

Вінницький національний технічний університет  
(повне найменування вищого навчального закладу)  
Факультет інформаційних технологій та комп'ютерної інженерії  
(повне найменування інституту, назва факультету (відділення))  
Кафедра програмного забезпечення  
(повна назва кафедри (предметної, циклової комісії))

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**  
на тему: «Розробка методів та програмних засобів системи управління стартап-проектами»

Виконала: студентка 2-го курсу, групи ЗПІ-22м  
спеціальності 121 – Інженерія програмного  
забезпечення

Маланчук А. В.  
(шифр і назва напрямку підготовки, спеціальності)  
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ:

Войтко В. В.  
(прізвище та ініціали)

« 12 » грудня 2023 р.

Опонент к.т.н., доц. каф. : Л. А.  
к.т.н., доц. каф. ОТ Савицька Л.А.

« 12 » грудня 2023 р.

Допущено до захисту  
Завідувач кафедри ПЗ  
к.т.н., проф. Романюк О.Н.  
(прізвище та ініціали)  
« 12 » грудня 2023 р.

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра програмного забезпечення  
Рівень вищої освіти II-й (магістерський)  
Галузь знань 12 – Інформаційні технології  
Спеціальність 121 – Інженерія програмного забезпечення  
Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Романюк О. Н.

« 19 » вересня 2023 р.

## ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЮ РОБОТУ СТУДЕНТУ

Маланчук Анні Василівні

1. Тема роботи – Розробка методів та програмних засобів системи управління стартап-проектами.

Керівник роботи: Войтко Вікторія Володимирівна, к.т.н., доц. кафедри ПЗ, затверджені наказом вищого навчального закладу від « 18 » вересня 2023 р. № 247.

2. Строк подання студентом роботи

5 грудня 2023 року

3. Вихідні дані до роботи: метод управління стартап-проектами; система керування базами даних – MySQL; мова запитів – SQL; підключення до використання бібліотеки JDBC; результат тестування.

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз стану розвитку систем управління стартап-проектами, порівняльний аналіз аналогів, аналіз методів розв'язання задачі, постановка задач дослідження; розробка загальної моделі системи, розробка методу управління стартап-проектами, розробка алгоритмів роботи системи; варіантний аналіз і обґрунтування вибору засобів для реалізації програмного засобу; розробка програмного модуля для реєстрації та авторизації користувачів; розробка програмного модуля для управління проектами; розробка програмного модуля аналізу проектів та

генерації звітів; тестування програмного додатку; економічна частина; висновки; список використаних джерел; додатки.

5. Перелік графічного матеріалу: моделі системи; блок-схеми алгоритмів роботи додатку; результати тестування додатку.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Войтко В.В. к.т.н., доцент кафедри ПЗ	19.09.23 <i>ВВ</i>	05.12.23 <i>ВВ</i>
5	Причепя І.В., к.е.н., доцент кафедри ЕПВМ	22.11.23 <i>ІВ</i>	01.12.23 <i>ІВ</i>

7. Дата видачі завдання 19 вересня 2023 р.

#### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз стану розвитку систем управління стартап-проектами	20.09.2023 – 30.09.2023	<i>вик</i>
2	Розробка методу, моделей та алгоритмів роботи системи управління стартап-проектами	01.10.2023 – 17.10.2023	<i>вик</i>
3	Програмна реалізація системи управління стартап-проектами	18.10.2023 – 04.11.2023	<i>вик</i>
4	Тестування програмного продукту	05.11.2023 – 21.11.2023	<i>вик</i>
5	Економічна частина	22.11.2023 – 01.12.2023	<i>вик</i>

Студентка

*Маланчук А. В.*  
(підпис) Маланчук А. В.  
(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи

*Войтко В. В.*  
(підпис) Войтко В. В.  
(прізвище та ініціали)

## Анотація

УДК 004.912.032.26

Маланчук А.В. – Розробка методів та програмних засобів системи управління стартап-проєктами Магістерська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2023. 147с.

На укр. мові. Бібліогр.: 47 назв; рисунків: 32; таблиць: 11.

У магістерській кваліфікаційній роботі розроблено додаток для управління стартап-проєктами. Було проведений детальний порівняльний аналіз існуючих аналогів системи, який дозволив визначити їхні переваги та недоліки. У роботі ретельно обґрунтовано доцільність розробки власного програмного продукту, вказано, які конкретні переваги та унікальні риси має додаток у порівнянні з існуючими аналогами.

Розроблено метод управління стартап-проєктами. Розроблено модель системи управління стартапами, яка враховує особливості організації роботи зі стартапами, нові стратегії моніторингу й оптимізації робочих процесів, що дозволяє покращити процес керування проєктами. Розроблено додаток з використанням інтегрованого середовища розробки IntelliJ IDEA та мови програмування Java. Системою управління базами даних обрано MySQL.

Результати магістерської кваліфікаційної роботи можуть бути використані для управління стартап-проєктами або управління проєктами в невеликих компаніях.

Ключові слова: ефективність, аналіз, управління, стартап, проєкт, невелика компанія.

## **Abstract**

Malanchuk A.V. Development of an educational game on the history of Ukraine for an Android application using adaptive learning methods. Master's thesis on the specialty 121 - software engineering, educational program - software engineering. Vinnytsia: VNTU, 2023. 147 p.

In Ukrainian speech Bibliogr.: 47 title; Drawings: 32; Tables: 11.

In the master's qualification work, an application for managing startup projects was developed. A detailed comparative analysis of existing analogues of the system was carried out, which made it possible to determine their advantages and disadvantages. The paper carefully substantiates the feasibility of developing one's own software product, it indicates what specific advantages and unique features the application has in comparison with existing analogues.

A method of managing startup projects has been developed. A model of the startup management system has been developed, which takes into account the peculiarities of the organization of work with startups, new strategies for monitoring and optimizing work processes, which allows improving the project management process. The application was developed using the integrated development environment IntelliJ IDEA and the Java programming language. MySQL was chosen as the database management system.

The results of the master's qualification work can be used to manage start-up projects or project management in small companies.

Keywords: efficiency, analysis, management, startup, project, small company.

## ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ СТАНУ РОЗВИТКУ СИСТЕМ УПРАВЛІННЯ СТАРТАП-ПРОЄКТАМИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ.....	8
1.1 Аналіз стану розвитку систем управління стартап-проєктами .....	8
1.2 Порівняльний аналіз аналогів .....	9
1.3 Аналіз методів розв’язання задачі.....	14
1.4 Постановка задач дослідження .....	16
1.5 Висновок.....	17
2 РОЗРОБКА МЕТОДУ, МОДЕЛЕЙ ТА АЛГОРИТМІВ РОБОТИ СИСТЕМИ УПРАВЛІННЯ СТАРТАП-ПРОЄКТАМИ .....	18
2.1 Аналіз інформаційного забезпечення системи.....	18
2.2 Розробка моделей системи управління стартапами .....	20
2.3 Розробка методу управління стартап-проєктами.....	27
2.4 Розробка алгоритмів роботи системи.....	35
2.5 Висновки .....	40
3 РОЗРОБКА ПРОГРАМИ УПРАВЛІННЯ СТАРТАП-ПРОЄКТАМИ....	41
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного засобу .....	41
3.2 Розробка програмного модуля для реєстрації та авторизації користувачів.....	49
3.3 Розробка програмного модуля для управління проєктами.....	53
3.4 Розробка програмного модуля аналізу проєктів та генерації звітів.....	57
3.5 Висновки .....	63
4 ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКУ .....	65
4.1 Аналіз методів тестування програмного забезпечення .....	65
4.2 Тестування системи управління стартап-проєктами .....	66
4.3 Розробка інструкції користувача.....	73
4.4 Висновки .....	75

5 ЕКОНОМІЧНА ЧАСТИНА.....	77
5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки .....	77
5.2 Розрахунок витрат на проведення науково-дослідної роботи .....	81
5.2.1 Витрати на оплату праці.....	81
5.2.2 Відрахування на соціальні заходи.....	84
5.2.3 Сировина та матеріали .....	84
5.2.4 Розрахунок витрат на комплектуючі .....	85
5.2.5 Спецустаткування для наукових (експериментальних) робіт.....	86
5.2.6 Програмне забезпечення для наукових (експериментальних) робіт.....	86
5.2.7 Амортизація обладнання, програмних засобів та приміщень.....	87
5.2.8 Паливо та енергія для науково-виробничих цілей .....	88
5.2.9 Службові відрядження .....	89
5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	89
5.2.11 Інші витрати .....	90
5.2.12 Накладні (загальновиробничі) витрати .....	90
5.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором .....	84
5.4 Висновки .....	88
ВИСНОВОК.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	93
Додаток А (обов'язковий). Технічне завдання <b>Помилка!</b> Закладку не визначено.	
Додаток Б (обов'язковий). Протокол перевірки на плагіат .....	<b>Помилка!</b>
<b>Закладку не визначено.</b>	
Додаток В (довідниковий). Лістинг програми .....	108
Додаток Г (довідниковий). Ілюстративна частина .....	135

## ВСТУП

**Обґрунтування вибору теми дослідження.** У сучасному світі стартапи стають джерелом інновацій та економічного зростання.

Малі компанії та стартапи часто стикаються з обмеженими ресурсами та обсягами роботи. Створення системи управління дозволить їм ефективно організувати завдання, використовувати ресурси та підвищити ефективність роботи команди [1].

Стартапи стають невід'ємною складовою економіки сучасного світу, що відіграють важливу роль у формуванні інновацій та забезпеченні економічного зростання. Однак, вони часто опиняються перед викликами обмежених ресурсів та потреби в ефективній організації та управлінні [2].

Розвиток інструментів для управління стартапами сприяє формуванню сприятливого середовища для нових підприємств. Це може включати в себе полегшення пошуку інвесторів, залучення та підтримку талановитих команд [3].

Система управління стартап-проєктами допомагає забезпечити стабільність та порядок у роботі, що є важливим елементом довіри як серед команд, так і серед інвесторів та партнерів [4].

Розробка такої системи відкриває можливості для дослідження методів управління проєктами в сфері стартапів та впровадження нових ідей та інновацій у цю галузь.

Основні завдання системи управління полягають у забезпеченні максимальної ефективності у процесі підтримки проєктів та керування стартапами, а також у впровадженні нових методів управління, що сприятимуть подальшому розвитку та успіху цих ініціатив.



Тому актуальним є створення системи, яка допомагатиме ефективно управляти стартап-проєктами, оскільки це сприятиме розвитку нових бізнес-ідей та підтримці підприємницької ініціативи.

Така система управління стартап-проєктами матиме потенціал значно покращити умови для розвитку та успіху стартапів у динамічному сучасному бізнес-середовищі [5].

**Зв'язок роботи з науковими програмами, планами, темами.** Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення

**Мета та завдання дослідження.** Метою магістерської кваліфікаційної роботи є полегшення та оптимізація процесів управління стартапами й малими компаніями шляхом розробки та впровадження спеціалізованої системи управління проєктами, що дозволить ефективний розподіл ресурсів для розв'язування поставлених задач.

У магістерській кваліфікаційній роботі планується створити інноваційний інструмент, який допоможе командам ефективно організовувати свої проєкти, використовувати ресурси та підвищувати ефективність роботи.

Основними задачами роботи є:

- визначити функціонал системи управління стартап-проєктами та методи і засоби її реалізації;
- розробити метод, моделі та алгоритми роботи системи;
- розробити програмне забезпечення системи управління стартап-проєктами;
- провести тестування роботи програмного продукту.

**Об'єктом дослідження** є процес створення системи управління стартап-проєктами.

**Предметом дослідження** є методи і засоби розробки системи управління стартап-проєктами.

**Методи дослідження.** У процесі досліджень використовувалися методи дослідження:

- методи автоматизації процесів управління для автоматизації процесів керування ресурсами;
- методи теорії інформації для обробки вхідного контенту;
- методи організації баз даних для організації процесу збереження даних на сервері;
- математичне та комп'ютерне моделювання процесів управління для створення середовища системи управління проектами.

**Наукова новизна отриманих результатів:**

- Подальшого розвитку дістав метод управління стартап-проектами, який, на відміну від існуючих, враховує індивідуальні потреби та особливості малих компаній, що дозволить системі адаптуватися до сучасного динамічного бізнес-середовища.
- Подальшого розвитку дістала модель системи управління стартапами, яка, на відміну від існуючих, враховує особливості організації роботи зі стартапами, сучасні стратегії моніторингу й оптимізації робочих процесів, формування графіків і звітів, що дозволяє покращити процес керування проектами.

**Практична цінність отриманих результатів.** Практичне значення магістерської кваліфікаційної роботи полягає у можливості практичного використання розробленої системи управління стартап-проектами у приватному користуванні чи в спеціалізованих навчальних цілях.

**Особистий внесок здобувача.** Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто. У науковій роботі, опублікованій у співавторстві, автору належать розроблені моделі системи управління стартап-проектами.

**Апробація матеріалів магістерської кваліфікаційної роботи.**

Результати магістерської кваліфікаційної роботи доповідалися та обговорювалися на міжнародній науково-практичній Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ - 2023».

**Публікації.** Основні результати досліджень опубліковані в збірнику матеріалів міжнародної науково-практичної Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ - 2023» [6].

**Структура та обсяг роботи.** Магістерська кваліфікаційна робота складається зі вступу, п'яти розділів, висновків, списку використаних джерел, що містить 47 найменування, 4 додатки. Робота містить 32 ілюстрації, 11 таблиць.

# **1 АНАЛІЗ СТАНУ РОЗВИТКУ СИСТЕМ УПРАВЛІННЯ СТАРТАП-ПРОЄКТАМИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ**

## **1.1 Аналіз стану розвитку систем управління стартап-проєктами**

Системи управління проєктами (Project Management Systems, PMS) стали однією з ключових складових сучасного бізнесу. Їхнє значення постійно зростає завдяки постійним змінам у бізнес-середовищі та вимогам до ефективного керування проєктами для досягнення бажаних результатів.

Серед тенденцій розвитку PMS особливо слід відзначити перехід від традиційних локальних систем управління проєктами до онлайн-платформ. Це рішення дозволяє користувачам працювати в будь-якому місці, забезпечуючи доступність проєктної інформації для всіх членів команди, навіть якщо вони знаходяться на різних континентах. Інтеграція з різними іншими інструментами, такими як електронна пошта та календарі, сприяє покращенню комунікації та координації роботи всієї команди.

Залучення інструментів аналітики та бізнес-інтелекту до PMS стає звичним явищем. За допомогою аналітичних засобів користувачі можуть проводити глибокий аналіз даних про проєкти, що дозволяє приймати більш обґрунтовані управлінські рішення та планувати подальші дії.

Мобільні додатки для управління проєктами роблять системи PMS більш доступними та зручними. Вони дозволяють користувачам вести облік та керувати проєктами прямо зі своїх смартфонів і планшетів, надаючи їм можливість реагувати на поточні події та виправляти ситуацію в режимі реального часу. Це особливо важливо в умовах сучасного бізнесу, де важко передбачити, де саме буде потрібно втручання.

Останньою, але не менш важливою тенденцією є збільшення кількості користувачів PMS. Більше компаній та стартапів визнають важливість ефективного управління проєктами для досягнення своїх цілей та збільшення

конкурентоспроможності. Тому системи управління проектами стають важливими інструментами для підвищення продуктивності та ефективності бізнесу.

Зростання складності проєктів також впливає на розвиток PMS. Сучасні проєкти вимагають більше уваги до управління ресурсами, бюджетами та виконанням завдань. PMS стають все більше інтегрованими системами, які допомагають ефективно керувати цими аспектами.

Отже, системи управління проектами постійно змінюються та вдосконалюються, адаптуючись до зростаючих потреб сучасного бізнесу. Їхня роль вдосконалення процесів управління проектами і підвищення конкурентоспроможності компаній стає все більшою, і вони продовжують впливати на розвиток бізнесу.

## **1.2 Порівняльний аналіз аналогів**

Одним із найпоширеніших методів наукового дослідження є метод порівняння. Коли маємо справу з об'єктами, які мають певну подібність або спільні характеристики, метод порівняння стає важливим інструментом. Під час використання цього методу окремо розглядаються об'єкти, які потрібно порівняти, і виділяються певні критерії та ознаки, за якими можна здійснити порівняння.

Під час порівняння визначаються як спільні риси, так і відмінності між об'єктами. Проаналізувавши результати дослідження, дослідник може зробити висновки щодо переваг та недоліків об'єктів, що порівнюються. Це дозволяє ідентифікувати аспекти, які потребують подальшого вдосконалення.

Порівняльний аналіз аналогів системи управління стартап-проєктами допоможе розглянути різні аспекти кожної системи більш детально. Оцінюючи переваги і обмеження кожного інструмента, можна прийняти більш інформоване

рішення щодо вибору системи, яка найкраще відповідає потребам вашого стартапу.

Trello, завдяки своєму простому та інтуїтивному інтерфейсу, є ідеальним вибором для малих команд та стартапів з невеликим бюджетом [7]. За допомогою дошок, списків та карток користувачі можуть легко створювати та відстежувати завдання (рисунок 1.1).

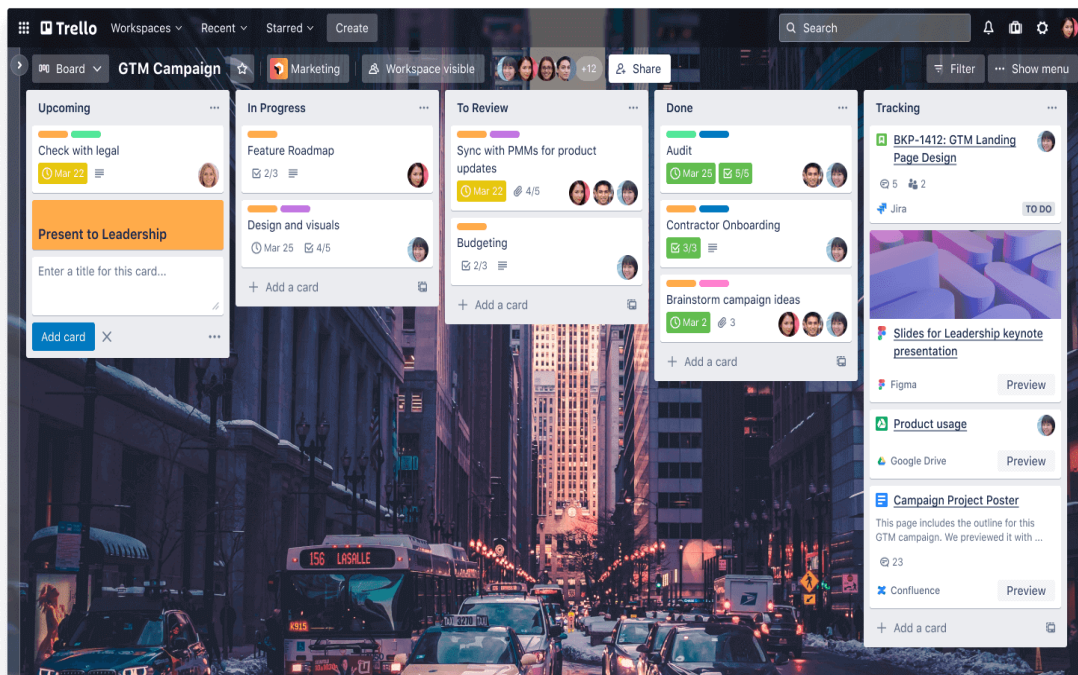


Рисунок 1.1 – Інтерфейс програми Trello

Ця система безкоштовна для базового використання, що важливо для малих підприємств, але вона також пропонує платні тарифи для додаткової функціональності.

Однак, важливо зазначити, що Trello може бути недостатньою для великих чи більш складних проєктів, і вона не надає засобів для управління ресурсами та аналітики.

Asana – це потужна система управління проєктами, яка надає розширені можливості для планування та відстеження проєктів (рисунок 1.2).

Вона функціонально підходить для середніх та великих підприємств, а також для стартапів, які мають більші амбіції [8].

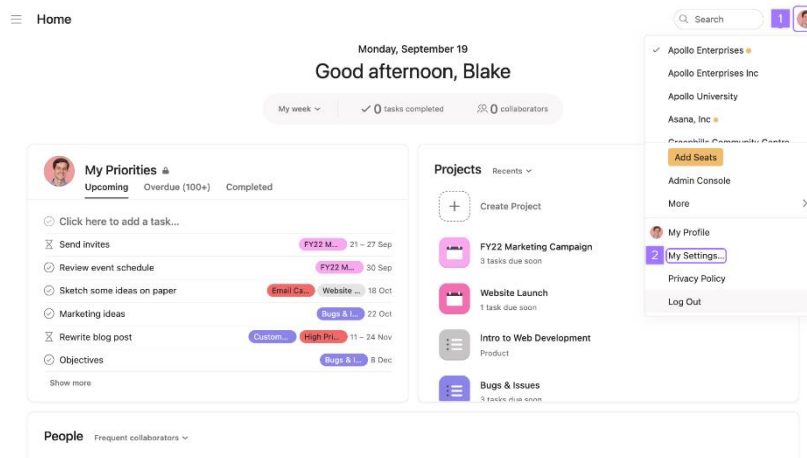


Рисунок 1.2 – Інтерфейс програми Asana

Asana дозволяє керувати ресурсами, встановлювати бюджети та вести облік витрат, що є важливими аспектами для більш вимогливих проєктів. Проте вища ціна на користування системою може бути фінансово обтяжливою для невеликих стартапів.

Monday.com [9] відзначається своєю гнучкістю та можливістю адаптації до різних видів проєктів (рисунок 1.3). Ця система надає засоби для створення кастомізованих дошок та відстеження завдань.

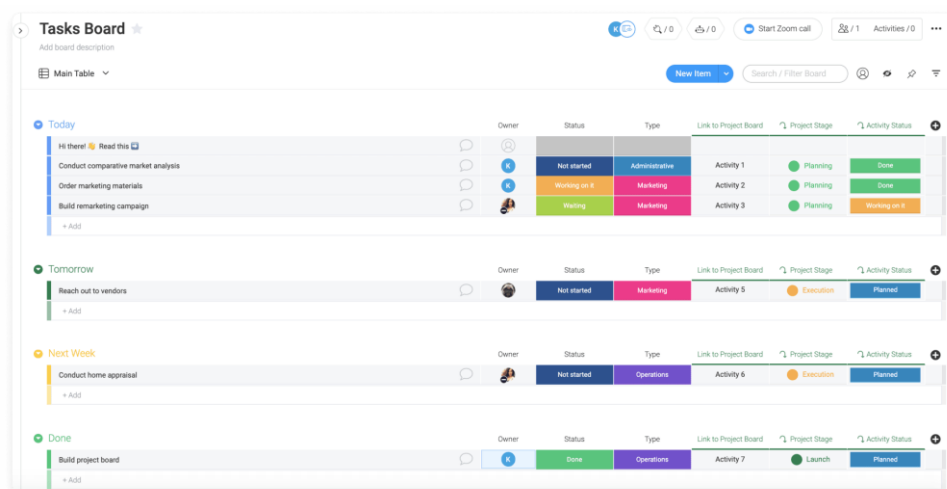


Рисунок 1.3 – Інтерфейс програми Monday.com

Вона пропонує широкий вибір шаблонів та готових рішень, що спрощує початок роботи. Monday.com також включає інструменти для співпраці команди, що робить її привабливою для колективної роботи. Але вищі витрати на користування можуть змушувати деякі стартапи шукати більш бюджетні рішення, особливо, коли вони тільки починають свою діяльність.

Після ознайомлення з аналогами стало очевидним, що кожен додаток має свої особливості та тонкощі. Проте, багато функцій є спільними для більшості програмних продуктів. Тому можна виділити найнеобхідніші критерії для порівняння цих продуктів. Результати порівняльного аналізу були внесені в таблицю 1.1.

Таблиця 1.1 – Порівняльний аналіз аналогів

Характеристика	"Система управління стартап-проектами"	Trello	Asana	Monday.com
Спеціалізація	Стартапи і малі компанії	Організації та невеликі команди	Об'ємні стартап команди	Малі стартапи та малі команди
Простота використання	Так	Так	Так	Так



Продовження таблиці 1.1

<b>Функціональність</b>	Управління проектами, облік ресурсів, аналітика, облік бюджету	Базовий облік завдань	Розширений облік завдань, управління ресурсами, аналітика	Гнучкий облік завдань, управління ресурсами
<b>Характеристика</b>	"Система управління стартап-проектами"	<b>Trello</b>	<b>Asana</b>	<b>Monday.com</b>
<b>Ціна</b>	Залежно від планів	Безкоштовний тариф	Безкоштовний тариф	Вищий ціновий поріг
<b>Інтеграція з іншими додатками</b>	Кілька розширених інтеграцій	Обмежена кількість	Багато інтеграцій	Багато інтеграцій
<b>Спеціалізовані інструменти</b>	Оповідення	Відсутні	Відсутні	Відсутні
<b>Гнучкість і адаптованість</b>	Потенційно висока	Відсутні	Потенційно висока	Потенційно висока

Аналіз таблиці порівняння характеристик дає можливість зробити висновок, що розробка власного програмного продукту є доцільною. Цей підхід дозволить створити продукт, який перекидає недоліки існуючих реалізацій та має більше функціональних можливостей.

### **1.3 Аналіз методів розв'язання задачі**

Розробка методів розв'язання поставленої задачі є важливим завданням, оскільки стартапи часто опиняються в умовах невизначеності та ризику. Ефективна система управління дозволяє стартапу краще планувати, виконувати та вдосконалювати свої проєкти.

Проведемо аналіз деяких методів та засобів, які можна використовувати для розробки системи управління стартап-проєктами:

#### **1. Методології управління проєктами:**

- Agile. Agile-підходи, такі як Scrum та Kanban, широко використовуються в стартапах [10]. Вони дозволяють гнучко реагувати на зміни та швидко адаптувати проєкт до реальних умов.
- Lean Startup. Цей підхід спрямований на мінімізацію витрат та максимізацію вивчення відгуків користувачів [11]. Lean Startup рекомендує швидко створювати прототипи та тестувати їх на ринку[12].

#### **2. Програмні засоби для управління проєктами:**

- JIRA. Це популярний інструмент для управління проєктами та задачами [13]. Він дозволяє створювати задачі, розподіляти їх, відстежувати прогрес та спільно працювати над проєктом.
- Trello. Ще один інструмент для управління завданнями, який дозволяє використовувати дошки для організації роботи та переміщення завдань між колонками.

#### **3. Методи аналізу та планування:**

- SWOT-аналіз. Допомагає визначити сильні та слабкі сторони стартапу, а також можливості та загрози на ринку [14].
  - Бізнес-планування (Business Dashboard) [15]. Розробка детального бізнес-плану допомагає зрозуміти стратегію розвитку стартапу та розподіл ресурсів [16].
4. Моніторинг та вимірювання результатів:
- KPI (Key Performance Indicators) [17]. Визначення ключових показників результативності дозволяє стежити за прогресом та оцінювати успішність проєкту.
  - Пульс бізнесу (Business Dashboard) [18]. Створення панелі інструментів для візуалізації даних допомагає приймати управлінські рішення на основі об'єктивних даних.
5. Комунікація та співпраця:
- Збори та звіти. Регулярні зустрічі та звіти дозволяють команді стежити за прогресом і обговорювати проблеми [19].
6. Ризик-менеджмент:
- Аналіз ризиків. Визначення можливих ризиків та розробка планів їх управління [20].
7. Навчання та підвищення кваліфікації:
- Навчальні програми та курси. Забезпечення команди стартапу можливістю навчання та розвитку навичок.
  - Система зворотного зв'язку. Сприяє навчанню на основі досвіду та помилок.
8. Управління ресурсами:
- Бюджетування та фінансовий контроль. Визначення і оптимізація витрат.
  - Ресурси людського капіталу. Ефективне управління командою та персоналом.

Важливою властивістю такої системи є можливість стартапу вибирати методи та інструменти, які найкраще відповідають його потребам та стратегії.

Гнучкість у підході до управління є ключовою, оскільки вона дозволяє підлаштовувати систему відповідно до конкретних умов і стратегії стартапу. Такий підхід дозволяє стартапам ефективно впроваджувати інновації та швидко адаптуватися до змін у зовнішньому середовищі.

Постійна адаптація та аналіз системи управління проектом стають необхідними умовами для досягнення успіху в стартап-середовищі. Активний моніторинг та оновлення системи управління допомагають виявляти та усувати можливі проблеми, а також підтримують постійне вдосконалення процесів.

Такий підхід до управління проектами в стартапі сприяє створенню адаптивної та ефективної системи, яка допомагає досягати максимальних результатів в умовах невизначеності та динамічності стартап-середовища.

#### **1.4 Постановка задач дослідження**

Проаналізувавши питання розробки системи управління стартап-проектами, було визначено завдання, які необхідно виконати для розробки програмного додатку.

Основними задачами роботи є:

- визначити функціонал системи для управління стартап-проектами та засоби її реалізації;
- розробити метод управління стартап-проектами;
- розробити модель системи управління стартапами;
- розробити алгоритми роботи системи;
- розробити програмне забезпечення системи для управління стартап-проектами;
- провести тестування роботи системи.

## **1.5 Висновок**

У першому розділі було проведено аналіз стану розвитку систем управління проектами на сучасному етапі. Розглянуто існуючі аналоги, зокрема, додатки «Trello», «Asana» та «Monday.com», проведено їх порівняння з розроблюваним програмним продуктом.

Результати порівняння дозволили зробити висновок про доцільність розробки власного програмного додатку. Основними перевагами розроблюваного продукту в порівнянні з існуючими є підтримка української мови, надання користувачам простого та інтуїтивного інтерфейсу, розробка системи з урахуванням конкретних потреб стартапів та малих компаній.

Також у першому розділі було проведено аналіз існуючих методів для вирішення задачі. В цьому контексті були сформульовані основні завдання, які необхідно виконати для успішної розробки програмного додатку.

## **2 РОЗРОБКА МЕТОДУ, МОДЕЛЕЙ ТА АЛГОРИТМІВ РОБОТИ СИСТЕМИ УПРАВЛІННЯ СТАРТАП-ПРОЄКТАМИ**

### **2.1 Аналіз інформаційного забезпечення системи**

Система управління стартап-проєктами спеціалізується на обслуговуванні стартапів та малих компаній. Це означає, що вона розроблена так, щоб відповідати конкретним потребам цієї аудиторії, яка, зазвичай, має обмежений бюджет та необхідність раціонально використовувати ресурси. Система спрощує управління проєктами та завданнями для стартапів, які можуть не мати великого досвіду у цій галузі.

Однією з ключових переваг системи є її простота використання. Інтерфейс системи має бути інтуїтивно зрозумілим, що дозволяє користувачам швидко орієнтуватися та розпочати роботу без необхідності витратити значний час на навчання. Це особливо важливо для стартапів, де швидкість початку роботи має критичне значення.

Система управління стартап-проєктами пропонує широкий функціонал для керування проєктами. Користувачі можуть створювати проєкти та додавати до них завдання та підзавдання. Деталі проєкту, такі як описи, бюджети та строки, можуть бути зручно вказані. Додатково система дозволяє відстежувати прогрес виконання завдань, використовуючи графіки та діаграми.

Забезпечення безпеки та конфіденційності даних є важливим аспектом для стартапів, особливо тих, які працюють над інноваційними проєктами та можуть мати конфіденційну інформацію. Система має забезпечувати високий рівень захисту даних та мати відповідні механізми для зберігання й обробки конфіденційної інформації.

Проєкт повинен бути розроблений так, щоб мати потенціал для зростання та адаптації до змінних потреб та розміру стартапів. Необхідно передбачити

можливість розширити функціональність системи та внести зміни для відповіді на конкретні потреби аудиторії.

Планування регулярних оновлень та надання якісної підтримки для користувачів є важливим аспектом для підтримки задоволеності клієнтів та розвитку продукту.

Система має підтримувати інтеграцію з багатьма популярними інструментами та додатками, що полегшує обмін даними та співпрацю з іншими системами, такими як електронна пошта, календарі, чат-системи та інші. Це зробить систему більш універсальною та зручною для користувачів.

Необхідно врахувати потреби стартапів у спеціалізованих інструментах, які допомагають вести облік робочого часу, витрат та ресурсів. Це сприятиме більш ефективному управлінню ресурсами та бюджетом та зробить систему більш привабливою для стартапів [21].

Дана система повинна відзначатися гнучкістю та адаптованістю, що дозволить користувачам налаштовувати її під свої потреби. Необхідною є можливість налаштовувати завдання, проєкти та інші елементи відповідно до конкретних потреб та процесів свого стартапу.

Система створюється з урахуванням співпраці та комунікації між учасниками проєктів. Вона має надавати засоби для обговорення завдань, коментування та прикріплення файлів, що полегшить обмін інформацією та спільну роботу команди. Робота над проєктами стає більш спільною та ефективною завдяки цим можливостям.

Усі ці аспекти зроблять дану систему управління стартап-проєктами цінною для стартапів та малих компаній, допомагаючи їм успішно організувати, виконувати та керувати своїми проєктами.

## 2.2 Розробка моделей системи управління стартапами

Створення та ефективне управління проектами визнаються стратегічною основою для досягнення високого рівня конкурентоспроможності та стійкості в умовах сучасного бізнес-середовища.

Цей комплексний підхід стає не просто інструментом вирішення завдань, але й керованим процесом, який формує культуру інновацій та підтримує стратегічні цілі компанії.

Створення та впровадження проєктів є ключовим механізмом для адаптації організації до постійних змін у бізнес-середовищі. Це дозволяє компанії не лише реагувати на виклики, але й активно формувати своє майбутнє, впроваджуючи інновації та забезпечуючи довгостроковий успіх.

Проєктний підхід допомагає зміцнювати внутрішню комунікацію та взаємодію між відділами.

Колективна робота над проєктами сприяє обміну знаннями та досвідом, що сприяє кращому розумінню завдань та формуванню сильних, спроможних команд [22].

Інтеграція проєктного менеджменту визнається не лише як стратегічний елемент, але і як фундамент для розвитку агільної та готової до змін організації. Цей підхід дозволяє не просто виживати в умовах змін, але і приводить до створення організації, яка приречена на постійний успіх та лідерство в своїй галузі.

Загальна модель роботи системи управління стартап-проєктами наведена на рис. 2.1.



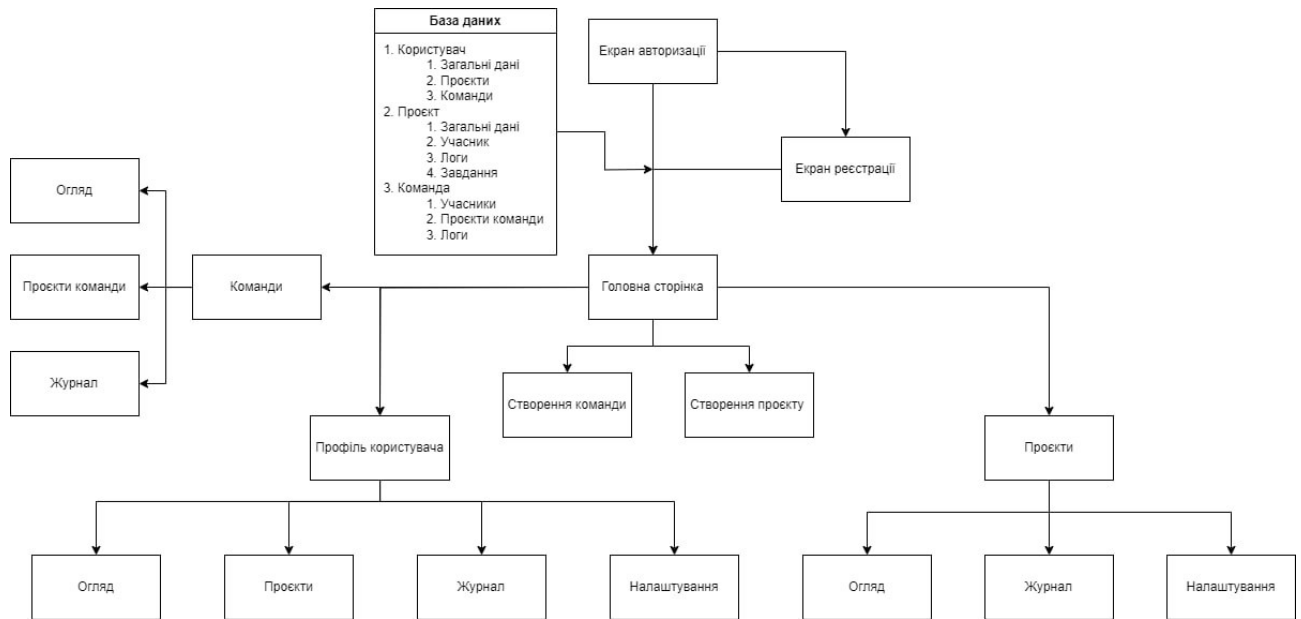


Рисунок 2.1 – Загальна модель роботи системи управління стартап-проектами

Розроблена модель системи управління стартап-проектами розпочинається зі збору важливих даних проєктів: інформації про учасників, статус виконання завдань, витрати та інші деталі. Отримані дані обробляються для подальшого використання.

Далі, на основі оброблених даних, генеруються звіти та графіки, які відображають стан проєктів. Ці звіти можуть містити інформацію про прогрес, фінансові витрати, а графіки – візуально демонструвати динаміку розвитку проєктів.

Після генерації звітів та графіків проводиться аналіз отриманих даних для оцінки ефективності проєктів. Цей етап дозволяє визначити досягнення поставлених цілей, виявити ризики та проблеми, а також знайти можливості для покращення [23].

Після успішного аналізу результатів, вони подаються у вигляді звітів, презентацій, тобто формі, зручній для розуміння та прийняття управлінських

рішень. Це дозволяє зацікавленим сторонам оцінити стан проєктів та вжити необхідних заходів для покращення їх ефективності.

Блок – схема алгоритму роботи розробленої моделі системи управління стартап-проєктами зображена на рисунку 2.2.

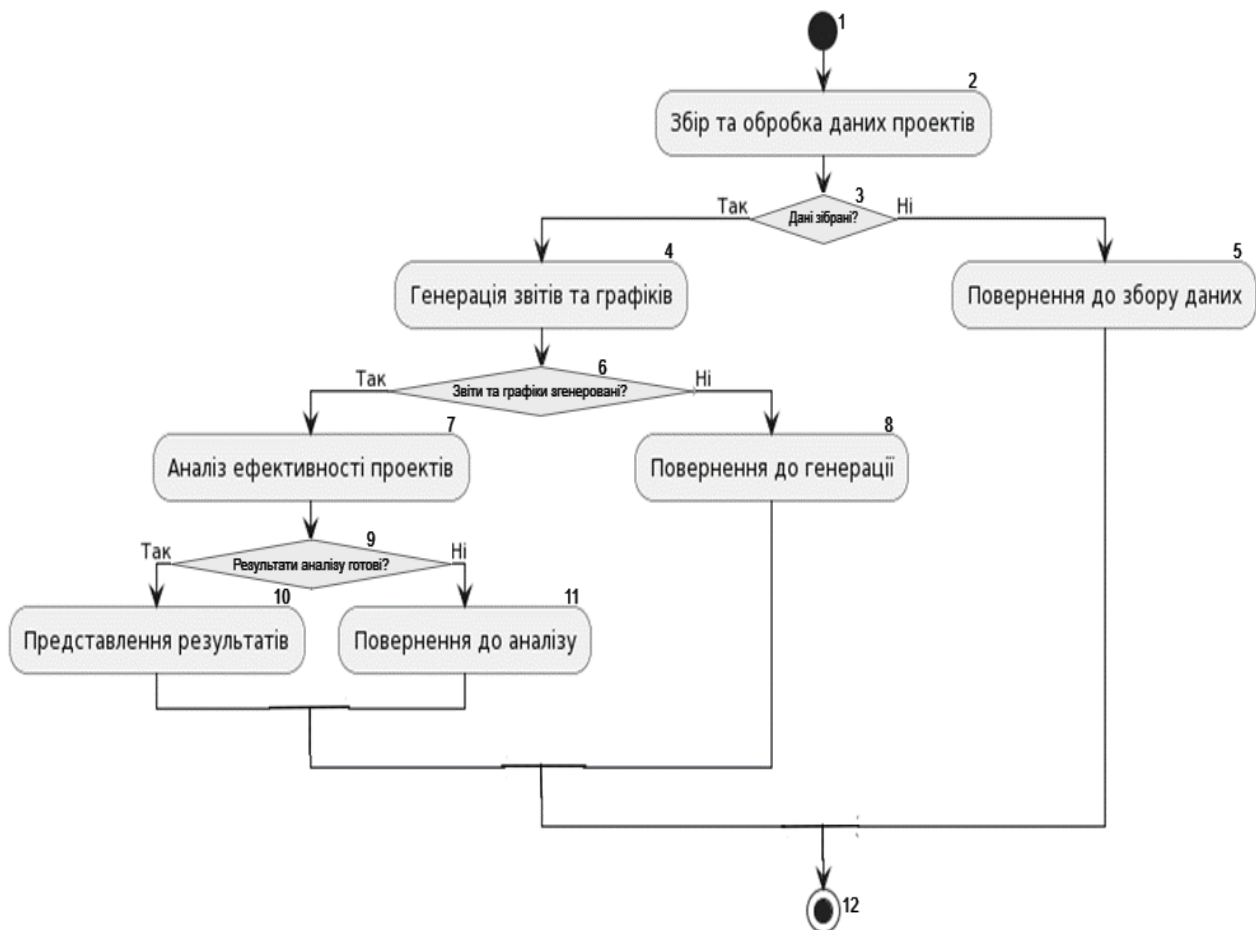


Рисунок 2.2 – Блок-схема алгоритму роботи моделі системи управління стартап-проєктами

Для взаємодії системи управління з базою даних було обрано Hibernate та JBDC.

На рисунку 2.3 наведено узагальнену модель бази даних розроблюваної системи.

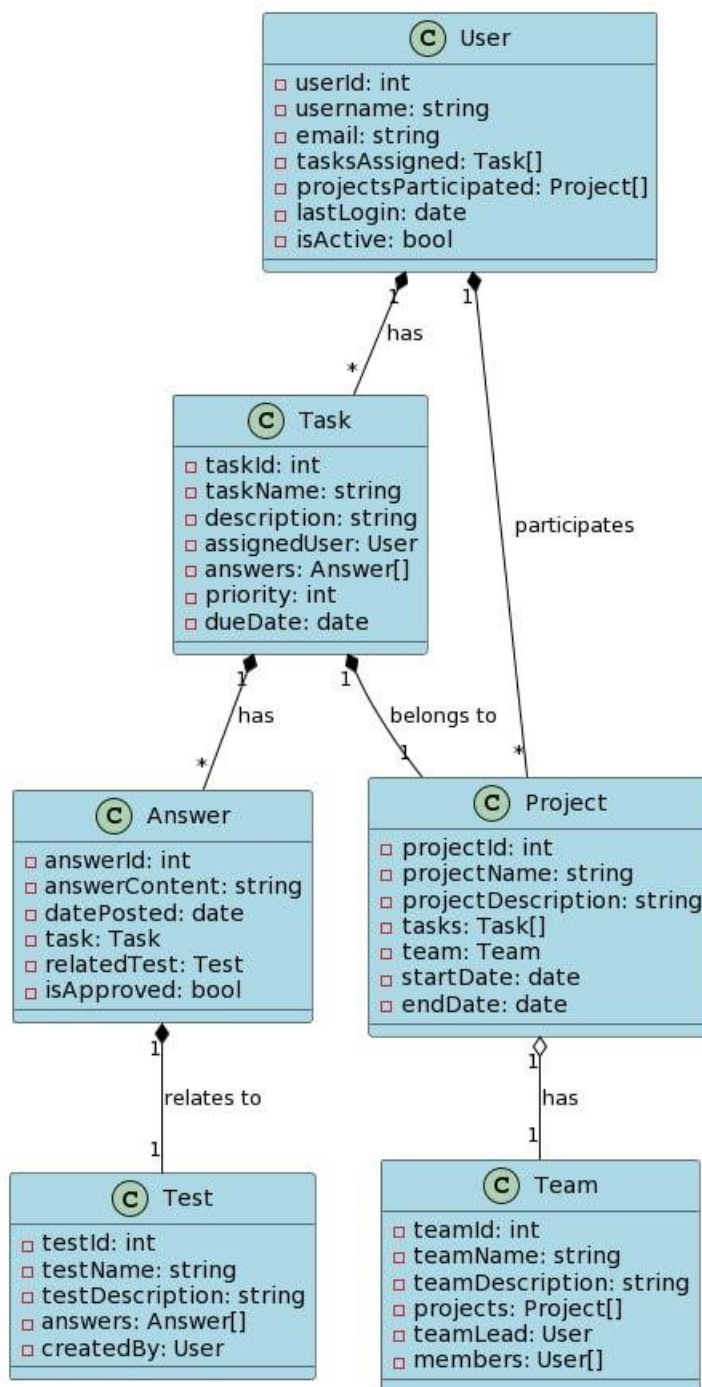


Рисунок 2.3 – Узагальнена модель бази даних програми

Розробка моделі взаємозв'язків у базі даних є важливим етапом у створенні "Системи управління стартап-проектами", оскільки вона визначає, як дані взаємодіють між собою та як вони зберігаються в базі даних [24]. Модель взаємозв'язків визначає структуру даних, таблиці та їхні зв'язки, що допомагає

ефективно організувати і зберігати інформацію про проекти, завдання, користувачів та інші сутності системи (рисунок 2.4).

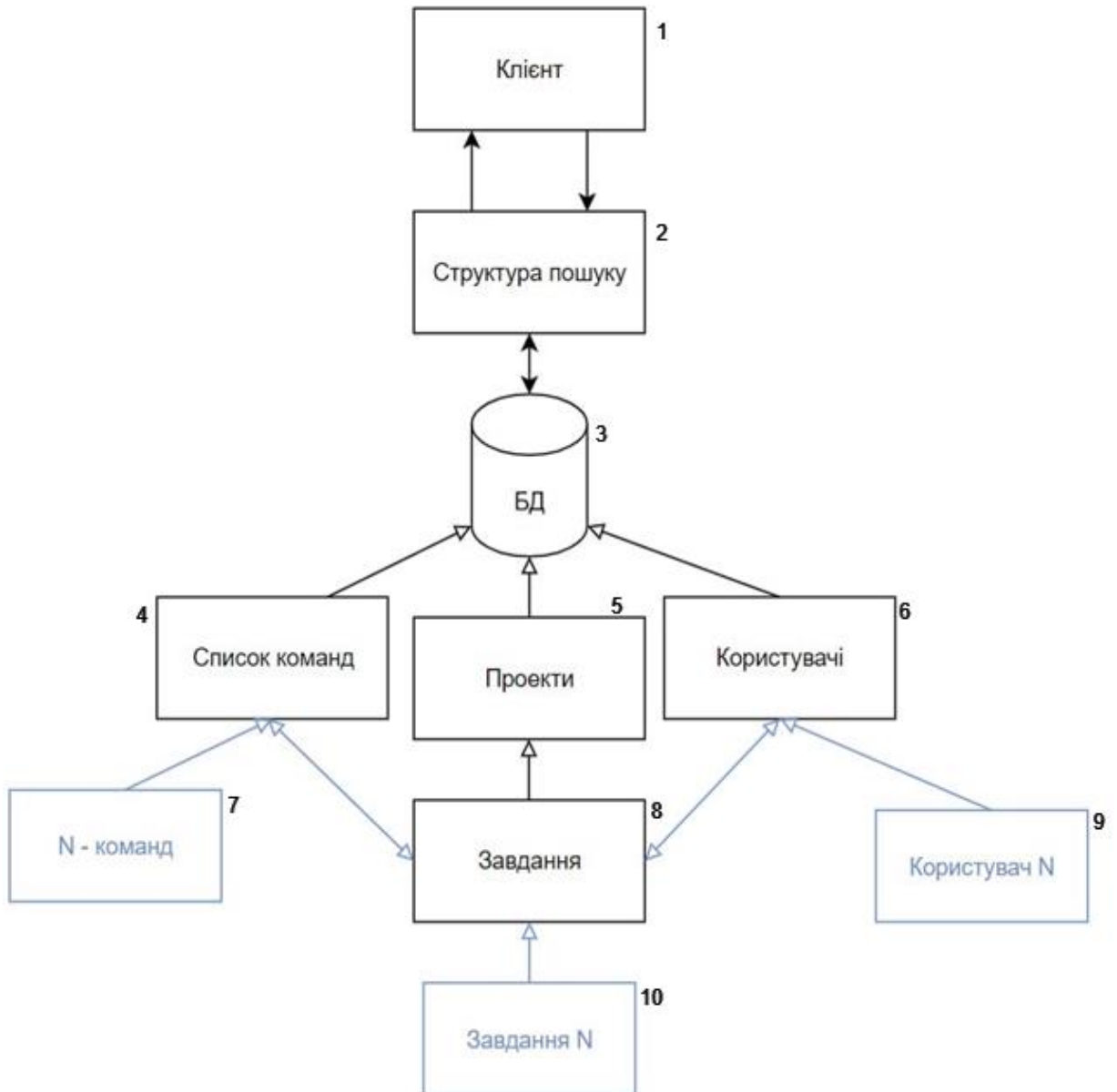


Рисунок 2.4 – Структура пошуку даних

Першим кроком у розробці моделі взаємозв'язків є ідентифікація основних сутностей, які включаються в систему. В системі управління стартап-проектами це є "Проекти", "Завдання", "Користувачі", "Команди" та інші [25].

Розглянемо деякі ключові сутності, їхні атрибути та взаємозв'язки:

1. Проекти. Ця сутність включає окремі стартап-проекти і має атрибути, такі як назва проекту, опис, бюджет, дата створення та інші характеристики.
2. Управління проектами. Завдання пов'язані з проектами і мають атрибути, які описують їхню природу та стан виконання. Це включає в себе назву завдання, опис, дедлайн виконання, статус та інші параметри.
3. Користувачі. Користувачі системи мають свої власні атрибути, такі як ім'я, електронна пошта, пароль для авторизації та роль (наприклад, менеджер проекту або звичайний користувач).
4. Команди. Деякі проекти можуть мати команди, які включають різних користувачів. Ця сутність акумулює в собі назву команди, список учасників та інші характеристики команди.
5. Взаємозв'язки. Взаємозв'язки визначають способи, якими ці сутності взаємодіють. Наприклад, кожне завдання пов'язане з конкретним проектом і відповідальним користувачем. Команди можуть бути пов'язані з проектами, де один проект може мати багато команд.

Далі, потрібно визначити атрибути для кожної сутності, тобто ті дані, які необхідно зберігати для кожної сутності. Наприклад, для сутності "Проекти" це може бути назва проекту, опис, бюджет та інші важливі властивості.

Після цього потрібно визначити взаємозв'язки між сутностями. Наприклад, "Проекти" можуть бути пов'язані з "Завданнями", де кожен проект може мати багато завдань, а кожне завдання може бути призначене конкретному користувачу. Такі взаємозв'язки визначаються за допомогою зовнішніх ключів, які вказують на зв'язок між записами в різних таблицях бази даних.

Для забезпечення нормалізації та ефективного пошуку даних в базі даних, важливо дотримуватися правил реляційної моделі даних. Це включає в себе розбиття даних на окремі таблиці, уникання повторення даних та правильний вибір типів даних для кожного атрибуту [26].

Останнім кроком є створення самої бази даних, яка відповідає розробленій моделі взаємозв'язків (рисунок 2.5). Це включає в себе створення таблиць, визначення індексів та зовнішніх ключів, а також налаштування обмежень і правил для забезпечення цілісності та безпеки даних.

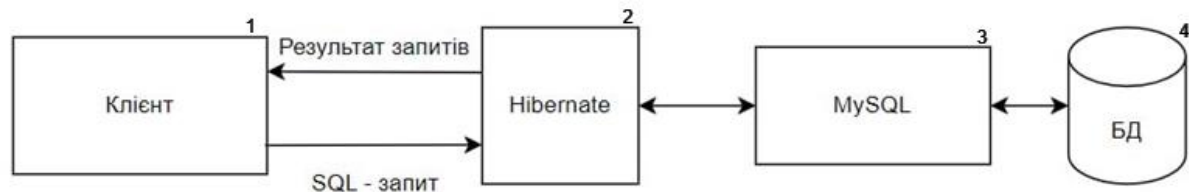


Рисунок 2.5 – Модель SQL-запитів

Розробка моделі взаємозв'язків у базі даних для системи управління стартап-проектами відіграє критичну роль у визначенні структури та взаємодії даних. Цей процес встановлює спосіб організації та зберігання інформації, що прямо впливає на продуктивність та функціональність системи.

Під час розробки моделі визначаються ключові зв'язки між різними компонентами системи, уникнення повторення даних та забезпечення їх цілісності. Це допомагає створити структуровану базу даних, яка враховує особливості стартап-середовища, де швидкі зміни та гнучкість важливі для успішного управління проектами.

Розробка моделі взаємозв'язків враховує потреби стартапу в ефективному зберіганні та обробці даних. Цей процес покликаний створити логічну структуру, яка дозволяє забезпечити ефективний облік та аналіз проектних параметрів.

Такий підхід до розробки бази даних сприяє покращенню якості управління проектами в стартап-середовищі та забезпечує гнучкість системи для швидкої адаптації до змін в умовах розвитку бізнесу.

### 2.3 Розробка методу управління стартап-проектами

Розроблений метод – це система принципів постановки цілей, яка дозволяє формулювати цілі таким чином, щоб вони були конкретними, вимірюваними, досяжними, відповідними та обмеженими у часі.

Специфічність полягає в тому, що ціль повинна бути чіткою, конкретною і ясною для всіх учасників проекту. Це допомагає уникнути розбіжностей у розумінні того, що саме має бути досягнуто.

Вимірюваність полягає у можливості кількісно вимірювати прогрес у досягненні цілі. Чітко визначені метрики або параметри дозволяють об'єктивно оцінювати прогрес і визначати, наскільки близько до досягнення мети знаходиться проект.

Досяжність вказує на реалістичність та досяжність цілі. Ціль повинна бути реалістичною в умовах, які доступні для досягнення, враховуючи ресурси та обмеження.

Відповідність означає, що ціль має відповідати стратегії та загальним цілям проекту або організації. Цілі мають бути узгоджені та співвідносні з загальними напрямками діяльності.

Обмеженість у часі вказує на те, що цілі повинні бути зв'язані з чітко визначеними строками. Це стимулює відчуття терміновості та відповідальності у досягненні цілей у визначений період часу .

Використання розробленого методу роботи системи управління стартапами сприяє більш точному, узгодженому та систематичному підходу до постановки цілей, що полегшує їхнє досягнення та контроль над процесом реалізації проектів чи завдань.

Запропонований метод роботи системи управління стартап-проектами презентує технологію ефективної постановки та досягнення цілей для проектів. Вона відображає послідовність кроків, необхідних для успішного досягнення поставлених завдань та мети проекту.

На рисунку 2.6 зображено алгоритм методу управління стартап-проєктами за допомогою візуальної презентації кроків, які необхідно виконати для ефективного управління та досягнення мети проєкту.

Починаючи з визначення мети проєкту, алгоритм (рис. 2.6) відображає перехід до аналізу поточного стану проєкту для кращого розуміння його вихідних позицій.

Далі процес переходить до специфікації цілей за розробленим методом – встановлення конкретних, вимірюваних, досяжних, відповідних та часово обмежених цілей.

Алгоритм процесу управління проєктами передбачає встановлення мети проєкту, моніторинг роботи над проєктом, аналіз поточного стану, специфікацію цілей на всіх етапах розробки проєкту, уточнення деталей в ітераційному режимі.

Формування звіту за проміжний період дозволить скорегувати цілі проєкту та вжити своєчасних заходів для успішного їх досягнення.

Важливу роль тут відіграє можливість відслідковування вимірюваного процесу, який засобами програми можемо візуалізувати у вигляді графіків щоденної активності виконавців проєкту і успішності виконання ними поставлених задач.

Розподіл завдань між учасниками проєкту з урахуванням їх спеціалізацій дозволить команді ефективно працювати над проєктом та сприятиме успішному розвитку стартапу.

Оцінка результатів виконання проєкту на проміжних етапах його розвитку дозволяє проаналізувати причини невдач і зорієнтувати персонал на активацію ресурсів для усунення недоліків.

Корегування щоденних завдань для виконавців проєкту з урахуванням попереднього досвіду дає змогу підвищити продуктивність їх роботи.



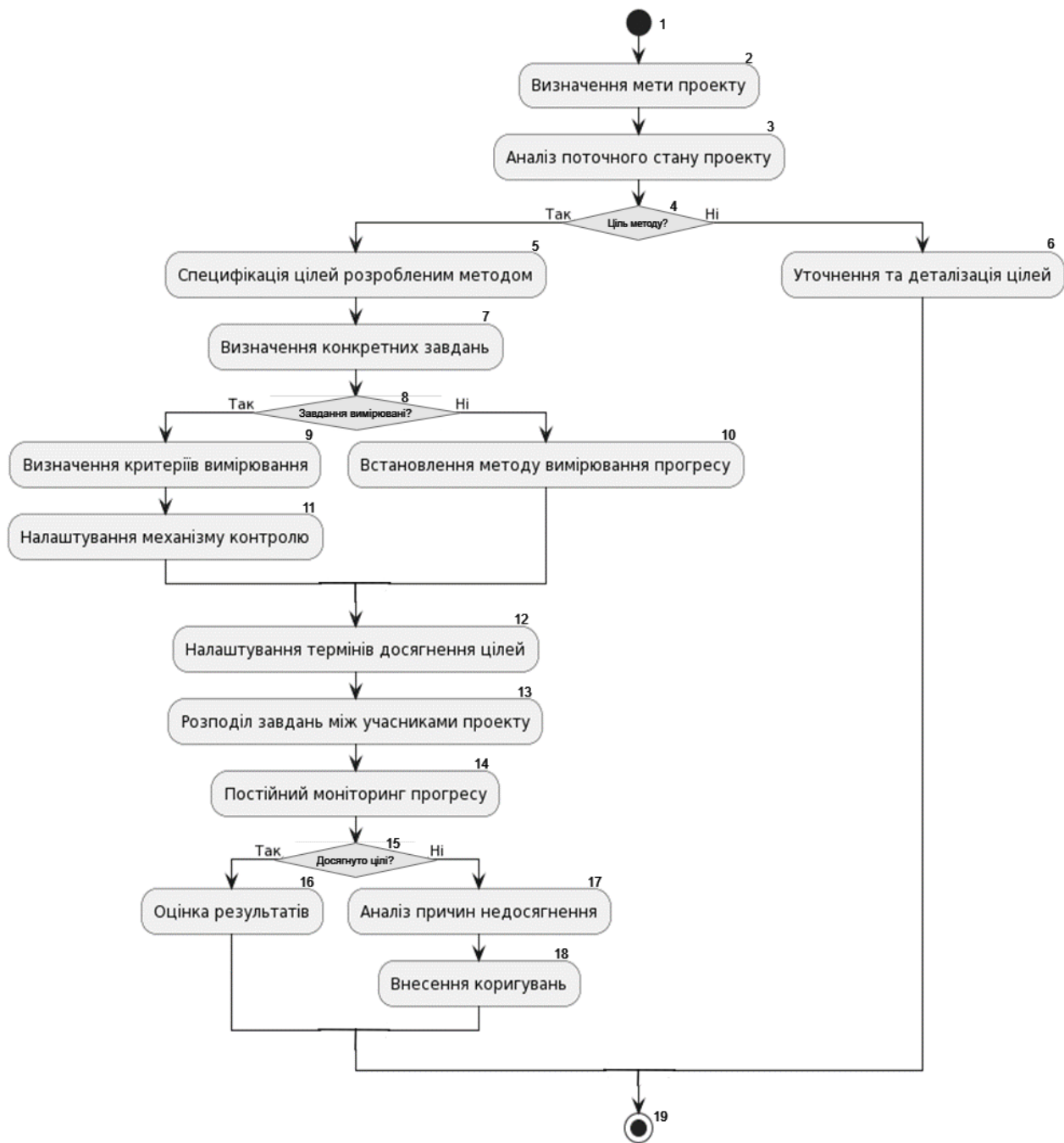


Рисунок 2.6 – Блок-схема алгоритму роботи методу управління стартап-проектами

Далі відбувається визначення конкретних завдань, які необхідно виконати для досягнення цілей, і встановлення критеріїв вимірювання прогресу у їхньому досягненні.

Потім здійснюється налаштування термінів досягнення цілей та розподіл завдань між учасниками проекту, що допомагає кожному знати свою роль та обов'язки.

Останні кроки включають постійний моніторинг прогресу та оцінку результатів з подальшим коригуванням та внесенням змін, якщо це необхідно для досягнення цілей проекту.

Запропонований підхід у сфері систем управління стартап-проектами включає в себе такі елементи:

1. Емоційний інтелект. Додано врахування аспектів емоційного інтелекту у процес управління. Враховано мотивацію команди шляхом підтримки позитивного робочого середовища та стресостійкості.
2. Гнучкі методики. Використані гнучкі методи управління проектами, що дозволить ефективніше взаємодіяти зі змінами та швидко адаптуватися до нових умов.
3. Інноваційний менталітет. Введені етапи аналізу та імплементації інновацій, спрямовані на підвищення ефективності та конкурентоспроможності.
4. Взаємодія з зацікавленими сторонами. Задіяно активну взаємодію з усіма зацікавленими сторонами, включаючи замовників, інвесторів та користувачів. Встановлено механізми для збору та оцінки їхнього фідбеку, який впливає на процес прийняття рішень.
5. Культура інноваційного навчання. Створено етапи для постійного навчання та розвитку команди. Система підтримує культуру навчання та обміну знаннями, що дозволяє команді ефективно реагувати на зміни та вдосконалювати свої навички.

Метод управління стартап-проєктами включає такі дії:

1. Перший етап включає створення нового проєкту та заповнення його даними через виклик методів `createNewProject()` та `setTextFieldsData()`. Кнопка "Створити новий проєкт" ініціює цей процес.
2. Далі обираємо тип проєкту зі списку за допомогою методу `selectProjectType()`, а також встановлюємо часу робот за допомогою `MaterialTimePicker` через метод `showTimePicker()`.
3. Наступний етап включає додавання тегів до проєкту з використанням відповідного `ScrollState` через метод `addTags()`. Також включаємо вибір фотографій з галереї та їх завантаження на сервер за допомогою методу `pickAndUploadPhotos()`.
4. Отримання URL завантажених фотографій реалізуємо за допомогою функції `getPhotoDownloadUrls()`. Після цього відбувається створення проєкту з деталями та посиланнями на фотографії через метод `createProjectWithDetails()`. Також включаємо навігацію на інший екран після створення проєкту та збереження його в базі даних.
5. Якщо відбулося створення проєкту, учасникам за допомогою методу `notifyParticipantsAboutNewProject()` прийде відповідне повідомлення. Також включаємо генерацію посилання для спільного доступу до проєкту через метод `generateProjectLink()`. Перевірка прав доступу до проєкту здійснюється методом `checkAccessRights()`.
6. Проводиться перевірка прав доступу до проєкту перед його відкриттям з використанням методу `checkAccessRights()`. Це гарантує, що лише користувачі з відповідними правами матимуть доступ до проєкту.

Почати новий проєкт можна за допомогою головного вікна, достатньо натиснути на відповідну іконку.

Відкривається відповідне вікно з широким вибором можливостей для налаштування.

Метод додавання нового проєкту в базу передбачає послідовність виконання таких дій:

1. Створення нового проєкту через виклик методу `createNewProject()` при натисканні кнопки "Створити новий проєкт".
2. Заповнення даних проєкту через використання методу `setTextFieldsData()` для збереження даних проєкту в `MutableStateFlow`.
3. Обрання типу проєкту з `DropDownMenu` за допомогою методу `selectProjectType()` для вибору типу проєкту зі списку.
4. Вибір часу роботи за допомогою `MaterialTimePicker` та `showTimePicker()` для встановлення часу початку та завершення роботи.
5. Перевірка заповненості полів і активація кнопки "Продовжити", що робить метод `checkFieldsCompletion()` для перевірки оновленого стану і активації кнопки.
6. Додавання тегів з використанням відповідного `ScrollState` та `addTags()` для додавання тегів з можливістю прокрутки.
7. Вибір фотографій з галереї та їх завантаження на сервер, використовуючи `pickAndUploadPhotos()` для обрання та завантаження фотографій.
8. Отримання URL завантажених фотографій функцією `getPhotoDownloadUrls()` для отримання URL завантажених фотографій.
9. Створення проєкту з введеною інформацією і посиланнями на фотографії `createProjectWithDetails()` для створення проєкту з даними та посиланнями на фотографії.
10. Навігація на екран "Профіль" після створення проєкту, виклик методу `navigateToProfileScreen()` для переходу на відповідний екран після успішного створення проєкту.
11. Збереження створеного проєкту в базі даних для подальшого використання.

12. Відправка повідомлення про створення проєкту учасникам .`notifyParticipantsAboutNewProject()` для сповіщення учасників про створення нового проєкту.
13. Генерація посилання методом `generateProjectLink()` для спільного доступу до проєкту.
14. Проведення перевірки прав доступу до проєкту. Використовуємо метод `checkAccessRights()` для перевірки прав доступу до проєкту перед відкриттям.

Приклад коду реалізації додавання нового проєкту в базу даних подано на рисунку 2.7.

```

public void addNewProject(, String projectType, String[] projectTags, String[] projectPhotos) {
    if (projectName != null && !projectName.isEmpty() && projectType != null && !projectType.isEmpty()) {
        Project newProject = new Project(projectName, projectType, startTime, endTime);
        if (projectTags != null && projectTags.length > 0) {
            newProject.addTags(projectTags);
        }
        if (projectPhotos != null && projectPhotos.length > 0) {
            for (String photo : projectPhotos) {
                Photo uploadedPhoto = uploadPhotoToServer(photo);
                if (uploadedPhoto != null) {
                    newProject.addPhoto(uploadedPhoto);
                }
            }
        }
        boolean projectSaved = saveProjectToDatabase(newProject);

        if (projectSaved) {
            notifyUser("Проект успішно створено!");
        } else {
            notifyUser("Помилка при збереженні проєкту!");
        }
    }
}

```

Рисунок 2.7 – Модуль додавання нового проєкта в базу даних

Метод `addNewProject` призначений для створення нового проєкту у системі управління проєктами. Він приймає різні параметри, такі як назва проєкту, тип, теги, час роботи, фотографії тощо.

Метод `addNewProject` перевіряє обов'язкові поля та викликає відповідні функції для збереження основних даних проєкту, додавання тегів, завантаження фотографій на сервер і збереження проєкту в базі даних. Він також відправляє

повідомлення користувачеві про результат створення проекту (успішно чи з помилкою).

Блок-схема алгоритму додавання нового проекту в базу даних зображено на рисунку 2.8.

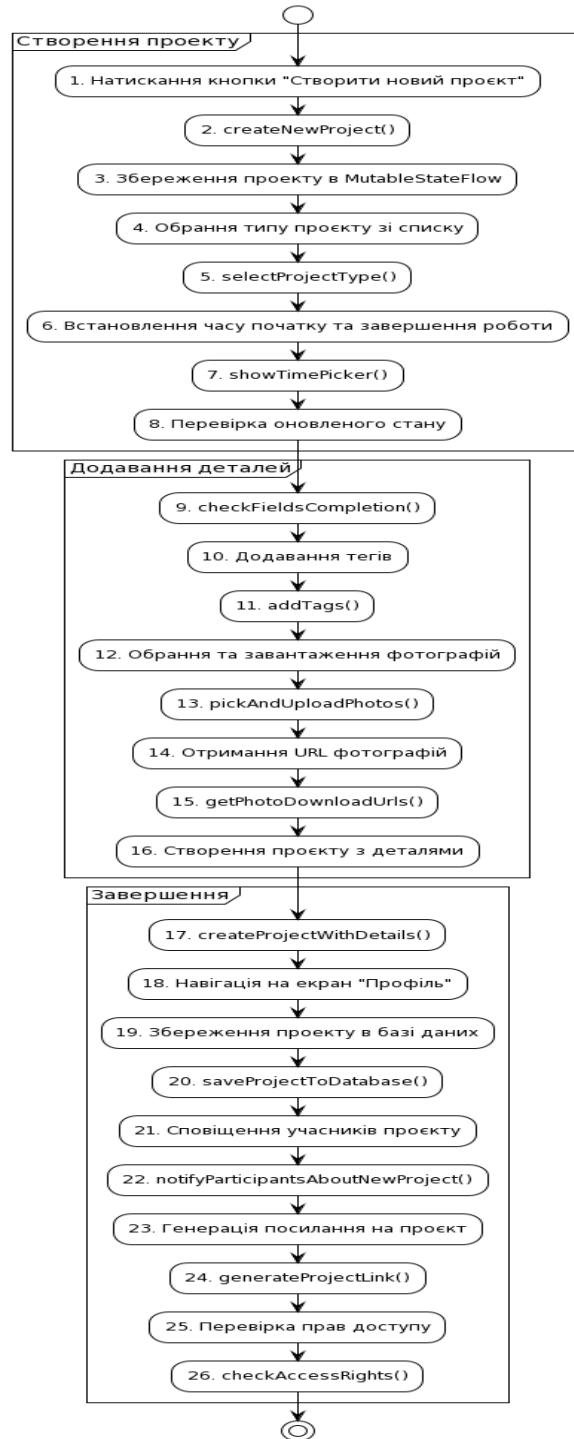
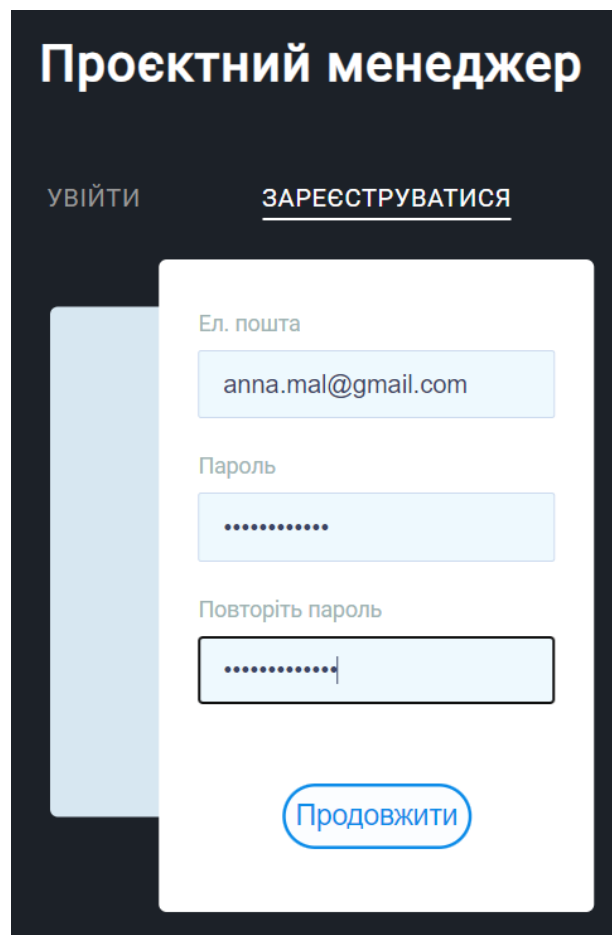


Рисунок 2.8 – Блок-схема алгоритму додавання нового проекту в базу даних

## 2.4 Розробка алгоритмів роботи системи

У розробці алгоритму реєстрації для системи управління стартап-проектами вирішальним є забезпечення безпеки, зручності та простоти процесу реєстрації. На рисунку 2.9 зображено вікно реєстрації користувача.



The image shows a user registration window titled "Проектний менеджер" (Project Manager). At the top, there are two buttons: "УВІЙТИ" (Log In) and "ЗАРЕЄСТРУВАТИСЯ" (Register), with the latter being underlined. The registration form contains three input fields: "Ел. пошта" (Email) with the value "anna.mal@gmail.com", "Пароль" (Password) with masked characters, and "Повторіть пароль" (Repeat password) with masked characters. A blue button labeled "Продовжити" (Continue) is located at the bottom of the form.

Рисунок 2.9 – Вікно реєстрації користувача

Алгоритм реєстрації користувача включає кілька етапів, починаючи зі створення облікового запису, де користувач вводить особисті дані, такі як електронна пошта та пароль. Система перевіряє унікальність цих даних у базі

даних, щоб уникнути конфліктів. Після цього користувач отримує електронний лист з посиланням для підтвердження реєстрації, що забезпечує активацію облікового запису.

На рисунку 2.10 зображено блок-схему алгоритму реєстрації користувача.

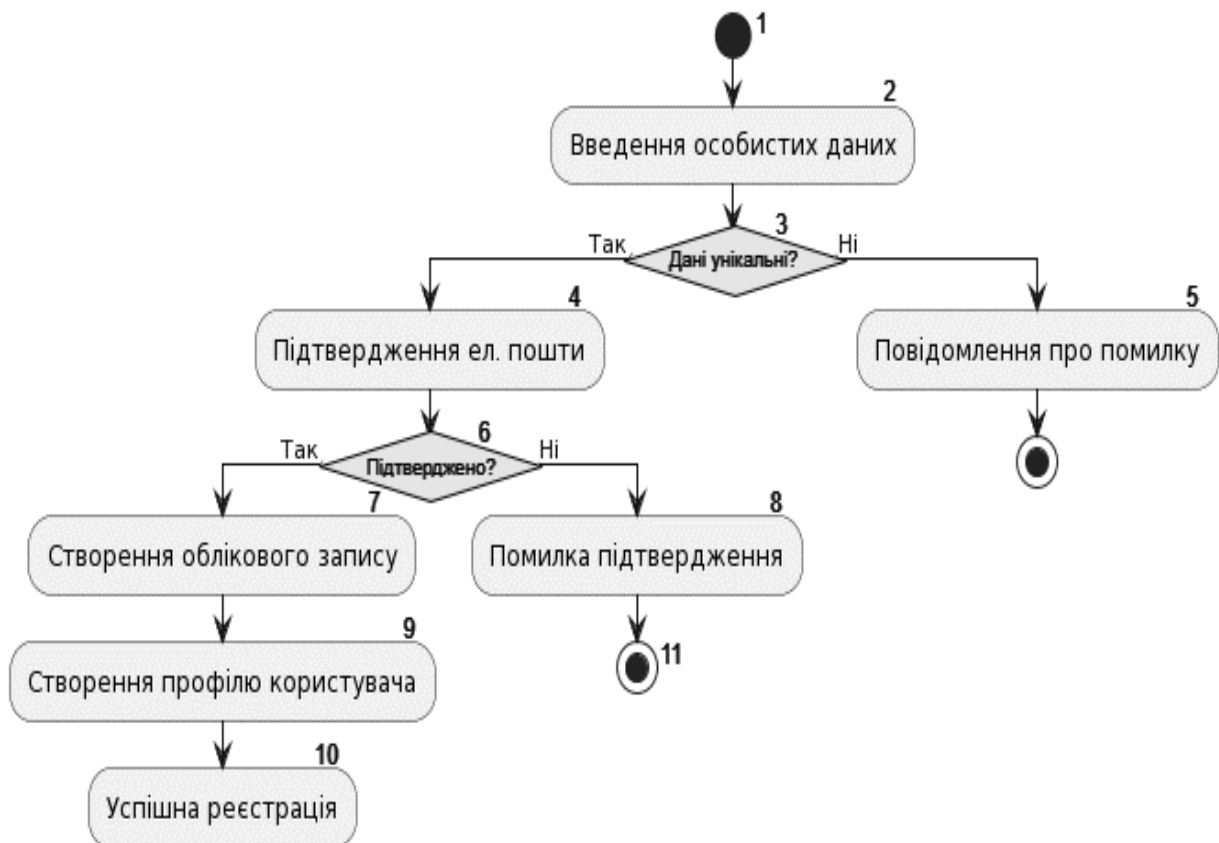


Рисунок 2.10 – Алгоритм реєстрації користувача

Авторизовані користувачі отримують доступ до ресурсів системи залежно від наданих прав з урахуванням їх заявлених ролей в команді.

Крім того, важливим етапом є створення профілю користувача, де користувач може додати додаткові особисті дані та управляти ними через панель управління (рис. 2.11).



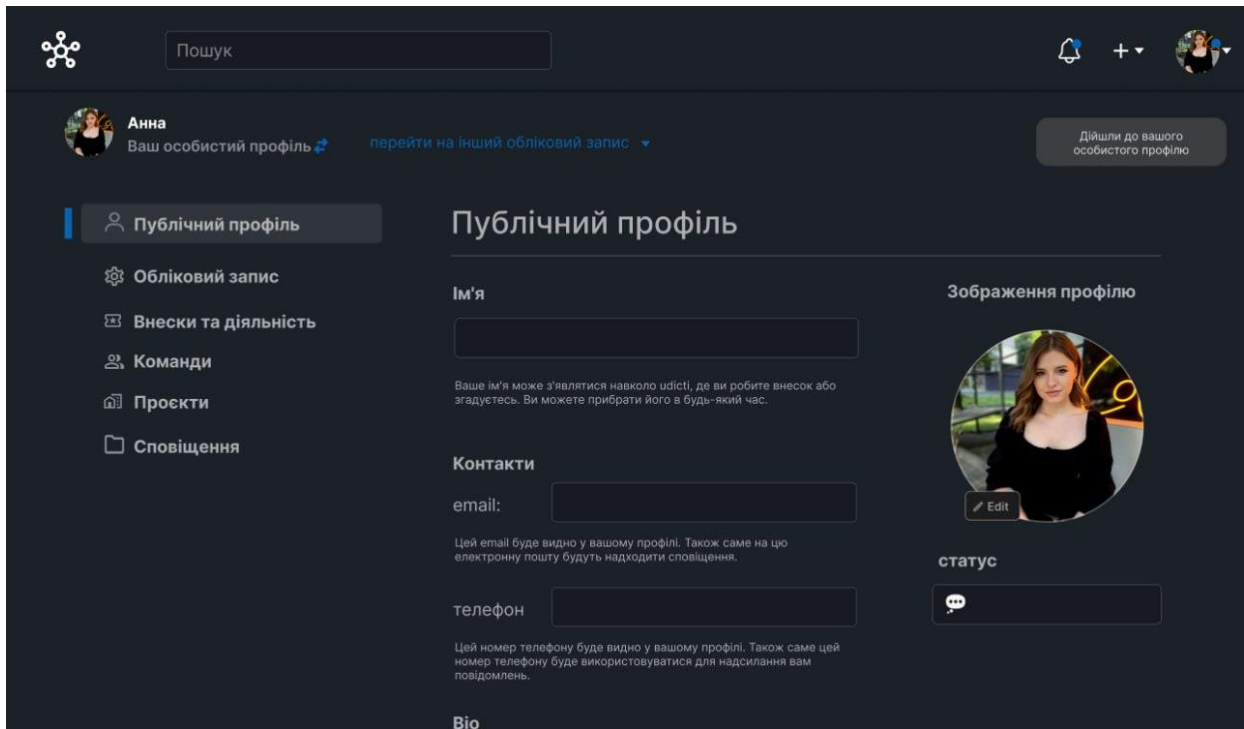


Рисунок 2.11 – Налаштування профілю

Алгоритм реєстрації повинен бути простим для користувача і водночас забезпечувати високий рівень безпеки та відповідати вимогам законодавства щодо зберігання та обробки особистих даних.

У процесі розробки алгоритму реєстрації використовуються різні технології та методи для забезпечення ефективності та безпеки процесу. Наприклад, використання шифрування даних під час передачі через мережу за допомогою протоколів HTTPS або TLS, а також використання капчі для підтвердження, що реєстрація виконується реальним користувачем, є одними з важливих аспектів безпеки. Підтвердження реєстрації через електронну пошту, використання безпечних паролів та захист від SQL ін'єкцій також є ключовими елементами безпеки [27].

Зокрема, для запобігання можливим атакам важливо використовувати підготовлені запити та параметризовані запити. Крім того, використання методів мультифакторної аутентифікації та створення зручного та інтуїтивного

інтерфейсу також є важливими кроками у забезпеченні безпеки та комфорту користувача під час реєстрації в системі управління стартап-проєктами.

Авторизація користувача у системі управління стартап-проєктами визнається важливим етапом забезпечення безпеки та конфіденційності. Цей процес включає в себе підтвердження та контроль доступу до особистої облікової запису в системі, забезпечуючи лише правомірному користувачеві можливість взаємодії з функціоналом та ресурсами платформи.

Під час авторизації користувач вводить свої унікальні облікові дані, такі як логін та пароль. Ця інформація використовується для підтвердження особистості користувача та перевірки його прав на доступ до конкретних функцій та даних системи. Логін і пароль, що обирає користувач, відіграють ключову роль у забезпеченні безпеки і визначенні обсягу його можливостей в системі.

Успішна авторизація визначає, які ресурси та функціонал стануть доступними користувачеві після входу до системи. Цей процес сприяє виключенню несанкціонованих осіб та забезпеченню захищеності конфіденційної інформації. Таким чином, авторизація відіграє критичну роль у забезпеченні безпеки та правильного функціонування системи управління стартап-проєктами. [28].

Перевірка введених облікових даних здійснюється через порівняння з даними, які зберігаються в базі даних системи. При відповідності введених даних збереженим в базі, користувачу надається доступ до системи. Для кращої безпеки та відстеження активності користувача, в системі створюється унікальний токен (сесія), який використовується для подальшого доступу та ідентифікації користувача під час його взаємодії з системою.

Сесія, що надається користувачеві, використовується для відстеження активності та забезпечення безпеки особистих даних. Система веде контроль за часом дії сесії та необхідністю повторної авторизації для підтвердження ідентифікації користувача в разі закінчення часу дії токenu або зміни стану сесії.

Блок-схема алгоритму процесу авторизації користувача в системі зображена на рисунку 2.12.



Рисунок 2.12 – Алгоритм процесу авторизації користувача в системі

Авторизація в системі стартап-проектів є важливим елементом забезпечення конфіденційності та безпеки особистих даних користувача. Головною метою цього процесу є захист інформації та забезпечення доступу тільки авторизованим особам.

Під час авторизації користувача система перевіряє його облікові дані, такі як логін та пароль, для переконання в тому, що особа, яка намагається отримати доступ, є правомірним користувачем. Цей процес гарантує, що лише користувачі з валідними обліковими даними можуть використовувати функціонал системи та отримувати доступ до особистих даних[29].

Забезпечення конфіденційності особистих даних є важливим етичним та правовим аспектом управління інформацією. Авторизація гарантує, що чутливі дані користувачів залишаються захищеними від несанкціонованого доступу та використання. Такий підхід є фундаментальним для збереження довіри користувачів до системи та підтримання високих стандартів безпеки у сфері управління стартап-проєктами.

## **2.5 Висновки**

Отже, в другому розділі було розроблено метод, моделі та алгоритми роботи системи управління стартап-проєктами, серед яких особливе місце відводилося алгоритмам пошуку даних і авторизації в програмі. Алгоритм пошуку даних створено з урахуванням оптимальності та швидкості знаходження інформації, забезпечуючи ефективну роботу програми. У свою чергу, алгоритм авторизації в програмі був спроектований з урахуванням надійності і захищеності, забезпечуючи безпечний доступ до функціоналу та даних. Ці алгоритми стали важливою складовою реалізації програмного продукту, забезпечуючи його надійність та продуктивність.

## **3 РОЗРОБКА ПРОГРАМИ УПРАВЛІННЯ СТАРТАП-ПРОЄКТАМИ**

### **3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного засобу**

Перед початком роботи зі створення системи управління стартап-проектами потрібно визначитися із засобами розробки. До яких належать: мова програмування, фреймворки, інтегроване середовище розробки та система керування базами даних.

IDE або інтегроване середовище розробки (Integrated Development Environment) – це програмний пакет, який надає зручний інтерфейс та набір інструментів для програмістів для розробки, тестування та відлагодження програмного забезпечення. Це надає одному місцю доступ до різних інструментів, які допомагають у всіх етапах розробки програм [30]. Інтегроване середовище розробки зазвичай включає в себе текстовий редактор для написання коду, компілятор або інтерпретатор для перетворення коду у виконуваний файл, а також інші інструменти, такі як відлагоджувач (debugger), система керування версіями, консоль для введення команд, та інші допоміжні функції .

У якості середовища розробки було обрана IntelliJ IDEA. IntelliJ IDEA – це потужне інтегроване середовище розробки, розроблене компанією JetBrains. Воно спрямоване на полегшення роботи програмістів з різними мовами програмування, зокрема, Java, Kotlin, Groovy та Scala.

Основна сильна сторона IntelliJ IDEA – це розумний відлагоджувач, який дозволяє відлагоджувати програми на льоту та виводити значущі дані під час виконання коду. Інструмент автоматичного завершення коду робить процес написання коду ефективнішим, надаючи пропозиції завершення під час введення коду.

Крім цього, IntelliJ IDEA пропонує розширений аналіз коду, що включає виявлення помилок та підказки щодо поліпшення коду. Інтеграція з різними

системами керування версіями, такими як Git, Subversion та Mercurial, дозволяє ефективно керувати версіями проєктів.

Не останню роль відіграє підтримка мови Kotlin, яка є продуктом самої компанії JetBrains. Це забезпечує високий рівень інтеграції та продуктивності для розробників, які вибирають Kotlin для своїх проєктів. IntelliJ IDEA також володіє вбудованими інструментами для роботи з різними фреймворками, що спрощує розробку під конкретні технології.

Загалом, це інтегроване середовище розробки визначається своєю високою продуктивністю, широким функціоналом та зручним інтерфейсом, роблячи його популярним вибором серед розробників у світі програмування.

Розробка "Системи управління стартап-проєктами" передбачає створення програмного забезпечення з використанням Java та JavaFX для розробки графічного інтерфейсу (GUI).

Java є однією з найважливіших та найпопулярніших мов програмування в сучасному програмуванні. Вона відзначається своєю виразністю, надійністю та переносимістю, що робить її дуже популярною серед розробників.

Однією з ключових переваг Java є її крос-платформеність. Це означає, що програми, написані на Java, можуть працювати на різних операційних системах без жодних змін. Це особливо важливо для веб-додатків та програмного забезпечення, яке має бути доступним для користувачів з різних платформ.

JavaFX, в свою чергу, є набором бібліотек та інструментів для розробки графічного інтерфейсу (GUI) для Java-додатків [31]. Вона надає засоби для створення стильних та інтуїтивно зрозумілих інтерфейсів, що полегшує користувачам взаємодію з програмою.

JavaFX дозволяє розробникам створювати вікна, кнопки, меню та інші графічні компоненти з використанням Java-коду. Вона підтримує різні стилі та теми, що дозволяє створювати інтерфейси, які відповідають сучасним дизайнерським стандартам.

Використання Java та JavaFX для розробки "Системи управління стартап-проектами" надає безліч переваг. Java забезпечує надійну та безпечну роботу програми, забезпечує її високу продуктивність та простоту управління ресурсами. JavaFX допомагає створити привабливий та користувацький інтерфейс, що покращує взаємодію з програмою та забезпечує високий рівень задоволення користувачів.

Java обрана як мова програмування завдяки своїй надійності та переносимості. Вона є ідеальним вибором для проєктів, де стійкість та надійність важливі. Java також має широку спільноту розробників, що спрощує розробку та підтримку проєкту.

JavaFX використовується для розробки графічного інтерфейсу. Ця технологія надає зручні інструменти для створення стильних та інтуїтивно зрозумілих інтерфейсів, які важливі для користувачів.

MySQL – це безкоштовна та відкрита система управління базами даних (СУБД), яка підходить для проєктів з обмеженими бюджетами. Вона славиться своєю швидкістю та простотою використання (рисунок 2.1). Однак вона може бути менш потужною в порівнянні з PostgreSQL та MS SQL Server у випадках, коли потрібні високі рівні безпеки та підтримки великих обсягів даних.

MySQL була обрана як система управління базами даних з кількох причин. По-перше, вона є відкритою та безкоштовною, що особливо важливо для стартапів з обмеженими бюджетами. MySQL також легко інтегрується з Java за допомогою JDBC (Java Database Connectivity), що робить її відмінним вибором для зберігання та керування даними вашої системи. Крім того, вона доступна на багатьох платформах, що відповідає основним перевагам Java щодо переносимості.

PostgreSQL – це потужна та відкрита об'єктно-реляційна система управління базами даних (СУБД), яка відзначається високою надійністю,

розширюваністю та функціональністю. Вона є однією з найпопулярніших СУБД у світі і часто використовується для великих та складних програмних проєктів.

Однією з ключових переваг PostgreSQL є її здатність обробляти великі обсяги даних та високі навантаження. Вона володіє потужними механізмами оптимізації запитів, які дозволяють виконувати запити швидко та ефективно, навіть в умовах великого обсягу даних.

PostgreSQL підтримує багато різних типів даних, включаючи географічні та текстові дані, що робить її ідеальним вибором для проєктів, як "Система управління стартап-проєктами", де може бути необхідно зберігати різноманітні дані.

Важливо відзначити, що PostgreSQL є вільним та відкритим програмним забезпеченням, що означає, що ви можете використовувати його безкоштовно, а також модифікувати та розповсюджувати відповідно до вашого вибору. Це робить PostgreSQL доступним та витратоспоживаючим рішенням для багатьох проєктів.

Завдяки своїй відкритій природі та активній спільноті розробників, PostgreSQL постійно оновлюється та покращується, що забезпечує його актуальність та конкурентоспроможність серед інших СУБД.

Узагальнюючи, PostgreSQL є потужною та надійною СУБД, яка підходить для великих та складних проєктів, а також відзначається відкритістю та доступністю для широкого кола розробників.

Microsoft SQL Server – це комерційна система управління базами даних (СУБД), розроблена корпорацією Microsoft, і вона відзначається високою функціональністю, продуктивністю та надійністю. Вона є однією з найпоширеніших СУБД у світі та широко використовується в корпоративному середовищі.

Microsoft SQL Server надає різноманітні можливості для роботи з даними, включаючи розширені можливості зберігання, оптимізації та аналізу даних. Вона



підтримує велику кількість типів даних, включаючи географічні, та має потужні інструменти для роботи з ними.

Однією з ключових переваг MS SQL Server є її інтеграція з іншими продуктами Microsoft, такими як Windows Server, Active Directory, та інші. Це робить її ідеальним вибором для організацій, які вже використовують продукти Microsoft, оскільки ця інтеграція дозволяє покращити продуктивність та ефективність роботи.

MS SQL Server також підтримує велику кількість різних інструментів та мов програмування, включаючи T-SQL, яка дозволяє виконувати запити та зберігати процедури.

Важливо відзначити, що MS SQL Server є комерційним продуктом, і використання його може вимагати значних витрат на ліцензії та підтримку. Однак ця СУБД надає розширену функціональність та надійність, що робить її привабливою для великих підприємств та організацій, де безпека та продуктивність є критичними факторами.

Загалом, Microsoft SQL Server є потужною та розширеною СУБД, яка надає розширені можливості роботи з даними та інтегрується з іншими продуктами Microsoft для забезпечення ефективної роботи корпоративних систем.

Порівняльний аналіз показує, що PostgreSQL – це потужна СУБД з високим рівнем надійності та розвинутою функціональністю, але вона може бути складніше налаштована та вимагає більше уваги до деталей. MS SQL Server від Microsoft підходить для великих корпоративних проєктів, але вона є платною, що може бути важливим чинником для стартапів.

Крім того, можуть бути використані такі програмні засоби для створення архітектури та структури програмного застосунку: Spring Framework, Hibernate, Apache Maven.

Spring Framework є однією з найпопулярніших та потужних технологій для розробки додатків на платформі Java. Ця рамка (framework) була створена з

метою спростити розробку великих та складних програмних проєктів, а також полегшити управління компонентами додатку та бізнес-логікою.

Однією з ключових переваг Spring є інверсія керування (Inversion of Control, IoC), яка полягає в тому, що вона визначає, коли і які об'єкти створюються та управляються контейнером Spring. Це дозволяє вам сконцентруватися на розробці функціональності додатку, не витрачаючи час на управління життєвим циклом об'єктів.

Spring також надає широкий спектр модулів та інструментів для розробки, такі як Spring Security для забезпечення безпеки додатку, Spring Data для роботи з різними базами даних, Spring Boot для швидкого створення самостійних додатків та багато інших.

Крім того, Spring пропонує вбудовану підтримку для тестування додатків, що допомагає забезпечити якість та надійність програмного забезпечення. Велика спільнота розробників та велика кількість документації роблять Spring однією з перших технологій, яку розглядають при розробці додатків на Java.

Spring Framework допомагає розробникам створити добре структуровані та підтримувані додатки, що легко розширюються та відповідають сучасним стандартам розробки програмного забезпечення.

Hibernate є потужним інструментом для роботи з базами даних в контексті Java-додатків, і він широко використовується у сфері розробки програмного забезпечення. Його основна перевага полягає в тому, що він дозволяє взаємодіяти з базою даних, використовуючи об'єктно-орієнтований підхід, що спрощує роботу з даними та реляційними базами даних.

Hibernate забезпечує високий рівень абстракції від деталей роботи з базою даних, що означає, що розробники можуть працювати з об'єктами та класами, не вдаючись у складну SQL-логіку.

Однією з ключових особливостей Hibernate є використання мапування об'єктів на таблиці бази даних. Це означає, що ви можете визначити модель

даних у вигляді класів Java, і Hibernate автоматично створить та управлятиме відповідними таблицями в базі даних. Це полегшує розробку та підтримку додатків, оскільки ви можете працювати з об'єктами, як звичайними об'єктами Java.

Hibernate також надає механізми для встановлення зв'язків між об'єктами, а також можливості виконувати запити до бази даних за допомогою Hibernate Query Language (HQL) або навіть чистого SQL, якщо це необхідно.

Важливо відзначити, що Hibernate підтримує різні бази даних, що робить його універсальним інструментом для розробки, незалежно від того, яка СУБД використовується.

З використанням Hibernate, розробники можуть створити потужні та ефективні додатки, які взаємодіють з базами даних в об'єктно-орієнтованому стилі, що полегшує розробку та підтримку програмного забезпечення.

Apache Maven – це потужний інструмент для автоматизації управління процесом збирання та побудови проєктів програмного забезпечення. Він широко використовується розробниками для ефективного управління залежностями, збиранням, тестуванням та розгортанням додатків.

Однією з ключових переваг Maven є його система керування залежностями. Завдяки конфігураційному файлу `pom.xml`, Maven дозволяє визначити всі необхідні залежності для проєкту, та автоматично завантажувати їх з центральних репозиторіїв. Це спрощує роботу з бібліотеками та забезпечує їхню актуальність.

Maven надає стандартизований та консистентний спосіб збирання та побудови проєктів. Завдяки цьому, розробники можуть легко виконувати різні завдання, такі як компіляція коду, створення JAR-файлів, виконання тестів, генерація звітів та багато інших, використовуючи заздалегідь налаштовані цілі.

Maven також підтримує роботу з різними фазами життєвого циклу проєкту, такими як `clean`, `compile`, `test`, `package`, `install`, що дозволяє розробникам визначити по+слідовність дій та оптимізувати процес збирання та розгортання.

Однією з інших переваг Maven є можливість роботи з різними типами проєктів, включаючи Java, .NET, Ruby, та інші.

Це робить його універсальним інструментом для будь-якого типу розробки програмного забезпечення.

Загалом, Apache Maven спрощує та автоматизує процеси збирання та управління проєктами, полегшуючи життя розробників та забезпечуючи стабільність та надійність у роботі з програмним забезпеченням.

Обрана мова Java визначається не лише її популярністю та переносимістю коду, але і можливістю використання широкого спектру бібліотек та фреймворків. Вона дозволяє побудувати масштабовані системи, що є ключовим фактором для управління стартап-проєктами.

Для розробки графічного інтерфейсу було використано JavaFX через його спрощену розробку графічного інтерфейсу та можливість інтеграції з Java. Його модульність та зручність використання дозволяють створювати інтуїтивно зрозумілі та естетично привабливі інтерфейси для користувачів.

Системою управління базами даних обрано MySQL через її широку популярність та надійність. Її здатність ефективно взаємодіяти з даними та підтримувати транзакції дозволяє забезпечити високий рівень цілісності даних у "Системі управління стартап-проєктами".

Ця реляційна база даних використовує мову SQL (Structured Query Language) для взаємодії з даними. MySQL підтримує велику кількість операцій збереження, пошуку та маніпулювання даними, забезпечуючи ефективність та надійність у роботі з базами даних.

## 3.2 Розробка програмного модуля для реєстрації та авторизації користувачів

Програмний модуль для реєстрації та авторизації функціонує на монолітній архітектурі, спільно з базою даних. Для реєстрації та авторизації використовуються Rest api запити «/register» та «/login». Для успішної авторизації користувач повинен надіслати JSON-запит із даними користувача (логіном або поштою) та паролем (рис. 3.1).

Коли бекенд сервер отримує запит на авторизацію, він починає наступну процедуру. Спочатку він виділяє потрібні параметри із отриманих даних, за ключами використовуючи JSONObject із бібліотеки org.json і метод .getString(). Далі проводиться попередня перевірка цих даних на предмет спаму або очевидно некоректних даних. Якщо дані викликають питань, сервер відправляє ці дані в модуль керування базою даних користувачів UserDatabase.

```
@PostMapping("/login")
public String loginUser(@RequestBody String loginRequest) {
    JSONObject answer = new JSONObject();
    try {
        String user_login = new JSONObject(loginRequest).getString( key: "login");
        String user_password = new JSONObject(loginRequest).getString( key: "password");
        User user = userDatabase.getUserLogin(user_login,user_password);
        String sessionId = SessionManager.createSession(user);
        answer.put("code",sessionId);
        return answer.toString();
    } catch (Exception e) {
        answer.put("error",e.getMessage());
        return answer.toString();
    }
}
```

Рисунок 3.1 – Демонстрація методу авторизації

Модуль авторизації при отриманні запиту створює з'єднання з базою даних за допомогою JDBC . Якщо з'єднання було успішне то модуль авторизації робить запит у базу даних, і якщо дані вірні, повністю об'єкт користувача відправляється

назад в модуль спілкування. Якщо ні, повертається помилка, що користувача не знайдено. Після чого програма закриває з'єднання з базою даних і відправляє виконану роботу в логи. Якщо модуль авторизації успішно отримує об'єкт користувача, він створює нову сесію та повертає тимчасовий код сесії користувачу в форматі JSON. У випадку помилки від модуля керування даними, користувач отримує відповідний повідомлення із описом проблеми, після чого проводить логування даних (рис.3.2).

```

public User getUserLogin(String email, String password) {
    ResultSet resultSet = selectUser( columnName: "email", email);
    try {
        if (resultSet.next()) {
            int id = resultSet.getInt( columnName: "id");
            String name = resultSet.getString( columnName: "name");
            String passwordCheck = resultSet.getString( columnName: "password");
            if (password.equals(passwordCheck)) {
                return new User(name, email, id);
            } else {
                throw new IllegalArgumentException("Incorrect password");
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    throw new IllegalArgumentException("User with email " + email + " does not exist");
}

```

Рисунок 3.2 – Демонстрація функції авторизації користувача базою даних

Якщо сервер отримує запит на реєстрацію, виконується інша послідовність дій (рис. 3.3). Дані, отримані від користувача, розбиваються на логін, пошту, пароль і підтвердження паролю. Після розбиття даних відбувається їх валідація та перевірка на наявність спаму. Для валідних даних потрібно, щоб логін був довший за 3 символи і складався тільки з букв, пошта відповідала шаблону, пароль був довший за 8 символів і співпадав з підтвердженням паролю.

```

@PostMapping("/register")
public String createUser(@RequestBody UserRequest userRequest) {
    JSONObject answer = new JSONObject();
    try {
        userDatabase.addUser(userRequest);
        answer.put("message", "user successfully registered");
        return answer.toString();
    } catch (Exception e) {
        answer.put("error", e.getMessage());
        return answer.toString();
    }
}

```

Рисунок 3.3 – Демонстрація алгоритму реєстрації користувача

На рисунку 3.4 зображено загальний алгоритм авторизації користувача в системі управління стартап-проектами.

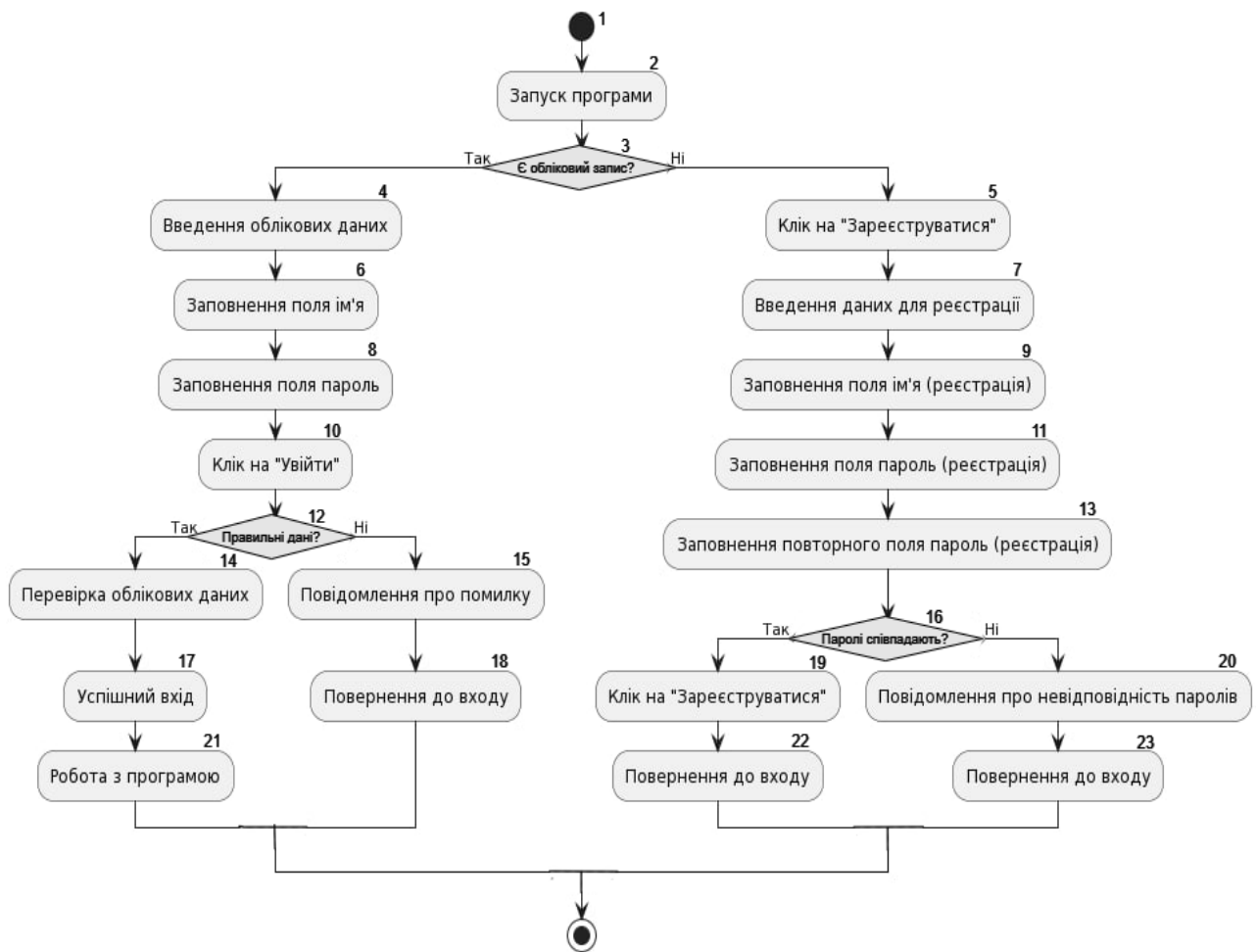


Рисунок 3.4 – Загальний алгоритм авторизації в програмі

Якщо валідація успішна, дані відправляються в модуль керування даними користувачів. Модуль реєстрації, отримавши дані, перевіряє, чи пошта, яка була отримана від модуля спілкування, вільна для реєстрації користувача.

Якщо пошта вільна, в базу даних відправляється запит на додавання нового користувача, виконується необхідна операція і повертається об'єкт користувача (рис. 3.5).

Якщо пошта зайнята чи виникла інша причина, що завадила додаванню користувача, модуль повертає відповідну помилку.

```
public void addUser(UserRequest user) throws Exception {
    if (!isValidEmail(user.getEmail())) {
        throw new Exception("Incorrect email address");
    }
    if (!isEmailFree(user.getEmail())) {
        throw new Exception("Email is already in use");
    }
    if (!isValidName(user.getName())) {
        throw new Exception("Name is shorter than 3 letters or has symbols");
    }
    if (!isValidPassword(user.getPassword(), user.getConfirmPassword())) {
        return;
    }

    String[] columns = {"name", "email", "password"};
    Object[] values = {user.getName(), user.getEmail(), user.getPassword()};
    database.insertData( tableName: "users", columns, values);
    System.out.println("SUCCESS");
}
```

Рисунок 3.5 – Код методу додавання користувача в базу даних

Якщо модуль спілкування отримує об'єкт користувача, він авторизує його та створює нову тимчасову сесію, після чого повертає код сесії. У випадку отримання помилки від модуля спілкування, користувач отримає відповідне повідомлення про помилку.



### 3.3 Розробка програмного модуля для управління проектами

Для ефективного управління проектами розроблено модуль управління, який відзначається своєю здатністю забезпечити ефективну та зручну взаємодію користувача з системою.

Процес авторизації та створення сесійного коду надає безпечний доступ до функціональності системи, гарантуючи, що тільки правомірні користувачі матимуть доступ до важливих функцій управління проектами.

Використання JDBC для взаємодії з базою даних надає надійний та ефективний механізм обміну інформацією.

Це дозволяє системі оперативно отримувати, зберігати та оновлювати дані, що є важливим для швидкої реакції на зміни в управлінні проектами.

Десктопна версія інтерфейсу, описана як приємна та спрощена, робить взаємодію користувача з системою інтуїтивно зрозумілою та приємною.

Це не лише полегшує процеси управління проектами, а й дозволяє користувачеві зосередитися на конкретних завданнях, уникнувши зайвих відволікань.

Важливою особливістю системи є той факт, що користувач може ефективно використовувати систему, навіть не розуміючи всіх деталей обміну даними між його комп'ютером та сервером.

Це створює додатковий комфорт для кінцевого користувача та сприяє більш ефективному впровадженню системи у практиці управління проектами.

На рисунку 3.6 зображено діаграму взаємодії модуля управління проектами.

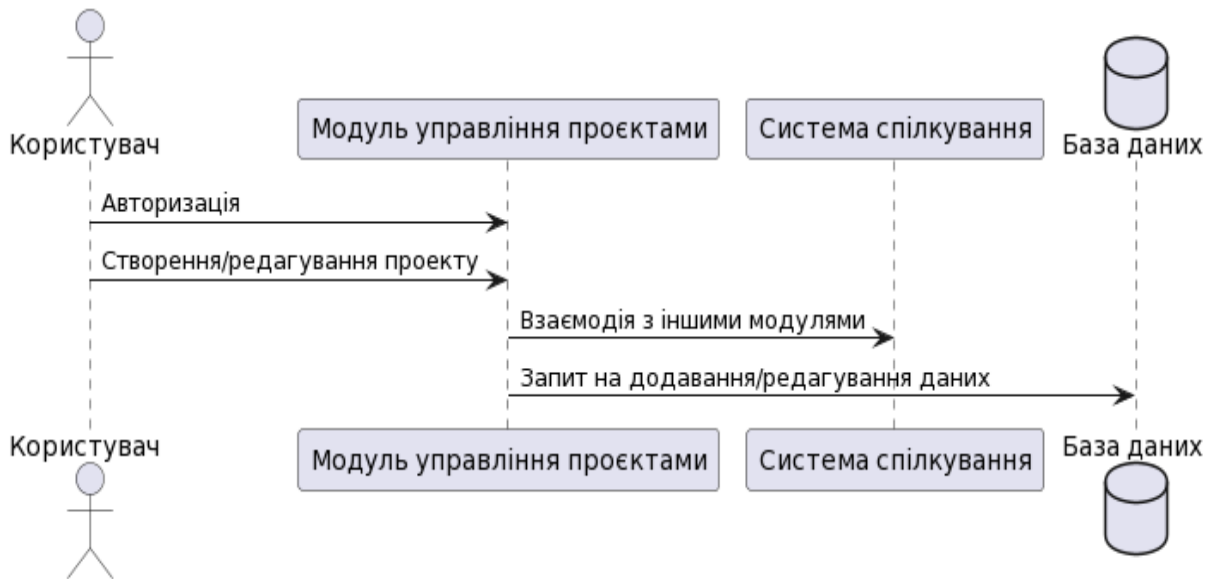


Рисунок 3.6 – Діаграма взаємодії модуля управління проєктами

Коли модуль спілкування отримує запит, він спочатку перевіряє його на можливість спаму. Якщо випадок спаму не підтверджується, відбувається парсинг отриманих даних від користувача і витягування з них коду. Потім відбувається запит у модуль сесій, де проводиться перевірка того, чи користувач зараз авторизований. Якщо сесійний модуль підтверджує, що користувач авторизований, реалізуються наступні дії залежно від виду запиту.

Якщо відбувається запит про створення проєкту, система перевіряє кількість проєктів, яку користувач може ще створити. Якщо користувач вже створив максимальну кількість проєктів, він отримає відповідне повідомлення. У разі, якщо користувач ще не досягнув ліміту проєктів, система аналізує дані від користувача щодо імені, опису та бюджету проєкту. Якщо такі дані відповідають вимогам, відправляється запит в базу даних для додавання нового проєкту з відповідними даними. У випадку успішної операції користувач отримає відповідне повідомлення. Після цих операцій у клієнті користувач може переглядати проєкт із зазначеними даними та своєю участю у ролі адміністратора.

На рисунку 3.7 зображено діаграму обробки запитів модулем управління проектами:

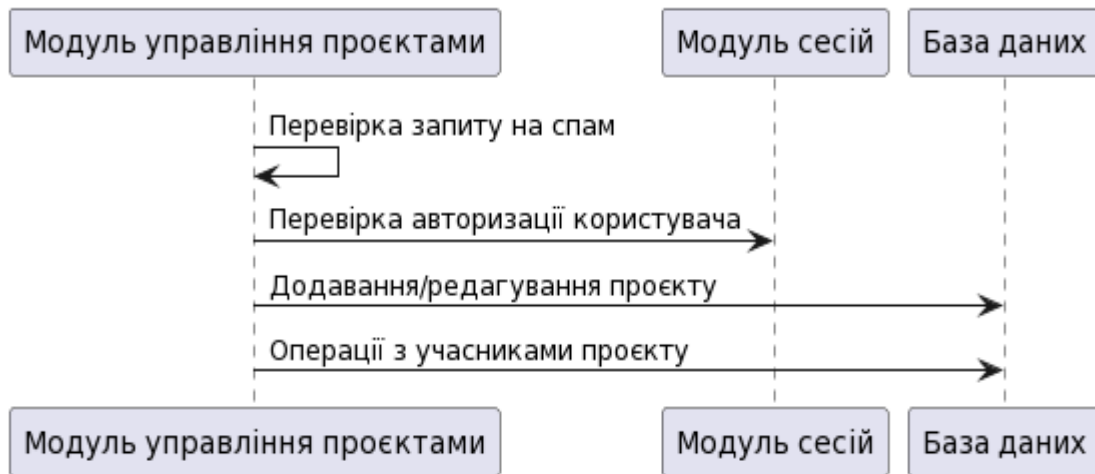


Рисунок 3.7 – Діаграма обробки запитів модулем управління проектами

У випадку, коли запит стосується зміни даних, кількості учасників або видалення, відбуваються аналогічні операції. Початково система ідентифікує користувача, який робить ці запити, і перевіряє, до яких проектів він відноситься. Після отримання списку проектів система перевіряє, чи існує проект, над яким планується провести операцію, та яку роль має користувач.

Якщо база даних підтверджує існування проекту та повертає роль користувача, і ця роль має необхідні права для виконання відповідної операції, система переходить до виконання операції. У випадку відсутності проекту в базі даних або невідповідних прав користувача, відбувається повернення помилки.

Виконання операцій над проектом відбувається аналогічно. Якщо йдеться про зміну параметрів або додавання користувача, відправляється запит в базу даних, який виконує операцію заміни або додавання даних.

У випадку видалення проекту, проект архівується на певний строк, а з бази даних видаляється. Дані проекту архівуються для можливості відновлення

проєкту. Якщо запит на відновлення проєкту не надійде протягом визначеного терміну, архівований проєкт буде остаточно видалений.

На рисунку 3.8 зображена діаграма зміни ролей користувачів.

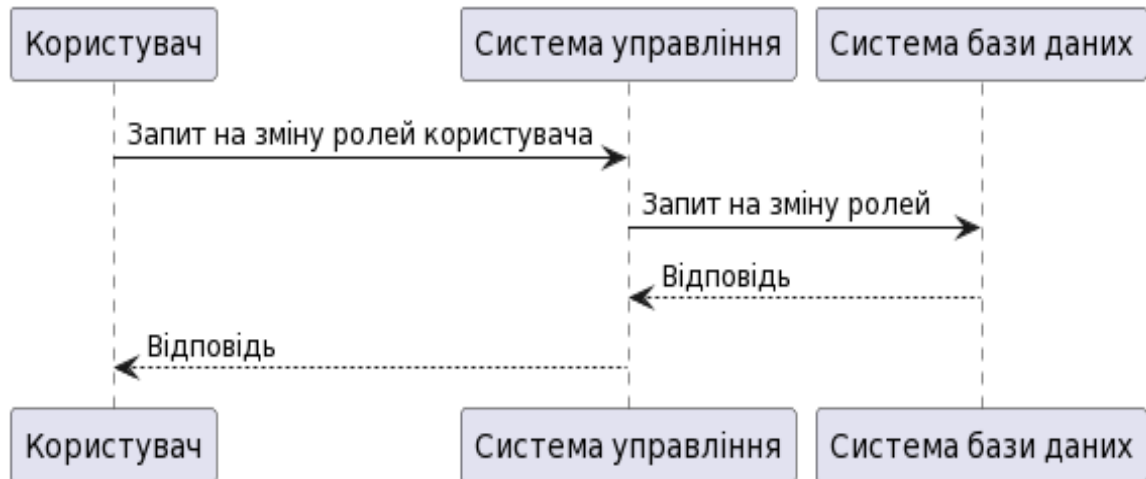


Рисунок 3.8 – Діаграма зміни ролей користувачів

Кожен проєкт має стандартизовану систему ролей, яку можна адаптувати відповідно до вимог конкретного проєкту. Ці ролі призначені для регулювання можливостей користувачів.

Після створення проєкту користувач, який його створив, отримує роль власника проєкту. Ця роль має всі права і призначена тільки для одного користувача в межах проєкту.

Коли власник проєкту додає нового користувача, у нього створюється порожня роль, яка дозволяє лише перегляд проєкту. Для того, щоб виконувати певні операції, власник проєкту повинен призначити одну зі стандартних ролей або створити свою, визначивши відповідні можливості.

Якщо користувач має певні права для взаємодії з іншими користувачами, такі як блокування або зміна їхніх ролей, він не може здійснити ці дії з користувачем, якщо ролі останнього перебувають на вищому рівні в дереві ролей.

### 3.4 Розробка програмного модуля аналізу проєктів та генерації звітів

Модуль звітності в системі управління стартап-проєктами — це компонент системи, призначений для систематичного та структурованого аналізу та представлення інформації щодо різних аспектів виконання проєктів, головною метою якого є забезпечення основи для прийняття управлінських рішень, надання зрозумілих та інформативних звітів для керівництва та учасників проєктів, відіграє важливу роль у забезпеченні ефективності та прозорості в управлінні.

Основні функції та характеристики модуля звітності включають:

1. Збір даних — модуль автоматично або вручну збирає дані з різних джерел, таких як база даних, інші модулі системи, зовнішні API, електронні таблиці тощо.
2. Аналіз та обробка інформації — зібрана інформація піддається аналізу та обробці для витягування ключових показників ефективності, прогресу, витрат та інших параметрів.
3. Представлення даних — модуль створює різноманітні звіти, графіки, діаграми та інші візуалізації для наглядної представлення результатів аналізу. Це може включати звіти про витрати, графіки прогресу, аналітику ризиків та інше.
4. Гнучкі налаштування — модуль зазвичай має можливість гнучкої настройки для адаптації до конкретних потреб та вимог кожного проєкту чи організації.
5. Регулювання доступу — забезпечення контролю доступу до різних рівнів інформації в залежності від ролі користувача. Наприклад, керівництво може мати доступ до більш широкого спектру звітів порівняно зі звичайними учасниками проєктів.

6. Моніторинг в реальному часі — модуль може забезпечувати можливість моніторингу подій та змін в реальному часі, щоб оперативно реагувати на зміни в ситуації.

Цей модуль здатний зібрати різноманітні дані, пов'язані з виконанням завдань, прогресом робіт, витратами ресурсів та іншою важливою інформацією. Після цього він забезпечує аналіз цих даних, надаючи керівництву та учасникам проєктів можливість зрозуміти ситуацію та приймати обґрунтовані рішення.

Одним з ключових аспектів модуля звітності є надання деталізованих звітів та статистики, що глибоко аналізують різні аспекти виконання проєктів. Цей аспект забезпечує об'єктивний погляд на різноманітні аспекти та показники, які є ключовими для оцінки успішності та ефективності проєктів, а також для прийняття управлінських рішень.

Модуль звітності дозволяє створювати деталізовані звіти з різних сфер проєкту. Це включає в себе аналіз витрат, який не лише розкриває загальні витрати, але і розподіляє їх за різними категоріями, що дає можливість виявити основні джерела витрат та здійснювати більш ефективне фінансове планування.

Терміни виконання завдань також стають об'єктом уваги модуля, надаючи змогу аналізувати пунктуальність та дотримання графіка робіт. Це не лише дозволяє визначити потенційні затримки, але й надає можливість приймати своєчасні корективні заходи.

Прогрес кожного етапу проєкту включає в себе ретельний аналіз досягнень та виявлення можливих проблем чи труднощів. Це допомагає визначити, чи відповідає виконання завдань передбаченому графіку та чи потрібні зміни в стратегії виконання.

Ефективність використання ресурсів, що оцінюється через модуль звітності, включає в себе аналіз використання робочого часу, матеріальних ресурсів та інших активів. Ця інформація є критичною для оптимізації витрат та підвищення ефективності всього процесу виконання проєкту.

Такий деталізований підхід до звітності забезпечує вищу точність та повноту інформації, що є важливим елементом для керівництва та команди проекту в процесі вирішення різних завдань та прийняття стратегічних рішень.

Загалом, модуль звітності сприяє покращенню процесів управління, допомагаючи здійснювати моніторинг та контроль за проектами, а також забезпечуючи аналіз для подальших вдосконалень та прийняття стратегічних рішень.(рис. 3.9).

Послідовність дій виконання модуля звітності:

1. Вибір параметрів звіту:

- Визначення періоду звіту (день, тиждень, місяць).
- Обрання конкретного стартап-проекту чи групи проектів для аналізу.

2. Збір інформації:

- Завантаження даних з бази даних MySQL, пов'язаних із стартапами, їх завданнями, витратами та прогресом виконання.
- Фільтрація та сортування даних відповідно до обраних параметрів.

3. Аналіз та формування звіту:

- Обчислення ключових показників ефективності, таких як витрати, терміни виконання та досягнення мети.
- Створення графіків, діаграм та інших візуальних елементів для наочного представлення інформації.

4. Генерація звіту:

- Створення комплексного звіту з результатами аналізу.
- Забезпечення можливості експорту звіту для подальшого використання та обміну.

5. Збереження звіту в базу даних:

- Запис отриманих результатів звіту в спеціальну таблицю бази даних для подальшого архівування та доступу.

C ReportModule	
□	connection: Connection
●	connectToDatabase(): void
●	selectReportParameters(project: String, period: String): void
●	gatherInformation(): void
●	analyzeAndGenerateReport(): void
●	saveReportToDatabase(): void
●	disconnectFromDatabase(): void
●	main(args: String[]): void

Рисунок 3.9 – Вміст модуля формування звітності

Зв'язок модуля формування звітності з базою даних MySQL становить ключовий елемент у справлянні з великим обсягом інформації та забезпеченні ефективної обробки та подання даних. Цей зв'язок забезпечує надійний та швидкий доступ до різноманітних даних, що дозволяє ефективно керувати та аналізувати інформацію, необхідну для прийняття важливих управлінських рішень.

Модуль звітності взаємодіє з базою даних за допомогою SQL-запитів, які є ефективним інструментом для витягування конкретних даних з таблиць бази даних. Використання SQL дозволяє точно визначити критерії вибірки, здійснювати фільтрацію та агрегацію даних, що робить процес отримання необхідної інформації більш гнучким та налаштованим під конкретні потреби.

Важливо відзначити, що дані, такі як витрати, прогрес виконання завдань та інші ключові параметри, систематично зберігаються в таблицях бази даних MySQL. Це створює структуровану основу для подальшої обробки та аналізу, забезпечуючи послідовність та цілісність інформації.

Такий підхід допомагає в уникненні втрати даних, сприяє швидкому доступу до необхідної інформації та дозволяє з легкістю взаємодіяти з базою



даних для отримання вичерпного аналізу стану справ та динаміки різних параметрів управління проєктами.

Модуль формування звітності визначається своєю здатністю використовувати різноманітні SQL-запити для ефективної взаємодії з базою даних MySQL. Ці SQL-запити виступають ключовим інструментом для отримання та обробки інформації, що стосується стану проєктів та їхнього виконання.

Операції витягування, включені в SQL-запити модуля, дозволяють отримувати конкретні дані з бази даних. Це може бути інформація щодо витрат, прогресу виконання завдань, стану ресурсів та інші важливі параметри. Використання операцій витягування робить можливим точне визначення необхідних даних для подальшої обробки та аналізу.

Фільтрація в SQL-запитах дозволяє вибирати певні записи з бази даних відповідно до заданих умов. Цей підхід є корисним для визначення конкретних аспектів проєктів чи виділення певних подій, які варто включити до звітності.

Агрегація — процес обчислення сумарних чи статистичних значень з певних даних, дозволяє отримати зведену інформацію про велику кількість даних, надаючи загальний погляд на їхню характеристику, включає в себе обчислення сумарних значень чи статистичних показників на основі вибраних даних. Цей етап є важливим для формування комплексних звітів, які відображають загальний стан проєктів та їхню ефективність.

Такий підхід до використання SQL-запитів у модулі формування звітності дозволяє не лише отримувати потрібну інформацію, але й ефективно обробляти її з метою створення інформативних та легко зрозумілих звітів для користувачів системи управління проєктами.

Поєднання модуля формування звітності та бази даних MySQL визначає новий стандарт у точності та достовірності інформації в системі управління проєктами. Цей симбіоз відзначається централізованим та структурованим

зберіганням даних, що створює надійну основу для подальшого аналізу та прийняття стратегічних рішень.

Цей тісний зв'язок має низку важливих переваг. Спершу, він забезпечує високу точність даних, оскільки вони зберігаються в централізованому репозиторії, що уникає можливості розбіжностей або дублювання інформації. Дані, таким чином, є єдиною джерельною правдивою інформацією для всіх користувачів системи.

Другий аспект полягає в швидкому доступі до даних. Це дозволяє користувачам ефективно використовувати інформацію без зайвих затримок чи складнощів. Безперервний доступ до централізованих даних сприяє оперативному реагуванню на зміни та події в проєкті.

Крім того, цей зв'язок створює умови для ефективного аналізу даних. SQL-запити, використовувані для взаємодії з базою даних MySQL, дозволяють витягувати потрібні дані та виконувати складні аналітичні операції. Це важливо для отримання інсайтів та формування стратегій на основі зібраних даних.

Отже, використання SQL-запитів у модулі формування звітності з базою даних MySQL є необхідним етапом для забезпечення високої якості та функціональності в сфері збору та аналізу даних у системі управління проєктами.

SQL-запити в модулі формування звітності взаємодіють із базою даних, використовуючи різноманітні оператори та конструкції, що дозволяють отримувати точні та необхідні дані. Вони враховують специфіку проєктної діяльності, дозволяючи користувачам отримувати інформацію про витрати, прогрес виконання завдань, ефективність використання ресурсів та інші ключові показники.

Поєднання модуля формування звітності з використанням SQL-запитів забезпечує високу точність та достовірність інформації. Запити можуть включати операції витягування для отримання конкретних даних, фільтрацію

для вибору певних записів та агрегацію для обчислення сумарних значень чи статистичних показників.

Цей підхід дозволяє модулю формування звітності не лише надавати користувачам необхідну інформацію, але й виконувати складні аналітичні завдання. Це важливо для прийняття управлінських рішень, розуміння ефективності проєктів та виявлення можливостей для оптимізації діяльності в майбутньому.

Таким чином, використання SQL-запитів стає невід'ємною частиною системи управління проєктами, сприяючи вдосконаленню процесів та досягненню стратегічних цілей компанії.

### **3.5 Висновки**

У третьому розділі роботи було ретельно розглянуто та обґрунтовано вибір ключових технологій для реалізації системи управління стартап-проєктами. Визначено, що мова програмування Java виявилася найбільш вдалою з точки зору кросплатформеності та надійності, що дозволяє забезпечити швидку та ефективну розробку.

Інтегроване середовище розробки IntelliJ IDEA було обране завдяки своїй високій продуктивності та зручному інтерфейсу, що сприяє комфортному процесу програмування.

Окрема увага приділялася вибору системи управління базами даних, і в результаті обрано MySQL. Це рішення було обрано на підставі його високої надійності та широких можливостей у розробці програмного забезпечення. MySQL володіє широкими функціональними характеристиками, які відповідають вимогам розроблюваної системи управління стартапами.

Обрані інструменти не лише відповідають вимогам проєкту, але й максимізують його потенціал. Їх комбінація забезпечує стабільність та продуктивність у всьому процесі розробки.

У результаті проведеної роботи успішно реалізовано базові модулі системи управління стартап-проєктами, що вказує на високий рівень ефективності обраних технологій.

Такий підхід дозволяє спростити та прискорити розробку продукту, що є важливим кроком у досягненні бізнес-цілей.

## 4 ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

### 4.1 Аналіз методів тестування програмного забезпечення

Тестування – це процес перевірки виконання програмного забезпечення порівняно з очікуваними результатами в умовах штучно створених умов. Його мета – довести, що готовий продукт відповідає вимогам і цілям, а також виявлення дефектів, якщо вони є.

Основним завданням тестування є пошук неполадок, надання інформації для прийняття рішень, підвищення якості програмного продукту та запобігання майбутнім помилкам. Види тестування включають функціональне, нефункціональне, тестування архітектури, тестування змін.

Класифікація тестування може відбуватися за різними критеріями, такими як знання будови системи, об'єкт та суб'єкт тестування, час проведення, критерій позитивності сценаріїв.

Тестування білої скриньки оцінює внутрішню будову програми та її коду, вимагаючи від тестувальника знання структури коду. Воно допомагає виявляти логічні помилки, друкарські та проєктні помилки.

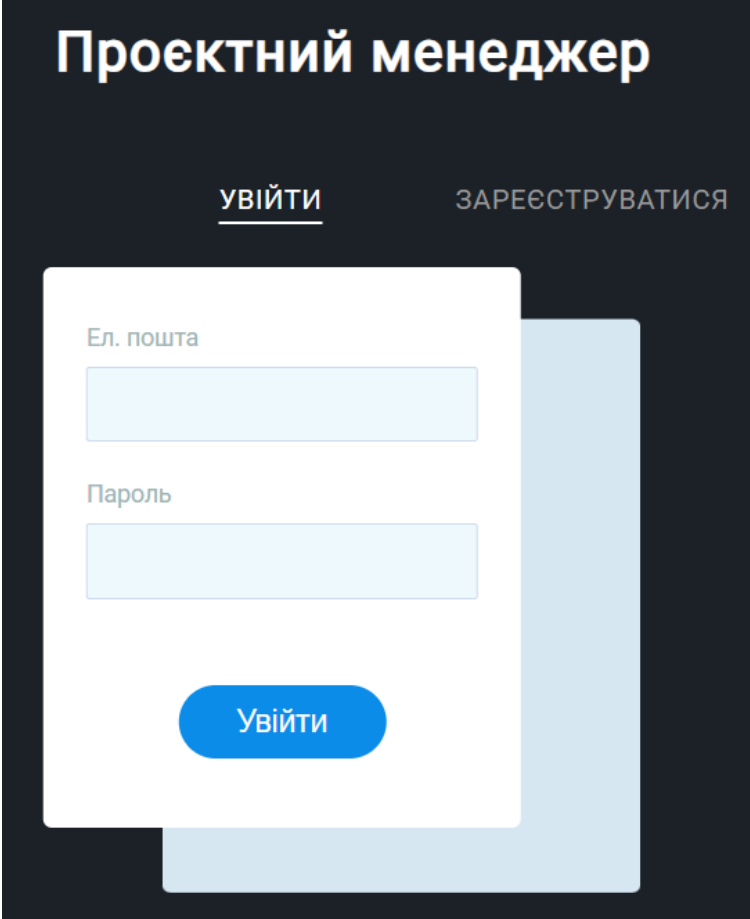
Тестування чорної скриньки порівнює вхідні та вихідні значення без знань про внутрішній механізм системи. Цей метод дозволяє виявляти дефекти на ранніх етапах тестування, але може пропустити певні сценарії через відсутність знань про внутрішню будову системи.

Тестування сірої скриньки поєднує особливості білої та чорної скриньки, не вимагаючи повного знання коду. Його перевага полягає в тому, що ймовірність пропуску сценаріїв тестування зменшується.

Для тестування системи управління стартап-проєктом обрано метод сірої скриньки. Це забезпечить перевірку з боку користувача і підтвердить успішність операцій та коректність дій з даними в базі даних.

## 4.2 Тестування системи управління стартап-проектами

Під час запуску програми користувачу необхідно буде здійснити авторизацію в системі для подальшої роботи з нею. Після запуску програми перед користувачем з'являється вікно авторизації (рис. 4.1).



The image shows a login window titled "Проектний менеджер" (Project Manager) on a dark background. At the top, there are two links: "УВІЙТИ" (Login) and "ЗАРЕЄСТРУВАТИСЯ" (Register). Below these is a white form with two input fields: "Ел. пошта" (Email) and "Пароль" (Password). A blue button labeled "Увійти" (Login) is positioned below the password field.

Рисунок 4.1 – Вікно авторизації

Система надасть сповіщення про відсутність акаунта у разі, коли введений логін не збігається з даними у базі. Якщо обліковий запис користувача відсутній, він може створити його, перейшовши до вікна реєстрації у вкладці "Sign up" (рис.4.2). Під час реєстрації важливо заповнити усі наявні поля. У разі введення некоректних даних, користувач отримає сповіщення про помилку. Після успішного підтвердження реєстрації система відобразить повідомлення про

успішне збереження даних. Далі користувачеві необхідно буде авторизуватися у новому вікні.

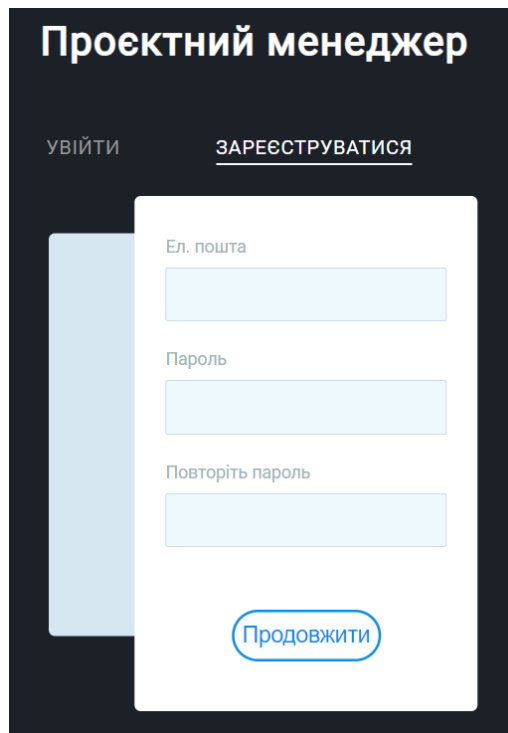
The image shows a registration form titled "Проектний менеджер" (Project Manager) on a dark background. At the top, there are two tabs: "УВІЙТИ" (Log In) and "ЗАРЕЄСТРУВАТИСЯ" (Register), with the latter being the active tab. The form itself is white and contains three input fields: "Ел. пошта" (Email), "Пароль" (Password), and "Повторіть пароль" (Repeat password). Below these fields is a blue button labeled "Продовжити" (Continue).

Рисунок 4.2 – Вікно реєстрації нового користувача

Якщо користувач вирішив зареєструватися в системі, його персоналізація може бути ще більш повною і зручною завдяки можливості додаткового заповнення профілю даними.

Цей етап реєстрації дозволяє збагатити інформацію, яку система використовує для надання більш індивідуалізованих послуг та покращення користувацького досвіду.

Безумовно, збільшення рівня персоналізації викликає не лише позитивні моменти для користувача, але і для системи в цілому.

Запропонована можливість редагування профілю в налаштуваннях облікового запису (рис. 4.3) у подальшому полегшить взаємодію з системою,

дозволяючи користувачам активно контролювати та оновлювати свої особисті дані відповідно до наявних потреб чи умов.

Цей функціонал не лише підвищує рівень індивідуалізації, але також слугує інструментом гнучкості та адаптабельності, щоб система могла краще враховувати та задовольняти потреби користувачів у динамічному середовищі.

Такий підхід до управління персональними даними покращує загальний досвід використання системи, роблячи його більш зручним та відповідальним.

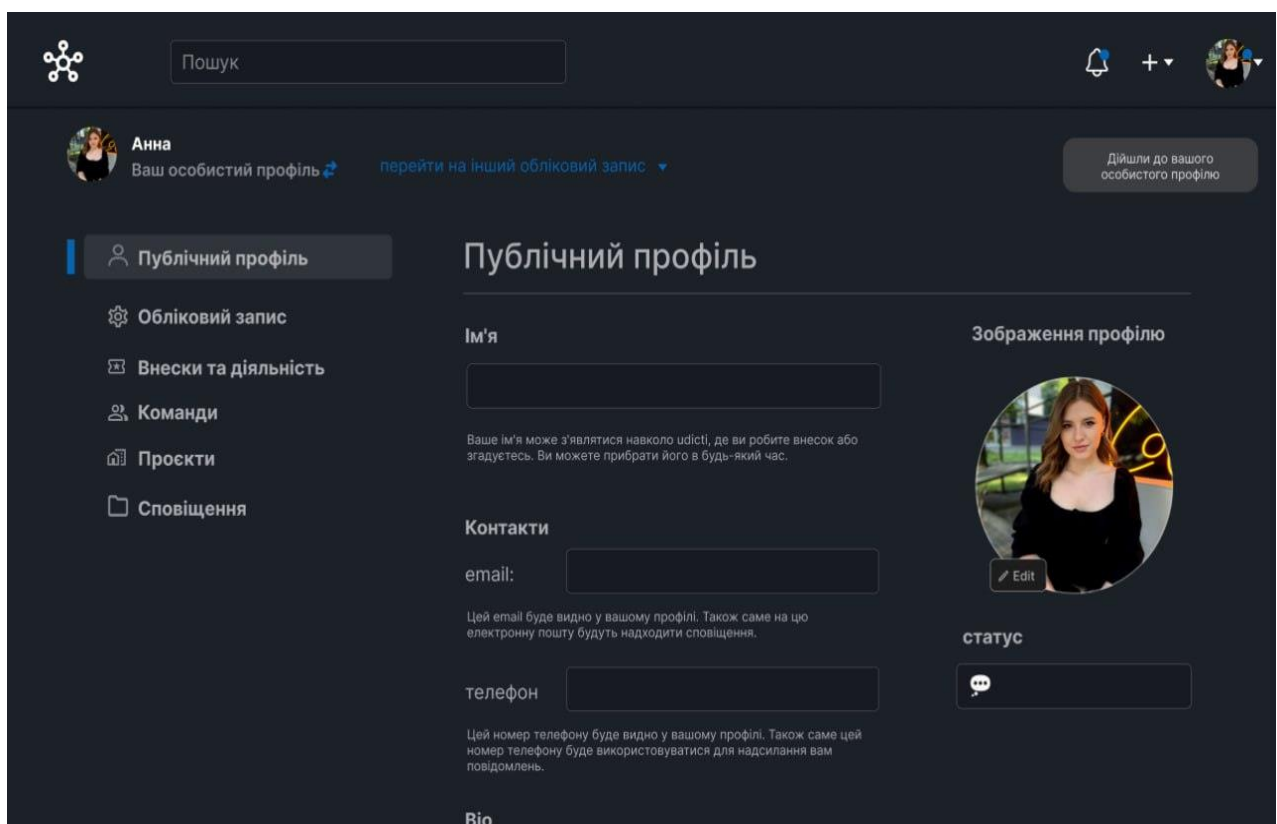


Рисунок 4.3 – Налаштування облікового запису

Після успішної авторизації користувача, відкривається головне вікно програми. Приклад вигляду головного вікна зображено на рисунку 4.4.



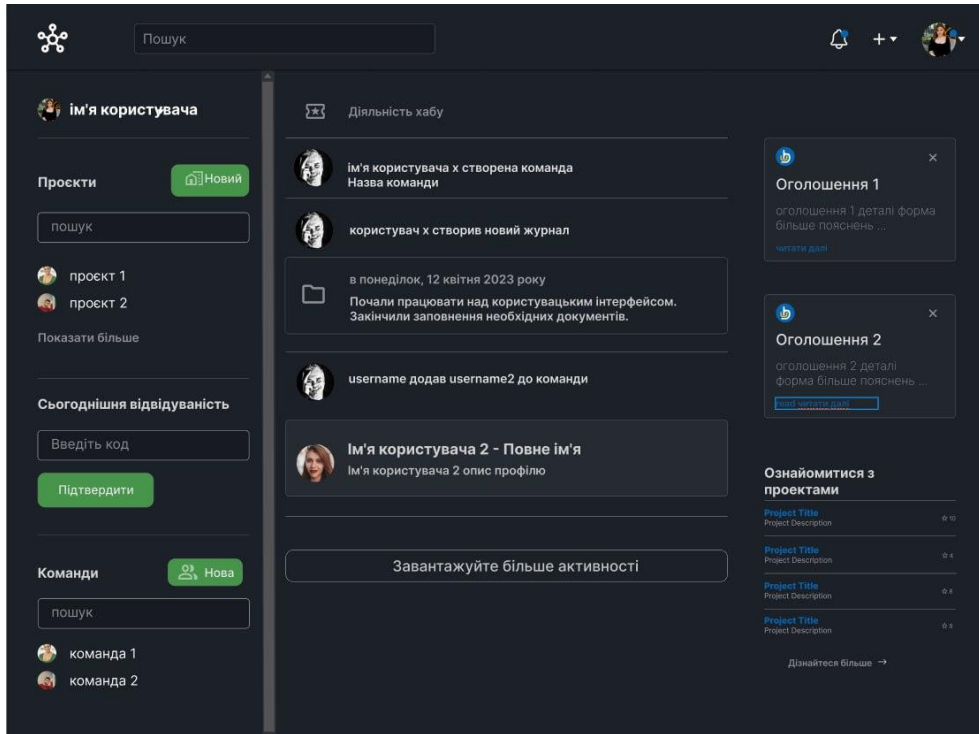


Рисунок 4.4 – Головний екран програми

Натиснувши на аватар користувача, можна перейти до його профілю в системі (рис. 4.5).

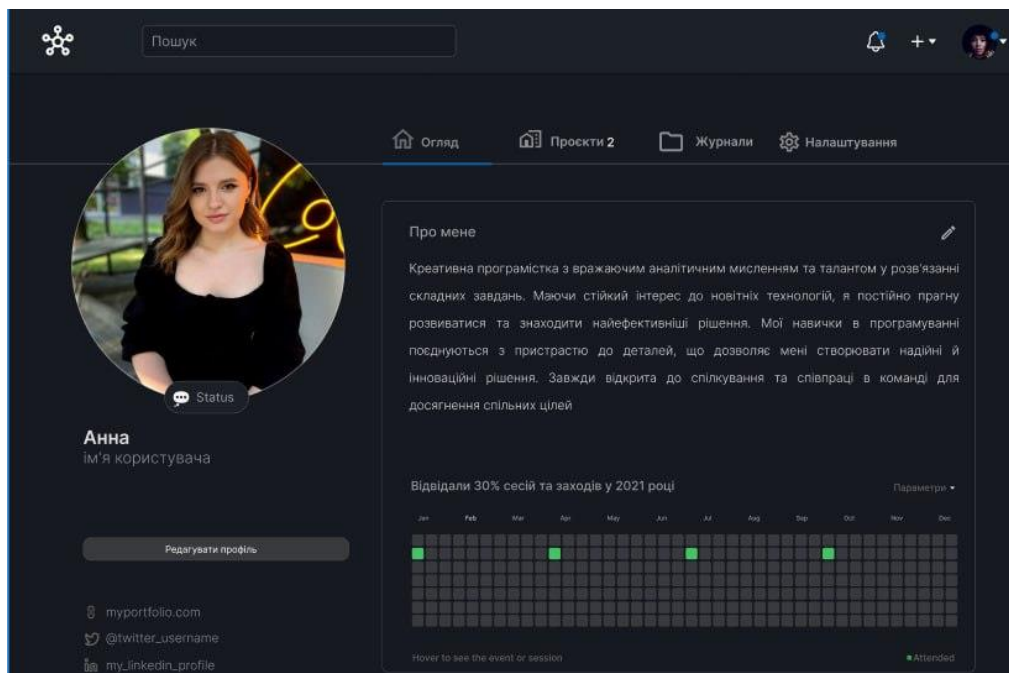
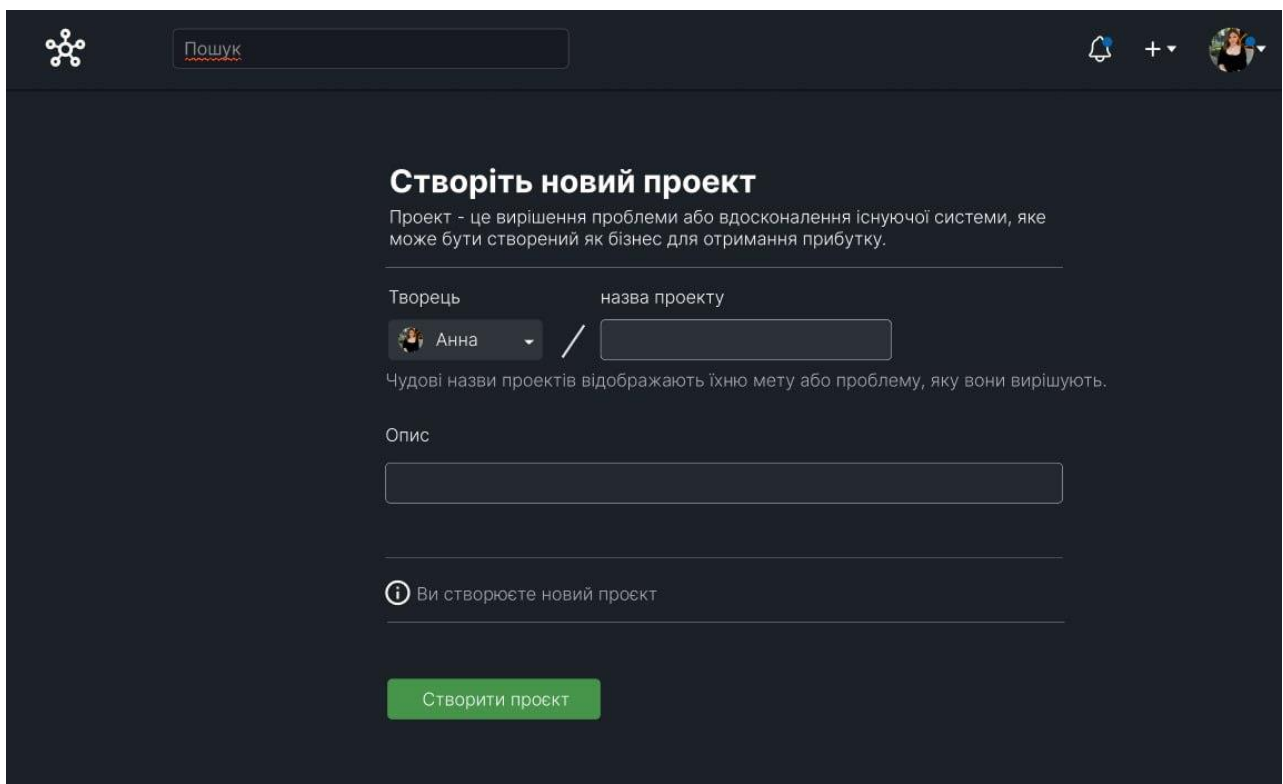


Рисунок 4.5 – Вікно з інформацією про користувача

Після натискання на кнопку "Новий" у розділі "Проект" на головному екрані користувач має можливість відкрити вікно, в якому доступна можливість створення нового розділу для проекту (рис. 4.6). Це інтуїтивно зрозумілий та легкий у використанні інтерфейс, що дозволяє зручно керувати створенням нових проектів. Відображення даного вікна створення розділу не лише полегшує користувачеві початковий етап роботи з платформою, але й створює зручні умови для організації та керування проектами.

Усі необхідні опції та налаштування доступні в цьому вікні, що сприяє ефективному створенню нових розділів та подальшій продуктивній роботі з проектами.

Такий деталізований та інтуїтивний процес створення розділу покращує користувацький досвід та забезпечує зручність у використанні програмного забезпечення для управління проектами.



Пошук

### Створіть новий проект

Проект - це вирішення проблеми або вдосконалення існуючої системи, яке може бути створений як бізнес для отримання прибутку.

Творець: Анна

назва проекту:

Чудові назви проектів відображають їхню мету або проблему, яку вони вирішують.

Опис:

Ви створюєте новий проект

Створити проект

Рисунок 4.6 – Вікно створення нового проекту

На рисунку 4.7 зображено вікно перегляду інформації про проєкт, де також можна редагувати інформацію про нього.

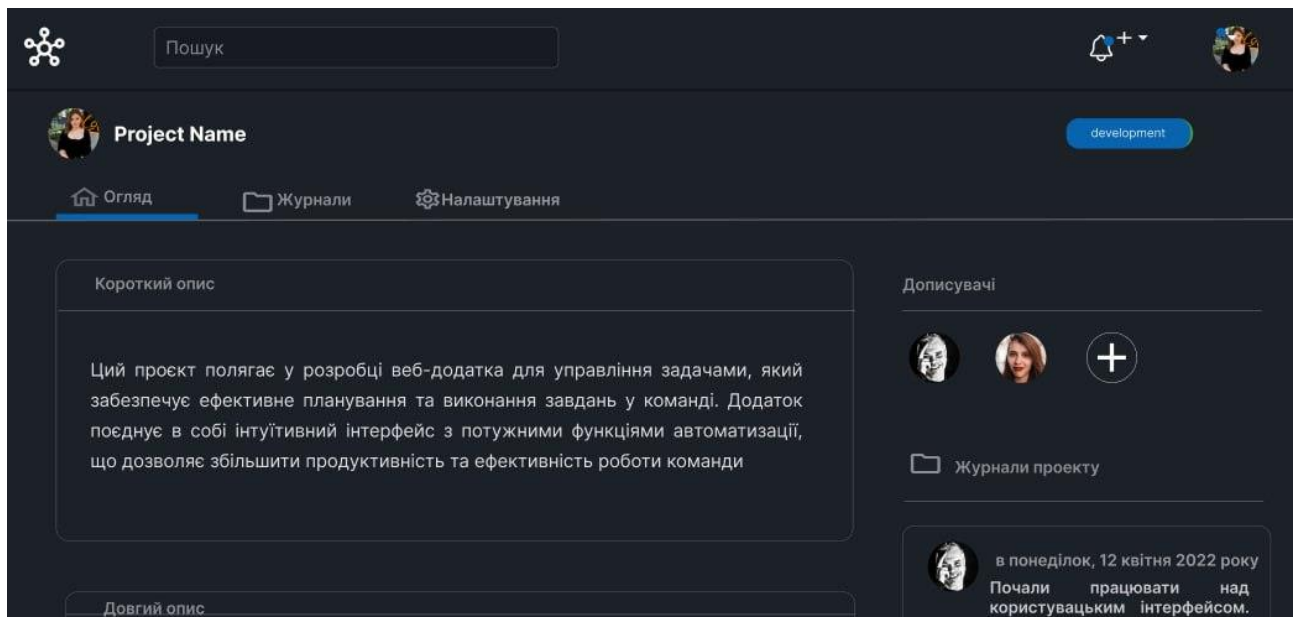


Рисунок 4.7 – Вікно проєкта

З метою візуалізації та аналізу ключових аспектів проєктів було розроблено вікно звітності, яке ілюструє функціональність створеної моделі. На рисунку 4.8 можна спостерігати інтерфейс, який сприяє створенню детальних звітів, графіків та проведенню аналізу даних, спрямованого на оцінку ефективності проєктів.

Однією з ключових особливостей цього вікна є можливість відображення комплексної інформації, що включає в себе якісні показники, так і кількісні параметри проєктів.

Графічне представлення даних сприяє швидкому сприйняттю та порівнянню результатів, дозволяючи приймати обґрунтовані управлінські рішення.

Крім того, зазначена модель аналізу проєктів може бути використана для автоматизації процесу створення звітів, що значно економить час та знижує

ймовірність помилок. Вона дозволяє оперативно оцінювати ефективність проєктів, що є критично важливим у сучасному управлінні проєктами.

Вікно звітності визначається як ключовий інструмент для керівників та аналітиків, який відіграє значущу роль у підвищенні ефективності управління проєктами.

Завдяки вікну звітності керівники можуть отримувати деталізовані та систематизовані звіти, які охоплюють різні аспекти проєктів, такі як витрати, прогрес виконання завдань, терміни, використання ресурсів та інші ключові показники. Ця аналітична інформація дозволяє керівникам отримати повний обсяг управлінської перспективи та швидко реагувати на будь-які зміни чи виклики, що виникають у процесі виконання проєктів.

Оперативний доступ до звітів у вигляді вікна сприяє зручності та ефективності роботи з аналітичною інформацією. Керівники можуть швидко переглядати, аналізувати та порівнювати дані, необхідні для прийняття важливих управлінських рішень.

Після проведеного тестування системи управління стартап-проєктами можна визначити, що програма проявила високий рівень ефективності та надійності.

Весь функціонал був виконаний відповідно до технічного завдання, і програма продемонструвала свою здатність взаємодіяти з різноманітними завданнями управління стартап-проєктами.

Зокрема, система виявила здатність ефективно аналізувати та відстежувати хід виконання проєктів, надавати зрозумілу та конкретну інформацію про стан завдань, а також забезпечувати ефективний обмін даними між учасниками проєкту.

Таким чином, результати тестування підтвердили, що система працює стабільно, відповідає стандартам якості та враховує всі наявні вимоги технічного завдання.

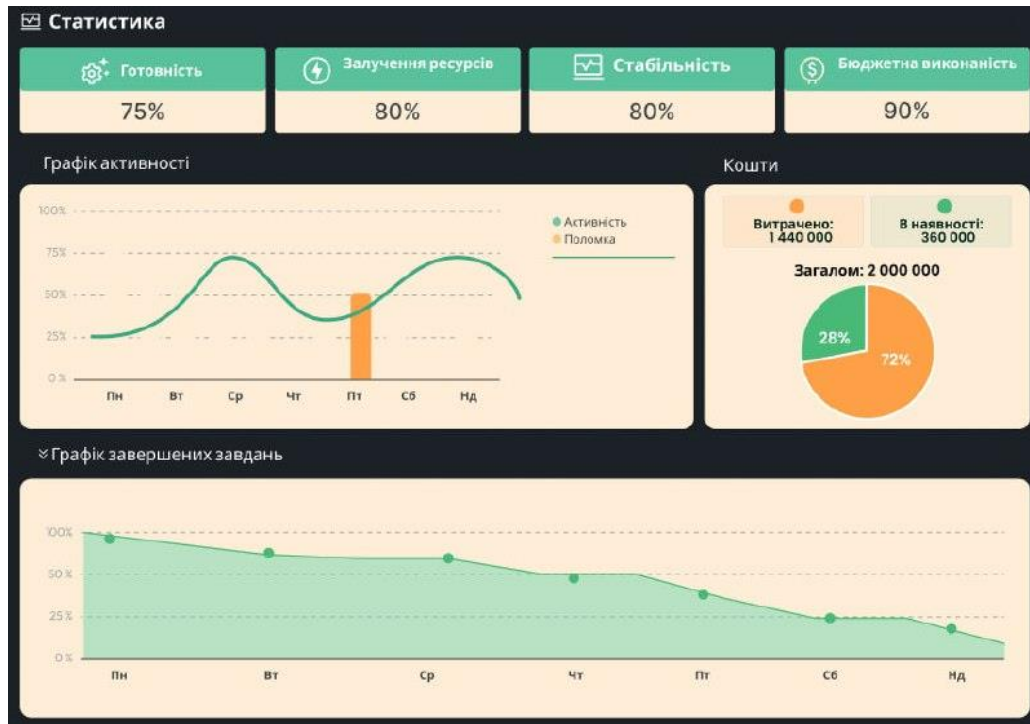


Рисунок 4.8 – Вікно звітності проєкту

Загалом, цей успішний етап тестування вказує на те, що розроблена система має потенціал стати ефективним інструментом для управління стартап-проєктами, сприяючи покращенню процесів прийняття рішень та забезпеченню успішності розвитку проєктів.

### 4.3 Розробка інструкції користувача

Інструкція користувача визнається невід’ємною складовою технічної документації, яка виявляється ключовою для надання вичерпної інформації користувачеві з метою забезпечення ефективного використання програмного продукту.

Починаючи з визначення мінімальних технічних вимог, інструкція дозволяє користувачеві встановити сумісність своєї системи з програмою, гарантуючи стабільне функціонування.

Особлива увага приділяється розділу, що стосується конфігурації персонального комп'ютера. Тут користувач отримує детальну інформацію про вимоги до обладнання, включаючи обсяг оперативної пам'яті, тип процесора, об'єм вільного простору на жорсткому диску та інші технічні параметри. Це надає можливість користувачеві готувати свій комп'ютер до ефективного використання програми.

Детальний опис системи управління стартап-проектами є ключовою складовою інструкції. Він надає користувачеві необхідні знання та навички для успішної взаємодії з програмним продуктом. Описуються основні функції, інтерфейс та інші аспекти, що можуть виникнути під час використання системи.

Інструкція користувача стає невід'ємною частиною процесу впровадження програмного продукту та гарантує його успішну експлуатацію, допомагаючи користувачеві максимально використовувати функціонал програми відповідно до його потреб і вимог.

Ця важлива інформація подана в таблиці 4.1, що створює основу для забезпечення ефективного використання програмного продукту та задоволення потреб користувача.

Для належного встановлення програмного продукту слід ретельно виконати послідовні дії, а саме запустити файл «PM.exe».

Після виконання цієї операції розпочне роботу інсталятор, в якому важливо уважно керуватися інструкціями з метою успішної установки додатку.

Після завершення цього процесу на робочому столі користувача автоматично з'явиться ярлик встановленого додатку.

Активуючи цей ярлик, користувач отримає доступ до вікна програми, відкривши яке зможе безпосередньо взаємодіяти з усіма функціональними можливостями програмного продукту.

Таблиця 4.1 – Мінімальна конфігурація комп'ютера

Характеристика	Опис
Операційна система	Windows 8.1/10/11
Процесор	32-розрядний (x86) або 64-розрядний (x64) процесор з тактовою частотою 2 ГГц
Оперативна пам'ять	Мінімум 1 ГБ RAM
Вільне місце	Мінімум 1 ГБ

Такий простий та ефективний процес встановлення програми забезпечить зручність користування системою управління стартап-проєктами та надійність її роботи.

#### 4.4 Висновки

У четвертому розділі були розглянуті особливості тестування програмного забезпечення системи, обрано методом тестування метод сірої скриньки, який був успішно використаний для перевірки працездатності готового програмного продукту та його відповідності технічному завданню.

Проведене комплексне тестування системи управління стартап-проєктами підтвердило її працездатність і відповідність поставленим вимогам.

Також була визначена мінімальна конфігурація персонального комп'ютера і розроблена інструкція користувача з процесу встановлення додатку.

Інструкція спрощує взаємодію користувачів з програмним продуктом, надаючи їм чіткі та послідовні кроки для використання різноманітних функцій та можливостей.

Такий підхід сприяє зручності використання та позитивному досвіду користувачів під час взаємодії з розробленою системою.



## 5 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота за темою «Розробка методів та програмних засобів системи управління стартап-проектами» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проєкту і переконати його в економічній доцільності такого кроку [32].

Для наведеного випадку мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

### **5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки**

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка методів та програмних засобів системи управління стартап-

проектами» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1 [32,33].

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Продовження таблиці 5.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 5.2.

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	4	3	4
3. Ринкові переваги (ціна продукту)	4	4	3
4. Ринкові переваги (технічні властивості)	4	3	3
5. Ринкові переваги (експлуатаційні витрати)	4	4	3
6. Ринкові перспективи (розмір ринку)	4	4	4
7. Ринкові перспективи (конкуренція)	4	3	4
8. Практична здійсненність (наявність фахівців)	3	3	3
9. Практична здійсненність (наявність фінансів)	4	4	4
10. Практична здійсненність (необхідність нових матеріалів)	3	3	3
11. Практична здійсненність (термін реалізації)	3	3	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	45	42	43
Середньоарифметична сума балів $СБ_c$	43,3		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3 [32,33].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка методів та програмних засобів системи управління стартап-проектами» становить 43,3 бали, що, відповідно до таблиці 5.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

## 5.2 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка методів та програмних засобів системи управління стартап-проектами», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

### 5.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними

ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою 5.1 [32,33]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.1)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дні;

$T_p$  – середнє число робочих днів в місяці,  $T_p=22$  дні.

$$Z_o = 30000,00 \cdot 63 / 22 = 85909,09 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.4.

Таблиця 5.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проєкту	30000	1363,64	63	85909,09
Інженер-розробник програмного забезпечення	25000	1136,36	63	71590,91
Консультант	17000	772,73	63	48681,82
Всього				206181,82

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР розраховуємо за формулою 5.2:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.2)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою 5.3:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.3)$$

де  $M_M$  – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо  $M_M=6700,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду [32,33];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  – середнє число робочих днів в місяці, приблизно  $T_p = 22$  дні;

$t_{зм}$  – тривалість зміни, год.

$$C_1 = 6700,00 \cdot 1,10 \cdot 1,65 / (22 \cdot 8) = 69,09 \text{ грн.}$$

$$З_{р1} = 69,09 \cdot 6,00 = 414,56 \text{ грн.}$$

Величина витрат на основну заробітну плату робітників наведена в табл. 5.5.

Таблиця 5.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
Підготовка робочого місяця дослідника	6	2	1,1	69,09	414,56

Інсталяція програмного забезпечення	4	5	1,7	106,78	427,13
Тестування	8	5	1,7	106,78	854,25
Всього					1695,94

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою 5.4:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (5.4)$$

де  $H_{\text{дод}}$  – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (206181,82 + 1695,94) \cdot 11 / 100\% = 22866,55 \text{ грн.}$$

### 5.2.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою 5.5:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (5.5)$$

де  $H_{\text{зн}}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (206181,82 + 1695,94 + 22866,55) \cdot 22 / 100\% = 50763,75 \text{ грн.}$$

### 5.2.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень.

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою 5.6:



$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (5.6)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$C_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$C_{ej}$  – вартість відходів  $j$ -го найменування, грн/кг.

$M_1 = 3 \cdot 195,00 \cdot 1,1 - 0,000 \cdot 0,00 = 643,5$  грн.

Проведені розрахунки зведемо до таблиці 5.6.

Таблиця 5.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за од	Норма витрат, од	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір	195	3	0	0	643,5
Канцелярське приладдя (набір офісного працівника)	3	0	0	511,5	3
Flesh-пам'ять	150	3	0	0	495
Всього					1650

#### 5.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_e$ ), які використовують при проведенні науково-дослідної роботи на тему «Розробка методів та програмних засобів системи управління стартап-проектами» відсутні.

### 5.2.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Витрати на «Спецустаткування для наукових (експериментальних) робіт» в роботі на тему «Розробка методів та програмних засобів системи управління стартап-проектами» відсутні.

### 5.2.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою 5.7:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{инрг}} \cdot C_{\text{прог.и}} \cdot K_i, \quad (5.7)$$

де  $C_{\text{инрг}}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог.и}}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{\text{прог}} = 15000,00 \cdot 1 \cdot 1,12 = 50400 \text{ грн.}$$

Отримані результати зведемо до таблиці 5.7.

Таблиця 5.7 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
IntelliJ IDEA	1	5500	6160
Операційна система Microsoft Windows 10 Pro for Workstations	3	15000	50400
Всього			56560

### 5.2.7 Амортизація обладнання, програмних засобів та приміщень

У спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою 5.8:

$$A_{обл} = \frac{Ц_б}{T_г} \cdot \frac{t_{вик}}{12}, \quad (5.8)$$

де  $Ц_б$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_г$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (25000,00 \cdot 3) / (4 \cdot 12) = 1562,5 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.8.

Таблиця 5.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
ПК DELL OptiPlex 7010 SFF (N001O7010SFF UBU), монітор	25 000	4	3	1562,50

ПК Expert PC Ultimate (I11400F.16.H1S2.16 50.A3836), монітор	26 000	4	3	1625,00
ПК 2E Integer Intel C J4005, монітор	12 000	4	3	750,00
Оргтехніка	7000	4	3	437,50
Операційна система Microsoft Windows 10 Pro for Workstations	50400	1	3	12600,00
IntelliJ IDEA	6160	1	3	1540,00
Всього				18515,00

### 5.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою 5.9:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (5.9)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 7,50$  грн;

$K_{eni}$  – коефіцієнт, що враховує використання потужності,  $K_{eni} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = 0,3 \cdot 480,0 \cdot 7,50 \cdot 0,95 / 0,97 = 1057,73 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.9.

Таблиця 5.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн

ПК DELL OptiPlex	0,3	480	1057,73
ПК Expert PC Ultimate	0,3	504	1110,62
ПК 2E Integer Intel C J4005	0,2	400	587,63
Оргтехніка	0,45	15	49,58
Всього			2805,56

### 5.2.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою 5.10:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (5.10)$$

де  $H_{cv}$  – норма нарахування за статтею «Службові відрядження», прийmemo  $H_{cv} = 20\%$ .

$$B_{cv} = (206181,82 + 1695,94) \cdot 20 / 100\% = 41575,55 \text{ грн.}$$

5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою 5.11:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (5.11)$$

де  $H_{\text{сп}}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{\text{сп}} = 35\%$ .

$$B_{\text{сп}} = (206181,82 + 1695,94) \cdot 35 / 100\% = 72757,21 \text{ грн.}$$

### 5.2.10 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою 5.12:

$$I_{\text{с}} = (Z_{\text{o}} + Z_{\text{p}}) \cdot \frac{H_{\text{іс}}}{100\%}, \quad (5.12)$$

де  $H_{\text{іс}}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{\text{іс}} = 70\%$ .

$$I_{\text{с}} = (206181,82 + 1695,94) \cdot 70 / 100\% = 145514,43 \text{ грн.}$$

### 5.2.11 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою 5.13:

$$B_{\text{нзв}} = (Z_{\text{o}} + Z_{\text{p}}) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (5.13)$$

де  $H_{\text{нзв}}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{\text{нзв}} = 120\%$ .

$$B_{нзв} = (206181,82 + 1695,94) \cdot 120 / 100\% = 249\,453,31 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи розраховуємо як суму всіх попередніх статей витрат за формулою 5.14:

$$B_{заг} = Z_o + Z_p + Z_{ood} + Z_n + M + K_{\epsilon} + B_{спец} + B_{прг} + A_{обл} + B_e + B_{св} + B_{сп} + I_{\epsilon} + B_{нзв}. \quad (5.14)$$

$$B_{заг} = 964939,12 \text{ грн.}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою 5.15:

$$ZB = \frac{B_{заг}}{\eta}, \quad (5.15)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta = 0,7$ .

$$ZB = 964939,12 / 0,7 = 1378484,46 \text{ грн.}$$

### **5.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором**

У ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку [33].

Результати дослідження передбачають комерціалізацію протягом 3-х років реалізації на ринку.

У цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$\Delta N$  – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

1-й рік – 1000 проєктних груп;

2-й рік – 3000 проєктних груп;

3-й рік – 2500 проєктних груп.

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 5000 проєктних груп;

$C_o$  – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 250000,00 грн;

$\pm\Delta C_o$  – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 20000,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta\Pi_i$  для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою 5.16 [32,33]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\mathcal{G}}{100}\right), \quad (5.16)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту. Приймемо  $\rho = 25\%$ ;

$\mathcal{G}$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році  $\mathcal{G} = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (20000,00 \cdot 5000,00 + 270000 \cdot 1000) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 63183050$$

грн.

Збільшення чистого прибутку 2-го року:



$\Delta\Pi_2 = (20000,00 \cdot 5000,00 + 270000 \cdot (1000 + 3000)) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) =$   
201502700 грн.

Збільшення чистого прибутку 3-го року:

$\Delta\Pi_3 = (20000,00 \cdot 5000,00 + 270000 \cdot (1000 + 3000 + 2500)) \cdot 0,83 \cdot 0,25 \cdot (1 -$   
 $0,18/100\%) = 316769075$  грн.

Приведена вартість збільшення всіх чистих прибутків  $ПП$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки, знаходиться за формулою 5.17:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (5.17)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,3$ ;

$t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 63183050 / (1 + 0,3)^1 + 201502700 / (1 + 0,3)^2 + 316769075 / (1 + 0,3)^3 =$$
  
312017268,78 грн.

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки, знаходиться за формулою 5.18:

$$PV = k_{инв} \cdot 3B, \quad (5.18)$$

де  $k_{inv}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{inv}=4$ ;

$ZB$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 1378484,46 грн.

$$PV = k_{inv} \cdot ZB = 4 \cdot 1378484,46 = 5513937,821.$$

Абсолютний економічний ефект  $E_{abc}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме за формулою 5.19:

$$E_{abc} = III - PV \quad (5.19)$$

де  $III$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 312017268,78 грн;

$PV$  – теперішня вартість початкових інвестицій, 5513937,821 грн.

$$E_{abc} = III - PV = 312017268,78 - 5513937,821 = 306503330,95 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, знаходиться за формулою 5.20:

$$E_g = T_{ж} \sqrt[1 + \frac{E_{abc}}{PV}]{} - 1, \quad (5.20)$$

де  $E_{abc}$  – абсолютний економічний ефект вкладених інвестицій, 306503330,95 грн;

$PV$  – теперішня вартість початкових інвестицій, 5513937,821 грн;

$T_{ж}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_g = \sqrt[3]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 306503330,95 / 5513937,821)^{1/3} = 2,84.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{min}$ , знаходиться за формулою 5.21:

$$\tau_{min} = d + f, \quad (5.21)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = 0,11$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{min} = 0,11 + 0,25 = 0,36 < 2,84$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка методів та програмних засобів системи управління стартап-проектами» доцільно.

Період окупності інвестицій  $T_{ок}$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розрахуємо за формулою 5.22:

$$T_{ок} = \frac{1}{E_g}, \quad (5.22)$$

де  $E_g$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 2,84 = 0,35 \text{ року.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

#### **5.4 Висновки**

За проведеними дослідженнями рівень комерційного потенціалу розробки за темою «Розробка методів та програмних засобів системи управління стартап-проектами» становить 43,3 бали, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

Також термін окупності становить 0,35 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка методів та програмних засобів системи управління стартап-проектами».

## ВИСНОВОК

Під час виконання магістерської кваліфікаційної роботи було розроблено систему управління стартап-проектами.

Робота виконана в середовищі розробки IntelliJ IDEA з використанням мови програмування Java та системи управління базами даних MySQL, що забезпечило ефективну реалізацію системи та швидкий доступ до необхідної інформації.

Магістерська кваліфікаційна робота оформлена згідно методичних вказівок [34].

Проведено аналіз сучасного стану розвитку систем управління стартап-проектами, ідентифіковані основні конкурентні продукти, зокрема, додатки «Trello», «Asana» та «Monday.com», визначено їхні переваги та недоліки. Здійснено детальне порівняння цих застосунків із розробленим власним програмним продуктом. У результаті аналізу конкурентів розкриті ключові аспекти, які вирізняють розроблений продукт у сфері управління стартапами в порівнянні з існуючими аналогами.

При виконанні магістерської кваліфікаційної роботи визначено функціонал системи для управління стартап-проектами та засоби її реалізації, розроблено метод, моделі та алгоритми роботи системи, розроблено програмне забезпечення системи та проведено тестування її роботи.

Програмна реалізація системи управління стартап-проектами включає розробку її базових програмних модулів, зокрема, модуля для реєстрації та авторизації користувачів, модуля управління проектами, модуля аналізу проектів та генерації звітів.

Подальшого розвитку дістав метод управління стартап-проектами, який враховує індивідуальні потреби та особливості малих компаній, що дозволить системі адаптуватися до сучасного динамічного бізнес-середовища.

Подальшого розвитку дістала модель системи управління стартапами, яка враховує особливості організації роботи зі стартапами, сучасні стратегії моніторингу й оптимізації робочих процесів, формування графіків і звітів, що дозволяє покращити процес керування проектами.

Під час тестування системи підтверджена її повна функціональність та відповідність визначеним технічним вимогам. Результати тестування свідчать про ефективність функціоналу та стабільну роботу програмного забезпечення.

Розроблена інструкція користувача описує процес використання системи управління стартап-проектами.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Digital transformation to develop an effective startup system [Електронний ресурс] URL: <https://www.emerald.com/insight/content/doi/10.1108/TG-01-2023-0006/full/html>
2. Algorithm Development Management [Електронний ресурс] URL: [https://www.researchgate.net/publication/331613056\\_Algorithm\\_for\\_Innovative\\_Development\\_Management\\_of\\_a\\_Project-Oriented\\_Organization](https://www.researchgate.net/publication/331613056_Algorithm_for_Innovative_Development_Management_of_a_Project-Oriented_Organization)
3. A Project Management System [Електронний ресурс] URL: [https://www.academia.edu/25086781/A\\_Web\\_Based\\_Project\\_Management\\_System](https://www.academia.edu/25086781/A_Web_Based_Project_Management_System)
4. System Development Life Cycle [Електронний ресурс] URL: <https://svitla.com/blog/system-development-life-cycle>
5. Mathematical Tools for Automation Systems I: Modeling and Simulation [Електронний ресурс] URL: [https://www.researchgate.net/publication/251217201\\_Mathematical\\_Tools\\_for\\_Automation\\_Systems\\_I\\_Modeling\\_and\\_Simulation](https://www.researchgate.net/publication/251217201_Mathematical_Tools_for_Automation_Systems_I_Modeling_and_Simulation)
6. Войтко В. В., Круподьорова Л.М., Денисюк А.В., Маланчук А. В. Розробка методів та засобів системи управління стартап-проектами // Електронні інформаційні ресурси: створення, використання, доступ. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2023 р. – Суми/Вінниця: НІКО/ КЗВО «Вінницька академія безперервної освіти», 2023. – С. 71-75.
7. Trello [Електронний ресурс]. URL: <https://trello.com/uk/education>
8. Asana [Електронний ресурс]. URL: <https://asana.com/>
9. Monday.com [Електронний ресурс]. URL: <https://monday.com/lang/com>
10. Agile – методологія гнучкої розробки [Електронний ресурс] URL: <https://qagroup.com.ua/publications/agile-info/>

11. Lean Startup – Методологія концепції підприємництва [Електронний ресурс] URL: <https://www.mbastrategy.ua/content/view/4718/461/>
12. Ries, E. (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Books. – p.75-90.
13. JIRA – Система управління проектами [Електронний ресурс] URL: <https://iampm.club.ua/blog/shho-take-jira-i-yak-z-neyu-praczuivati/>
14. SWOT – Ефективний інструмент бізнес-планування [Електронний ресурс] URL: <https://business.diia.gov.ua/handbook/marketing/so-take-swot-analiz>
15. Бізнес планування (Business Dashboard) – [Електронний ресурс] URL: <https://elearn.nubip.edu.ua/mod/book/view.php?id=373455>
16. Osterwalder, A., & Pigneur, Y. (2010). Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons. – p. 105-120.
17. KPI – фінансова та нефінансова система оцінки [Електронний ресурс] URL: <https://business.diia.gov.ua/handbook/prodazi/so-take-kpi-prikladi>
18. Chesbrough, H. W. (2013). Open Innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business Press. – p. 180-195.
19. Morris, M. H., Kuratko, D. F., & Covin, J. G. (2011). Corporate entrepreneurship & innovation. Cengage Learning. – p. 10-25.
20. Аналіз ризиків у стартап-проектах - [Електронний ресурс] URL: <https://www.yarlan.com.ua/articles/10-analiz-rizikiv>
21. Blank, S. G., & Dorf, B. (2012). The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company. K & S Ranch. – p. 35-50.
22. Gans, J. S., Stern, S., & Ebling, M. (2016). The Scale-Up Opportunity: A New Model for Innovation. Pearson. – p. 275-290.
23. Brown, T. (2009). Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. Harper Business. – p. 310-325.
24. How are relationships modeled in a database? [Електронний ресурс] URL:



- <https://www.quora.com/How-are-relationships-modeled-in-a-database>
25. Database and Information Systems Development [Електронний ресурс] URL: <https://highervista.medium.com/database-and-information-systems-development-data-models-an-introduction-to-3-essential-models-ba60d5413d7d>
  26. Machine Learning Model Development and Model Operations: Principles and Practices [Електронний ресурс] URL: <https://www.kdnuggets.com/2021/10/machine-learning-model-development-operations-principles-practice.html>
  27. Designing A Login Register and User Authentication System [Електронний ресурс] URL: <https://medium.com/@mycodingmantras/designing-a-login-register-and-user-authentication-script-in-python-326a11821504>
  28. Spring Boot User Registration and Login Example Tutorial [Електронний ресурс] URL: <https://www.javaguides.net/2018/10/user-registration-module-using-springboot-springmvc-springsecurity-hibernate5-thymeleaf-mysql.html>
  29. Software Project Management and Development [Електронний ресурс] URL: <https://www.ecs.soton.ac.uk/module/COMP6204/>
  30. The different types of software testing [Електронний ресурс] URL: <https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing>
  31. Software Testing Methodologies Guide: A High-Level Overview [Електронний ресурс] URL: <https://www.parasoft.com/blog/software-testing-methodologies-guide-a-high-level-overview/>
  32. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В.В.Кавецький. – Вінниця: ВНТУ, 2021. – 42 с.

- 33.Кавецький В. В., Козловський В. О., Причепя І. В. Економічне обґрунтування інноваційних рішень: практикум. – Вінниця: ВНТУ, 2016. – 113 с.
- 34.Методичні вказівки до виконання магістерської кваліфікаційної роботи для студентів спеціальності 121 «Інженерія програмного забезпечення» / уклад.: О. Н. Романюк, Г. О. Черноволик. – Вінниця: ВНТУ, 2022. – 50 с.

**Додаток А (обов'язковий). Технічне завдання**  
Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії


ЗАТВЕРДЖУЮ

д.т.н., проф. О. Н. Романюк


"19" вересня 2023 р.

**Технічне завдання**  
на магістерську кваліфікаційну роботу «Розробка методів та програмних  
засобів системи управління стартап-проеєктами» за спеціальністю  
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:

 к.т.н., доц. В.В. Войтко  
" 19 " 09 2023 р.

Виконала:

 студентка гр.3ПІ-22м А.В. Маланчук  
" 19 " 09 2023 р.

Вінниця – 2023 р.

### **1. Найменування та галузь застосування**

Магістерська кваліфікаційна робота: «Розробка методів та програмних засобів системи управління стартап-проектами».

Галузь застосування – додаток для малих компаній та організацій.

### **2. Підстава для розробки**

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ ректора № 247 від 18 вересня 2023 р. по ВНТУ про закріплення тем МКР.

### **3. Мета та призначення розробки**

Метою магістерської кваліфікаційної роботи є полегшення та оптимізація процесів управління стартапами й малими компаніями шляхом розробки та впровадження спеціалізованої системи управління проектами, що дозволить ефективний розподіл ресурсів для розв'язування поставлених задач.

Призначення роботи – розробка методів і засобів реалізації програмних системи управління стартап-проектами.

### **4 Вихідні дані для проведення НДР**

Перелік основних літературних джерел, на основі яких буде виконуватися

магістерська кваліфікаційна робота:

1. Войтко В. В., Круподьорова Л.М., Денисюк А.В., Маланчук А. В. Розробка методів та засобів системи управління стартап-проектами // Електронні інформаційні ресурси: створення, використання, доступ. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2023р. – Суми/Вінниця: НІКО/ КЗВО «Вінницька академія безперервної освіти», 2023. – С. 71-75.

2. Ries, E. (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Books. – p. 75-90.
3. Spring Boot User Registration and Login Example Tutorial [Електронний ресурс] URL: <https://www.javaguides.net/2018/10/user-registration-module-using-springboot-springmvc-springsecurity-hibernate5-thymeleaf-mysql.html>
4. Кавецький В. В., Козловський В. О., Причепя І. В. Економічне обґрунтування інноваційних рішень: практикум. – Вінниця: ВНТУ, 2016. – 113 с.

### 5. Технічні вимоги

Вихідні дані до роботи: середовище розробки IntelliJ IDEA, мова розробки Java, система керування базами даних – MySQL, операційна система – Windows10.

### 6. Конструктивні вимоги

Інтерфейс програмного застосунку повинен відповідати естетичним та ергономічним вимогам, програма має бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

**7. Перелік технічної документації, що пред'являється по закінченню робіт:**

- пояснювальна записка до магістерської кваліфікаційної роботи;
- технічне завдання;
- лістинг програми.

### 8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

### 9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз стану розвитку систем управління стартап-проектами	20.09.2023 – 30.09.2023	<i>век</i>
2	Розробка методу, моделей та алгоритмів роботи системи управління стартап-проектами	01.10.2023 – 17.10.2023	<i>век</i>
3	Програмна реалізація системи управління стартап-проектами	18.10.2023 – 04.11.2023	<i>век</i>
4	Тестування програмного продукту	05.11.2023 – 21.11.2023	<i>век</i>
5	Економічна частина	22.11.2023 – 01.12.2023	<i>век</i>

### 10. Порядок контролю та прийняття

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б (обов'язковий). Протокол перевірки на плагіат  
**ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ)  
 РОБОТИ**

Назва роботи: **Розробка методів та програмних засобів системи управління стартап-проектами**

Тип роботи: магістерська кваліфікаційна робота

Підрозділ : кафедра програмного забезпечення, ФІТКІ, ЗПІ – 22м

Науковий керівник: Войтко В.В.

Unicheck	
Оригінальність	97,5 %
Схожість	2,5 %

**Аналіз звіту подібності**

- **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**
  - Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
  - Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку



Черноволик Г. О.

Опис прийнятого рішення: **допустити до захисту**

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck

Автор роботи



Маланчук А.В.

Керівник роботи

Войтко В.В.

## Додаток В (довідниковий). Лістинг програми

```

import javafx.concurrent.Task;
import javafx.scene.Node;
import javafx.scene.image.Image;
import javafx.animation.Animation;
import javafx.animation.KeyFrame;
import javafx.animation.KeyValue;
import javafx.animation.Timeline;
import javafx.event.EventHandler;
import javafx.geometry.HPos;
import javafx.geometry.VPos;
import javafx.scene.control.SkinBase;
import javafx.scene.input.MouseEvent;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Rectangle;
import javafx.util.Duration;
public class Main extends Launcher {

    @Override
    public void build(Context context) {

        Layout layout = new Layout(context);
        context.setLayout(layout);

        Loader loadCircle = new LoadCircle("Starting..", "");
        Task<View> loadViews = new LoadViews(context, loadCircle); // Load View task

        Thread tLoadViews = new Thread(loadViews);
        tLoadViews.setDaemon(true);
        tLoadViews.start();

        layout.setContent((Node) loadCircle);

        loadViews.setOnSucceeded(event -> {
            layout.setNav(context.routes().getView("drawer"));
            context.routes().nav("dash");
        });

        icons.add(new Image(context.getResource("style/img/logo_64.png").toExternalForm(), 128, 128, true,
true));

//    layout.setContent((Node) loadCircle);
    }

}
public abstract class Launcher extends Application {

```



```

private final IRoot root = new IRoot();
private final IContext context = new IContext(root, getHostServices());
protected ObservableList<Image> icons = FXCollections.observableArrayList();

private final Logger logger = Logger.getLogger("app");
@Override
public void start(Stage stage) {
    build(context);
    context.setStage(stage);
    Scene scene = new Scene(root);
    stage.setTitle("DashboardFx App!");
    stage.setScene(scene);
    stage.setMaximized(true);

    stage.getIcons().addAll(icons);

    stage.setMinWidth(300);
    stage.setMinHeight(500);

    scene.getStylesheets().addAll(
        clean("style/css/colors.css"),
        clean("style/css/skeleton.css"),
        clean("style/css/bootstrap.css"),
        clean("style/css/theme/simple_info.css"),
        clean("style/css/typographic.css"),
        clean("style/css/immersive_scroll.css"),
        clean("style/css/shapes.css"),
        clean("style/css/helpers.css"),
        clean("style/css/dash.css")
    );

    stage.show();
    CSSFX.start(stage);
//    ScenicView.show(scene);
    addLoggerHandler();
}

private void addLoggerHandler() {
    try {

        // This block configure the logger with handler and formatter
        FileHandler fh = new FileHandler("app.log");
        logger.addHandler(fh);
        SimpleFormatter formatter = new SimpleFormatter();
        fh.setFormatter(formatter);

        // the following statement is used to log any messages
        logger.setLevel(Level.OFF);
    } catch (SecurityException | IOException e) {
        e.printStackTrace();
    }
}

```

```

    }
}

private String clean(String c) {
    return context.getResource(c).toExternalForm();
}

public abstract void build(Context context);
}
public interface Context {

    URL getResource(String res);

    Layout layout();

    void setLayout(Layout layout);

    Routes routes();

    Wrapper wrapper();

    Flow flow();

    Stage stage();

    ObservableList<SearchItem> searchItems();

    @ApiStatus.Experimental
    Logger logger();

    /*****
    *
    * Util methods
    *
    *****/

    SnackBar createSnackBar();

    void openLink(String uri);

    ActionView controllerOf(String view);
}
public class ErrorPage extends ScrollPane {

    private final VBox body = new VBox();
    private final VBox boxTitle = new VBox();
    private final VBox boxInfo = new VBox();

    private final Text errorTitle = new Text();

```

```

private final Text title = new Text();
private final Text legend = new Text();
private final Text info = new Text();

private final TextFlow titleFlow = new TextFlow(title);
private final Hyperlink see = new Hyperlink("See Routes");
private final TextFlow infoFlow = new TextFlow(info, see);

public ErrorPage(String name) {
    errorTitle.setText("404");
    title.setText("The view was not encountered.");
    legend.setText("Sorry, system couldn't find the view.");
    info.setText("View \'' + name + '\'' doesn't exist.");
    configLayout();
}

private void configLayout() {

    errorTitle.getStyleClass().addAll("text-bold", "h3");
    errorTitle.setStyle("-fx-fill: -fx-accent;");
    title.getStyleClass().addAll("text-bold", "h1");
    title.getStyleClass().addAll("-fx-fill: -text-color;");
    legend.getStyleClass().addAll("-fx-fill: -text-color;");
    errorTitle.getStyleClass().addAll("h4");
    info.getStyleClass().addAll("h6");
    info.setStyle("-fx-fill: -text-color;");
    titleFlow.setTextAlignment(TextAlignment.CENTER);
    infoFlow.setTextAlignment(TextAlignment.CENTER);
    boxInfo.getChildren().addAll(infoFlow);

    body.getChildren().setAll(boxTitle, boxInfo);
    this.setContent(body);
    boxTitle.getChildren().setAll(errorTitle, titleFlow);

    body.setAlignment(Pos.CENTER);
    boxTitle.setAlignment(Pos.CENTER);
    boxInfo.setAlignment(Pos.CENTER);

    this.setFitToWidth(true);
    this.setFitToHeight(true);
}
}

public final class ControllerCastException extends Exception {

    private String code;

    public ControllerCastException(String code, String message) {
        super(message);
        this.setCode(code);
    }
}

```

```

    }

    public ControllerCastException(String code, String message, Throwable cause) {
        super(message, cause);
        this.setCode(code);
    }

    public String getCode() {
        return code;
    }

    private void setCode(String code) {
        this.code = code;
    }
}

public final class LoadViewException extends Exception {

    private String code;

    public LoadViewException(String code, String message) {
        super(message);
        this.setCode(code);
    }

    public LoadViewException(String code, String message, Throwable cause) {
        super(message, cause);
        this.setCode(code);
    }

    public String getCode() {
        return code;
    }

    private void setCode(String code) {
        this.code = code;
    }
}

public final class NavigationException extends Exception {

    private String code;
    private Context context;

    private final VBox blank = new VBox();
    private final Text message = new Text();
    private View errorView;

    public NavigationException(Context context, String code, String message) {
        super(message);
        this.setCode(code);
        this.context = context;
    }
}

```

```

        errorView = new SimpleView(
            "error_page", createRouteNotFound(null)
        );
    }

    public String getCode() {
        return code;
    }

    private void setCode(String code) {
        this.code = code;
    }

    private void setMessage(String _message) {
        message.setText(_message);
    }

    public StackPane createRouteNotFound(String message) {
        blank.setAlignment(Pos.BOTTOM_CENTER);
//        root.setPadding(new Insets(100));
        blank.setBackground(new Background(
            new BackgroundImage(
                new Image(context.getResource("style/img/404.png").toExternalForm()),
                BackgroundRepeat.NO_REPEAT,
                BackgroundRepeat.NO_REPEAT,
                BackgroundPosition.CENTER,
                new BackgroundSize(
                    500, 400, false, false,
                    false, false
                )
            )
        ));

        StackPane root = new StackPane(blank);
        if (!blank.getChildren().contains(this.message))
            blank.getChildren().add(this.message);

        root.setPadding(new Insets(100));

        return root;
    }

    public void getRouteNotFound(@NotNull Context context, String message) {
        if (context.routes().getView(errorView.getName()) == null) {
            setMessage(message);
            context.routes().put(errorView);
        }
//        try {

```

```

        context.routes().setView(errorView.getName());
//    } catch (NavigationException e) {
//        throw new RuntimeException(e);
//    }
}
}
public class CurvedChart<X, Y> extends AreaChart<Number, Number> {

    public CurvedChart() {
        this(new NumberAxis(), new NumberAxis());
    }

    public CurvedChart(@NamedArg("xAxis") Axis<Number> numberAxis, @NamedArg("yAxis")
Axis<Number> numberAxis2) {
        super(numberAxis, numberAxis2);
    }

    protected void layoutPlotChildren() {
        super.layoutPlotChildren();

        for(int seriesIndex = 0; seriesIndex < this.getDataSize(); ++seriesIndex) {
            Series<Number, Number> series = (Series)this.getData().get(seriesIndex);
            Path seriesLine = (Path)((Group)series.getNode()).getChildren().get(1);
            Path fillPath = (Path)((Group)series.getNode()).getChildren().get(0);
            smooth(seriesLine.getElements(), fillPath.getElements());
        }
    }

    private int getDataSize() {
        ObservableList<Series<Number, Number>> data = this.getData();
        return data != null ? data.size() : 0;
    }

    private static void smooth(ObservableList<PathElement> strokeElements,
ObservableList<PathElement> fillElements) {
        Point2D[] dataPoints = new Point2D[strokeElements.size()];

        for(int i = 0; i < strokeElements.size(); ++i) {
            PathElement element = (PathElement)strokeElements.get(i);
            if (element instanceof MoveTo) {
                MoveTo move = (MoveTo)element;
                dataPoints[i] = new Point2D(move.getX(), move.getY());
            } else if (element instanceof LineTo) {
                LineTo line = (LineTo)element;
                double x = line.getX();
                double y = line.getY();
                dataPoints[i] = new Point2D(x, y);
            }
        }
    }
}

```

```

double zeroY = ((MoveTo)fillElements.get(0)).getY();
strokeElements.clear();
fillElements.clear();
Pair<Point2D[], Point2D[]> result = calcCurveControlPoints(dataPoints);
Point2D[] firstControlPoints = (Point2D[])result.getKey();
Point2D[] secondControlPoints = (Point2D[])result.getValue();
strokeElements.add(new MoveTo(dataPoints[0].getX(), dataPoints[0].getY()));
fillElements.add(new MoveTo(dataPoints[0].getX(), zeroY));
fillElements.add(new LineTo(dataPoints[0].getX(), dataPoints[0].getY()));

for(int i = 1; i < dataPoints.length; ++i) {
    int ci = i - 1;
    strokeElements.add(new CubicCurveTo(firstControlPoints[ci].getX(),
firstControlPoints[ci].getY(), secondControlPoints[ci].getX(), secondControlPoints[ci].getY(),
dataPoints[i].getX(), dataPoints[i].getY()));
    fillElements.add(new CubicCurveTo(firstControlPoints[ci].getX(), firstControlPoints[ci].getY(),
secondControlPoints[ci].getX(), secondControlPoints[ci].getY(), dataPoints[i].getX(), dataPoints[i].getY()));
}

fillElements.add(new LineTo(dataPoints[dataPoints.length - 1].getX(), zeroY));
fillElements.add(new ClosePath());
}

public static Pair<Point2D[], Point2D[]> calcCurveControlPoints(Point2D[] dataPoints) {
    int n = dataPoints.length - 1;
    Point2D[] firstControlPoints;
    Point2D[] secondControlPoints;
    if (n == 1) {
        firstControlPoints = new Point2D[]{new Point2D((2.0 * dataPoints[0].getX() +
dataPoints[1].getX()) / 3.0, (2.0 * dataPoints[0].getY() + dataPoints[1].getY()) / 3.0)};
        secondControlPoints = new Point2D[]{new Point2D(2.0 * firstControlPoints[0].getX() -
dataPoints[0].getX(), 2.0 * firstControlPoints[0].getY() - dataPoints[0].getY())};
        return new Pair(firstControlPoints, secondControlPoints);
    } else {
        double[] rhs = new double[n];

        for(int i = 1; i < n - 1; ++i) {
            rhs[i] = 4.0 * dataPoints[i].getX() + 2.0 * dataPoints[i + 1].getX();
        }

        rhs[0] = dataPoints[0].getX() + 2.0 * dataPoints[1].getX();
        rhs[n - 1] = (8.0 * dataPoints[n - 1].getX() + dataPoints[n].getX()) / 2.0;
        double[] x = GetFirstControlPoints(rhs);

        for(int i = 1; i < n - 1; ++i) {
            rhs[i] = 4.0 * dataPoints[i].getY() + 2.0 * dataPoints[i + 1].getY();
        }

        rhs[0] = dataPoints[0].getY() + 2.0 * dataPoints[1].getY();
        rhs[n - 1] = (8.0 * dataPoints[n - 1].getY() + dataPoints[n].getY()) / 2.0;
        double[] y = GetFirstControlPoints(rhs);
    }
}

```

```

firstControlPoints = new Point2D[n];
secondControlPoints = new Point2D[n];

for(int i = 0; i < n; ++i) {
    firstControlPoints[i] = new Point2D(x[i], y[i]);
    if (i < n - 1) {
        secondControlPoints[i] = new Point2D(2.0 * dataPoints[i + 1].getX() - x[i + 1], 2.0 *
dataPoints[i + 1].getY() - y[i + 1]);
    } else {
        secondControlPoints[i] = new Point2D((dataPoints[n].getX() + x[n - 1]) / 2.0,
(dataPoints[n].getY() + y[n - 1]) / 2.0);
    }
}

return new Pair(firstControlPoints, secondControlPoints);
}
}

```

```

private static double[] GetFirstControlPoints(double[] rhs) {
    int n = rhs.length;
    double[] x = new double[n];
    double[] tmp = new double[n];
    double b = 2.0;
    x[0] = rhs[0] / b;

    int i;
    for(i = 1; i < n; ++i) {
        tmp[i] = 1.0 / b;
        b = (i < n - 1 ? 4.0 : 3.5) - tmp[i];
        x[i] = (rhs[i] - x[i - 1]) / b;
    }

    for(i = 1; i < n; ++i) {
        x[n - i - 1] -= tmp[n - i] * x[n - i];
    }

    return x;
}
}

```

```

public class DonutChart extends PieChart {

    private final Circle innerCircle;

    public DonutChart() {
        this(FXCollections.<Data>observableArrayList());
    }

    public DonutChart(ObservableList<Data> pieData) {
        super(pieData);
    }
}

```



```

this.getStyleClass().add("donut-chart");
innerCircle = new Circle();
innerCircle.getStyleClass().add("inner-circle");
Circle clip = new Circle();

// just styled in code for demo purposes,
// use a style class instead to style via css.
innerCircle.setFill(Color.WHITE);
innerCircle.setStroke(Color.TRANSPARENT);

innerCircle.setStrokeWidth(3);

// this.setClip(clip);
}

@Override
protected void layoutChartChildren(double top, double left, double contentWidth, double contentHeight) {
    super.layoutChartChildren(top, left, contentWidth, contentHeight);

    addInnerCircleIfNotPresent();
    updateInnerCircleLayout();
}

private void addInnerCircleIfNotPresent() {
    if (getData().size() > 0) {
        Node pie = getData().get(0).getNode();
        if (pie.getParent() instanceof Pane) {
            Pane parent = (Pane) pie.getParent();

            if (!parent.getChildren().contains(innerCircle)) {
                parent.getChildren().add(innerCircle);
            }
        }
    }
}

private void updateInnerCircleLayout() {
    double minX = Double.MAX_VALUE, minY = Double.MAX_VALUE;
    double maxX = Double.MIN_VALUE, maxY = Double.MIN_VALUE;
    for (Data data: getData()) {
        Node node = data.getNode();

        Bounds bounds = node.getBoundsInParent();
        if (bounds.getMinX() < minX) {
            minX = bounds.getMinX();
        }
        if (bounds.getMinY() < minY) {
            minY = bounds.getMinY();
        }
        if (bounds.getMaxX() > maxX) {

```

```

        maxX = bounds.getMaxX();
    }
    if (bounds.getMaxY() > maxY) {
        maxY = bounds.getMaxY();
    }
}

innerCircle.setCenterX(minX + (maxX - minX) / 2);
innerCircle.setCenterY(minY + (maxY - minY) / 2);

innerCircle.setRadius((maxX - minX) / 3);
}
}
public class GNAvatar extends Control {
    private final ObjectProperty<Image> image;
    private final DoubleProperty radius;

    public GNAvatar() {
        this(20.0);
    }

    public GNAvatar(double _radius) {
        this.image = new SimpleObjectProperty();
        this.radius = new SimpleDoubleProperty();
        this.radius.set(_radius);
        this.setPrefSize(_radius, _radius);
    }

    public GNAvatar(Image image, double _radius) {
        this.image = new SimpleObjectProperty<>();
        this.radius = new SimpleDoubleProperty();
        this.radius.set(_radius);
        this.setPrefSize(_radius, _radius);
        this.setImage(image);
    }

    protected Skin<?> createDefaultSkin() {
        return new GNAvatarSkin(this);
    }

    public Image getImage() {
        return (Image)this.image.get();
    }

    public ObjectProperty<Image> imageProperty() {
        return this.image;
    }

    public void setImage(Image image) {
        this.image.set(image);
    }
}

```

```

public double getRadius() {
    return this.radius.get();
}

public DoubleProperty radiusProperty() {
    return this.radius;
}

public void setRadius(double radius) {
    this.radius.set(radius);
}
}
@SuppressWarnings("unused")
public class ToggleSwitch extends Parent {

    private final BooleanProperty switchedOn = new SimpleBooleanProperty(false);

    private final TranslateTransition translateAnimation = new TranslateTransition(Duration.seconds(0.15));
    private final FillTransition fillAnimation = new FillTransition(Duration.seconds(0.15));
    private final ParallelTransition animation = new ParallelTransition(translateAnimation, fillAnimation);

    private BooleanProperty switchedOnProperty() {
        return switchedOn;
    }

    public ToggleSwitch() {
        getStyleClass().add("toggle-switch");
        Rectangle background = new Rectangle(40, 26);
        background.getStyleClass().add("background");
        background.setArcWidth(25);
        background.setArcHeight(25);
        background.setFill(Color.WHITE);
        background.setStroke(Color.LIGHTGRAY);
        Circle trigger = new Circle(10);
        trigger.setCenterX(12);
        trigger.setCenterY(13);
        trigger.setFill(Color.WHITE);
        trigger.setStroke(Color.LIGHTGRAY);
        getChildren().addAll(background, trigger);

        translateAnimation.setNode(trigger);
        fillAnimation.setShape(background);

        switchedOn.addListener((obs, oldState, newState) -> {
            boolean isOn = newState;
            translateAnimation.setToX(
                isOn ? 100 - 84 : 0
            );
            fillAnimation.setFromValue(isOn ? Color.WHITE : Color.web("#0096C9"));
            fillAnimation.setToValue(isOn ? Color.web("#0096C9") : Color.WHITE);
        });
    }
}

```

```

        animation.play();
    });

    setOnMouseClicked(event -> switchedOn.set(!switchedOn.get()));
}
}
import javafx.application.Platform;
import javafx.scene.Node;
import javafx.scene.control.ScrollPane;
import javafx.scene.control.TitledPane;
import javafx.scene.control.ToggleButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;

import java.util.ArrayList;
import java.util.List;

/**
 * @author Gleidson Neves da Silveira | gleidisonmt@gmail.com
 * Version 0.0.1
 * Create on 24/04/2023
 */
public class DrawerBehavior {

    private List<SearchItem> searchItems = new ArrayList<>();

    public DrawerBehavior(StackPane content, ToggleGroup group) {

        VBox body = (VBox) content.lookup("#drawer-content");

        ScrollPane scrollPane = (ScrollPane) body.lookup("#drawer-scroll");
        VBox box = (VBox) scrollPane.getContent();

        box.getChildren()
            .stream()
            .filter(fil ->
                fil instanceof ToggleButton && fil.getStyleClass().contains("drawer-item"))
            .map(item -> (ToggleButton) item)
            .forEach(each -> {
                each.addEventFilter(MouseEvent.MOUSE_PRESSED, event -> {
                    if (each.isSelected()) event.consume();
                });
            });
        // context.searchItems().add(SearchItemAdapter.adapter(each));
        searchItems.add(SearchItemAdapter.adapter(each));
    });

    box.getChildren()
        .stream()

```

```

        .filter(fill -> fill instanceof TitledPane
            && ((TitledPane) fill).getContent() instanceof VBox
        )
        .map(item -> (VBox) ((TitledPane) item).getContent())
        .forEach(each -> {
            for (Node fill : each.getChildren()) {
                if (fill instanceof ToggleButton e) {
                    e.setToggleGroup(group);
                    e.addEventFilter(MouseEvent.MOUSE_PRESSED, event -> {
                        if (e.isSelected()) {
                            event.consume();
                        }
                    });
                    e.selectedProperty().addListener((observable, oldValue, newValue) -> {
                        if (!newValue) {
                            e.getParent().getParent().getParent().setId(null);
                        } else {
                            e.getParent().getParent().getParent().setId("drawer-menu-selected");
                        }
                    });
                }
            }
        });

//                context.searchItems().add(SearchItemAdapter.adapter(e));
//                searchItems.add(SearchItemAdapter.adapter(e));
    }
}
});

}
}
}

public class LoadViews extends Task<View> {

    private ViewMap data;
    private final StringBuilder builder = new StringBuilder();
    private final Context context;
    private final Loader loader;

    public LoadViews(Context context, Loader loader) {
        this.context = context;
        this.loader = loader;

        InputStream inputStream =
        getClass().getResourceAsStream("/io.github.gleudsonmt.dashboardfx/views.yml");
        Yaml yaml = new Yaml(new Constructor(ViewMap.class));
        data = yaml.load(inputStream);
    }

    @Override
    protected View call() throws Exception {

        for (ViewComposer viewComposer : data.getViews()) {

```

```

Thread.sleep(100);

Platform.runLater() -> {
    loader.updateLegend("Loading view... " + viewComposer.getName());
    loadView(viewComposer);
});

Thread.sleep(1000);
}
return null;
}

private void loadView(ViewComposer view) {

    FXMLLoader loader = new FXMLLoader();
    URL location = null;

    String path = "/io.github.gleidsonmt.dashboardfx/views";

    if (view.getFolder() != null) {
        builder.append("/").append(view.getFolder());
    }

    // if (view.getViews() != null) {
    //     for (ViewComposer v : view.getViews()) {
    //         if (v.getFxml() != null) {
    //             //
    //             location = getClass().getResource(path + builder + "/"
    //                 + v.getFxml());
    //         }
    //         v.setRoot(view);
    //         loadView(v);
    //     }
    // }

    if (view.getFolder() == null) location = LoadViews.class.getResource(path + builder + "/"
        + view.getFxml());
    else if (view.getFxml() != null && view.getFolder() != null)
        location = getClass().getResource(path + builder + "/"
            + view.getFxml());

    if (view.getFolder() != null) {
        String act = builder.substring(builder.lastIndexOf("/") + 1, builder.length());
        if (act.equals(view.getFolder())) builder.delete(builder.lastIndexOf("/"), builder.length());
    }

    if (location != null && view.getFxml() != null) {

        loader.setLocation(location);
        loader.setCharset(StandardCharsets.UTF_8);
        // loader.setResources(App2.INSTANCE.getResourceBundle());
    }
}

```

```

    try {
        loader.load();
    } catch (IOException e) {
        e.printStackTrace();
    }

//    context.getRoutes().addView(new View(view, loader));
//    context.routes().put(new FXMLView(view, loader));

} else if (view.getFxml() != null) {
    IOException io = new IOException("The fxml with ["
        + view.getName() + "]" + " doesn't correspond.");
    io.printStackTrace();
}
}
}
public class SearchItem {
    private StringProperty name = new SimpleStringProperty();
    private ObjectProperty<EventHandler> action = new SimpleObjectProperty<>();
    public String getName() {
        return name.get();
    }

    public StringProperty nameProperty() {
        return name;
    }

    public void setName(String name) {
        this.name.set(name);
    }

    public void setAction(EventHandler event) {
        action.set(event);
    }

    public EventHandler getAction() {
        return action.get();
    }

    @Override
    public String toString() {
        return name.get();
    }
}
public class CounterLabel extends Label {

    private final IntegerProperty act = new SimpleIntegerProperty(0);
    private final IntegerProperty max = new SimpleIntegerProperty();

    public CounterLabel() {

```

```

    act.addListener((observable, oldValue, newValue) -> update());
    max.addListener((observable, oldValue, newValue) -> update());
}

private void update() {
    if (max.intValue() == -1) {
        setText(act.get() + "../");
    } else setText(act.get() + "/" + max.get());
}

public int getCount() {
    return act.get();
}

public IntegerProperty actProperty() {
    return act;
}

public void setCount(int act) {
    this.act.set(act);
}

public int getMax() {
    return max.get();
}

public IntegerProperty maxProperty() {
    return max;
}

public void setMax(int max) {
    this.max.set(max);
}
}
@DefaultProperty("children")
public class Editor extends TextField {

    private final BooleanProperty maskText    = new SimpleBooleanProperty(false); // used for password
text
    private final IntegerProperty maxText    = new SimpleIntegerProperty(Integer.MAX_VALUE); //
used for max text input

    public Editor() {

        getStyleClass().add("editor");
        setAlignment(Pos.CENTER_LEFT);
        setFocusTraversable(false);

        addEventFilter(KeyEvent.KEY_TYPED, event -> {
            if (getLength() >= maxText.get()) {
                positionCaret(getLength());
            }
        });
    }
}

```



```

        event.consume();
    }
});

}

@Override
public void paste() {
    final Clipboard clipboard = Clipboard.getSystemClipboard();

    if (clipboard.hasString()) {
        final String text = clipboard.getString();
        if (text != null) {
            int major = text.length();
            int comparator = maxText.get() - this.getLength();
            String sub ;
            if ( (major + this.getLength()) < maxText.get() ) {
                //
                this.replaceSelection(text);
            } else {
                sub = text.substring(0,
                    comparator);
                this.replaceSelection(sub);
            }
        }
    }
}

@Override
protected Skin<?> createDefaultSkin() {
    return new EditorSkin(this);
}

// @Override
// public String getUserAgentStylesheet() {
//     return
Objects.requireNonNull(FloatEditor.class.getResource("/io.github.gleidsonmt.gncontrols/agents/editor.css"))
.toExternalForm();
// }

public boolean isMaskText() {
    return maskText.get();
}

public BooleanProperty maskTextProperty() {
    return maskText;
}

public void setMaskText(boolean maskText) {
    this.maskText.set(maskText);
}

```

```

public int getMaxText() {
    return maxText.get();
}

public IntegerProperty maxTextProperty() {
    return maxText;
}

public void setMaxText(int maxText) {
    this.maxText.set(maxText);
}
}
public class EditorSkin extends TextFieldSkin {

    private final Editor control;

    public EditorSkin(Editor _control) {
        super(_control);
        this.control = _control;

        Parent inFocus; // Change the focus when the parent is gntextbox
        if (control.getParent() instanceof GNTextBoxBase) {
            inFocus = control.getParent();
        } else inFocus = control;

        registerChangeListener(inFocus.focusedProperty(), c -> {
            if ((boolean) c.getValue()) {
                control.requestFocus();
            }
        });
    }

    @Override
    protected String maskText(String txt) {

        if (getSkinnable() instanceof Editor editor) {
            if (editor.isMaskText()) {

                final char BULLET = '\u25cf';
                int n = txt.length();

                return String.valueOf(BULLET).repeat(n);
            } else {
                return editor.textProperty().getValueSafe();
            }
        } else return txt;
    }
}

```

```

    }
}

public class FloatEditorSkin extends EditorSkin {

    private static final PseudoClass FLOAT_PSEUDO_CLASS =
        PseudoClass.getPseudoClass("float");

    private final Label moveablePrompt = new Label("Float Prompt");

    private final FloatEditor control;
    private VPos          vPos;

    private final double aniVeloz = 100;

    private final Timeline upAnimation = new Timeline();
    private final Timeline downAnimation = new Timeline();

    public FloatEditorSkin(FloatEditor _control) {
        super(_control);
        this.control = _control;

        // Setting default Conventions
        moveablePrompt.getStyleClass().add("label-prompt");

        moveablePrompt.textProperty().bind(control.promptTextProperty());
        moveablePrompt.setFont(new Font(12));

        moveablePrompt.getStyleClass().add("prompt-text");
        moveablePrompt.setFocusTraversable(false);
        moveablePrompt.setMouseTransparent(true);
        moveablePrompt.toFront();

        // add prompt and hide old prompt
        getChildren().add(moveablePrompt);
        setPromptTextFill(Color.TRANSPARENT);

        if (_control.getText() != null && !_control.getText().isEmpty())
            vPos = getPos();
        else
            vPos = VPos.CENTER;

        Parent inFocus; // Change the focus when the parent is gntextbox
        if (control.getParent() instanceof GNTTextBoxBase) {
            inFocus = control.getParent();
        } else
            inFocus = control;
    }
}

```

```

// When focused start animation
registerChangeListener(inFocus.focusedProperty(), c -> {

    if ((boolean) c.getValue()) {

        control.requestFocus();

        if (downAnimation.getStatus() == Animation.Status.RUNNING) {
            downAnimation.stop();
            up();
        } else
            if (isPromptInCenter()) {
                up();
            }
    } else {

        if (upAnimation.getStatus() == Animation.Status.RUNNING) {
            upAnimation.stop();

            down();
        } else {

            if (isPromptInTop()) {
                down();
            }
        }
    }
});

registerChangeListener(control.floatAlignmentProperty(), c -> {
    vPos = (VPos) c.getValue();

    if (vPos == VPos.TOP) // It's for calcs.. use the padding for relocate the prompt in line of border
        control.setPadding(new Insets(0, 6,0,6));
    else {
        control.setPadding(new Insets(4, 7, 4, 5));
    }

});

if (control.getText() != null && !control.getText().isEmpty()) {
    moveablePrompt.setTranslateX(control.getDistanceX());
}

_control.textProperty().addListener((observable, oldValue, newValue) -> {
    if (newValue != null && !newValue.isEmpty()) {

```

```

        if (!isPromptInTop()) {
            up();
        }
    });
}

// Getting the middle of prompt
public double getPromptHalf() {
//    return (moveablePrompt.getHeight() / 2);
    return moveablePrompt.getHeight();
}

// Getting the top line from the center of this skin
public double getMaxTopPosition() {
    return - (control.getHeight() / 2);
}

// Convetion for define in center
private boolean isPromptInCenter() {
    return vPos == VPos.CENTER;
}

// Convention for define is in top
private boolean isPromptInTop() {
    return vPos == VPos.BASELINE || vPos == VPos.TOP;
}

private void configAnimation() {

}

@Override
protected void layoutChildren(double x, double y, double w, double h) {
    super.layoutChildren(x, y, w, h);

    if (downAnimation.getStatus() == Animation.Status.RUNNING ||
        upAnimation.getStatus() == Animation.Status.RUNNING) {

        positionInArea(moveablePrompt, x,
            y, w, h, -1, HPos.LEFT, VPos.CENTER);

        return;
    }

    moveablePrompt.setTranslateY(0);

    if (isPromptInTop()) {
        positionInArea(moveablePrompt, x, y, w, h, -1, HPos.LEFT, getPos());
    }
}

```

```

//      positionInArea(filledBorder, x, y, w, h , -1, HPos.LEFT, getPos());

    } else {
        positionInArea(moveablePrompt, x, y, w, h , -1, HPos.LEFT, VPos.CENTER);
    }

}

private VPos getPos() {
    if (control.getFloatAlignment() == FloatAlignment.TOP) return VPos.TOP;
    else return VPos.BASELINE;
}

// Animation for top
private void up() {

//      if (!control.getText().isEmpty()) return;

    double ty;

    if (getPos() == VPos.TOP) {
        ty = getMaxTopPosition() + snappedTopInset() ;
    } else {
        ty = getMaxTopPosition() ;
    }

}

upAnimation.getKeyFrames().setAll(
    new KeyFrame(Duration.ZERO, new KeyValue(
        moveablePrompt.translateYProperty(), moveablePrompt.getTranslateY()
    )),
    new KeyFrame(Duration.millis(aniVeloz), new KeyValue(
        moveablePrompt.translateYProperty(), ty
    )),
    new KeyFrame(Duration.ZERO, new KeyValue(
        moveablePrompt.translateXProperty(), control.getTranslateX()
    )),
    new KeyFrame(Duration.millis(aniVeloz), new KeyValue(
        moveablePrompt.translateXProperty(), control.getDistanceX()
    )),
    new KeyFrame(Duration.ZERO, new KeyValue(
        moveablePrompt.fontProperty(), moveablePrompt.getFont()
    )),
    new KeyFrame(Duration.millis(aniVeloz), new KeyValue(

```

```

        moveablePrompt.fontProperty(), new Font(11)
    ))
);

control.pseudoClassStateChanged(FLOAT_PSEUDO_CLASS, true);
upAnimation.setOnFinished(event -> vPos = getPos());

upAnimation.play();

// control.setClip(moveablePrompt);
}

// Animation for bottom
private void down() {

    if (control.getText() != null && !control.getText().isEmpty()) return;
//    if (!control.getText().isEmpty()) return;
//    if (animation.getStatus() == Animation.Status.RUNNING) return;

    double ty;
//
    if (getPos() == VPos.TOP) {
        ty = getMaxTopPosition() + snappedTopInset();

    } else {
        ty = getMaxTopPosition();
    }

    double startY = moveablePrompt.getTranslateY() >= 0 ? ty : moveablePrompt.getTranslateY();
//    startY = 0;

    downAnimation.getKeyFrames().setAll(
        new KeyFrame(Duration.ZERO, new KeyValue(
            moveablePrompt.translateYProperty(), startY
        )),
        new KeyFrame(Duration.millis(aniVeloz), new KeyValue(
            moveablePrompt.translateYProperty(), 0
        )),

        new KeyFrame(Duration.ZERO, new KeyValue(
            moveablePrompt.translateXProperty(), moveablePrompt.getTranslateX()
        )),

        new KeyFrame(Duration.millis(aniVeloz), new KeyValue(
            moveablePrompt.translateXProperty(), 0
        )),

        new KeyFrame(Duration.ZERO, new KeyValue(
            moveablePrompt.fontProperty(), moveablePrompt.getFont()
        )),
    );
}

```

```

        new KeyFrame(Duration.millis(aniVeloz), new KeyValue(
            moveablePrompt.fontProperty(), new Font(12)
        ))
    );

    control.pseudoClassStateChanged(FLOAT_PSEUDO_CLASS, false);

    downAnimation.setOnFinished(event -> {
        vPos = VPos.CENTER;

//        moveablePrompt.setTranslateY(0);
    });

    downAnimation.play();

}

@Override
protected double computeMinHeight(double width, double topInset, double rightInset, double
bottomInset, double leftInset) {
    return super.computeMinHeight(width, topInset, rightInset, bottomInset, leftInset) ;
}

@Override
protected double computeMinWidth(double height, double topInset, double rightInset, double
bottomInset, double leftInset) {
    return super.computeMinWidth(height, topInset, rightInset, bottomInset, leftInset) +
        (moveablePrompt.prefWidth(height) + (snappedRightInset() + snappedLeftInset()));
}

@Override
protected double computePrefHeight(double width, double topInset, double rightInset, double
bottomInset, double leftInset) {
    return super.computePrefHeight(width, topInset, rightInset, bottomInset, leftInset);
}

@Override
protected double computePrefWidth(double height, double topInset, double rightInset, double
bottomInset, double leftInset) {
    return super.computePrefWidth(height, topInset, rightInset, bottomInset, leftInset);
}
}

public class GNIconButtonSkin extends SkinBase<GNIconButton> {

    private final GNIconButton control;

    private final Rectangle clip = new Rectangle();
    private final IconContainer iconContainer = new IconContainer();

```



```

private final Timeline timeline = new Timeline();
private final Circle circle = new Circle();

public GNIconButtonSkin(GNIconButton _control) {
    super(_control);
    this.control = _control;

    if (_control.getIcon() == null || _control.getIcon() == Icons.NONE) {
        iconContainer.setContent(Icons.FAVORITE);
    } else
        iconContainer.setContent(_control.getIcon());

    _control.iconProperty().addListener((observable, oldValue, newValue) -> {
        if (newValue != null && newValue != Icons.NONE) {
            iconContainer.setContent(newValue.getContent());
        }
    });

    iconContainer.setFill(Color.WHITE);
    getChildren().add(iconContainer);

    this.clip.widthProperty().bind(this.control.widthProperty());
    this.clip.heightProperty().bind(this.control.heightProperty());

    this.control.widthProperty().addListener((observable, oldValue, newValue) -> {
        clip.setArcWidth(_control.getWidth() * 2);
        clip.setArcHeight(_control.getHeight() * 2);
    });

    this.control.heightProperty().addListener((observable, oldValue, newValue) -> {
        clip.setArcWidth(_control.getWidth() * 2);
        clip.setArcHeight(_control.getHeight() * 2);
    });

    _control.addEventHandler(MouseEvent.MOUSE_RELEASED, onPressed);

    _control.setClip(clip);
}

@Override
protected void layoutChildren(double contentX, double contentY, double contentWidth, double
contentHeight) {
    layoutInArea(iconContainer, contentX, contentY, contentWidth, contentHeight, -1, HPos.CENTER,
VPos.CENTER);
}

private final EventHandler<MouseEvent> onPressed = new EventHandler<>() {
    @Override
    public void handle(MouseEvent event) {

```

```

if (timeline.getStatus() == Animation.Status.RUNNING) {
    return;
}

circle.setRadius(0);
circle.setStrokeWidth(0);

circle.setFill(Color.WHITE);

circle.setLayoutX(event.getX());
circle.setLayoutY(event.getY());

circle.setOpacity(0.5);

circle.setMouseTransparent(true);
circle.setManaged(false);

// clip.setWidth( Math.floor(control.getWidth() / 6 ));
// clip.setHeight( Math.floor(control.getHeight() / 6 ));

getChildren().add(circle);

double diameter = Math.max(control.getWidth(), control.getHeight());
double radius = diameter / 2;

timeline.getKeyFrames().setAll(
    new KeyFrame(Duration.ZERO, new KeyValue(circle.radiusProperty(), 0)),
    new KeyFrame(Duration.millis(250), new KeyValue(circle.radiusProperty(), radius * 2 ))
);

timeline.play();

timeline.setOnFinished( (e) -> getChildren().remove(circle));
}
};
}

```

## Додаток Г (довідниковий). Ілюстративна частина

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра програмного забезпечення

### МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка методів та програмних засобів системи управління стартап-проєктами

Виконала:  
студентка групи ЗПІ -22м Маланчук А.В.

Науковий керівник:  
к.т.н., доцент кафедри ПЗ Войтко В.В.

Рисунок Г.1 – Назва роботи

### Мета, предмет та об'єкт дослідження

- **Метою магістерської кваліфікаційної роботи** є полегшення та оптимізація процесів управління стартапами й малими компаніями шляхом розробки та впровадження спеціалізованої системи управління проєктами, що дозволить ефективний розподіл ресурсів для розв'язування поставлених задач.
- **Об'єктом дослідження** є процес створення системи управління стартап-проєктами.
- **Предметом дослідження** є методи і засоби розробки системи управління стартап-проєктами.

Рисунок Г.2 – Мета, об'єкт і предмет дослідження

## Постановка задач:

- Визначити функціонал системи управління стартап-проектами та методи і засоби її реалізації .
- Розробити метод, моделі та алгоритми роботи системи.
- Розробити програмне забезпечення системи управління стартап-проектами.
- Провести тестування роботи програмного продукту .

Рисунок Г.3 – Постановка задач

## Наукова новизна розробки

- Подальшого розвитку дістав метод управління стартап-проектами, який, на відміну від існуючих, враховує індивідуальні потреби та особливості малих компаній, що дозволить системі адаптуватися до сучасного динамічного бізнес-середовища
- Подальшого розвитку дістала модель системи управління стартапами, яка, на відміну від існуючих, враховує особливості організації роботи зі стартапами, сучасні стратегії моніторингу й оптимізації робочих процесів, формування графіків і звітів, що дозволяє покращити процес керування проектами



Рисунок Г.4 – Наукова новизна

## Практична цінність отриманих результатів

Практичне значення магістерської кваліфікаційної роботи полягає у можливості практичного використання розробленої системи управління стартап-проєктами у приватному користуванні чи в спеціалізованих навчальних цілях.

Рисунок Г.5 – Практична цінність

### Аналіз аналогів



Характеристика	"Система управління стартап проєктами"	Trello	Asana	Monday.com
Спеціалізація	Стартапи і малі компанії	Загальний ринок	Загальний ринок	Загальний ринок
Простота використання	Так	Так	Так	Так
Функціональність	Управління проєктами, облік ресурсів, аналітика, облік бюджету	Базовий облік завдань	Розширений облік завдань, управління ресурсами, аналітика	Гнучкий облік завдань, управління ресурсами
Ціна	Невідомо (залежно від планів)	Безкоштовний тариф	Безкоштовний тариф	Вищий ціновий поріг
Гнучкість і адаптованість	Потенційно висока	Відсутність	Потенційно висока	Потенційно висока



Рисунок Г.6 – Порівняння аналогів

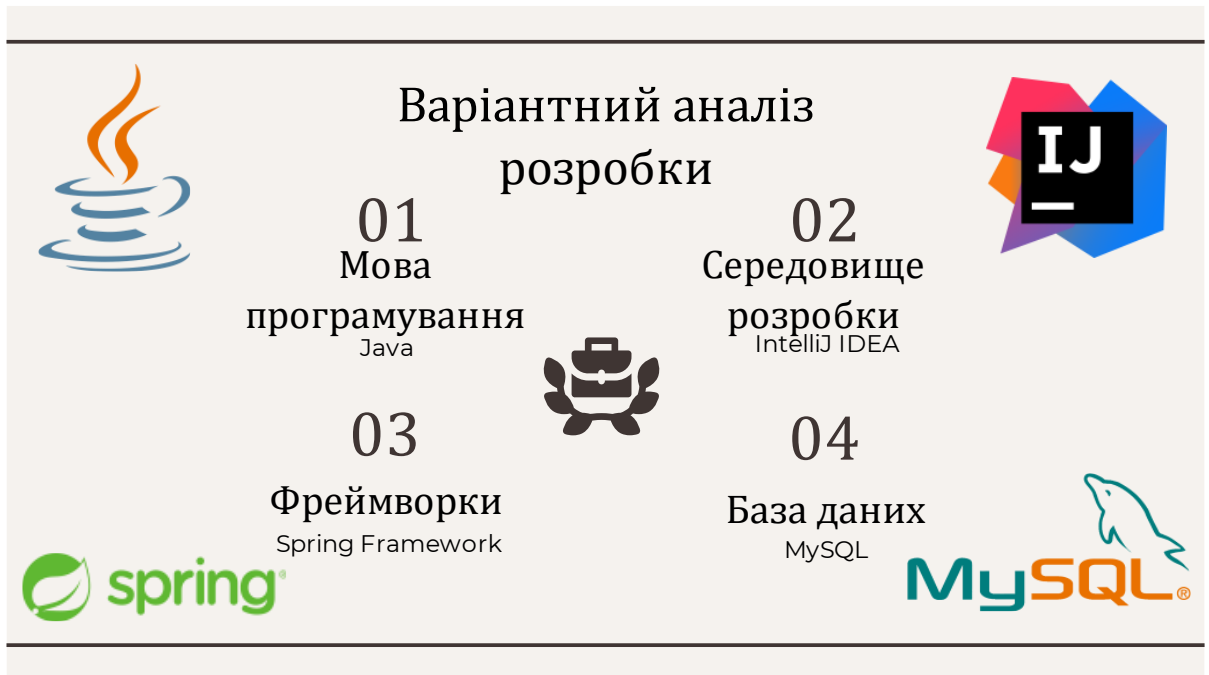


Рисунок Г.7 – Варіантний аналіз розробки



Рисунок Г.8 – Загальна модель роботи системи

### Блок- схема алгоритму роботи моделі системи управління стартап-проектами

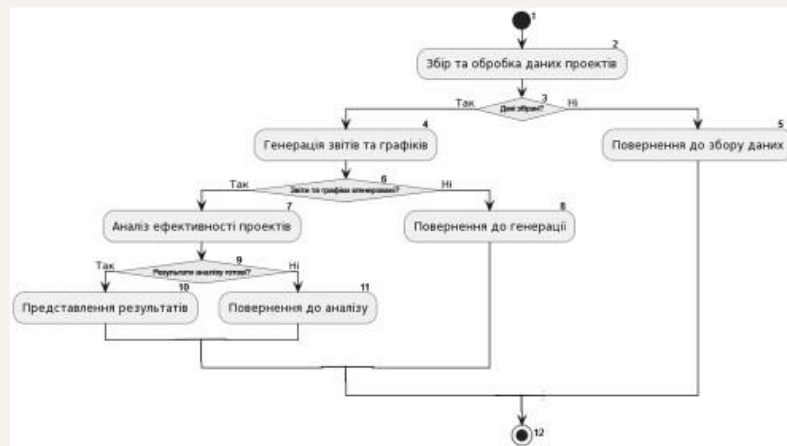


Рисунок Г.9 – Блок-схема алгоритму роботи моделі

### Метод управління стартап-проектами включає такі дії:

1. Перший етап включає створення нового проекту та заповнення його даними через виклик методів `createNewProject()` та `setTextFieldsData()`. Кнопка "Створити новий проект" ініціює цей процес.
2. Далі йде обирання типу проекту зі списку за допомогою методу `selectProjectType()`, а також встановлення часу роботи за допомогою `MaterialTimePicker` через метод `showTimePicker()`.
3. Наступний етап включає додавання тегів до проекту з використанням відповідного `ScrollState` через метод `addTags()`. Також включає вибір фотографій галереї та їх завантаження на сервер за допомогою методу `pickAndUploadPhoto()`.
4. При отриманні URL завантажених фотографій та допомогою функції `getPhotoDownloadUrl()`. Після цього створення проекту з деталями та посиланнями на фотографії через метод `createProjectWithDetails()`. Також включає навігацію на інший екран після створення проекту та збереження його в базі даних.
5. Якщо відбулося створення проекту учасникам, за допомогою методу `notifyParticipantsAboutNewProject()` приймає відповідне повідомлення. Також включає генерацію посилання для спільного доступу до проекту через метод `generateProjectLink()`. Перевірка прав доступу до проекту здійснюється методом `checkAccessRight()`.
6. Проводиться перевірка прав доступу до проекту перед його відкриттям методом `checkAccessRight()`. Це гарантує, що лише користувачі з відповідними правами матимуть доступ до проекту.

Рисунок Г.10 – Метод управління стартап-проектами

### Блок-схема алгоритму роботи методу управління стартап - проєктами

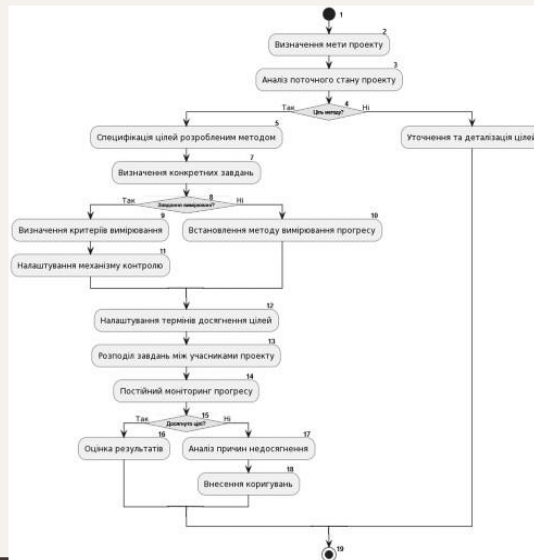


Рисунок Г.11 – Блок-схема алгоритму роботи методу

### Метод додавання нового проєкту в базу передбачає послідовність виконання таких дій:

1. Створення нового проєкту через виклик методу `createNewProject()` при натисканні кнопки "Створити новий проєкт".
2. Заповнення даних проєкту через використання методу `setTextFieldsData()` для збереження даних проєкту в `MutableStateFlow`.
3. Обрання типу проєкту з `DropDownMenu`за допомогою методу `selectProjectType()` для вибору типу проєкту зі списку.
4. Вибір часу роботи за допомогою `MaterialTimePicker`а `showTimePicker()` для встановлення часу початку та завершення роботи
5. Перевірка заповненості полів і активація кнопки "Продовжити", що робить метод `checkFieldsCompletion()` для перевірки оновленого стану і активації кнопки.
6. Додавання тегів з використанням відповідного `ScrollState` та `addTags()` для додавання тегів з можливістю прокрутки
7. Вибір фотографій з галереї та їх завантаження на сервер, використовуючи `pickAndUploadPhoto()` для обрання та завантаження фотографій

Рисунок Г.12 – Метод додавання нового проєкту



### Метод додавання нового проєкту в базу передбачає послідовність виконання таких дій ( продовження ) :

8. Отримання URL завантажених фотографій функцією `getPhotoDownloadUrls()` для отримання URL завантажених фотографій.
9. Створення проєкту з введеною інформацією і посиланнями на фотографії `createProjectWithDetails ()` для створення проєкту з даними та посиланнями на фотографії.
10. Навігація на екран "Профіль" після створення проєкту, виклик методу `navigateToProfileScreen ()` для переходу на відповідний екран після успішного створення проєкту.
11. Збереження створеного проєкту в базі даних для подальшого використання .
12. Відправка повідомлення про створення проєкту учасникам .`notifyParticipantsAboutNewProject ()` для сповіщення учасників про створення нового проєкту.
13. Генерація посилання методом `generateProjectLink ()` для спільного доступу до проєкту.
14. Проведення перевірки прав доступу до проєкту. Використовуємо метод `checkAccessRights ()` для перевірки прав доступу до проєкту перед відкриттям.

Рисунок Г.13 – Метод додавання нового проєкту (продовження)

### Блок-схема алгоритму методу додавання нового проєкту в базу даних

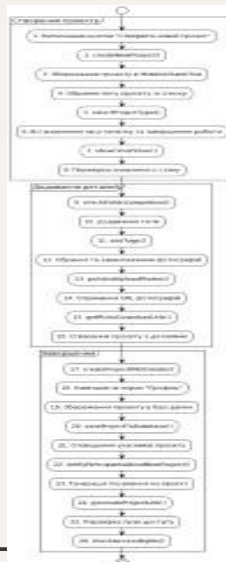


Рисунок Г.14 – Блок-схема алгоритму додавання нового проєкту в базу даних

### Загальний алгоритм авторизації в програмі

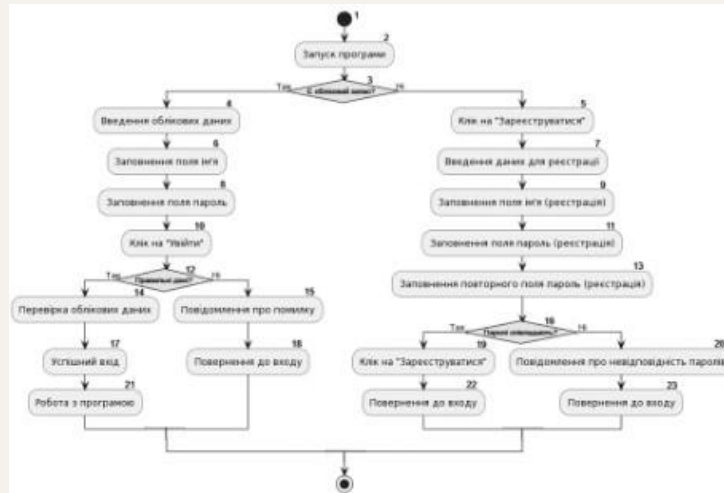


Рисунок Г.15 – Загальний алгоритм авторизації в програмі

### Діаграма взаємодії модуля управління проєктами



Рисунок Г.16 – Діаграма взаємодії модуля управління проєктами

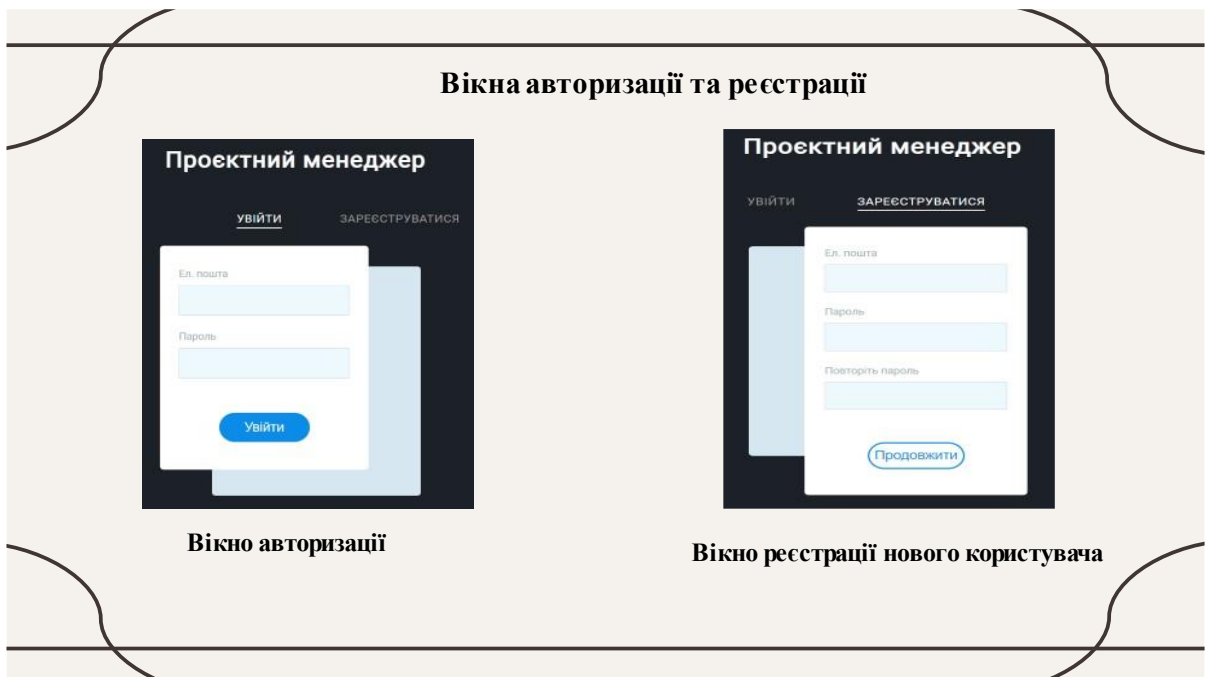


Рисунок Г.17 – Вікна реєстрації та авторизації

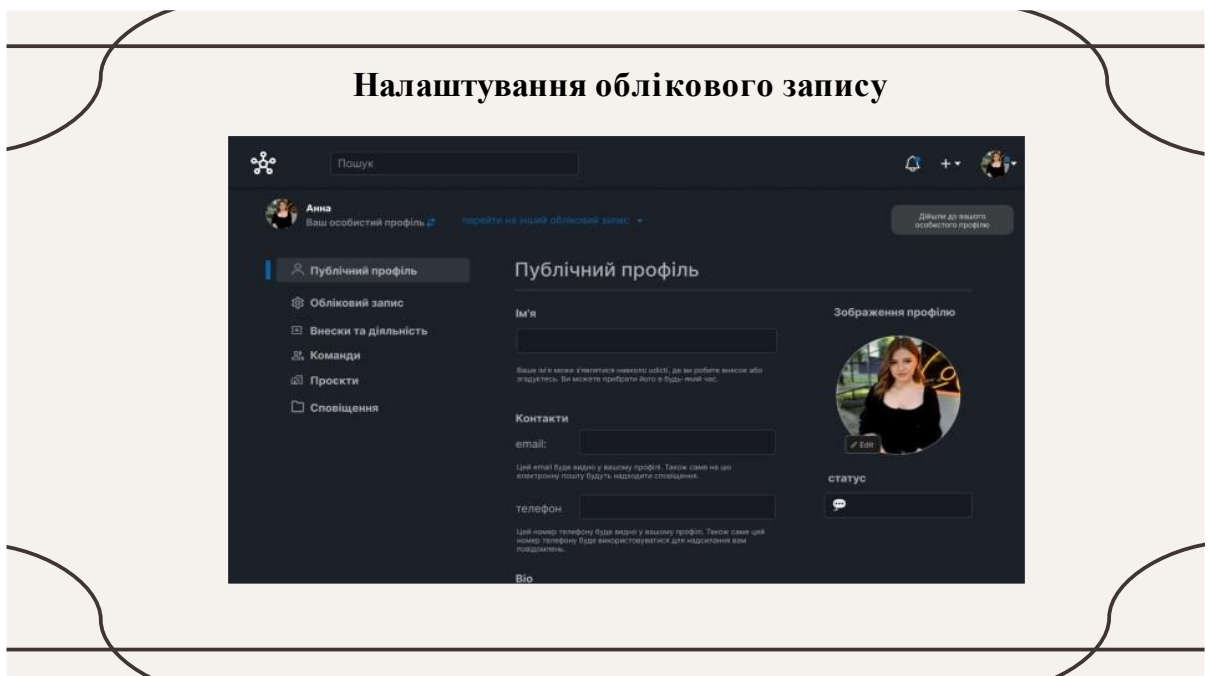


Рисунок Г.18 – Налаштування облікового запису

## Головний екран програми

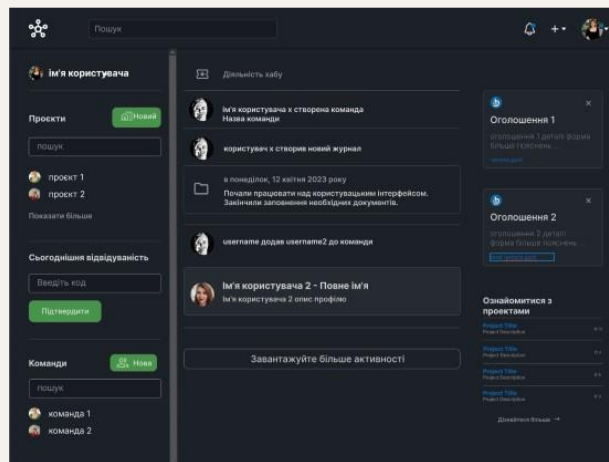


Рисунок Г.19 – Головний екран програми

## Вікно з інформацією про користувача

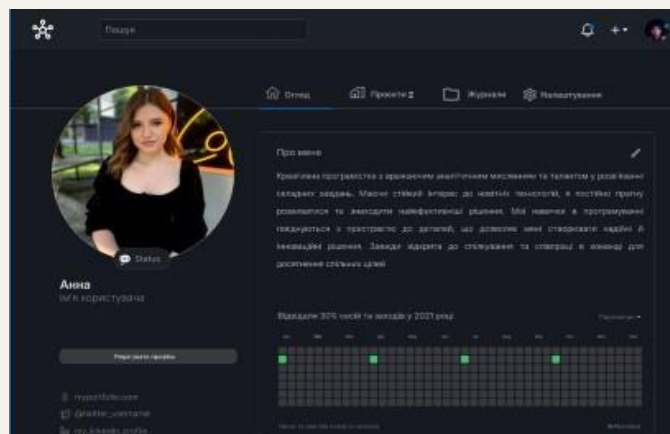


Рисунок Г.20 – Вікно з інформацією про користувача

## Вікно створення нового проєкта

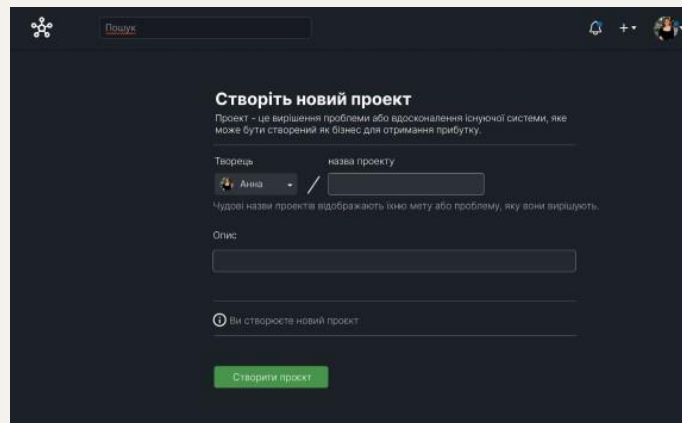


Рисунок Г.21– Вікно створення нового проєкта

## Вікно додавання опису проєкта

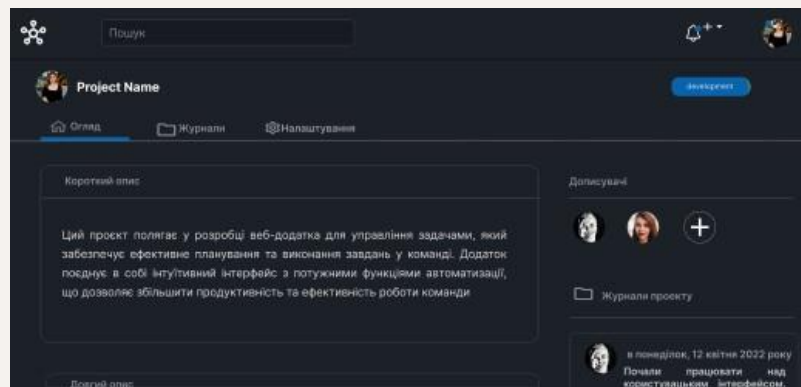


Рисунок Г.22 – Вікно додавання опису проєкта

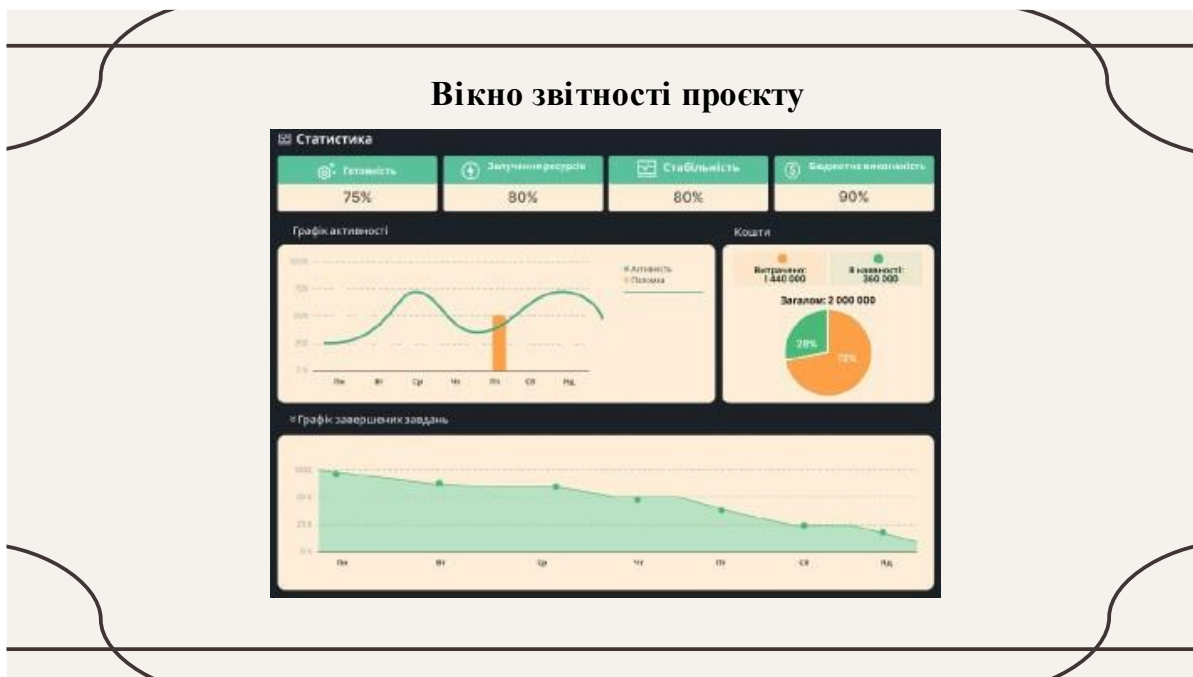


Рисунок Г.23– Вікно звітності проєкту

## Апробації та публікації

- **Апробація матеріалів магістерської кваліфікаційної роботи.** Результати магістерської кваліфікаційної роботи доповідалися та обговорювалися на міжнародній науково -практичній Інтернет -конференції «Електронні інформаційні ресурси: створення, використання , доступ - 2023».
- **Публікації** Основні результати дослідження опубліковані в тезах доповіді на міжнародній науково -практичній Інтернет -конференції «Електронні інформаційні ресурси: створення, використання , доступ - 2023»

Рисунок Г.24 –Апробації та публікації

## Висновки

- Під час виконання магістерської кваліфікаційної роботи було визначено функціонал системи управління стартап-проектами та методи і засоби її реалізації .
- Розроблено метод, моделі та алгоритми роботи системи.
- Розроблено програмне забезпечення системи управління стартап-проектами.
- Проведено тестування роботи програмного продукту.

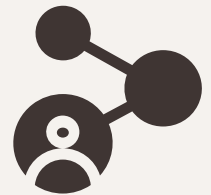


Рисунок Г.25 – Висновки

# Дякую за увагу



Рисунок Г.26 – Дякую за увагу