UX Designer: Працює лише при необхідності, приблизно 10% від часу для періодичних оновлень інтерфейсу. Витрати: \$160 на місяць.
- QA Engineer: Часткова зайнятість (25%) для періодичних тестувань після оновлень. Витрати складають \$240-\$400 на місяць.
- Business Analyst: Мінімальна залученість, близько 10% для моніторингу та періодичних аналітичних перевірок. Витрати: \$90-\$140 на місяць.

Загалом, витрати на підтримку проекту у 7-24 місяці суттєво нижчі і складають приблизно \$1830-\$2760 на місяць.

### **2.3.3. Матеріальні та технічні витрати**

Для успішної реалізації проекту система управління відгуками вимагає відповідного обладнання та програмного забезпечення. Матеріальні витрати поділяються на серверне забезпечення для зберігання та обробки відгуків користувачів, а також ліцензії для інструментів розробки, тестування і проєктного управління.

Оскільки в цьому проєкті використовується залучення внутрішніх трудових ресурсів компанії, всі співробітники вже мають робочі ПК з необхідними операційними системами, тому немає необхідності включати витрати на комп'ютерне обладнання до бюджету проєкту. Це оптимізує бюджет і дозволяє зосередити фінансування на спеціалізованому програмному забезпеченні та технічних ресурсах, необхідних для реалізації системи управління відгуками.

*Серверні витрати.* Основна потреба в серверних потужностях – зберігання коментарів та відгуків з додатків. Передбачається використання HDD для зберігання даних з пропускнуою здатністю 300 MB/s та вартістю \$30 за 1 ТБ. За оцінками, цього обсягу місця достатньо для зберігання коментарів за весь термін життя проєкту (10 додатків). Витрати на зберігання становитимуть близько \$30 щомісяця.

*Інструменти для розробки.* Для розробки Front-end і Back-end частин проєкту обрано Visual Studio Professional, яке надає універсальні можливості для створення багатокomпонентних систем. Цей інструмент підтримує основні мови та фреймворки, такі як JavaScript, TypeScript, C#, .NET, а також інтеграцію з базами даних. Це забезпечує ефективну реалізацію як інтерфейсу, так і серверної частини. Щомісячна вартість підписки на Visual Studio Professional становить \$45 на користувача, що робить його доступним і функціональним рішенням для розробників проєкту [47].

*Інструменти для тестування та автоматизації.* Для забезпечення якості проєкту передбачено інструменти тестування, які допоможуть виявити помилки та оптимізувати продуктивність. Щомісячні витрати на тестувальні фреймворки оцінюються приблизно в \$50 [48].

*Система управління проєктами.* Для управління проєктом обирається Jira, яка забезпечує функції для спільної роботи, відстеження задач та підтримки Agile-методології. Згідно з планом для команди з 7 осіб, вартість становить близько \$52.71 щомісяця на плані Standard [49].

Серверні витрати забезпечують усі потреби проєкту в зберіганні даних, а використання онлайн-сервісів для розробки та тестування знімає необхідність у придбанні фізичного обладнання. Придбання ліцензій на програмне забезпечення також проводиться за підпискою, що дозволяє зберігати гнучкість та уникати великих одноразових витрат.

Ці дані зведено у таблиці 2.4.

Таблиця матеріальних витрат на кожен місяць (1-6 та 7-24 місяці)

Місяці	Серверне зберігання	Інструменти для розробки	Інструменти для тестування	Система управління проєктами (Jira)	Загальні витрати
1-6 місяці	\$30	\$45	\$50	\$52.71	\$177.71
7-24 місяці	\$30	\$45	\$50	\$52.71	\$177.71

Загалом, матеріальні витрати на кожен місяць проєкту становлять \$177.71 і залишаються постійними протягом всього терміну реалізації та підтримки.

### 2.3.4. Непередбачені витрати та резерв бюджету

Проєкти з розробки програмного забезпечення часто стикаються з непередбачуваними витратами, які можуть виникнути через затримки, додаткову потребу в ресурсах або необхідність у розширеному тестуванні. Для забезпечення фінансової стабільності проєкту виділяється резервний бюджет, який допоможе покрити можливі витрати.

З огляду на складність проєкту, доцільно виділити резервний бюджет у розмірі 10-15% від загального бюджету [50], [51]. Такий відсоток дозволяє мати фінансовий буфер для покриття непередбачених витрат без суттєвого збільшення загальних витрат [52], [53].

Непередбачені витрати можуть виникати з різних причин:

- Затримки у виконанні задач: можливі затримки можуть бути пов'язані з труднощами в інтеграції компонентів, складністю тестування або недостатньою координацією між командами. Такі затримки можуть збільшити витрати на трудові ресурси.
- Додаткові ресурси: проєкт може вимагати додаткового залучення спеціалістів для вирішення технічних проблем, впровадження нового функціоналу або оптимізації системи.
- Додаткове тестування: під час реалізації проєкту можуть виникнути додаткові потреби у тестуванні, особливо якщо виявляються неочікувані

помилки або збої, що потребують тривалого тестування, зокрема навантажувального або регресійного.

Резервний бюджет відіграє ключову роль у підтримці стабільності проєкту, оскільки він допомагає команді уникнути фінансового тиску в разі непередбачених витрат. Наявність резерву гарантує безперервність робочого процесу, дозволяючи проєкту тривати без затримок чи збоїв, які можуть виникнути через нестачу фінансування. Завдяки фінансовій гнучкості резерву команда може оперативнo реагувати на нові потреби проєкту без необхідності перерозподіляти основний бюджет. Крім того, резервний бюджет сприяє збереженню якості проєкту, оскільки додаткові ресурси можна спрямувати на підтримку й оптимізацію процесів, що важливо для досягнення кінцевих цілей. Таким чином, резерв у розмірі 10-15% від загального бюджету стає невід'ємним елементом для успішної реалізації проєкту, забезпечуючи фінансову стійкість і стабільність на всіх етапах виконання.

### **2.3.5. Підсумковий бюджет проєкту**

Проєкт зі створення системи управління відгуками для мобільних додатків потребує детального планування бюджету, який враховує всі основні категорії витрат на кожному етапі реалізації. У цьому розділі представлено зведену таблицю загальних витрат проєкту з місячною розбивкою, що дає змогу чітко відстежувати структуру витрат. Також наведено аналіз прогнозу прибутковості, який показує, коли проєкт зможе досягти окупності та які фінансові перспективи можуть відкритись після завершення етапів розробки.

Зведена таблиця охоплює витрати на трудові ресурси, програмне забезпечення, серверне зберігання та резервний бюджет. Таблиця структурована за місяцями, показуючи як базові витрати на перші 6 місяців, так і значно зменшені витрати на 7-24 місяці (період підтримки проєкту). Такий підхід забезпечує повну прозорість у розподілі коштів та полегшує аналіз фінансових показників проєкту.

Загальний бюджет проєкту представлено у вигляді таблиці 2.5.

Таблиця 2.5

**Зведена таблиця загальних витрат проєкту**

Місяць	Трудові ресурси (мін.-макс.)	Програмне забезпечення	Серверне зберігання	Резерв (10%)	Загальні витрати (мін.-макс.)
1	\$7,160 - \$12,200	\$177.71	\$30	\$736.77 - \$1,240	\$8,104.48 - \$13,647.71
2	\$7,160 - \$12,200	\$177.71	\$30	\$736.77 - \$1,240	\$8,104.48 - \$13,647.71
3	\$7,760 - \$12,040	\$177.71	\$30	\$796.77 - \$1,205	\$8,764.48 - \$13,452.71
4	\$7,760 - \$12,040	\$177.71	\$30	\$796.77 - \$1,205	\$8,764.48 - \$13,452.71
5	\$4,720 - \$7,360	\$177.71	\$30	\$492.77 - \$736	\$5,420.48 - \$8,303.71
6	\$4,720 - \$7,360	\$177.71	\$30	\$492.77 - \$736	\$5,420.48 - \$8,303.71
7 - 24	\$4,320 - \$6,880	\$177.71	\$30	\$452.77 - \$688	\$4,980.48 - \$7,775.71
<b>Загальний бюджет на 24 місяці</b>					<b>\$134,227.52 - \$210,771.04</b>

Отже, загальний бюджет проєкту можна визначити у межах від \$134,227.52 до \$210,771.04.

Перші шість місяців витрати становлять від \$49,559 до \$81,573, що покриває весь цикл розробки та тестування системи. Створення внутрішньої системи управління відгуками дозволить компанії уникнути витрат на сторонні сервіси, які зазвичай коштують \$200-\$1,000 на місяць на одного користувача, залежно від функціоналу. За період у 24 місяці такі сервіси могли б обійтись компанії в \$4,800 - \$24,000, що робить розробку власної системи економічно виправданою на довгостроковій основі.

Після завершення етапу розробки (перші шість місяців) система переходить у фазу підтримки з меншими щомісячними витратами — від \$4,980.48 до \$7,775.71. При виході продукту на зовнішній ринок як комерційного рішення прогнозується середній дохід у \$10,000 на місяць з можливістю зростання на 5% щомісяця завдяки залученню нових користувачів.

З огляду на щомісячні витрати та очікуваний дохід, проєкт може досягти точки беззбитковості на 18-20 місяці з моменту початку продажів на ринку. При

цьому, після окупності система може почати приносити стабільний прибуток, що додатково підвищує інвестиційну привабливість проєкту.

### **2.3. Архітектура інформаційної системи продукту проєкту**

Архітектура інформаційної системи AppReviewPro є багаторівневою, що забезпечує ефективне управління відгуками користувачів, їх аналіз і взаємодію з різними платформами. Основні компоненти архітектури можна умовно поділити на три рівні: представницький рівень (front-end), рівень бізнес-логіки (back-end) та рівень даних (data layer).

#### *1. Представницький рівень (Front-end)*

Цей рівень відповідає за взаємодію кінцевих користувачів із системою. Його основна функція — надання інтерфейсу для доступу до основних функцій системи, таких як перегляд відгуків, аналіз результатів, та управління коментарями. Ключовими елементами цього рівня є:

- Веб-додаток: призначений для співробітників компанії, що використовують систему для управління відгуками та аналізу зворотного зв'язку. Веб-додаток дозволяє отримувати доступ до даних у зручному інтерфейсі, працювати з аналітичними інструментами та реагувати на відгуки користувачів.
- Мобільний додаток: Забезпечує доступ до функцій системи на мобільних пристроях, що є важливим для користувачів, які працюють у динамічному середовищі або не мають постійного доступу до настільних комп'ютерів.

#### *2. Рівень бізнес-логіки (Back-end)*

Цей рівень забезпечує основні функції системи, такі як збір, обробка і аналіз даних, а також управління зворотним зв'язком. Він включає кілька важливих компонентів:

- Модуль збору відгуків: автоматично отримує відгуки з магазинів App Store та Google Play за допомогою API. Він забезпечує безперервний процес збору інформації в реальному часі, яка потім передається для обробки.

- Модуль аналізу відгуків: використовує алгоритми машинного навчання для аналізу тональності відгуків і виділення ключових проблем, що згадуються користувачами. Алгоритми обробляють велику кількість текстових даних і автоматично класифікують їх за різними параметрами.

- Інтерфейс для комунікації з користувачами: дозволяє користувачам системи реагувати на відгуки, створювати шаблони відповідей для типових ситуацій і автоматично надсилати відповіді на коментарі користувачів.

- Модуль інтеграції: забезпечує інтеграцію з іншими корпоративними інструментами, такими як CRM-системи, аналітичним модулем та іншими зовнішніми сервісами для покращення управління відгуками та даними.

### *3. Рівень даних (Data layer)*

Цей рівень відповідає за зберігання та управління всіма даними, які надходять до системи або генеруються нею. Він включає:

- База даних: центральне сховище для всіх зібраних відгуків, результатів аналізу та іншої інформації, необхідної для роботи системи. База даних підтримує швидкий доступ до даних для забезпечення безперебійної роботи системи.

- Аналітичний модуль: містить агреговані дані для створення звітів та проведення аналітики. Він дозволяє швидко формувати аналітичні звіти та візуалізувати результати аналізу відгуків.

- Система резервного копіювання: забезпечує збереження копій усіх даних для їх відновлення у випадку технічних збоїв або інших проблем.

### *4. Безпека та контроль доступу*

Окремим елементом архітектури є механізми безпеки, що забезпечують захист даних користувачів. Основними елементами безпеки є:

- Аутентифікація та авторизація: система використовує методи захисту, такі як OAuth 2.0, для перевірки та контролю доступу користувачів. Це гарантує, що доступ до чутливих даних мають тільки авторизовані особи.

- Шифрування даних: всі дані, що передаються між користувачами і системою, шифруються для забезпечення конфіденційності та захисту від несанкціонованого доступу.

- Модуль аудиту: відстежує всі дії, що виконуються в системі, та створює журнали для моніторингу дотримання правил безпеки.

Архітектура інформаційної системи AppReviewPro забезпечує гнучку, надійну та масштабовану платформу для управління відгуками користувачів. Вона включає рівні представницького інтерфейсу, бізнес-логіки та даних, а також забезпечує високий рівень безпеки. Така архітектура дозволяє ефективно обробляти великі обсяги інформації, інтегруватися з іншими системами та забезпечити користувачам зручний доступ до необхідних функцій.

## **2.4. Структура бази даних продукту проєкту**

Розробка бази даних для системи AppReviewPro включає створення концептуальної та логічної моделей для зберігання та обробки даних, що пов'язані з мобільними додатками, відгуками користувачів, і взаємодією з іншими компонентами системи.

### **2.4.1. Концептуальна модель бази даних**

Концептуальна модель бази даних визначає основні сутності та їхні атрибути, що використовуються в системі AppReviewPro. Вона описує, які дані необхідні для роботи системи, але не вказує технічні деталі реалізації [54], [55]. Основними сутностями для цього проєкту є:

- Apps (Додатки): ця сутність містить інформацію про мобільні додатки, для яких збираються відгуки. Серед атрибутів сутності: назва додатку, платформа (iOS чи Android), кількість відгуків, рейтинг, версія додатку, та розробник.

- Users (Користувачі): сутність представляє всіх користувачів системи, включаючи розробників додатків, аналітиків та менеджерів підтримки. Основні

атрибути включають ім'я, прізвище, роль у системі, електронну пошту та дату останньої активності.

- **Comments (Коментарі):** ця сутність представляє відгуки користувачів, які збираються із платформ App Store та Google Play. Атрибути включають текст коментаря, дату створення, дату відповіді, мову коментаря, версію додатку, та статус обробки.

- **Templates (Шаблони відповідей):** шаблони використовуються для автоматизації відповідей на типові коментарі. Сутність включає текст шаблону, мову, дату створення, та категорію запиту.

- **Integrations (Інтеграції):** ця сутність відображає інтеграції системи з іншими сервісами, такими як CRM, маркетингові інструменти та системи аналітики. Атрибути включають назву сервісу, тип інтеграції та статус підключення.

- **Tags (Теги):** ця сутність використовується для маркування відгуків і коментарів, що дозволяє згрупувати їх за категоріями для подальшого аналізу. Атрибути включають назву тегу та його опис.

- **Reply Data (Дані відповіді):** містить інформацію про відповіді на коментарі, включаючи текст відповіді, дату відповіді та мову коментаря. Ця сутність пов'язана із Comments та Templates.

- **Rating (Оцінки):** сутність для відображення оцінок, наданих користувачами додатком. Атрибути включають оцінку у вигляді числа (1-5), текст відгуку та відповідний додаток.

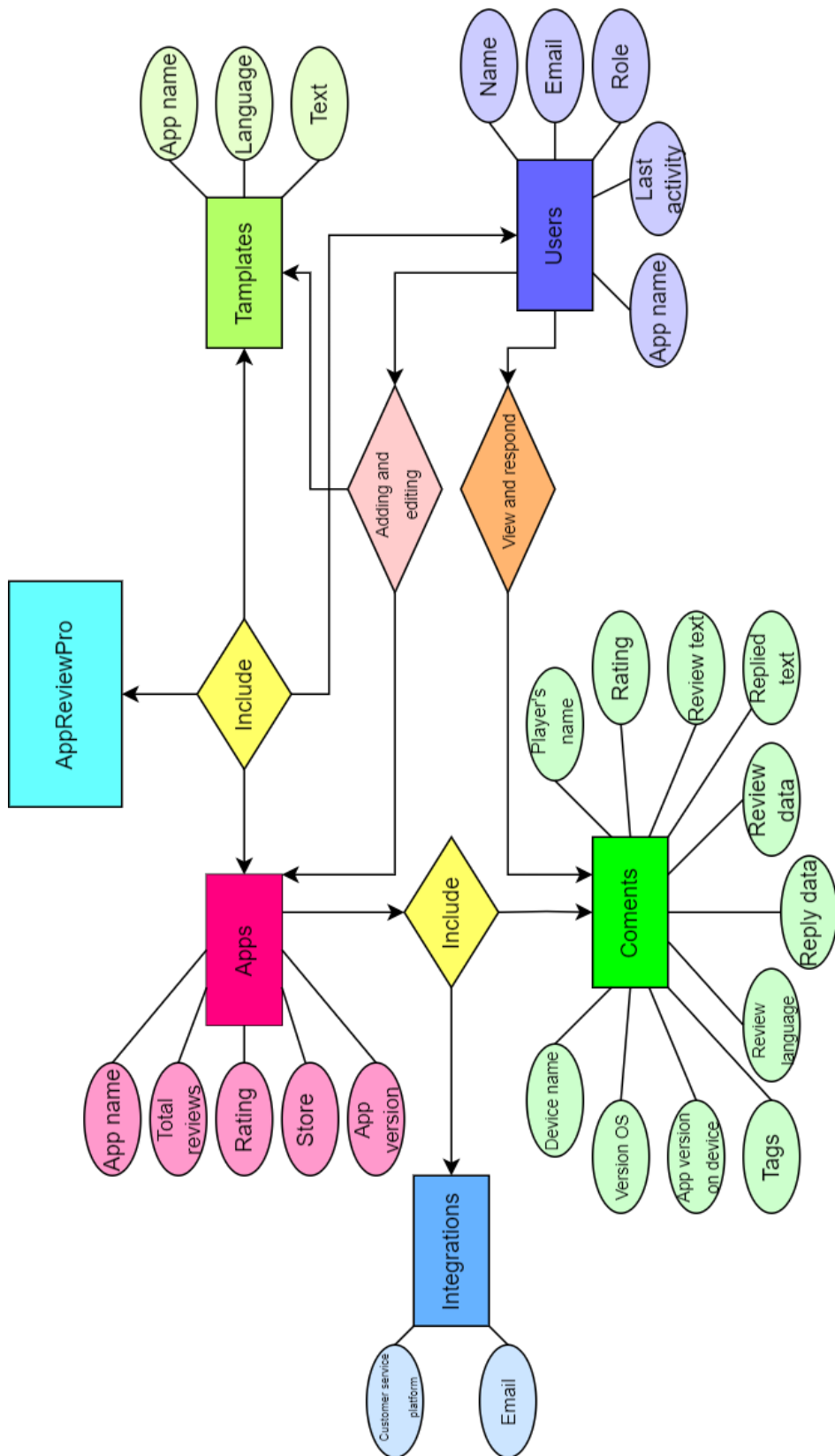


Рис. 2.3 Концептуальна модель бази даних

## 2.4.2. Логічна модель бази даних

Логічна модель бази даних деталізує концептуальну модель, додаючи до неї структуру таблиць, поля та зв'язки між ними [56], [57], [58].

Основні таблиці та їх ключі включають:

- Таблиця Apps: містить інформацію про мобільні додатки.

Основними ключами є:

- AppID — унікальний ідентифікатор додатку;
- AppVersion — номер версії додатку.

- Таблиця Users: містить дані про користувачів системи. Основний

ключ:

- UserID — унікальний ідентифікатор користувача.

- Таблиця Comments: зберігає інформацію про відгуки користувачів.

Основні ключі:

- CommentID — унікальний ідентифікатор коментаря;
- AppID — зовнішній ключ, який посилається на додаток, до якого належить коментар;
- UserID — зовнішній ключ, який пов'язує коментар із користувачем, що його залишив.

- Таблиця Templates: зберігає шаблони відповідей на коментарі.

Основний ключ:

- TemplateID — унікальний ідентифікатор шаблону;
- AppID — зовнішній ключ для прив'язки шаблону до конкретного додатку.

● Таблиця Integrations: зберігає інформацію про інтеграції з іншими сервісами. Основний ключ:

- IntegrationID — унікальний ідентифікатор інтеграції;
- AppID — зовнішній ключ, що пов'язує інтеграцію з додатком.

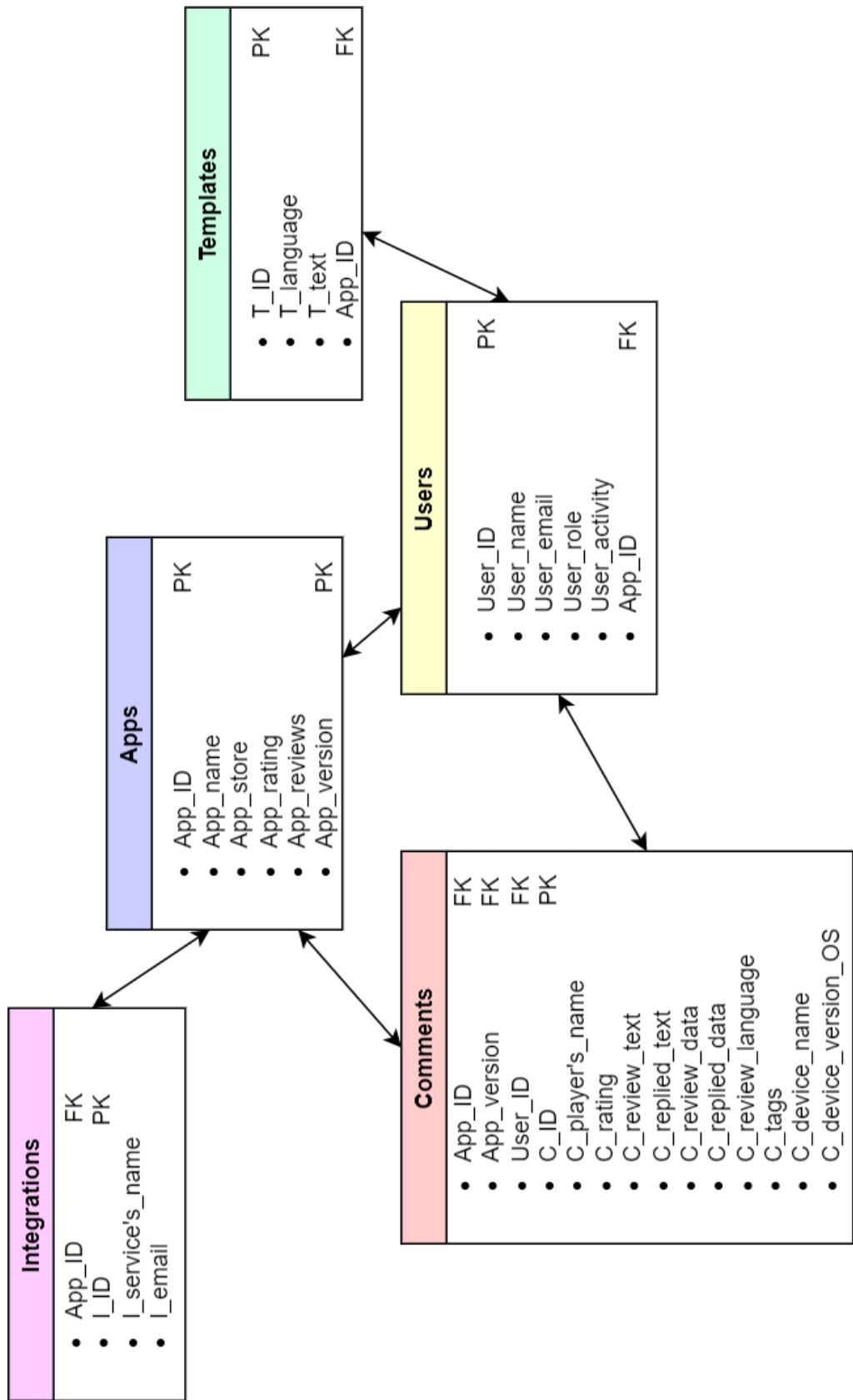


Рис. 2.4. Логічна модель бази даних

### 2.4.3. Зв'язки між сутностями у базі даних

Зв'язки між сутностями бази даних визначають взаємодію між різними компонентами системи AppReviewPro. Основні типи зв'язків між таблицями:

- *"Один-до-багатьох" між Apps і Comments:* один додаток може мати багато коментарів. Це забезпечується через зовнішній ключ AppID у таблиці Comments.
- *"Один-до-багатьох" між Users і Comments:* один користувач може залишати багато коментарів. Зв'язок встановлюється через зовнішній ключ UserID у таблиці Comments.
- *"Один-до-багатьох" між Apps і Templates:* один додаток може мати багато шаблонів для відповідей на коментарі. Цей зв'язок забезпечується через зовнішній ключ AppID у таблиці Templates.
- *"Один-до-багатьох" між Apps і Integrations:* один додаток може мати багато інтеграцій з іншими сервісами. Цей зв'язок встановлюється через зовнішній ключ AppID у таблиці Integrations.

Завдяки цим зв'язкам база даних забезпечує цілісність даних, дозволяючи зберігати, обробляти і зв'язувати велику кількість інформації про відгуки, додатки та користувачів.

### 2.4.4. Технічні особливості бази даних

Для забезпечення безперебійної та продуктивної роботи системи AppReviewPro, база даних повинна відповідати певним технічним вимогам. Основними вимогами є:

1. *Масштабованість:* оскільки кількість відгуків користувачів, які надходять через магазини App Store та Google Play, може бути великою, важливо, щоб база даних була здатна до масштабування. Це дозволить обробляти великі обсяги даних без втрат продуктивності, що досягається за рахунок використання розподілених систем управління базами даних, наприклад, NoSQL баз даних (MongoDB) або реляційних баз даних з масштабованими сервісами (PostgreSQL).

2. *Швидкий доступ до даних*: час відповіді на запити до бази даних має бути мінімальним, навіть при великій кількості користувачів та одночасних запитах. Для цього важливе використання індексів на ключових полях (наприклад, AppID, CommentID, UserID) для прискорення пошуку даних.

3. *Відмовостійкість та надійність*: База даних повинна бути захищена від збоїв і втрати даних. Для цього необхідно передбачити регулярне резервне копіювання даних і використання кластерних рішень для мінімізації простоїв і втрати інформації у разі збоїв.

4. *Безпека даних*: оскільки система працює з персональними даними користувачів та коментарями, важливо забезпечити надійне шифрування даних як під час їх зберігання, так і при передачі через інтернет. Це можна реалізувати за допомогою SSL/TLS для передачі даних та AES для зберігання.

5. *Інтеграція з зовнішніми системами*: для забезпечення можливості надсилання звітів та збирання аналітичних даних база даних повинна підтримувати інтеграцію з зовнішніми аналітичними сервісами, такими як Google Analytics або Firebase, що дозволяє проводити більш глибокий аналіз поведінки користувачів і результативності продуктів.

6. *Резервне копіювання*: для захисту від втрати даних, важливо налаштувати регулярне резервне копіювання бази даних, що забезпечить можливість відновлення після аварій або збоїв. Це стосується як щоденних автоматичних резервних копій, так і можливості відновлення даних за попередні періоди.

#### **2.4.5. Оптимізація продуктивності бази даних**

Для забезпечення високої продуктивності бази даних системи AppReviewPro необхідно застосовувати кілька важливих методів оптимізації. Одним із ключових підходів є використання індексації на критично важливих полях, таких як AppID, CommentID та UserID. Індексация дозволяє значно

прискорити пошук і фільтрацію даних, що особливо важливо при роботі з великими обсягами інформації.

Ще одним ефективним методом є кешування даних. За допомогою технологій кешування, таких як Redis або Memcached, можна зберігати часто запитувані дані в оперативній пам'яті, що знижує навантаження на базу даних і пришвидшує час відповіді на запити.

У випадках, коли обсяг даних стає занадто великим для одного сервера, можна використовувати шардінг — горизонтальне розділення даних на кілька серверів. Цей підхід дозволяє розподілити навантаження між кількома базами даних, забезпечуючи вищу продуктивність та відмовостійкість системи.

Баланс між гнучкістю структури даних та швидкістю виконання запитів можна підтримувати за рахунок нормалізації даних для уникнення надлишковості, а також денормалізації у випадках, коли це допомагає зменшити кількість об'єднань таблиць та покращити швидкість виконання запитів.

Таким чином, для забезпечення оптимальної продуктивності бази даних необхідно використовувати індексацію, кешування, шардінг, а також комбінувати нормалізацію та денормалізацію даних залежно від специфіки запитів і потреб системи. Це забезпечить швидку та надійну роботу бази даних навіть при великих навантаженнях.

## **РОЗДІЛ 3. УПРАВЛІННЯ ЗМІСТОМ ТА ПРОДУКТОМ ПРОЄКТУ. ІНФОРМАЦІЙНІ РЕСУРСИ В УПРАВЛІННІ ПРОЄКТОМ**

### **3.1. Управління змістом проєкту. WBS і роботи проєкту**

Управління змістом проєкту охоплює всі процеси, що дозволяють чітко визначити і контролювати обсяг робіт, необхідний для досягнення цілей проєкту, зокрема створення інформаційної системи “AppReviewPro”. Це управління забезпечує узгодженість між початковими вимогами замовників, специфікаціями системи та кінцевими результатами, зменшуючи ризик відхилень від затвердженого плану. Основною метою управління змістом є організація процесів таким чином, щоб забезпечити виконання робіт в межах визначених ресурсів, строків і обсягу.

Перший етап управління змістом – планування змісту проєкту, яке включає аналіз та фіксацію основних вимог до системи. Це визначення є критичним для розуміння того, що саме потрібно створити і яких результатів очікують від проєкту. На цьому етапі уточнюються технічні та функціональні характеристики майбутньої системи, перелік основних функцій та вимог до її продуктивності, а також особливості, які повинні задовольняти кінцевих користувачів і забезпечити комфортну роботу з відгуками з App Store і Google Play Market. Планування змісту допомагає визначити і формалізувати цілі проєкту, створюючи чіткі критерії для оцінки успішності його реалізації.

Після планування змісту проєкту виконується процес створення ієрархічної структури робіт (WBS), що дозволяє представити проєкт у вигляді системи взаємопов'язаних завдань. Ієрархічна структура робіт є інструментом, який розбиває загальний обсяг робіт на менші компоненти, що підлягають виконанню, формуючи основну структуру планування і контролю. Кожен рівень WBS деталізує роботу, яка необхідна для реалізації системи, а також дає можливість ефективніше розподілити відповідальність за виконання конкретних завдань між членами команди [59], [60]. Зокрема, у проєкті AppReviewPro це

може включати розробку компонентів, таких як інтерфейс користувача, модуль аналізу відгуків, інтеграційні рішення з іншими системами тощо. Використання WBS дозволяє послідовно відстежувати прогрес виконання робіт і вчасно виявляти можливі проблеми у виконанні запланованих завдань [61], [62].

Завершальним етапом є контроль змісту проєкту, що забезпечує відповідність виконаних робіт початково визначеним вимогам та специфікаціям. Цей процес передбачає регулярну перевірку та оцінку виконаних завдань, аналіз їх відповідності плану та, при необхідності, внесення коригувань. Контроль змісту допомагає запобігти розширенню обсягу робіт, яке не було передбачене початковими домовленостями і може призвести до перевищення бюджету або затримки завершення проєкту. Для цього у проєкті AppReviewPro передбачаються регулярні ревію виконаних робіт, порівняння з очікуваними результатами та визначення відповідності поточного прогресу з графіком реалізації проєкту.

Загалом, управління змістом проєкту є основоположним процесом, який дозволяє створити чітку структуру завдань, забезпечити прозорість і передбачуваність у роботі команди та гарантувати, що всі зусилля будуть спрямовані на досягнення визначених цілей у встановлені строки та з оптимальним використанням ресурсів.

Далі опишемо поняття WBS, його роль у проєктному менеджменті, рівні деталізації для проєкту “AppReviewPro”, а також основні етапи та компоненти з ієрархічною структурою.

WBS (Work Breakdown Structure) — це ієрархічна структура, яка поділяє проєкт на менші, керовані завдання або компоненти, що полегшують організацію, планування і контроль робіт. Структура WBS допомагає чітко визначити обсяг проєкту, розподілити відповідальність за виконання конкретних завдань, а також забезпечити узгодженість між командою та цілями проєкту. Для проєкту AppReviewPro, WBS є важливим інструментом, який дозволяє

деталізувати усі основні роботи, зменшуючи ризик пропущених завдань і сприяючи ефективному розподілу ресурсів.

Рівні деталізації WBS для проєкту AppReviewPro забезпечують розподіл проєкту на чіткі та керовані завдання, що дозволяє команді ефективно контролювати процес реалізації на кожному з етапів. Для AppReviewPro структура WBS має багаторівневу ієрархію, де кожен рівень деталізації дає змогу розбити складні процеси на зрозумілі компоненти.

Згідно з WBS, структура деталізації для проєкту AppReviewPro може бути описана на чотирьох рівнях, де кожен рівень надає додаткову деталізацію для кожного етапу, підзадач і операційних дій. Ось детальний опис кожного рівня з акцентом на різні етапи реалізації цього проєкту.

#### *Рівень 1: Основні фази проєкту*

Цей рівень є найзагальнішим і охоплює основні фази життєвого циклу проєкту AppReviewPro. Він складається з чотирьох головних компонентів.

Управління проєктом - включає організаційні аспекти, планування та контроль над виконанням проєкту.

Розробка програмного забезпечення - охоплює всі етапи створення функціонального продукту, від аналізу вимог до тестування та впровадження.

Оптимізація та розвиток - пов'язана з вдосконаленням, внесенням змін та адаптацією продукту під нові вимоги.

Впровадження та підтримка - фаза, яка забезпечує розгортання продукту та його технічну підтримку після випуску.

#### *Рівень 2: Ключові підфази*

На другому рівні кожна з основних фаз проєкту деталізується на підфази, які охоплюють більш конкретні етапи роботи.

Управління проєктом поділяється на наступні елементи:

- Планування проєкту - розробка плану проєкту, який включає графік робіт і визначення ключових ресурсів.

- Визначення потреб та обговорення вимог - збір і аналіз вимог для точного визначення функцій та цілей системи.
- Організація команди - розподіл ролей та обов'язків серед членів команди.
- Створення робочого графіку - визначення часових рамок для кожного етапу та підетапу.
- Моніторинг та контроль виконання - регулярний контроль за виконанням завдань для забезпечення відповідності плану.

Розробка програмного забезпечення включає такі підфази:

- Аналіз вимог - вивчення потреб користувачів та специфікацій, які повинна виконувати система.
- Проєктування архітектури системи - визначення основних архітектурних рішень, які забезпечують стабільність і функціональність системи.
- Розробка функціональності системи - створення компонентів системи, включаючи систему збору даних, аналітичну систему, інтерфейс користувача, систему зберігання даних та інтеграцію з магазинами додатків.
- Тестування та якість - забезпечення якості та стабільності через планування тестування, виконання тестів та виправлення помилок.
- Реліз та розгортання - випуск системи в робоче середовище та її впровадження для кінцевих користувачів.

Оптимізація та розвиток охоплює наступні підфази:

- Вдосконалення функціоналу - аналіз результатів використання системи та внесення покращень.
- Внесення змін для покращення роботи системи - адаптація системи під нові вимоги та зовнішні зміни.
- Оптимізація інтерфейсу та функціоналу - удосконалення зручності та ефективності використання системи відповідно до оновлень і змін у вимогах.

Впровадження та підтримка містить дві основні підфази:

- Впровадження в експлуатацію - процес розгортання системи для кінцевих користувачів.

- Технічна підтримка та оновлення - забезпечення безперервної підтримки та адаптації системи після її впровадження.

### *Рівень 3: Завдання та підзавдання*

Цей рівень включає конкретні завдання, які входять до кожної підфази. Для кожної підфази визначено набір основних завдань.

Для Управління проєктом: планування етапів проєкту, проведення зустрічей для уточнення вимог, контроль виконання проміжних завдань.

Для Розробки програмного забезпечення: наприклад, в системі збору даних включено збір та обробку відгуків користувачів, у аналітичній системі - налаштування та аналіз статистичних даних.

Для Оптимізації та розвитку: завдання, спрямовані на покращення зручності інтерфейсу та продуктивності функціоналу.

Для Впровадження та підтримки: налагодження технічної підтримки, підготовка документації та адаптація системи під нові оновлення.

### *Рівень 4: Оперативні дії*

На цьому рівні детально розписуються окремі дії, які здійснюються в рамках кожного завдання. Це можуть бути конкретні кроки для реалізації окремих функцій системи або технічні процеси, такі як тестування, налагодження, документування.

Управління проєктом: наприклад, проведення регулярних зустрічей для оцінки статусу виконання завдань, коригування плану за потреби.

Розробка програмного забезпечення: виконання модульного тестування, перевірка на відповідність вимогам.

Оптимізація та розвиток: збір відгуків від користувачів, аналіз їхньої задоволеності, внесення коригувань в інтерфейс та функціонал.

Впровадження та підтримка: встановлення оновлень, підтримка системи після запуску, оперативне виправлення проблем.

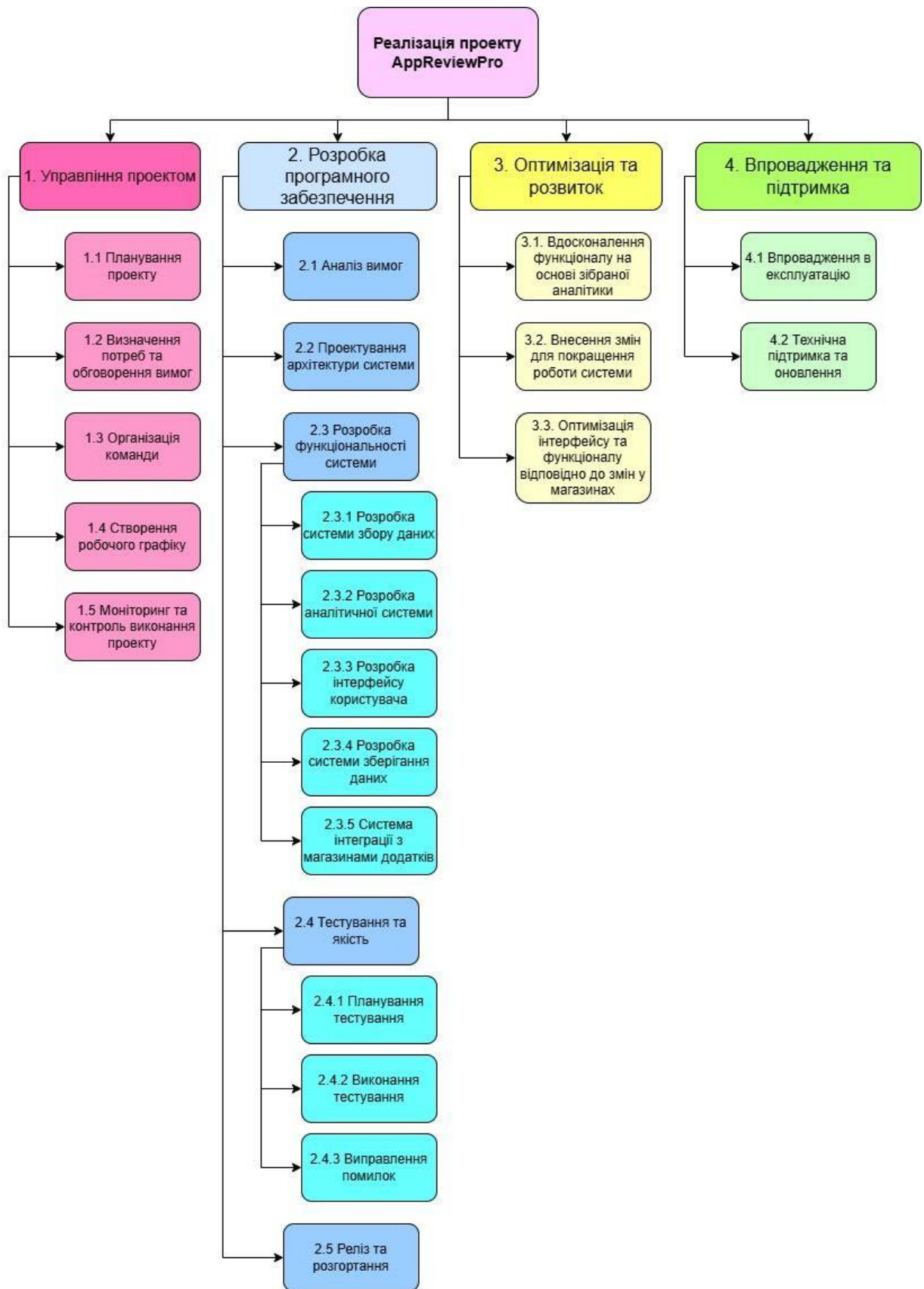


Рис. 3.1. WBS проекту AppReviewPro

Ідентифікація ключових задач для проєкту AppReviewPro є важливим етапом, оскільки дозволяє визначити задачі, від яких залежить успішність реалізації системи. Ключові задачі виділяються через їх критичний вплив на основні функції, які система повинна забезпечувати для користувачів і партнерів. До них належать етапи, пов'язані з проєктуванням, розробкою, тестуванням та впровадженням компонентів системи, які забезпечують її ефективне функціонування.

Однією із перших важливих задач є аналіз вимог. Ця задача впливає на всю систему, оскільки саме на його основі формується розуміння потреб користувачів, а також визначаються функціональні й технічні вимоги до системи. Якість проведення цієї задачі напряду впливає на те, наскільки система відповідатиме очікуванням кінцевих користувачів та її інтеграцію в бізнес-процеси партнерів.

Наступною ключовою задачею є проєктування архітектури системи. Архітектура визначає структуру компонентів системи та їх взаємодію, що є критично важливим для забезпечення надійності, масштабованості й можливості подальшого розширення системи. Від правильного проєктування архітектури залежить, наскільки легко система зможе підтримувати великі обсяги даних та надавати швидкий доступ до інформації користувачам.

Розробка функціональних модулів, зокрема, модулів збору та обробки даних, також є ключовою задачею. Модулі збору даних дозволяють системі взаємодіяти з магазинами додатків, отримувати інформацію про відгуки користувачів та інші дані, що є основою для роботи AppReviewPro. Модулі обробки даних, у свою чергу, забезпечують аналіз інформації, що дозволяє користувачам отримувати аналітичні інсайти.

Не менш важливим є етап тестування системи, який включає перевірку працездатності всіх компонентів. Ця задача має значний вплив на загальну

реалізацію проєкту, оскільки дозволяє виявити й виправити помилки до моменту впровадження, що мінімізує ризики збоїв у роботі системи після запуску.

Завершальною ключовою задачею є впровадження системи в експлуатацію, що включає налаштування серверів, підготовку середовища для роботи та навчання користувачів. Впровадження визначає, наскільки швидко й ефективно система зможе почати функціонувати в реальних умовах.

Методика пріоритезації задач для проєкту AppReviewPro ґрунтується на двох основних критеріях: важливість і терміновість. Ці критерії дозволяють ефективно розподілити ресурси й зосередити увагу команди на тих задачах, які мають найбільший вплив на успіх проєкту та його своєчасне виконання.

*Критерій важливості* визначає, наскільки конкретна задача впливає на основні цілі проєкту та якість кінцевого продукту. Задачі, які мають критичний вплив на реалізацію ключових функцій системи або без яких система не зможе функціонувати належним чином, отримують вищий пріоритет.

*Критерій терміновості* враховує часові обмеження, а також строки, необхідні для виконання задачі. Задачі з короткими дедлайнами або ті, що необхідні для подальшого виконання інших етапів проєкту, мають вищу терміновість.

Для проєкту AppReviewPro завдання можна розподілити за матрицею важливості та терміновості наступним чином:

*Висока важливість та висока терміновість* — завдання, які потрібно виконати негайно, оскільки вони критично впливають на успіх проєкту та мають жорсткі часові обмеження.

- **Аналіз вимог:** необхідно виконати на початковому етапі, щоб забезпечити точність у розумінні функцій та можливостей, які має надати система.

- Проєктування архітектури системи: цей етап є важливим для побудови надійної структури, яка підтримуватиме всі компоненти та модулі AppReviewPro.

- Розробка модуля збору даних: оскільки цей модуль є ключовим для отримання інформації з магазинів додатків, його потрібно впровадити одним із перших.

- Базове тестування модулів (особливо модулів збору та обробки даних): виявлення помилок на ранньому етапі дозволить запобігти значним ризикам.

*Висока важливість та низька терміновість* — завдання, які мають значний вплив на проєкт, але не мають жорстких часових рамок. Їх потрібно виконати після термінових завдань.

- Вдосконалення аналітичного модуля: після завершення основних функцій системи його можна адаптувати для кращого аналізу та відображення відгуків.

- Оптимізація інтерфейсу користувача: покращення зручності роботи із системою збільшує користувацький досвід, але може бути виконане після базової реалізації функціоналу.

- Документування процесів та підготовка матеріалів для навчання користувачів: необхідно для забезпечення підтримки та навчання, але не є критичним на ранніх етапах.

*Низька важливість та висока терміновість* — завдання, що мають бути виконані швидко, але менш критичні для кінцевого результату. Їх можна доручити менш завантаженим членам команди або виконувати паралельно з важливішими завданнями.

- Налаштування інфраструктури для розробки та тестування: це забезпечить безперебійний робочий процес і комфорт для розробників.

- Підготовка середовища тестування та прототипування: дозволяє розробникам і тестувальникам швидко проводити перевірки.

*Низька важливість та низька терміновість* — завдання, що можуть бути відкладені або виконані після всіх інших. Такі завдання не є критичними й можуть виконуватися у випадку наявності додаткового часу.

- Дизайн та доопрацювання візуальних деталей інтерфейсу: можна провести вже після того, як система функціонує стабільно.

- Внесення незначних змін та вдосконалень на основі додаткових вимог: це може виконуватися, коли всі критичні функції вже реалізовані.

- Проведення додаткових користувацьких тестів для збору фідбеку після основного запуску: корисно для подальшого вдосконалення, але не є критичним для початкового запуску системи.

### **3.2. Вигляд фінального продукту**

В цьому підрозділі представлено опис фінального продукту системи управління відгуками, розробленої для ефективного моніторингу, аналізу та взаємодії з користувацькими відгуками мобільних додатків. Метою є продемонструвати, як інтерактивні елементи інтерфейсу та функціональні можливості системи сприяють досягненню головних цілей проекту: підвищення ефективності роботи команди підтримки, швидкого реагування на відгуки, поліпшення взаємодії з користувачами та підтримки високої репутації додатків на ринку.

Фінальний продукт включає зручний інтерфейс користувача з розширеними функціями аналітики, автоматизовані механізми обробки відгуків та інтеграційні можливості для роботи з платформами App Store та Google Play Market. Це забезпечує цілісний процес управління відгуками, спрощуючи роботу та зменшуючи час на виконання рутинних завдань, що в результаті підвищує загальну продуктивність команди та якість обслуговування клієнтів.

Система управління відгуками для мобільних додатків надає користувачам розширений набір функцій, що забезпечують зручний та ефективний процес обробки відгуків. Нижче описані основні функціональні можливості системи, які допомагають досягти високого рівня автоматизації та підвищення продуктивності.

1. *Збір та аналіз відгуків*: система автоматично збирає відгуки з різних джерел, зокрема з платформ App Store та Google Play Market, і зберігає їх у єдиній базі даних. Це дозволяє користувачам бачити всі коментарі до одного додатку, незалежно від того, на якій платформі вони були опубліковані. Завдяки цій функції, зменшується час на пошук та обробку відгуків вручну. Аналіз відгуків включає визначення їхньої тональності (позитивної, негативної чи нейтральної) та виявлення ключових тенденцій у відгуках користувачів. Це дає змогу швидко оцінити настрої користувачів і реагувати на них відповідно.

2. *Автоматизація відповідей*: система включає модуль для створення шаблонних відповідей, які можна використовувати для швидкого реагування на типові коментарі. Це особливо важливо для підтримки високої швидкості обробки відгуків та зменшення навантаження на команду підтримки. Автоматизація відповідей дозволяє налаштувати автоматичну відправку повідомлень у відповідь на певні типи відгуків або використовувати готові шаблони для напівручного управління відповідями.

3. *Побудова звітів*: окрім збору та аналізу, система має функціонал для створення детальних звітів про обробку відгуків. Це дозволяє командам оцінювати ефективність своєї роботи, відстежувати прогрес у взаємодії з користувачами та покращувати процеси на основі отриманих даних. Звіти можна налаштовувати за різними параметрами, такими як період, тип відгуків чи ефективність відповіді.

4. *Інтеграції з App Store та Google Play Market*: Система підтримує інтеграцію з App Store та Google Play Market, що є основною функцією для об'єднання відгуків з різних платформ у єдиній базі. Це дозволяє розробникам і

менеджерам зручно переглядати всі відгуки про один додаток в одному місці, аналізувати їх та забезпечувати послідовну комунікацію з користувачами. Інтеграції забезпечують можливість автоматичного імпорту відгуків, що значно спрощує процес моніторингу зворотного зв'язку.

*5. Можливості роботи з даними:* Система пропонує розширені функції для роботи з відгуками, включаючи фільтрацію, сортування та аналіз. Користувачі можуть швидко фільтрувати відгуки за різними параметрами, такими як тональність, дата публікації, ключові слова, країна та інше. Сортування допомагає впорядковувати відгуки за важливістю або іншими критеріями, що полегшує управління великими обсягами інформації. Завдяки цим можливостям користувачі можуть легко знаходити релевантні дані та працювати з ними, покращуючи процес взаємодії з користувачами та прийняття рішень.

Інтерфейс системи управління відгуками розроблений з урахуванням зручності використання та інтуїтивної зрозумілості для користувачів. Система містить кілька основних модулів, які забезпечують ефективний доступ до функціональності та підтримку управління відгуками на високому рівні.

Основні елементи інтерфейсу включають:

- *Панель керування (початковий екран):* цей модуль надає користувачу оглядову інформацію про загальний стан відгуків, їх кількість, середній рейтинг та статус відповідей. На панелі відображаються ключові метрики, такі як середня оцінка відгуків та кількість відповідних коментарів.

- *Модуль аналізу відгуків:* користувач може переглядати детальну інформацію про відгуки за різні періоди часу та здійснювати аналіз їхньої тональності. Це дозволяє ідентифікувати позитивні, нейтральні та негативні відгуки, оцінюючи загальний настрій користувачів.

- *Шаблони для відповідей:* інтегрований модуль шаблонів дозволяє створювати та використовувати попередньо підготовлені відповіді для швидкої

обробки відгуків. Це значно знижує час, необхідний для відповіді на типові питання та коментарі.

- *Робоча зона для відповідей:* у цьому модулі відображаються конкретні відгуки, на які потребують відповіді, разом із додатковою інформацією про автора, країну, версію додатка тощо. У цьому розділі користувачі можуть використовувати шаблони, редагувати відповіді та надсилати їх.

- *Фільтри та сортування:* у системі передбачені зручні інструменти фільтрації та сортування відгуків за рейтингом, датою, країною, та іншими параметрами, що дозволяє швидко знаходити потрібні дані та працювати з ними.

На рисунку 3.2 відображено інтерфейс панелі керування системою управління відгуками. Ця панель є початковим екраном, який надає користувачу оглядову інформацію про стан додатків та ключові метрики. Вона дозволяє користувачу бачити список додатків, які знаходяться під моніторингом, а також основні показники, такі як загальний рейтинг додатків, кількість відгуків та інших ключових метрик. З лівого боку панелі розташовані навігаційні елементи, що забезпечують швидкий доступ до основних розділів системи, включаючи аналіз відгуків, відповіді на коментарі, аналіз рейтингів та шаблони відповідей.

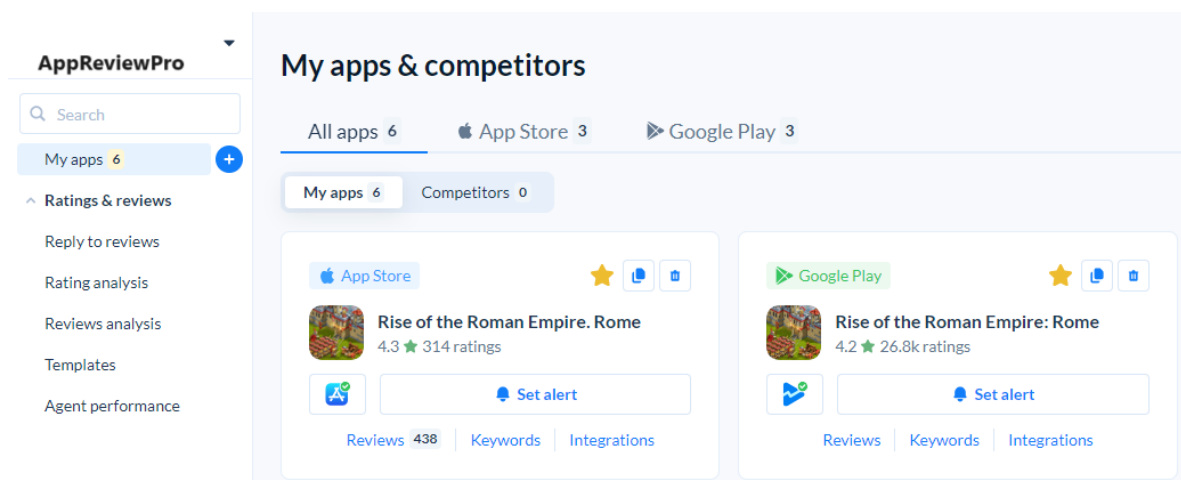


Рис. 3.2. Панель керування застосунку (початковий екран)

На рисунку 3.3 представлено Модуль аналізу відгуків, який є важливою складовою системи управління відгуками. Цей модуль забезпечує користувачам

можливість аналізувати відгуки за різними показниками, що допомагає приймати обґрунтовані рішення для покращення продукту.

У верхній частині зображення знаходиться огляд ключових метрик для аналізованого додатку, таких як загальна кількість відгуків, середній рейтинг відгуків та розподіл відгуків за оцінками (кількості зірок). Ця інформація дозволяє оцінити поточний стан користувацького сприйняття додатку.

На рисунку 3.3 представлена діаграма розподілу відгуків за оцінками на певний період часу. Ця діаграма допомагає візуалізувати зміни в оцінках користувачів, що дозволяє визначати дні з найвищими чи найнижчими оцінками та аналізувати причини таких змін.

Модуль також включає функції фільтрації за різними параметрами, такими як дата, мова відгуків, рейтинг тощо. Це спрощує аналіз великих обсягів даних та дозволяє зосередитися на релевантних відгуках, які потребують більш детального вивчення або реагування.

На рисунках А.1 та А.2 з Додатку А, відображено Модуль шаблонів відповідей, який дозволяє створювати, редагувати та організовувати шаблони відповідей на відгуки користувачів. Цей модуль допомагає підвищити ефективність роботи команди підтримки, забезпечуючи швидкі та узгоджені відповіді на повторювані запити.

Рисунок А.1 демонструє організацію шаблонів у вигляді папок, що надає можливість структурувати відповіді за різними категоріями або мовами. Наприклад, у системі можна створювати папки для різних ігор або типів запитів, що значно полегшує навігацію та використання шаблонів.

На рисунку А.2 показано процес створення нового шаблону відповіді, де користувач може ввести назву шаблону, текст відповіді, вибрати додатки, для яких цей шаблон буде використовуватись, та додати теги для швидкого пошуку. Це дозволяє створювати стандартні відповіді, що використовуються для частих запитів від користувачів.

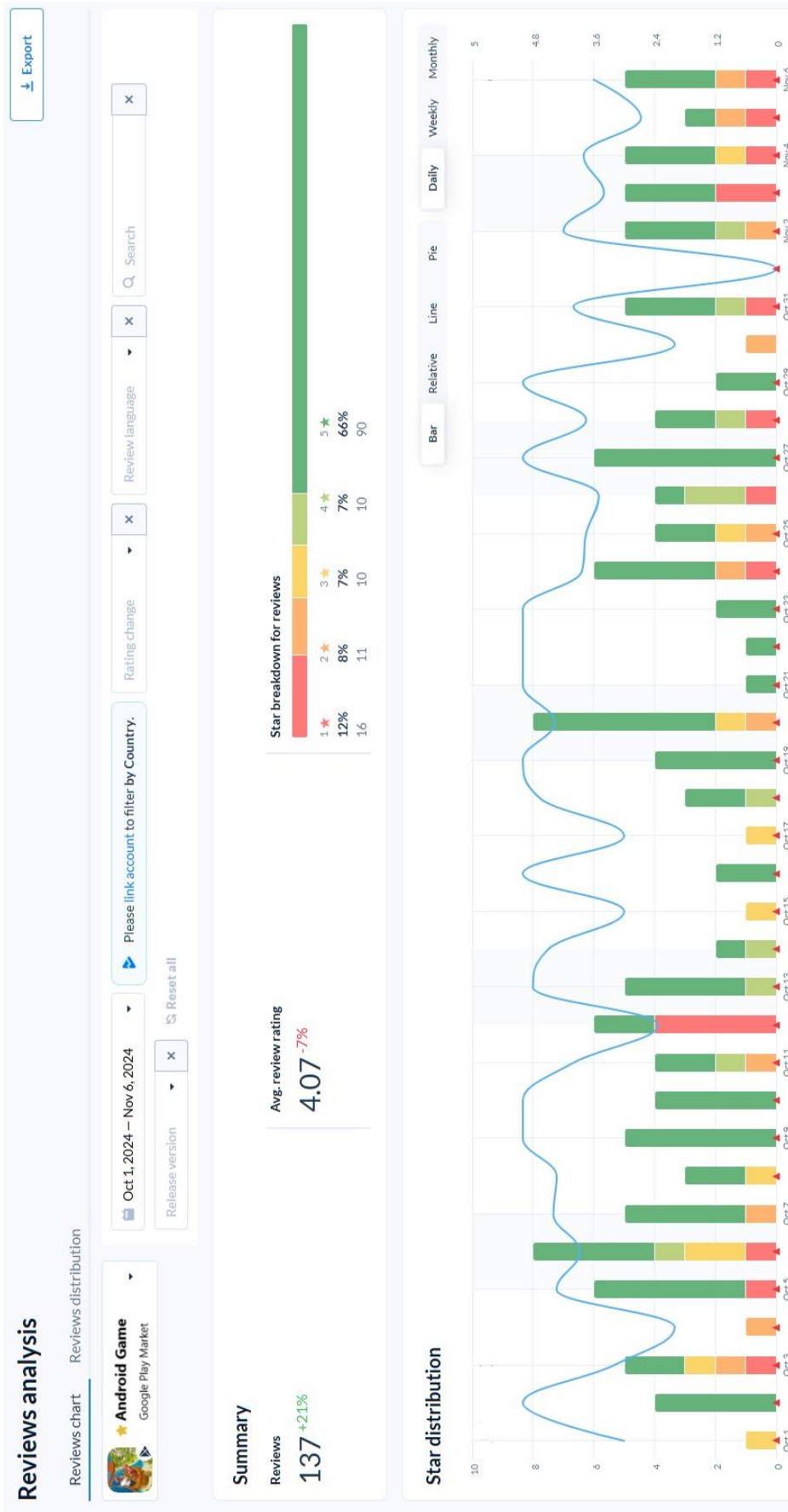


Рис. 3.3 Модуль аналізу відгуків

Модуль також підтримує можливість імпорту та експорту шаблонів, що робить його зручним для використання у великих командах або при масштабуванні роботи. Загалом, цей інструмент забезпечує стандартизацію відповідей, економію часу та підвищення якості обслуговування користувачів.

На рисунку А.3 з Додатку А, представлена сторінка перегляду відгуків, яка демонструє можливості роботи з відгуками користувачів та управління відповідями. Користувач має доступ до функціональних елементів, що включають панель з відгуками, де відображаються коментарі з детальною інформацією про пристрій, мову, версію додатку та інші дані, важливі для аналізу.

Також на зображенні видно поле для введення відповіді на відгук та можливість вибору шаблонів відповідей, що значно полегшує процес комунікації з користувачами. Ця функція підтримує створення індивідуальних або автоматизованих відповідей, що сприяє оперативному реагуванню. Додатково користувач може використовувати теги для класифікації відгуків, що полегшує їхню організацію та аналіз.

На рисунку А.4. з Додатку А представлена панель фільтрації та сортування, яка використовується для перегляду та обробки відгуків користувачів. Дана панель дозволяє користувачам вибирати період, за який вони хочуть переглянути відгуки (наприклад, останні 7 днів, 30 днів, 90 днів або індивідуально заданий період). Також присутня можливість фільтрації відгуків за рейтингом, що дає змогу виділяти відгуки з певними оцінками (наприклад, від 1 до 5 зірок), мовою написання відгуку та типом відповіді (наприклад, без відповіді або старі відповіді).

Крім цього, у панелі передбачено поле для пошуку тексту в межах відгуків і вибору тегів для подальшої сортування. Це забезпечує зручність в управлінні відгуками та полегшує процес знаходження необхідної інформації, що сприяє більш ефективній роботі зі зворотним зв'язком і підвищенню якості обслуговування користувачів.

На зображенні А.5. з Додатку А, представлена панель звітності, яка демонструє підсумкові показники аналізу відгуків для ігор, розміщених на платформах Google Play Market та App Store. У верхній частині інтерфейсу відображаються налаштування фільтрів для вибору періоду аналізу та географічної локалізації відгуків.

Основні блоки інформації включають розділ "Резюме рейтингу", де надано середні оцінки для кожної гри, а також порівняння з ринковим показником. Це дозволяє оцінити, як додатки виглядають на фоні конкурентів за кількістю та середнім рейтингом відгуків.

Розділ "Метрики зворотного зв'язку" містить детальні показники, такі як кількість нових та загальних відгуків, частота оновлення відгуків, ефективність відповідей та середній час відповіді на коментарі. Ці метрики допомагають розробникам аналізувати якість взаємодії з користувачами, визначати, як швидко команда реагує на коментарі та наскільки ефективні відповіді у покращенні оцінки додатку.

### **3.3. Управління часом проєкту. Діаграма Ганта**

Управління часом є ключовим аспектом ефективного реалізації будь-якого проєкту, адже від нього залежить своєчасне виконання завдань, узгодження робіт між командами та загальна успішність проєкту. Часовий контроль дозволяє уникнути затримок, оптимізувати використання ресурсів та забезпечити дотримання дедлайнів. У цьому контексті діаграма Ганта виступає одним із найбільш ефективних інструментів для управління графіком робіт. Вона дозволяє наочно відобразити послідовність та тривалість завдань, побачити взаємозв'язки між ними та своєчасно виявляти критичні точки проєкту. Завдяки своїй простоті та візуальній інформативності, діаграма Ганта допомагає координувати процеси, виявляти можливі ризики та своєчасно ухвалювати управлінські рішення для підтримки стабільного прогресу.

Процес створення діаграми Ганта розпочинається з ідентифікації основних етапів проєкту та їх деталізації на підзадачі з відповідними тривалістю та логічними зв'язками між ними [63]. На основі даних, отриманих із файлів користувача, можна виділити ключові завдання проєкту, їхню тривалість та залежності між етапами. Діаграма дозволяє візуально відстежувати взаємозв'язки між завданнями, а також визначити, які з них можуть виконуватись паралельно, а які залежать від завершення попередніх етапів [64], [65].

Для створення діаграми використовувався інструмент Jira, який забезпечує зручний інтерфейс для побудови графіків і управління проєктом. Jira дозволяє не тільки створювати детальну діаграму Ганта, але й зберігати гнучкість у процесі планування, вносячи зміни в залежності від потреб проєкту.

Проєкт розробки системи управління відгуками складається з кількох основних етапів, кожен з яких має свою тривалість та взаємозв'язки з іншими етапами. Загальна структура проєкту включає етапи від дослідження та аналізу до впровадження та підтримки.

Опис кожного етапу та його взаємозв'язок з іншими етапами:

- *Дослідження та аналіз* : включає збір вимог від стейкхолдерів, аналіз конкурентних систем та вивчення можливостей API. Цей етап є критичним для подальшого планування і визначає цілі проєкту.

- *Планування та ресурси*: включає визначення мети та обсягу проєкту, розподіл ресурсів, оцінку бюджету та інші підготовчі завдання. Цей етап залежить від результатів дослідження і є основою для етапу розробки.

- *Розробка та проєктування* : включає проєктування архітектури системи, розробку бази даних та інтерфейсу користувача. Цей етап розпочинається після оцінки тривалості задач та планування.

- *Реалізація та тестування* : включає написання коду, тестування системи та виправлення помилок. Етап тестування розпочинається після завершення основного написання коду, а оптимізація та виправлення помилок відбувається паралельно.

- *Впровадження та підтримка* : завершальний етап включає впровадження системи та навчання персоналу користуванню.

Усі задачі проєкту відображені у таблиці 3.1. Вона містить перелік основних задач проєкту з вказанням їхньої тривалості та зв'язків, які визначають послідовність виконання етапів роботи.

Для опису взаємозв'язків між задачами використовуються такі типи зв'язків:

- FS (Finish-to-Start) — залежність типу «закінчення-початок», яка означає, що наступна задача не може початися, поки не завершиться попередня. Цей тип зв'язку найбільш поширений і визначає послідовне виконання задач.

- SS (Start-to-Start) — залежність «початок-початок», яка означає, що дві задачі можуть початися одночасно, але можуть виконуватися паралельно. Цей тип зв'язку дозволяє задачам, що взаємодіють, починатися одночасно.

- FF (Finish-to-Finish) — залежність «закінчення-закінчення», яка вказує, що дві задачі мають завершитися приблизно в один і той самий час, хоча початок задач може різнитися.

Крім того, у таблиці зазначаються затримки або випередження для кожного з типів зв'язків, які впливають на графік виконання задач. Затримки вказуються у днях і позначають необхідність відкласти початок або завершення наступної задачі, а випередження — дозволяють розпочати задачу раніше.

Критичний шлях є важливим елементом управління проєктом, оскільки він визначає мінімальний час, необхідний для завершення всіх завдань. У рамках цього проєкту, що охоплює розробку системи управління відгуками, критичний шлях включає завдання, які не можуть бути відкладені без впливу на загальний термін виконання проєкту [66].

Таблиця 3.1

## Найменування задач, їх тривалість та зв'язки з іншими задачами

№	Попередня задача	Тип зв'язку	Затримка (випередження / запізнення)	№	Наступна задача та її тривалість
	Початок (08.01.2025)	SS / relates to		1.1.	Збір вимог від стейкхолдерів (1 день)
1.1.	Збір вимог від стейкхолдерів (1 день)	SS / relates to		1.2.	Аналіз функціональності конкуруючих систем (1 день)
1.2.	Аналіз функціональності конкуруючих систем (1 день)	FS / is blocked by		1.3.	Вивчення API та можливостей інтеграції з App Store та Google Play Market (1 день)
1.3.	Вивчення API та можливостей інтеграції з App Store та Google Play Market (1 день)	FS / is blocked by	+1	2.1.	Визначення мети та обсягу проєкту. (1 день)
2.1.	Визначення мети та обсягу проєкту. (1 день)	SS / relates to		2.2.	Визначення потреб у людських ресурсах, обладнанні та програмному забезпеченні (2 дні)
2.2.	Визначення потреб у людських ресурсах, обладнанні та програмному забезпеченні (2 дні)	FS / is caused by		2.3.	Оцінка бюджету проєкту та розподіл ресурсів (2 дні)
1.1.	Збір вимог від стейкхолдерів (1 день)	FS / is blocked by	+2	2.4.	Розробка плану комунікації зі стейкхолдерами (1 день)
2.3.	Оцінка бюджету проєкту та розподіл ресурсів (2 дні)	SS / relates to	+1	2.5.	Розробка специфікацій вимог до програмного забезпечення (2 дні)

Продовження таблиці 3.1

2.5.	Розробка специфікацій вимог до програмного забезпечення (2 дні)	SS / relates to		2.6.	Визначення функціональних блоків та їх взаємодії (2 дні)
2.6.	Визначення функціональних блоків та їх взаємодії (2 дні)	FS / is blocked by	1	2.7.	Розподіл завдань (2 дні)
2.7.	Розподіл завдань (2 дні)	SS / relates to	1	2.8.	Оцінка тривалості окремих етапів та задач (1 день)
2.8.	Оцінка тривалості окремих етапів та задач (1 день)	FS / is caused by		3.1.	Проектування архітектури системи (20 днів)
2.8.	Оцінка тривалості окремих етапів та задач (1 день)	FS / is caused by		3.2.	Розробка бази даних для зберігання коментарів та відгуків (20 днів)
2.8.	Оцінка тривалості окремих етапів та задач (1 день)	FS / is caused by		3.3.	Розробка інтерфейсу користувача для спілкування з коментарями (10 днів)
3.1.	Проектування архітектури системи (20 днів)	FS / is caused by	-1	4.1.	Написання програмного коду для функціональності системи (30 днів)
4.1.	Написання програмного коду для функціональності системи (30 днів)	FF / is caused by	-5	4.2.	Тестування роботи системи з різними мобільними застосунками (20 днів)
4.2.	Тестування роботи системи з різними мобільними застосунками (20 днів)	FF / is caused by	-5	4.3.	Виправлення помилок та оптимізація роботи системи (14 днів)
4.3.	Виправлення помилок та оптимізація роботи системи (14 днів)	FS / is blocked by		5.1.	Впровадження системи в роботу (1 день)
5.1.	Впровадження системи в роботу (1 день)	SS / relates to		5.2.	Навчання персоналу користуватись системою (1 день)
5.2.	Навчання персоналу користуватись системою (1 день)	FF / is caused by			Фінал розробки (07.04.2025)

Критичний шлях складається з послідовності взаємопов'язаних завдань, які мають найбільшу тривалість у проєкті. Визначення цього шляху здійснюється шляхом аналізу всіх завдань та їх тривалості з урахуванням логічних зв'язків між ними. Завдання, що входять до критичного шляху, мають нульовий резерв часу, тобто будь-яке їх відхилення від плану призводить до затримки завершення проєкту.

Критичний шлях є ключовим для управління часом, оскільки його завдання не мають резерву часу [67]. Це означає, що будь-яка затримка в їх виконанні призведе до затримки всього проєкту. Управління критичним шляхом передбачає постійний моніторинг цих завдань та контроль за їх своєчасним виконанням.

У випадку виявлення ризику затримок, управління критичним шляхом може включати застосування методів прискорення, таких як перерозподіл ресурсів або залучення додаткових працівників. Це дозволяє компенсувати можливі відхилення і забезпечити виконання проєкту у встановлений термін.

Завдання, що входять до критичного шляху:

1.1 → 1.2 → 1.3 → 2.1 → 2.2 → 2.3 → 2.8 → 3.1 → 4.1 → 4.2 → 4.3 → 5.1

Також цей критичний шлях та тривалість усіх задач можна відслідкувати на діаграмі Ганта, побудованій у системі Jira.



Рис. 3.4. Діаграма Ганта

## РОЗДІЛ 4. УПРАВЛІННЯ РИЗИКАМИ ПРОЄКТУ

### 4.1. Ідентифікація ризиків на етапі планування та реалізації

Управління ризиками є критично важливим аспектом ефективного управління проєктом, особливо в контексті розробки інформаційної системи для роботи з сервісами App Store та Google Play Market. На етапах планування та реалізації можуть виникати різні ризики, що можуть негативно вплинути на успішність проєкту. У цьому підпункті проведено ідентифікацію основних ризиків, які потенційно можуть виникнути, та їх категоризацію. [68], [69]

Всі ризики, пов'язані з реалізацією проєкту розробки системи для роботи з відгуками в App Store та Google Play Market, було ідентифіковано та класифіковано за їхньою природою. Таблиця 4.1. містить опис ризиків, їх потенційні наслідки та етапи, на яких вони можуть виникати. Це дозволяє чітко окреслити можливі проблеми та зосередитися на їхньому попередженні.

Таблиця 4.1.

Ризики проєкту

Номер ризику	Тип ризику	Опис ризику	Етап виникнення	Можливі наслідки
1	Технічний	Помилки інтеграції з API App Store та Google Play Market	Розробка	Затримка проєкту через необхідність виправлення коду або адаптації до нових вимог API.
2	Технічний	Технічна складність бази даних	Проектування	Низька продуктивність системи, затримки через необхідність переробки архітектури.
3	Технічний	Недостатнє тестування	Тестування	Пропуск критичних помилок, які знижують якість системи та потребують значних витрат на їх виправлення.
4	Технічний	Залежність від сторонніх бібліотек, фреймворків або програмного забезпечення	Розробка	Збої в роботі системи через несумісність оновлених сторонніх інструментів.

Продовження таблиці 4.1

5	Технічний	Труднощі забезпечення кросплатформної сумісності	Розробка інтерфейсу	Неправильне відображення системи на різних пристроях, що може викликати негативну реакцію користувачів.
6	Організаційний	Недостатнє планування ресурсів	Планування	Перевантаження одних працівників і простої інших, затримки у виконанні задач.
7	Організаційний	Проблеми комунікації в команді	Реалізація	Дублювання задач, пропуск важливих вимог або аспектів проекту.
8	Організаційний	Втрата ключових співробітників	Усі етапи	Затримки через потребу заміни фахівців або адаптації нових учасників до проекту.
9	Організаційний	Невідповідність змінам у вимогах	Реалізація	Додаткові витрати та затримки через необхідність внесення змін у вже виконані частини проекту.
10	Організаційний	Відсутність регулярного моніторингу	Усі етапи	Затримки через несвоєчасне виявлення проблем.
11	Процесний	Низька якість початкової документації	Планування	Неправильне трактування вимог, що впливає на кінцевий результат.
12	Процесний	Неефективне управління змінами	Реалізація	Збої у виконанні задач, хаотичне оновлення планів.
13	Процесний	Недостатня автоматизація процесу тестування	Тестування	Залишення критичних помилок у системі.
14	Процесний	Відсутність централізованої системи зберігання документації	Усі етапи	Втрата важливих даних, помилки у прийнятті рішень через використання застарілої інформації.
15	Процесний	Недостатнє управління знаннями всередині команди	Усі етапи	Повторення помилок, втрата часу на вирішення вже розв'язаних раніше питань.

Продовження таблиці 4.1

16	Зовнішній	Зміни в політиках App Store та Google Play Market	Усі етапи	Затримки через необхідність адаптації до нових вимог платформ.
17	Зовнішній	Обмеження у доступі до сторонніх API або сервісів	Тестування	Ускладнення перевірки функціональності, затримки у виконанні завдань.
18	Зовнішній	Економічна нестабільність	Усі етапи	Перевищення бюджету через підвищення цін або коливання валют.
19	Зовнішній	Юридичні зміни (наприклад, GDPR)	Реалізація, тестування	Необхідність переробки функціональності для дотримання законодавства, штрафи.
20	Зовнішній	Посилення конкурентів	Реалізація	Зниження попиту на продукт через появу конкурентних рішень.

Наступна таблиця 4.2. містить ранжування ідентифікованих ризиків за силою їхнього впливу на проєкт і рівнем керованості. Це дозволяє визначити критичні ризики, що потребують найбільшої уваги та превентивних заходів. Керованість визначена на основі можливості контролювати або зменшувати вплив ризику через управлінські рішення.

Таблиця 4.2.

**Ранжування ризиків за впливом і керованістю**

Номер ризику	Опис ризику	Вплив (Незначний, Середній, Високий, Дуже високий, Критичний)	Керованість (низька, середня, висока)
1	Помилки інтеграції з API App Store та Google Play Market	Критичний вплив	Середня
6	Недостатнє планування ресурсів	Критичний вплив	Висока
11	Низька якість початкової документації	Критичний вплив	Висока
19	Юридичні зміни (наприклад, GDPR)	Критичний вплив	Низька
2	Технічна складність бази даних	Дуже високий вплив	Середня

Продовження таблиці 4.2

10	Відсутність регулярного моніторингу	Дуже високий вплив	Висока
16	Зміни в політиках App Store і Google Play Market	Дуже високий вплив	Низька
3	Недостатнє тестування	Дуже високий вплив	Висока
7	Проблеми комунікації в команді	Високий вплив	Висока
12	Неефективне управління змінами	Високий вплив	Висока
13	Недостатня автоматизація процесу тестування	Високий вплив	Середня
5	Труднощі забезпечення кросплатформної сумісності	Середній вплив	Середня
8	Втрата ключових співробітників	Середній вплив	Середня
9	Невідповідність змінам у вимогах	Середній вплив	Середня
14	Відсутність централізованої системи зберігання документації	Середній вплив	Висока
15	Недостатнє управління знаннями всередині команди	Середній вплив	Середня
20	Посилення конкурентів	Середній вплив	Низька
4	Залежність від сторонніх бібліотек або фреймворків	Незначний вплив	Середня
17	Обмеження у доступі до сторонніх API або сервісів	Незначний вплив	Низька
18	Економічна нестабільність	Незначний вплив	Низька

#### 4.2. Оцінка ризиків. Використання матриці ймовірності та впливу ризиків.

Матриця ймовірності та впливу ризиків є ключовим інструментом для ефективного управління ризиками в проєкті. Вона дозволяє структурувати та оцінити потенційні ризики, які можуть виникнути під час реалізації проєкту, шляхом визначення їхньої ймовірності виникнення та можливого впливу на проєкт.

Основною метою використання матриці є надання командам управління інструменту для пріоритизації ризиків. Замість того, щоб рівноцінно ставитися до всіх потенційних загроз, матриця дозволяє зосередитися на найбільш критичних ризиках, які мають найвищу ймовірність і суттєвий вплив на ключові показники проєкту: строки, бюджет, якість та кінцевий результат [70].

Ймовірність та вплив ризиків:

1. Помилки інтеграції з API App Store та Google Play Market:
  - Ймовірність: висока (4 з 5), через складність інтеграції та часті оновлення API.
  - Вплив: критичний (5 з 5), оскільки помилки можуть затримати впровадження системи.
2. Технічна складність бази даних:
  - Ймовірність: середня (3 з 5), залежно від обраної архітектури.
  - Вплив: дуже високий (4 з 5), через можливе зниження продуктивності або необхідність значних змін.
3. Недостатнє тестування:
  - Ймовірність: середня (3 з 5), якщо команда не забезпечить якісне тестування.
  - Вплив: дуже високий (4 з 5), через ризик пропуску критичних помилок.
4. Залежність від сторонніх бібліотек або фреймворків:
  - Ймовірність: середня (3 з 5), якщо використовуються популярні інструменти.
  - Вплив: незначний (1 з 5), оскільки збої можуть бути швидко виправлені.
5. Труднощі забезпечення кросплатформної сумісності:
  - Ймовірність: середня (3 з 5), враховуючи різноманітність пристроїв.
  - Вплив: середній (2 з 5), через негативний вплив на користувацький досвід.
6. Недостатнє планування ресурсів:
  - Ймовірність: середній (2 з 5), через негативний вплив на користувацький досвід.

- Вплив: критичний (5 з 5), оскільки це може призвести до затримок.

7. Проблеми комунікації в команді:

- Ймовірність: середня (3 з 5), залежно від інструментів і процесів комунікації.

- Вплив: високий (3 з 5), через дублювання задач або пропуск вимог.

8. Втрата ключових співробітників:

- Ймовірність: низька (2 з 5), якщо співробітники задоволені умовами роботи.

- Вплив: середній (2 з 5), через затримки в заміні.

9. Невідповідність змінам у вимогах:

- Ймовірність: середня (3 з 5), залежно від стейкхолдерів.

- Вплив: середній (2 з 5), через додаткові витрати.

10. Відсутність регулярного моніторингу:

- Ймовірність: середня (3 з 5), якщо не впроваджені чіткі процеси контролю.

- Вплив: дуже високий (4 з 5), через затримки у виявленні проблем.

11. Низька якість початкової документації:

- Ймовірність: висока (4 з 5), через недостатню увагу до деталей.

- Вплив: критичний (5 з 5), через потенційні непорозуміння.

12. Неefективне управління змінами:

- Ймовірність: середня (3 з 5), через складність адаптації.

- Вплив: високий (3 з 5), через хаос у роботі.

13. Недостатня автоматизація процесу тестування:

- Ймовірність: середня (3 з 5), залежно від доступних інструментів.
  - Вплив: високий (3 з 5), через пропуск критичних помилок.
14. Відсутність централізованої системи зберігання документації:
- Ймовірність: середня (3 з 5), залежно від обраних систем.
  - Вплив: середній (2 з 5), через втрату важливих даних.
15. Недостатнє управління знаннями всередині команди:
- Ймовірність: середня (3 з 5), через недостатнє використання інструментів.
  - Вплив: середній (2 з 5), через втрату продуктивності.
16. Зміни в політиках App Store і Google Play Market:
- Ймовірність: середня (3 з 5), через часті оновлення політик.
  - Вплив: дуже високий (4 з 5), через затримки впровадження.
17. Обмеження у доступі до сторонніх API або сервісів:
- Ймовірність: низька (2 з 5), через стабільність платформ.
  - Вплив: незначний (1 з 5), через затримки в тестуванні.
18. Економічна нестабільність:
- Ймовірність: середня (3 з 5), залежно від зовнішніх умов.
  - Вплив: незначний (1 з 5), через перевищення бюджету.
19. Юридичні зміни (наприклад, GDPR):
- Ймовірність: низька (2 з 5), залежно від регіону впровадження.
  - Вплив: критичний (5 з 5), через необхідність переробки системи.
20. Посилення конкурентів:
- Ймовірність: середня (3 з 5), через активний ринок.
  - Вплив: середній (2 з 5), через зменшення попиту.

Матрицю ймовірності та впливу ризиків представлено у вигляді таблиці 4.3.

Таблиця 4.3.

### Матриця ймовірності та впливу ризиків

Ймовірність / Вплив	Незначний вплив (1)	Середній вплив (2)	Високий вплив (3)	Дуже високий вплив (4)	Критичний вплив (5)
Низька ймовірність (2)	Ризик 17	Ризик 8			Ризик 19
Середня ймовірність (3)	Ризик 4 Ризик 18	Ризик 5 Ризик 9 Ризик 14 Ризик 15 Ризик 20	Ризик 7 Ризик 3 Ризик 13	Ризик 2 Ризик 3 Ризик 10 Ризик 16	
Висока ймовірність (4)					Ризик 1 Ризик 6 Ризик 11

Відсутність ризиків із "дуже високою" (5) або "дуже низькою" (1) ймовірністю може бути наслідком декількох причин.

Проект, який розробляється, передбачає детальне планування і контроль, тому більшість ризиків оцінюються як такі, що мають середню або високу ймовірність. Дуже високий рівень ймовірності (5) притаманний ризикам, що майже гарантовано реалізуються, наприклад, в умовах невизначеності чи нестабільності, що не було зазначено для цього проєкту.

Більшість ризиків мають середню ймовірність через їхню загальну природу (наприклад, затримки або помилки інтеграції), і вони оцінюються на основі минулого досвіду або реалістичних припущень.

"Дуже низька" ймовірність (1) характерна для ризиків, які мало ймовірно існують через існування превентивних заходів або слабкий вплив зовнішніх факторів. Наприклад, обмеження доступу до API мало ймовірно, якщо платформи стабільно працюють.

У даному випадку оцінка ймовірності враховує баланс між ймовірністю та впливом. Оскільки проєкт знаходиться в середовищі з достатньо

контрольованими умовами (внутрішня команда, чіткий план, доступ до ресурсів), екстремальні оцінки ймовірності менш застосовні.

### 4.3. Протиризикові заходи

Управління ризиками є ключовим компонентом успішного виконання проєкту. Для кожного визначеного ризику розроблено заходи, які спрямовані на його мінімізацію, раннє виявлення та ефективне реагування. У таблиці 4.4. представлено систематизовану інформацію про профілактичні дії, симптоматику виникнення ризиків, а також стратегії реагування на кожному етапі — від виявлення симптомів до подолання проблеми. Такий підхід забезпечує проактивне управління ризиками, дозволяючи уникати значних втрат ресурсів і часу під час реалізації проєкту.

Таблиця 4.4.

#### Протиризикові заходи проєкту

Номер ризику	Опис ризику	Профілактичний захід	Виявлення симптоматики	Протиризиковий захід при симптомі	Протиризиковий захід при проблемі
1	Помилки інтеграції з API App Store та Google Play Market	Регулярне оновлення знань про API, створення тестового середовища	Невідповідність результатів тестів	Повторне тестування, перевірка документації	Оновлення коду відповідно до вимог API, звернення до підтримки
2	Технічна складність бази даних	Використання надійної архітектури бази даних	Зниження продуктивності під час тестів	Оптимізація запитів, перевірка наявних індексів	Зміна архітектури або перехід на іншу базу даних
3	Недостатнє тестування	Розробка детального тест-плану, використання автоматизації тестів	Пропущені помилки під час першого тестування	Додати нові сценарії тестування	Повторне тестування всієї системи
4	Залежність від сторонніх бібліотек	Використання перевірених бібліотек, створення резервного плану	Збої в роботі бібліотеки	Використання резервного коду, звернення до підтримки	Перехід на альтернативні бібліотеки

Продовження таблиці 4.4

5	Труднощі забезпечення кросплатформної сумісності	Створення тестового середовища для різних платформ	Некоректне відображення даних на одній із платформ	Додаткове тестування для конкретних платформ	Адаптація коду для кожної платформи
6	Недостатнє планування ресурсів	Залучення експертів для оцінки потреб, створення резервного фонду	Невідповідність обсягу роботи плану та реальності	Перерозподіл ресурсів між задачами	Найм додаткових фахівців або закупівля додаткового обладнання
7	Проблеми комунікації в команді	Впровадження регулярних зустрічей, використання Jira	Пропуск важливих аспектів, дублювання задач	Налагодження регулярних звітів	Створення чітких каналів комунікації
8	Втрата ключових співробітників	Створення детальної документації, збереження знань	Різне зниження продуктивності в певних задачах	Перерозподіл завдань між іншими членами команди	Найм нового фахівця
9	Невідповідність змінам у вимогах	Розробка чітких процесів управління змінами	Пропущені ключові аспекти змін	Створення оновлених технічних завдань	Переробка коду відповідно до змін
10	Відсутність регулярного моніторингу	Впровадження системи звітності	Відставання від графіка виконання задач	Проведення термінового аудиту	Реорганізація етапів роботи
11	Низька якість початкової документації	Створення чітких стандартів для документування	Невідповідність між описаними вимогами та виконаними задачами	Редагування документації	Залучення зовнішніх експертів для оцінки документації
12	Неефективне управління змінами	Впровадження затвердженого процесу управління змінами	Збої у плануванні через зміни	Проведення зустрічі для узгодження змін	Перегляд плану та термінів реалізації
13	Недостатня автоматизація тестування	Використання сучасних інструментів автоматизації тестів	Збільшення часу на ручне тестування	Пошук нових інструментів автоматизації	Найм додаткових тестувальників
14	Відсутність централізованої системи зберігання документації	Впровадження системи для управління документацією	Втрата доступу до необхідної інформації	Відновлення даних через резервні копії	Створення нової системи документообігу
15	Недостатнє управління знаннями всередині команди	Проведення тренінгів, збереження документації	Збільшення кількості повторюваних помилок	Організація регулярних навчальних сесій	Перепланування завдань для навчання

Продовження таблиці 4.4

16	Зміни в політиках App Store і Google Play Market	Регулярний моніторинг оновлень платформ	Несумісність із новими вимогами	Проведення термінового оновлення	Повна адаптація продукту до нових вимог
17	Обмеження у доступі до сторонніх API або сервісів	Використання альтернативних API	Нестабільний доступ до даних API	Звернення до технічної підтримки	Перехід на інші API
18	Економічна нестабільність	Створення резервного фонду	Підвищення вартості ліцензій або сервісів	Скорочення витрат	Пошук додаткових джерел фінансування
19	Юридичні зміни (наприклад, GDPR)	Проведення юридичної експертизи	Невідповідність новим законам	Адаптація функціональності системи	Перегляд процесів роботи
20	Посилення конкурентів	Регулярний моніторинг ринку	Втрата частини потенційних клієнтів	Проведення маркетингових акцій	Адаптація цінової політики

Реалізація запропонованих заходів сприятиме мінімізації потенційних втрат часу, ресурсів і фінансів. Проактивний підхід до управління ризиками дозволяє не лише запобігати проблемам, а й ефективно реагувати на непередбачені обставини. Це забезпечує стабільність реалізації проекту та підвищує ймовірність досягнення поставлених цілей.

## ВИСНОВКИ

У кваліфікаційній роботі магістра на тему «Дослідження моделей управління проектом з розробки інформаційної системи для роботи з сервісами App Store та Google Play Market» було проаналізовано, адаптовано та запропоновано рішення для ефективного управління проектом. У дослідженні особливу увагу приділено методологіям управління, організації командної роботи, плануванню ресурсів, ідентифікації ризиків та розробці протиризикових заходів.

Робота охоплює як теоретичний аналіз сучасних моделей управління проектами, так і практичну реалізацію ключових етапів проекту з використанням фреймворку Kanban. Основні висновки сформульовано за результатами аналізу, планування та впровадження підходів до управління проектом.

1. Проведено всебічний аналіз сучасних моделей управління проектами, включаючи традиційні, гнучкі та гібридні методології. Визначено, що гнучкий підхід, зокрема Kanban, найбільш ефективний для управління проектами у сфері ІТ завдяки його здатності швидко адаптуватися до змін і забезпечувати прозорість процесів на всіх етапах реалізації.

2. Обґрунтовано вибір фреймворку Kanban як основи для управління проектом. У ході дослідження враховано специфіку реалізації проекту зі створення інформаційної системи для роботи з сервісами App Store і Google Play Market, що передбачає високу динаміку змін у вимогах і потребу в оптимальній координації команди.

3. Розроблено концепцію інформаційної системи, яка включає визначення структури команди, ключових модулів системи та архітектури бази даних. Визначено, що система буде спрямована на автоматизацію обробки відгуків користувачів, що сприятиме підвищенню якості обслуговування та репутації мобільних додатків компанії.

4. Здійснено оцінку фінансових аспектів проєкту, включаючи формування бюджету, розрахунок витрат на трудові ресурси, технічне й програмне забезпечення. Розроблено фінансову модель, яка демонструє можливість виходу проєкту на самоокупність протягом 13 місяців, завдяки чітко сформованій стратегії монетизації.

5. Продемонстровано практичне застосування фреймворку Kanban у процесі управління проєктом. Розроблено ієрархічну структуру робіт (WBS), побудовано діаграму Ганта, реалізовано управління задачами та ресурсами за допомогою інструменту Jira, що забезпечило чітке планування й контроль виконання задач.

6. Розроблено модель управління ризиками, що включає ідентифікацію, оцінку ризиків і формування матриці ймовірності та впливу. Для кожного ризику сформульовано протиризові заходи, що забезпечують зниження негативного впливу на реалізацію проєкту, підвищуючи його успішність.

Варто зазначити, що у цій кваліфікаційній роботі магістра було досліджено моделі управління проєктами, адаптовані найефективніші методи до конкретного проєкту та реалізований план, що забезпечує його успішне виконання. Запропоновані рішення дозволяють розробити інформаційну систему, яка забезпечить ефективне управління відгуками, покращить взаємодію з користувачами та зміцнить позиції ІТ компанії на ринку. Отже загалом можна стверджувати, що мета кваліфікаційної роботи досягнута, а задачі – вирішені.

## ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Управління проєктами: процеси планування проєктних дій [Текст]: підручник / І.В. Чумаченко, В.В. Морозов, Н.В. Доценко, А.М. Чередніченко. - К.: Університет економіки та права «КРОК», 2014. - 673 с.
2. Управління проєктами, програмами та проєктно-орієнтованим бізнесом: Колективна монографія. Том 3 / В.В. Морозов, Є.Д. Кузнецов, О.Б. Данченко та інші; за наук. ред. В.В.Морозова - К.: ВНЗ «Університет економіки та права «КРОК», 2012. - 238 с.
3. Marketing Theories - PESTEL Analysis: [Електронний ресурс]. Режим доступу: <https://www.professionalacademy.com/blogs/marketing-theories-pestel-analysis/>
4. What is a PESTLE Analysis?: [Електронний ресурс]. Режим доступу: <https://mooncamp.com/glossary/pestle-analysis#what-is-a-pestle-analysis>
5. Marketing Theories - Swot Analysis: [Електронний ресурс]. Режим доступу: <https://www.professionalacademy.com/blogs/marketing-theories-swot-analysis/>
6. Stakeholder analysis: [Електронний ресурс]. Режим доступу: <https://www.pmi.org/learning/library/stakeholder-analysis-pivotal-practice-projects-8905>
7. Stakeholder Analysis: [Електронний ресурс]. Режим доступу: <https://www.productplan.com/glossary/stakeholder-analysis/>
8. What are the roles of scenario planning in project management?: [Електронний ресурс]. Режим доступу: <https://www.pmi.org/learning/library/scenario-planning-project-management-6858>
9. Finding the competitive edge: [Електронний ресурс]. Режим доступу: <https://www.pmi.org/learning/library/organizational-tools-finding-competitive-edge-4016>

10. The role of user feedback in improving your mobile app: [Электронный ресурс]. Режим доступа: <https://moldstud.com/articles/p-the-role-of-user-feedback-in-improving-your-mobile-app>
11. Android Apps and User Feedback: A Dataset for Software Evolution and Quality Improvement: [Электронный ресурс]. Режим доступа: [https://www.researchgate.net/publication/318317595\\_Android\\_Apps\\_and\\_User\\_Feedback\\_A\\_Dataset\\_for\\_Software\\_Evolution\\_and\\_Quality\\_Improvement](https://www.researchgate.net/publication/318317595_Android_Apps_and_User_Feedback_A_Dataset_for_Software_Evolution_and_Quality_Improvement)
12. The Importance of User Feedback in Mobile App Development: [Электронный ресурс]. Режим доступа: <https://www.linkedin.com/pulse/importance-user-feedback-mobile-app-development-houssam-el-hatimi/>
13. How To Analyze App Store Customer Reviews With AI?: [Электронный ресурс]. Режим доступа: <https://www.anecdoteai.com/blog/how-to-analyze-app-store-customer-reviews-with-ai>
14. Why MonkeyLearn?: [Электронный ресурс]. Режим доступа: <https://help.monkeylearn.com/en/articles/2173773-why-monkeylearn>
15. Transform Complex Text Documents into Data, Insights, & Value: [Электронный ресурс]. Режим доступа: <https://www.lexalytics.com/>
16. Explore AppFollow: [Электронный ресурс]. Режим доступа: <https://support.appfollow.io/hc/en-us/categories/360003753697-Explore-AppFollow>
17. Service that's up close and personal: [Электронный ресурс]. Режим доступа: <https://www.zendesk.com/service/>
18. Google Data Studio: [Электронный ресурс]. Режим доступа: <https://elit-web.ua/ua/blog/google-data-studio>
19. The Best Customer Feedback Management Tools for 2024: [Электронный ресурс]. Режим доступа: <https://www.getbeamer.com/blog/customer-feedback-management-tools>
20. Global Feedback Management Software Market Size, Share & Industry Trends Analysis Report By Component, By Type (Customer, Employee), By

Deployment Mode (On-premises and Cloud), By Organization Size, By Vertical, By Regional Outlook and Forecast, 2022 - 2028 : [Электронный ресурс]. Режим доступа: <https://www.kbvresearch.com/feedback-management-software-market/>

21. Feedback And Reviews Management Software Market Size And Forecast: [Электронный ресурс]. Режим доступа: <https://www.verifiedmarketresearch.com/product/feedback-and-reviews-management-software-market/>

22. What Is CRM (Customer Relationship Management)? : [Электронный ресурс]. Режим доступа: <https://www.salesforce.com/crm/what-is-crm/>

23. Critical evaluation of waterfall project management methodology: A case study of digital management conference project: [Электронный ресурс]. Режим доступа:

[https://www.researchgate.net/publication/361092440\\_Critical\\_evaluation\\_of\\_waterfall\\_project\\_management\\_methodology\\_A\\_case\\_study\\_of\\_digital\\_management\\_conference\\_project](https://www.researchgate.net/publication/361092440_Critical_evaluation_of_waterfall_project_management_methodology_A_case_study_of_digital_management_conference_project)

24. Rick Sherman Project Management: Business Intelligence Guidebook 2015. 449-492 с. Режим доступа:

<https://www.sciencedirect.com/science/article/abs/pii/B9780124114616000186>

25. A Guide to the Project Management Body of Knowledge (PMBOK Guide) - Fifth Edition.: Project Management Institute, Inc., 2013. - 614 с. Режим доступа: <https://www.pmi.org/>

26. Agile Project Management: A Communicational Workflow Proposal / Carina Loiro, Hélio Castro, Paulo Ávila, Maria Manuela Cruz-Cunha, Goran D. Putnik, Luís Ferreira: Procedia Computer Science Vol. 164, p. 485-490. Режим доступа: <https://www.sciencedirect.com/science/article/pii/S1877050919322574>

27. Manifesto for Agile Software Development: [Электронный ресурс]. Режим доступа: <https://agilemanifesto.org/iso/en/manifesto.html>

28. Maja Due Kadenic, Konstantinos Koumaditis, Louis Junker-Jensen Mastering scrum with a focus on team maturity and key components of scrum:

Information and Software Technology, Vol. 153. Режим доступа:  
<https://www.sciencedirect.com/science/article/pii/S0950584922001884>

29. Agile vs Scrum. How to choose the best methodology for you:  
[Электронный ресурс]. Режим доступа:  
<https://www.atlassian.com/agile/scrum/agile-vs-scrum>

30. Overview on Kanban Methodology and its Implementation:  
[Электронный ресурс]. Режим доступа:  
[https://www.researchgate.net/publication/280865949\\_Overview\\_on\\_Kanban\\_Methodology\\_and\\_its\\_Implementation](https://www.researchgate.net/publication/280865949_Overview_on_Kanban_Methodology_and_its_Implementation)

31. Hamzah Alaidaros, Mazni Omar, Rohaida Romli. The state of the art of agile Kanban method: challenges and opportunities 2021. Vol. 12, №8, p. 2535-2550.  
Режим доступа:  
[https://www.researchgate.net/publication/348960202\\_The\\_State\\_of\\_the\\_Art\\_of\\_Agile\\_Kanban\\_Method\\_Challenges\\_and\\_Opportunities](https://www.researchgate.net/publication/348960202_The_State_of_the_Art_of_Agile_Kanban_Method_Challenges_and_Opportunities)

32. The Hybrid Era: Project Management Embraces the Fit-for-Purpose Approach: веб-сайт. Режим доступа: <https://www.pmi.org/blog/project-management-embraces-the-fit-for-purpose-approach>

33. Scrumban: Mastering two Agile methodologies: [Электронный ресурс].  
Режим доступа: <https://www.atlassian.com/agile/project-management/scrumban>

34. Завербний А.С, Ільницький В.С. Lean production as a tool to increase the efficiency of project management: Market Infrastructure, DOI:10.32843/infrastructure65-12, p. 69-71. Режим доступа:  
[https://www.researchgate.net/publication/360854389\\_LEAN\\_PRODUCTION\\_AS\\_A\\_TOOL\\_TO\\_INCREASE\\_THE\\_EFFICIENCY\\_OF\\_PROJECT\\_MANAGEMENT](https://www.researchgate.net/publication/360854389_LEAN_PRODUCTION_AS_A_TOOL_TO_INCREASE_THE_EFFICIENCY_OF_PROJECT_MANAGEMENT)

35. Kanban. How the kanban methodology applies to software development:  
[Электронный ресурс]. Режим доступа: <https://www.atlassian.com/agile/kanban>

36. On the benefits and challenges of using kanban in software engineering: a structured synthesis study: [Электронный ресурс]. Режим доступа:  
<https://jserd.springeropen.com/articles/10.1186/s40411-018-0057-1>

37. Michael Polonsky Stakeholder management and the stakeholder matrix: Potential strategic marketing tools: Journal of Market-Focused Management Vol. 1, №3, p. 209-229. Режим доступу:  
[https://www.researchgate.net/publication/227107320 Stakeholder management and the stakeholder matrix Potential strategic marketing tools](https://www.researchgate.net/publication/227107320_Stakeholder_management_and_the_stakeholder_matrix_Potential_strategic_marketing_tools)
38. Using a Stakeholder Matrix: 3 Practical Examples: [Електронний ресурс]. Режим доступу: <https://simplystakeholders.com/stakeholder-matrix/>
39. Морозов В.В. Формування, управління та розвиток команди проекту / Морозов В.В., Чередніченко А.М., Шпильова Т.І. - Київ: Таксон, 2009. - 461с.
40. Wilemon, D. L. & Thamhain, H. J. Team Building in Project Management: Project Management Quarterly, Vol.14, № 2, p.73-81. Режим доступу:  
<https://www.pmi.org/learning/library/team-building-development-project-management-5707>
41. G. Smith Making the team [project team building and leadership]: IEE Review Vol. 47, №5, p. 33 - 36. Режим доступу:  
[https://www.researchgate.net/publication/3346479 Making the team project team building and leadership](https://www.researchgate.net/publication/3346479_Making_the_team_project_team_building_and_leadership)
42. Project Management and Teamwork: [Електронний ресурс]. Режим доступу:  
[https://www.researchgate.net/publication/247277075 Project Management and Teamwork](https://www.researchgate.net/publication/247277075_Project_Management_and_Teamwork)
43. Organizational Breakdown Structure (OBS) in Project Management: [Електронний ресурс]. Режим доступу: <https://pmiuk.co.uk/unlocking-project-success-the-power-of-organizational-breakdown-structure-obs/>
44. Управління проектами: процеси планування проектних дій / І.В. Чумаченко, В.В. Морозов, Н.В. Доценко, А.М. Чередніченко / Київ 2016.
45. Salary statistics for Ukraine: [Електронний ресурс]. Режим доступу:  
<https://www.work.ua/en/stat/>

46. Front End Developer Salary in Ukraine: [Электронный ресурс]. Режим доступа: <https://www.salaryexpert.com/salary/job/front-end-developer/ukraine>
47. Buy Visual Studio: [Электронный ресурс]. Режим доступа: <https://visualstudio.microsoft.com/vs/pricing/>
48. Plans and Pricing: [Электронный ресурс]. Режим доступа: <https://assertible.com/plans>
49. Simple, transparent pricing for every team: [Электронный ресурс]. Режим доступа: <https://www.atlassian.com/en/software/jira/pricing>
50. Shrivastava, N. K. A model to develop and use risk contingency reserve: Paper presented at PMI® Global Congress 2014—North America, Phoenix, AZ. Newtown Square, PA: Project Management Institute. Режим доступа: <https://www.pmi.org/learning/library/model-risk-contingency-reserve-9310>
51. England, K. & Moreci, J. Contingency—are you covered?: Paper presented at PMI® Global Congress 2012—North America, Vancouver, British Columbia, Canada. Newtown Square, PA: Project Management Institute. Режим доступа: <https://www.pmi.org/learning/library/contingency-are-covered-6099>
52. Cooper, M. Contingency when proposing IT service projects—the supplier’s viewpoint: Paper presented at Project Management Institute Annual Seminars & Symposium, San Antonio, TX. Newtown Square, PA: Project Management Institute. Режим доступа: <https://www.pmi.org/learning/library/contingency-proposing-service-projects-8508>
53. Contingency Funding: [Электронный ресурс]. Режим доступа: <https://kirkwood.pressbooks.pub/projectmanagementbasics/chapter/6-5-contingencies/>
54. Matthew West Some Types and Uses of Data Models: Developing High Quality Data Models 2011, p. 23-36. Режим доступа: <https://www.sciencedirect.com/science/article/abs/pii/B9780123751065000038>
55. What is Conceptual Data Modeling?: [Электронный ресурс]. Режим доступа: <https://erstudio.com/blog/conceptual-data-modeling/>

56. James V. Luisi Information Architecture: Pragmatic Enterprise Architecture, 2014. p. 189-261. Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/B9780128002056000044>

57. The logical data model : a new approach to database logic: [Електронний ресурс]. Режим доступу: [https://www.researchgate.net/publication/35973495\\_The\\_logical\\_data\\_model\\_a\\_new\\_approach\\_to\\_database\\_logic](https://www.researchgate.net/publication/35973495_The_logical_data_model_a_new_approach_to_database_logic)

58. What Is a Logical Data Model?: [Електронний ресурс]. Режим доступу: <https://www.gooddata.com/blog/how-build-logical-data-models-scale-analytical-applications/>

59. What is a work breakdown structure (WBS)? : [Електронний ресурс]. Режим доступу: <https://business.adobe.com/blog/basics/what-is-work-breakdown-structure#:~=The%20WBS%20is%20a%20hierarchical,into%20smaller%2C%20more%20manageable%20tasks.>

60. Work Breakdown Structure (WBS) In Project Management: [Електронний ресурс]. Режим доступу: <https://www.forbes.com/advisor/business/what-is-work-breakdown-structure/>

61. Webster, F. M. (1994). The WBS: PM Network, Vol. 8, №12, p. 40-46. Режим доступу: <https://www.pmi.org/learning/library/work-breakdown-structure-basic-principles-4883>

62. Brotherton, S. A., Fried, R. T., & Norman, E. S. Applying the work breakdown structure to the project management lifecycle: Paper presented at PMI® Global Congress 2008—North America, Denver, CO. Newtown Square, PA: Project Management Institute. Режим доступу: <https://www.pmi.org/learning/library/applying-work-breakdown-structure-project-lifecycle-6979>

63. Що таке Діаграма Ганта і як правильно користуватися?: [Електронний ресурс]. Режим доступу: <https://worksection.com/ua/blog/what-is-gantt-chart.html>

64. What are Gantt charts and how to create one: [Электронный ресурс]. Режим доступа: <https://www.atlassian.com/agile/project-management/gantt-chart#:~=Summary%3A%20A%20Gantt%20chart%20is,tasks%2C%20assignees%2C%20and%20more>.

65. Ramachandran K K Gantt Chart: An Important Tool of Management: International Journal of Innovative Technology and Exploring Engineering Vol. 8, №7с, р.2278-3075. Режим доступа: [https://www.researchgate.net/publication/358234055\\_Gantt\\_Chart\\_An\\_Important\\_Tool\\_of\\_Management](https://www.researchgate.net/publication/358234055_Gantt_Chart_An_Important_Tool_of_Management)

66. Ahmad Aliyu Maidamisa Project Management using Critical Path Method (CPM): A Pragmatic Study: Global Journal of Pure and Applied Sciences Vol. 18, №3-4. р. 197-206. Режим доступа: [https://www.researchgate.net/publication/272459738\\_Project\\_Management\\_using\\_Critical\\_Path\\_Method\\_CPM\\_A\\_Pragmatic\\_Study](https://www.researchgate.net/publication/272459738_Project_Management_using_Critical_Path_Method_CPM_A_Pragmatic_Study)

67. How to Use the Critical Path Method (CPM) in Project Management [2024]: [Электронный ресурс]. Режим доступа: <https://monday.com/blog/project-management/critical-paths/>

68. Green, S. D. & Dikmen, I. Narratives of Project Risk Management: From Scientific Rationality to the Discursive Nature of Identity Work: Project Management Journal, Vol. 53, р.608-624. doi: <https://doi.org/10.1177/87569728221124496>. Режим доступа: <https://www.pmi.org/learning/library/narratives-project-risk-management-13647>

69. Project risk management guidelines: managing risk in large projects and complex procurements / Dale F. Cooper, Stephen Grey, Geoffrey Raymond and Phil Walker. John Wiley & Sons Ltd, Chichester, West Sussex, England, 2005. - 384p.

70. Beyond probability-impact matrices in project risk management: A quantitative methodology for risk prioritization / Fernando Acebes, Adolfo Lopez-Paredes, José Manuel González-Varona, Javier Pajares: HUMANITIES AND SOCIAL SCIENCES COMMUNICATIONS 2024. p. 1-11.

DOI:10.48550/arXiv.2405.20679.

Режим

доступу:

[https://www.researchgate.net/publication/381108758\\_Beyond\\_probability-impact\\_matrices\\_in\\_project\\_risk\\_management\\_A\\_quantitative\\_methodology\\_for\\_risk\\_prioritisation](https://www.researchgate.net/publication/381108758_Beyond_probability-impact_matrices_in_project_risk_management_A_quantitative_methodology_for_risk_prioritisation)

# ДОДАТОК А

## Вигляд фінального продукту






Рис.А.1. Модуль шаблонів відповідей

## Add template ✕

**Template name**

**Template text** 0/5970

**Apps**

-  iOS Game 1 ✕
-  iOS Game 2 ✕ ▼
-  iOS Game 3 ✕

**Folder**

▼

**Tags**

▼

**Add to workspace**

▼

**Save**

Рис. А.2. Створення нового шаблону відповіді

**FEATURED**

★★★★★ Solangel Garcia Domas Nov 5, 2024, 7:07 AM Report ⋮

Muy buen juego. No he encontrado otro como este. Lo desintele. Y tuve la necesidad de volver a instalarlo. Me arrepenti. Porq no hay otro como este. Me encanta. Lo q hay que tener paciencia. Porq me he quedado admirada con su trabajo q hasta el progreso de mi juego está por dónde mismo lo deje. Gracias les doy. Y espero me disculpen. Los Felicitó. Buen trabajo para mí tienen 5 estrellas. Para mí es uno de los mejores juegos y lo extraña muchísimo.

[Show history](#)

Tags: [+ Thank you](#) [+ proposal](#) [+ negative](#) [+ update](#) [+ support](#) [+ Add tag](#)

**Liza Kliesman** Nov 8, 2024, 4:31 PM, status is replied

Agradecemos enormemente tu valoración y entusiasmo. Es genial saber que disfrutaste tanto del juego y que decidiste volver. ¡Nos alegra que estés aquí y que lo consideres uno de los mejores!

[Save reply as template](#)

Country: [Get access](#)  
 Device language: Spanish  
 Review language: Spanish  
 App version: 1.23.0  
 App version code: 7313  
 OS: Android 12.0  
 Device: TECNO (TECNO-BF7)

★☆☆☆☆ Dorinha Gomes Nov 8, 2024, 5:10 AM Report ⋮

Vou deixar aqui a minha opinião :está bugando bastante, toda dia mais de dez vezes reinicia quando toca em alguma coisa. A intenção foi boa para gastar dinheiro, ok 🤔 ? Todos os dias nasce planta, pedra, baú em cima das coisas, fica muito caro e feio a ilha, já fiz de tudo para manter organizada, impossível!! Desinstalando e não recomendo. Deixo aqui meu apelo para continuar se arrumar essa bagunça!

Tags: [+ Bug](#) [+ Add tag](#)

**Templates** Support ID Energy (1) ⋮ ✕

Enter reply text:

[Send reply](#)

Country: [Get access](#)  
 Device language: Portuguese  
 Review language: Portuguese  
 App version:  
 OS: Android 13.0  
 Device: POCO X3 Pro (vayu)

Рис. А.3. Сторінка зі списком відгуків

**Period**

Last 7 days

Last 30 days

Last 90 days

This year

Custom

**Review rating**

5 ★  4 ★

3 ★  2 ★  1 ★

**Review language**

e.g. English ▾

**Reply**

No reply × Outdated × ▾

**Review text**

Enter text

**Tag**

e.g. Bug ▾

Рис.А.4. Панель фільтрації та сортування