

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
ПІДТРИМКИ ПРОЦЕСУ УПРАВЛІННЯ ПРОЄКТАМИ
SOFTWARE TO SUPPORT
THE PROJECT MANAGEMENT PROCESS

Виконав: студент 4 курсу, групи КНТ-121сп
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

РЕКАШОВ І.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

РЕКАШОВА Ігоря Валерійовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення підтримки процесу управління проєктами. Software to Support the Project Management Process

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 03 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., професор	Підпис	Підпис
Нормоконтроль	КАМІНСЬКА Ж.К., асистент	Підпис	Підпис

7. Дата видачі завдання “24” квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

Підпис Ігор РЕКАШОВ
(підпис) (Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

Підпис Андрій ОЛІЙНИК
(підпис) (Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
103 с., 42 рис., 3 табл., 3 дод., 19 джерел.

ANDROID STUDIO, JAVA, МОБІЛЬНИЙ ЗАСТОСУНОК, МОВА
ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОЄКТ.

Об'єкт дослідження – процес розробки програмного забезпечення для управління проєктами.

Предмет дослідження – програмні засоби для підтримки процесу управління проєктами.

Мета роботи – розробка програмного забезпечення для підтримки процесу управління проєктами.

Матеріали, методи та технічні засоби: мова програмування Java, середовище розробки Android Studio.

Результати. Розроблено програмне забезпечення для підтримки процесу управління проєктами за допомогою мови програмування Java та середовища розробки Android Studio.

Висновки. Виконано проєктування програмного забезпечення для підтримки процесу управління проєктами. Розроблено програмне забезпечення для підтримки процесу управління проєктами. Здійснено тестування розробленого програмного забезпечення для підтримки процесу управління проєктами.

Галузь використання – управління проєктами та програмами.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 103 pages, 42 figures, 3 tables, 3 appendixes, 19 sources.

ANDROID STUDIO, JAVA, MOBILE APPLICATION, PROGRAMMING LANGUAGE, SOFTWARE, PROJECT.

The object of research is the process of developing software for project management.

The subject of the research is software for supporting the project management process.

The purpose of this work is to develop the software to support the project management process.

Materials, methods and technical tools: Java programming language, Android Studio development environment.

Results. The software tools to support the project management process using the Java programming language and the Android Studio development environment have been developed.

Conclusions. The software to support the project management process has been designed. The software to support the project management process has been developed. The developed software to support the project management process has been tested.

Scope of use – project management.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Програмне забезпечення для управління проєктами.....	11
1.2 Аналіз програмного забезпечення для підтримки процесу управління проєктами.....	24
1.3 Висновки за розділом 1	38
2 Матеріали і методи	40
2.1 Вибір мови програмування	40
2.2 Вибір середовища розробки для створення програмного забезпечення для підтримки процесу управління проєктами.....	42
2.3 Висновки за розділом 2	44
3 Опис програми	45
3.1 Структура програмного забезпечення для підтримки процесу управління проєктами.....	45
3.2 Функціонування програмного забезпечення для підтримки процесу управління проєктами	46
3.3 Особливості реалізації програмного забезпечення для підтримки процесу управління проєктами	50
3.4 Проєктування інтерфейсу програмного забезпечення для підтримки процесу управління проєктами	53
3.5 Висновки за розділом 3	59
4 Експлуатація, тестування та експериментальне дослідження програми.....	60
4.1 Призначення й умови застосування програми	60
4.2 Характеристики програми для підтримки процесу управління проєктами.....	61
4.3 Інструкція по експлуатації програми.....	61
4.3.1 Звернення до програми.....	61

4.3.2 Вхідні й вихідні дані	62
4.3.3 Повідомлення.....	62
4.4 Виконання програмного забезпечення для підтримки процесу управління проєктами	62
4.5 Тестування програмного забезпечення для підтримки процесу управління проєктами	70
4.6 Висновки за розділом 4	71
Висновки	72
Перелік джерел посилання	75
Додаток А Технічне завдання	77
Додаток Б Фрагмент тексту програми	81
Додаток В Слайди презентації	95

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CPM – Critical Path Method;

PRINCE – PRojects IN Controlled Environments;

БД – база даних;

ІТ – інформаційна технологія;

ПЗ – програмне забезпечення.

ВСТУП

В сучасному бізнес-середовищі, де проєкти стають все складнішими і мають багатоаспектну структуру, ефективне управління за допомогою відповідного програмного забезпечення стає критично важливим для досягнення успіху проєкту [1]-[3].

Сучасні проєкти часто мають складну структуру та велику кількість завдань і учасників. Використання програмного забезпечення для управління проєктами дозволяє ефективно розподіляти завдання, визначати та відстежувати терміни виконання, що сприяє кращій координації робіт [1]-[4].

Програмне забезпечення для управління проєктами забезпечує можливості спільної роботи та комунікації між членами команди, що призводить до зменшення ймовірності непорозумінь та покращує комунікацію між членами проєктної команди, що, в свою чергу, сприяє підвищенню командної ефективності [4]-[7]. Програмне забезпечення для управління проєктами дозволяє вести точний облік робочого часу, витрат і прогресу завдань, що допомагає забезпечити вчасне виконання проєктів, визначити можливі ризики та приймати вчасні коригування [1]-[7].

Використання інструментів автоматизації та інтеграції засобів управління проєктами дозволяє вдосконалити процеси управління проєктами, що збільшує продуктивність команди та знижує ризики помилок [4]-[6]. Програмне забезпечення для управління проєктами дозволяє збирати та аналізувати великі обсяги даних щодо прогресу проєктів. Це дає можливість приймати обґрунтовані рішення на основі об'єктивних даних [5]-[7].

Отже використання програмного забезпечення для управління проєктами дозволяє компаніям оптимізувати свою діяльність, зменшити ризики та підвищити ефективність в умовах зростаючої конкуренції та мінливих умов ринку [1]-[7].

Проте деякі програми для підтримки процесу управління проєктами можуть бути досить складними для впровадження, зокрема, великі та складні

програмні засоби управління проєктами можуть вимагати значних зусиль та часу для впровадження, що може бути проблематичним для невеликих компаній або проєктів з обмеженим бюджетом. Деяке програмне забезпечення для управління проєктами може бути дорогими в установці та супроводженні, що може бути важливим негативним фактором для малих підприємств або команд з обмеженим фінансуванням. Програмне забезпечення для управління проєктами підпорядковане технологічним рішенням і може потребувати постійного оновлення та підтримки. Деякі програмні продукти для управління проєктами можуть бути або занадто простими, або, навпаки, занадто складними для конкретного проєкту. Невірний вибір може викликати проблеми та знижувати ефективність управління проєктом та роботи проєктної команди. Деяке програмне забезпечення для управління проєктами може бути занадто стандартизованим та не надавати достатньої гнучкості для вирішення конкретних потреб певного проєкту або команди [1]-[7].

Тому актуальною є розробка програмного забезпечення для підтримки процесу управління проєктами. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для підтримки процесу управління проєктами.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підтримки процесу управління проєктами;
- здійснити проєктування програмного забезпечення для підтримки процесу управління проєктами;
- створити програмне забезпечення для підтримки процесу управління проєктами;
- виконати тестування розробленого програмного забезпечення для підтримки процесу управління проєктами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для управління проєктами

Управління проєктами полягає у застосуванні конкретних знань, вмінь, методологій і прийомів для досягнення визначених, вимірюваних цілей проєкту, включаючи успішне завершення процесу. Відмінність управління проєктами від звичайного менеджменту полягає в тому, що воно керує метою та завданнями проєкту опосередковано та з плином час [1]-[7].

Проєкт є сукупністю певних, обмежених за часом завдань, спрямованих на досягнення конкретної мети. Участь кількох членів команди в процесі реалізації проєкту допомагає досягти значущих результатів за визначений період. Оскільки проєкт включає роботу представників різних сфер, він потребує початкових інвестицій у час та ресурси [1]-[7].

Менеджери проєктів охоплюють усі етапи життєвого циклу проєкту, від керівництва діяльністю команди до розв'язання щоденних викликів та впровадження стратегій управління проєктами. Вони несуть кінцеву відповідальність за успішне втілення проєкту та досягнення ключових результатів і цілей проєкту [1]-[7].

Проаналізуємо обов'язки керівника проєкту, що дасть можливість поставити вимоги до розробки програмного забезпечення для управління проєктами [1]-[7]. Менеджери проєктів несуть відповідальність за різноманітні завдання, пов'язані із виконанням проєкту, включаючи: розробка та управління планом проєкту, призначення завдань і розподіл обов'язків між учасниками команди, взаємодія з ключовими зацікавленими сторонами, ефективне виконання кожного етапу проєкту, конфігурація командних інструментів і обладнання для управління проєктом (наприклад, використання програмного забезпечення для управління проєктами та створення діаграм Ганта), моніторинг бюджету та масштабів проєкту для уникнення перевитрат або відхилень від плану, виявлення та вирішення потенційних перешкод та проблем, планування та організація зустрічей, а також управління

обговореннями в команді, відстеження прогресу поточних проєктів, завершення життєвого циклу проєкту, включаючи огляд результатів та виконання всіх проєктних обов'язків [1]-[7].

Менеджери проєктів повинні мати широкий і гнучкий набір навичок для успішного виконання цих обов'язків [1]-[7].

Проаналізуємо етапи управління проєктом [1]-[7]. Початок проєкту: фаза ініціації проєкту фокусується на створенні високо рівневого бачення проєкту та отриманні офіційного схвалення від зацікавлених сторін. Цей етап не має за мету докладні деталі, а скоріше слугує для того, щоб надати узагальнений огляд того, що відбувається, і мотивувати команду розмислювати про майбутнє [1]-[7].

Планування проєкту: на етапі планування проєкту команда глибше опрацьовує вже створене бачення проєкту. Перш за все, команда повинна відповісти на кілька ключових питань: як будуть досягнуті поставлені цілі, як буде втілюватися проєкт, коли він розпочнеться і як вимірюватиметься його успіх. Отримавши відповіді на ці питання, команда може розпочати створення інфраструктури проєкту, яка охоплює такі ключові аспекти, як: обсяг проєкту, поставка, ключові етапи, цілі та завдання, основні ресурси (внутрішні та зовнішні), ролі в команді, тривалість проєкту, можливі ризики або труднощі, залежності [1]-[7].

Фаза реалізації проєкту є основною частиною процесу, де відбувається більшість роботи та досягаються ключові результати. На цьому етапі керівник проєкту навчає та керує командою для досягнення поставлених цілей проєкту, інформуючи зацікавлених сторін про ключові кроки та показники результативності, а також відстежуючи прогрес. Управління проєктом, таке як використання програмного забезпечення для керування проєктами, може полегшити цей процес, відслідковуючи терміни виконання та результати та служити платформою для співпраці між членами команди [1]-[7].

Під час етапу моніторингу ефективності виконання проєкту керівник проєкту активно спостерігає за загальним прогресом роботи та станом

команди. Незалежно від того, чи просувається команда в напрямку вражаючих результатів, чи стикається з викликами та труднощами, керівник проєкту може сприяти зневаженню стресу, вирішенню проблем та повідомленню ключовим зацікавленим сторонам про оновлення [1]-[7].

Заключним етапом життєвого циклу проєкту є період завершення проєктної діяльності, досягнення поставлених результатів та обговорення витоків і провалів протягом всього проєкту. Важливою є ефективна комунікація між членами команди на завершальних етапах роздумів і святкування успіхів [1]-[7].

Методологія та структура управління проєктами визначають набір інструкцій і принципів, які команди можуть використовувати для максимізації ефективності та досягнення найкращих результатів у проєкті. Кожна методологія розглядає управління проєктом зі свого унікального погляду, надаючи чіткі та повторювані кроки, які команди використовують протягом усього життєвого циклу проєкту. Кожен метод є конкретним і не може бути застосований одночасно з іншими методами [1]-[7].

Структура управління проєктом може служити методологічною основою та надавати більш цілеспрямований погляд на те, як впроваджувати та застосовувати методологічні вказівки. Структури та правила відповідають методологічним принципам, проте сама структура визначає, як і коли ці правила має застосовувати в контексті проєкту [1]-[7].

Дослідимо різні методології управління проєктами [1]-[7].

Метод гнучкого управління проєктами Agile зосереджений на ітеративному та дуже гнучкому підході до управління проєктами. Цей підхід акцентує поетапну реалізацію проєктів протягом всього життєвого циклу, а не в кінці проєкту. Гнучке управління проєктами дає можливість командам адаптуватися до викликів і вносити корективи більш гнучко, порівняно зі структурованими методологіями, наприклад, водоспадною моделлю [1]-[7].

Модель Agile управління проєктами підходить для проєктів, які характеризуються такими умовами [1]-[7]: група розробників програмного

забезпечення є невеликою; група розробників стикається з високим рівнем невизначеності; групи, що створюють прототипи, які вимагають кількох рівнів налаштувань та модифікацій; команда, яка тісно співпрацює з зовнішніми сторонами та зацікавленими сторонами [1]-[7].

Водоспадне управління проєктами – це традиційний спосіб управління проєктами, що передбачає чіткі послідовні фази проєкту. У моделі водоспаду кожна фаза проєкту повинна бути повністю завершена до початку наступної, а результати проєкту повідомляються лише в кінці проєкту [1]-[7].

Водоспадна модель управління проєктами підходить для проєктів, які характеризуються такими умовами: проєкти з конкретними цілями, проєкт з певним графіком, команди повинні визначити суворі вимоги до проєкту на ранній стадії, структура управління проєктом визначена [1]-[7].

Як видно з назви, управління проєктами за допомогою моделі Scrum ґрунтується на товарищескості та командній роботі, натхненні командною грою в регбі, в рамках гнучкої методології. Під керівництвом Scrum керівник команди заохочує вивчати досвід, само організовуватися для вирішення проблем і досягати прогресу протягом усього життєвого циклу проєкту [1]-[7]. Модель Scrum управління проєктами підходить для проєктів, які характеризуються такими умовами: невеликі групи працюють із багатьма невизначеними змінними, які постійно змінюються; групи, які розробляють прототипи, що потребують кількох рівнів налаштування та модифікацій; група, яка тісно взаємодіє із зовнішніми сторонами та зацікавленими сторонами [1]-[7].

Інфраструктура моделі управління проєктами Kanban є підсистемою гнучкої методології, що акцентує на постійному вдосконаленні та гнучкому управлінні завданнями. У системі Kanban команди користуються дошкою Kanban, вертикальною дошкою, що розділяє окремі картки із завданнями на категорії в залежності від статусу життєвого циклу проєкту (наприклад, не розпочато, виконується, завершено) [1]-[7]. Модель Kanban управління проєктами підходить для проєктів, які характеризуються такими умовами:

групи, які тільки починають свій шлях в управлінні проєктами та шукають просту та систематичну структуру; проєкти із кількома незалежними завданнями та дорученнями; групи, які потребують швидкого доступу до підсумків завдань та загального огляду стану виконання [1]-[7].

Метод критичного шляху (Critical Path Method, CPM) являє собою систему управління проєктами у водоспадному стилі, яка визначає критичні та некритичні завдання, встановлює їх пріоритети за важливістю та розглядає вирішення вузьких місць. Методи CPM підкреслюють значущість визначення зв'язків між завданнями та залежностями [1]-[7]. Метод критичного шляху для управління проєктами підходить для проєктів, які характеризуються такими умовами: групи, що управляють великими та складними проєктами; проєкти, які включають в себе кілька завдань з підзадачами та залежностями; групи, що прагнуть підвищити продуктивність та уникнути вузьких місць у після проєктному періоді [1]-[7].

Модель PProjects IN Controlled Environments (PRINCE2) є методологією каскадного управління проєктами, що наголошує на організації та контролі. Модель PRINCE2 керує ризиками та ресурсами, розбиваючи проєкти на невеликі управлінні підрозділи та чітко визначаючи командні ролі та відповідальність [1]-[7]. Модель PRINCE2 управління проєктами підходить для проєктів, які характеризуються такими умовами: групи з меншим досвідом управління проєктами (PRINCE2 надає чіткі та зрозумілі кроки); групи можуть чітко визначити конкретні обов'язки; визначені фази проєкту та розподіл завдань [1]-[7].

Дослідимо процес управління ІТ-проєктами. ІТ-проєкт – це вид проєкту, що включає в себе інфраструктуру інформаційних технологій, інформаційні системи або комп'ютери. Приклади таких проєктів охоплюють веброзробку, розробку програмного забезпечення, створення мобільних додатків, конфігурацію мережі, впровадження програмного забезпечення, установку апаратного забезпечення, управління базами даних та відновлення ІТ під час збоїв [1]-[7].

Управління проектами в галузі інформаційних технологій являє собою процес планування, розробки, впровадження, моніторингу та звітності про ІТ-проекти. Оскільки багато галузей ІТ сфокусовані саме на ІТ-проектах, багато з них не включають компоненти ІТ [1]-[7].

Оскільки сфера діяльності з розробки програмного забезпечення охоплює різні предметні області, менеджерам ІТ-проектів необхідно враховувати питання, такі як ризики, перехресна інтеграція, оновлення програмного забезпечення та інші. Таким чином, для успішної реалізації ІТ-проектів необхідно використовувати більше, ніж традиційні інструменти та методи управління проектами [1]-[7].

Відзначимо, що професійне програмне забезпечення для управління ІТ-проектами, яке включає діаграми Ганта, дошки Kanban, інформаційні панелі та звіти, надає необхідні інструменти для успішної реалізації ІТ-проектів [1]-[7].

Шість етапів ІТ-проектів ґрунтуються на етапах управління проектом, які використовуються в поєднанні з фазою управління проектом. Ці етапи виглядають наступним чином [1]-[7].

Перший етап в будь-якому проекті полягає у визначенні потреб проекту. Це питання необхідно вирішити, щоб з'ясувати мету проекту. Далі слід скласти проектну пропозицію, включаючи бізнес-план, який відповідає вимогам проекту. Також можна провести техніко-економічне обґрунтування для повного обґрунтування пропозиції [1]-[7].

Після схвалення проектною пропозиції проект переходить у фазу визначення. Тут остаточно формулюються цілі проекту та встановлюються вимоги для успішної реалізації проекту. На цьому етапі можна визначити обсяг проекту та розробити план його реалізації. Бюджет визначений, і кошторис складений [1]-[7].

На етапі створення проекту програмного забезпечення команда зосереджується на пошуку оптимальних рішень для досягнення своїх цілей, включаючи розробку різноманітних візерунків і дизайнів. Після вибору відповідного дизайну формулюються специфікації, які передаються команді

розробників [1]-[7]. На етапі прогресу назначаються завдання для команди розробників, вибираються інструменти управління проєктом, і визначаються технічні деталі. Тут також уточнюються різні аспекти. Головна мета на цьому етапі полягає у забезпеченні зрозумілого загального плану, щоб уникнути проблем на етапі впровадження [1]-[7].

На етапі впровадження відбувається реалізація кінцевих результатів проєкту. Цей етап часто виявляється найдовшим. Команди проєкту виконують завдання, а менеджери відстежують та контролюють виконання завдань, ресурси, витрати, якість і ризики [1]-[7].

На етапі виконання остаточний проєкт передається клієнтам або зацікавленим сторонам. Наступний етап включає всі завдання, які виконуються після реалізації проєкту, такі як створення команди підтримки, навчання кінцевих користувачів, проведення посмертного аналізу та, нарешті, закриття проєкту [1]-[7].

Більшість ІТ-проєктів та їх етапів керуються традиційною каскадною структурою. Однак гнучка структура може зменшити ризик при додаванні функцій. Керівник ІТ-проєктів має відповідальність за нагляд за діяльністю розробників та інших членів команди для вчасного та ефективного виконання проєктів у визначених рамках бюджету [1]-[7]. До обов'язків керівника ІТ-проєктів входять: визначення цілей проєкту та розробка плану для їх досягнення, моніторинг графіків та бюджетів проєктів та створення звітів про їхній статус, управління ресурсами, включаючи команди та обладнання, розподіл завдань серед членів команди, розробка стратегії для вчасного та бюджетного виконання проєктів, використання інструментів управління ІТ-проєктами для відстеження прогресу і продуктивності, оцінка ризиків проєкту та розробка стратегії управління ними, проведення регулярних зустрічей з командою та зацікавленими сторонами [1]-[7]. Керівники ІТ-проєктів повинні мати глибокі знання у сфері комп'ютерів, операційних систем, мереж та управління службою обслуговування. Також важливі навички включають високий рівень комунікабельності та здатність чітко пояснювати складні

технічні питання. Інші важливі навички включають досвід у плануванні, бюджетуванні та управлінні ресурсами [1]-[7].

Суть програмного забезпечення для управління ІТ-проєктами полягає в тому, щоб допомагати менеджерам в організації та управлінні процесами ІТ-проєктів. Так само, як і у випадку із всіма програмними інструментами, основна мета полягає в максимізації ефективності [1]-[7].

Програмне забезпечення для управління ІТ-проєктами підвищує ефективність, забезпечуючи користувачам необхідні функції для моніторингу та відстеження прогресу та продуктивності. Воно допомагає підтримувати ІТ-проєкти в рамках плану та виконувати жорсткі графіки та бюджети [1]-[7].

Основні функції, загальні для інструментів управління ІТ-проєктами, включають відстеження завдань і часу, реальні дані, необмежену пам'ять для зберігання файлів, кілька переглядів проєктів, планування, планування та звітування з підтримкою різних методів [1]-[7].

Не залежно від обраного програмного забезпечення для керування ІТ-проєктами, воно повинно спрощувало роботу. Це сприятиме організації завдань та плануванню виконання відповідно до конкретного бюджету. При розробці програмного забезпечення потрібні інструменти для об'єднання команди та зацікавлених сторін, забезпечуючи єдність підходу [1]-[7].

Використання програмного забезпечення для управління ІТ-проєктами має суттєві переваги. Проаналізуємо переваги програмного забезпечення для управління ІТ-проєктами має суттєві переваги [1]-[7]: планування проєкту за допомогою діаграми Ганта, відстеження статусу проєкту в режимі реального часу через інформаційну панель, програмне або портфельне управління ІТ-проєктами, керування командою проєктами цифрової трансформації, можливість відстеження часу, витраченого на виконання завдань, і контролювання прогрес [1]-[7].

Проаналізуємо функції програмного забезпечення для управління ІТ-проєктами [1]-[7]. Основною функцією відповідного програмного забезпечення є можливість керування проєктами. Існує багато способів

управління IT-проєктами, але деякі аспекти є універсальними. Кроки можуть трошки відрізнятися, але загальний напрямок однаковий [1]-[7]. Таке програмне забезпечення повинно підтримувати процес збору та аналізу вимог до розроблюваного проєкту: перед початком проєкту необхідна документація, необхідно визначити обсяг, створити бюджет і визначити вимоги зацікавлених сторін. Усі ці документи можна зберігати у необмеженій кількості файлів у вашому проєкті за допомогою нашого програмного забезпечення [1]-[7].

Зібравши вимоги до проєкту, необхідно створити команду з потрібними навичками та досвідом для виконання поставленого завдання. Цей процес необхідно інтегрувати з відповідним програмним забезпеченням для керування проєктами для безперебійного спілкування [1]-[7].

Після цього необхідно використовувати діаграми Ганта для додавання завдань до графіка, встановлення залежностей, визначення етапів та перегляду критичних шляхів. Діаграма Ганта рекомендована для менеджерів проєктів, що працюють у водоспадному середовищі. Проте дані проєкту можуть бути спільно використані в різних режимах перегляду, таких як списки завдань, календарі та дошки Kanban, що усуває потребу у використанні традиційних інструментів планування [1]-[7]. Використання дошки Kanban дозволяє відстежувати робочий процес під час виконання проєкту. Тому відповідне програмне забезпечення управління проєктами повинно дозволяти створювати робочі процеси, проводити спринти та працювати з гнучкими системами, не порушуючи графіку Ганта [1]-[7].

У процесі реалізації проєкту важливо, щоб фактичний прогрес відповідав запланованому. Тому відповідне програмне забезпечення повинно мати інформаційну панель у реальному часі, яка автоматично збирає оновлену інформацію [1]-[7]. IT-проєкти потребують розумного управління навантаженням, що також повинно контролюватися за допомогою відповідного програмного забезпечення [1]-[7].

Проєкти постійно розвиваються. Якщо з'являється запит на зміни від зацікавлених сторін, необхідно негайно приймати заходи. Відповідне

програмне забезпечення повинно надавати можливість планування та забезпечувати гнучкість команди [1]-[7].

Програмне забезпечення підтримки процесу управління проєктами повинно надавати можливість створювати докладні звіти для моніторингу прогресу та повідомлення зацікавлених сторін [1]-[7].

Керівництво ІТ-проєктами охоплює різноманітні питання, які в кінцевому підсумку впливають на різні аспекти організації, такі як управління, кадри, фінанси, інші відділи всередині організації та поза її межами [1]-[7]. Однією з проблем є те, що сфера ІТ є нестабільною, і темпи змін можуть бути захоплююче високими. Технології швидко застарівають, і керівництво ІТ-проєктів повинно бути готовим до неминучих трансформацій [1]-[7]. Іншим аспектом, який потрібно вирішити керівництву ІТ-проєктів, є взаємодія між командами. Це загальна проблема для всіх проєктів, але ІТ-спеціалісти ускладнюють ситуацію, організовуючи роботу віддаленими командами на різних часових поясах. Наявність чітких та ефективних каналів зв'язку є важливою умовою успішності будь-якого ІТ-проєкту [1]-[7]. Також важливою є прозорість, оскільки в галузі ІТ існує певна концентрація. Чіткі терміни та прозорість у всіх проєктах дозволяють командам ефективно інтегрувати нові технології та оперативно реагувати на будь-які зміни [1]-[7].

Управління ІТ-проєктами стикається з численними викликами, і одним з головних питань є те, що багато організацій не використовують методи управління ІТ-проєктами для своїх завдань. Без такого керівництва, проєкти можуть втратити напрямок та виходити за межі бюджету. Присутність керівника ІТ-проєкту, який визначає процеси, ролі та інструменти, є важливим кроком для досягнення успіху у проєкті [1]-[7].

Існує велика різноманітність методів управління проєктами, проте їх можна розділити на дві основні категорії: традиційні та нетрадиційні методи. Проаналізуємо традиційний підхід водоспадного типу, де робота над проєктом розподіляється на послідовні етапи, кожен з яких залежить від завершення попереднього [1]-[7].

Методологія водоспаду є стандартним підходом для багатьох ІТ-проектів. Незважаючи на те, що така методологія широко використовується в управлінні проектами для великих проектів, вона також успішно застосовується в ІТ-проектах з формальною та лінійною структурою [1]-[7]. У цій методології перш за все, визначаються, аналізуються та фіксуються вимоги у документі, який описує, що і як має бути виконано. Цей документ оцінюється зацікавленими сторонами [1]-[7]. На наступному етапі рішення, прийняті на попередньому етапі, фіксуються в документі, який включає усе необхідне для завершення проекту [1]-[7]. Після цього керівники проектів та розробляють проектні документи згідно зі специфікаціями, процедурами та графіками [1]-[7].

На етапі тестування результати проекту порівнюються з базовими документами та стандартами, встановленими зацікавленими сторонами, включаючи контроль якості проекту. У разі необхідності процес починається спочатку [1]-[7]. Якщо тестування успішне, проект готовий до доставки кінцевим користувачам, і на цьому етапі виріб повинен бути повністю функціональним [1]-[7]. Більшість ІТ-проектів не завершуються постачанням, і підтримку після інсталяції часто довіряють спеціалізованій команді, особливо якщо потрібні оновлення або модифікації [1]-[7].

У розробці програмного забезпечення також використовується гнучка методологія. Ця методологія спрямована на ітераційний підхід, який використовує короткі спринти і дозволяє гнучкі зміни всередині проекту, не обмежуючись жорсткими планами. Деякі команди впроваджують методологію Agile або його конкретні варіації у свої проекти [1]-[7].

Серед деяких команд розробників більш популярним, ніж гнучкий підхід, є гібридний підхід, що поєднує в собі водоспадний та гнучкий підходи для створення більш гнучкого, але структурованого методу, який підходить для ІТ-проектів. Залежно від параметрів конкретного проекту, такий підхід може бути оптимальним рішенням [1]-[7].

Програмне забезпечення для управління ІТ-проєктами повинно адаптуватися як для каскадних, так і для гібридних методологій. Надаючи можливість роботи з діаграмами Ганта для прихильників водоспадної моделі та дошки Kanban для прихильників гнучкої розробки. Крім того, інформаційні панелі в режимі реального часу надають керівникам ІТ-проєктів оновлену інформацію про прогрес, використовуючи фільтровані показники для зручного відображення та обміну цікавою інформацією [1]-[7].

Хоча вони рідко використовуються в сфері ІТ-проєктів, важливо взяти до уваги методи критичного шляху та управління критичним ланцюгом проєкту. Метод критичного шляху визначає завдання, які обов'язково потрібно виконати для досягнення цілей проєкту. Це досягається за допомогою структури розподілу робіт або діаграми Ганта з відповідними залежностями завдань, відображеними протягом життєвого циклу проєкту, щоб уникнути ситуацій з тупиками в роботі команди. Такі підходи дозволяють точно визначити, які завдання слід виконати та коли [1]-[7].

Управління критичним ланцюгом проєкту аналізує наслідки та користується попереднім досвідом для планування робіт, необхідних для завершення проєкту. Управління критичним ланцюгом проєкту є високоефективним способом використовувати ресурси, фокусуючись на кінцевій меті. Однак затримки можуть виникнути, і це є загальним явищем, що не завжди ідеально підходить для роботи над портфелем проєктів [1]-[7].

Інструменти управління ІТ-проєктами можуть допомогти подолати виклики, з якими зіштовхуються ІТ-проєкти, забезпечити керівникам більший контроль та сприяти ефективній співпраці команд, що призводить до підвищеної продуктивності [1]-[7]. Більшість розглянутих методів працюють з діаграмами Ганта, а також надають підхід дозволяє легко пов'язувати взаємопов'язані завдання [1]-[7]. Діаграми Ганта широко використовуються командами, особливо командами, які співпрацюють і працюють над завданнями. Це корисно на бізнес-рівні, оскільки можна інтерпретувати та сприймати інформацію. Крім того, ви можете прикріпити свою роботу до

документів чи зображень, додавати допоміжні матеріали та отримувати затвердження після завершення робіт [1]-[7].

Хмарне програмне забезпечення для управління проєктами надає панель звітів у режимі реального часу, актуальні оновлення статусу відображаються у режимі реального часу. Це означає, що інформаційні панелі в реальному часі надають деталі проєкту у режимі реального часу [1]-[7]. Програмне забезпечення для управління проєктами повинно надавати різні показники, такі як вартість, робоче навантаження, час і т.д., щоб відобразити саме той обсяг даних, який є необхідним у конкретній ситуації [1]-[7].

Ролі та обов'язки в управлінні ІТ-проєктами значною мірою віддзеркалюють ролі та обов'язки в інших галузях. Менеджер ІТ-проєктів відповідає за планування, закупівлю та реалізацію проєктів. Хоча більшість проєктів мають віддалені та локальні групи, групи є важливою частиною всього проєкту. Проте можна виділити три види груп в управлінні ІТ-проєктами [1]-[7]. Традиційна група управління проєктами не обмежується інформаційними технологіями, вони керуються та укомплектовуються відповідно до формальних методологій управління проєктами [1]-[7]. Іншою групою при розробці програмного забезпечення є професійна служба обслуговування, яка постачає технології зовнішнім клієнтам, що часто відбувається під час інсталяції програмного чи апаратного забезпечення. Зазвичай його керує керівник проєкту, але може очолювати віце-президент чи директор служби. Тим не менш, використовуються формальні методи управління проєктами. Також важливо розуміти, що група, яка керує поставкою та обслуговуванням технологій у організації, впроваджує нові системи, встановлює комп'ютери, монітори, телефони та інше обладнання для співробітників і управляє системою [1]-[7].

Визначено, що управління проєктами полягає у застосуванні конкретних знань, вмінь, методологій і прийомів для досягнення визначених, вимірюваних цілей проєкту, включаючи успішне завершення процесу. Відзначено, що проєкт є сукупністю певних, обмежених за часом завдань,

спрямованих на досягнення конкретної мети. Менеджери проєктів та відповідне програмне забезпечення охоплюють підтримку усіх етапів життєвого циклу проєкту, від керівництва діяльністю команди до розв'язання щоденних викликів та впровадження стратегій управління проєктами. Проаналізовано обов'язки керівника проєкту та етапи управління проєктом, що дасть можливість поставити вимоги до розробки програмного забезпечення для управління проєктами. Досліджено методології управління проєктами, що визначають набір інструкцій і принципів, які команди можуть використовувати для максимізації ефективності та досягнення найкращих результатів у проєкті. Досліджено процес управління ІТ-проєктами. Визначено, що управління проєктами в галузі інформаційних технологій являє собою процес планування, розробки, впровадження, моніторингу та звітності про проєкти програмного забезпечення.

1.2 Аналіз програмного забезпечення для підтримки процесу управління проєктами

Проаналізуємо програмне забезпечення для підтримки процесу управління проєктами [8]-[15].

Програмне забезпечення Asana (рис. 1.1) є однією з широко використовуваних програм для управління проєктами на ринку, яка пропонує кілька режимів перегляду, що можуть бути настроєні для кожного учасника команди. Безкоштовна версія програмного забезпечення Asana володіє характеризується багатьма функціями, які полегшують роботу менеджера проєктів, йому доступний необмежений простір для зберігання, можливість пошуку даних з попередніх проєктів і відправлення повідомлень на платформі. Якщо проєктна команда невелика, то безкоштовна версія підходить для довгострокового використання [8], [9], [13].

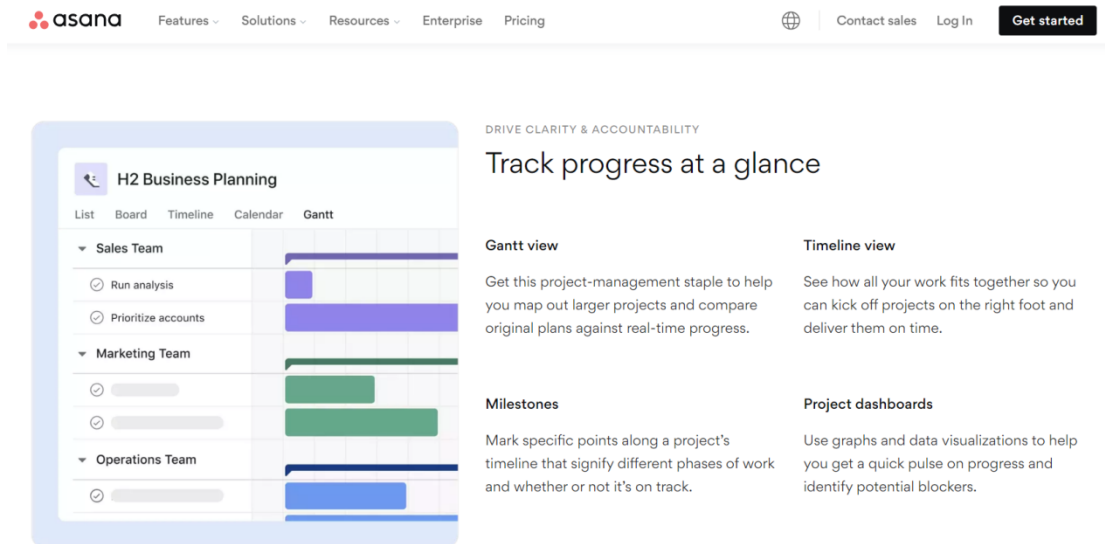


Рисунок 1.1 – Програмне забезпечення Asana [13]

Програмне забезпечення Asana пропонує широкий спектр варіантів перегляду проєктів. Програмне забезпечення Asana надає можливість використовувати його у вигляді карток, для відображення проєктів і термінів у календарі або для відображення робочого навантаження в цілому. Можливість перемикається між цими типами видів дозволяє членам команди бути більш гнучкими та взаємодіяти з багатьма зацікавленими сторонами [8], [9], [13].

Програмне забезпечення Asana підтримує ряд автоматизованих процесів та інтеграцій, які можуть сприяти підвищенню ефективності вашої команди. Програмне забезпечення Asana дозволяє встановлювати крайні терміни та отримувати автоматичні сповіщення, коли певні частини розроблюваного проєкту завершуються. Команди можуть легко переходити до наступної фази проєкту [8], [9], [13].

Програмне забезпечення Asana виходить за межі звичайного програмного забезпечення для управління проєктами, надаючи повноцінне рішення для керування завданнями, яке адаптується до динамічних та зростаючих потреб бізнесу. Програмне забезпечення Asana не лише допомагає керувати своїми завданнями, але також пов'язує всі аспекти бізнесу з цілями та стратегією бізнесу [8], [9], [13].

Програмне забезпечення Asana інтегрується з різними програмними інструментами та системами, упорядковуючи роботу на єдиній платформі. Автоматизація завдань, планування, призначення завдань і т.д. спрямована на зменшення часу, витраченого на завдання з низьким пріоритетом [8], [9], [13]. Готові шаблони проєктів дозволяють швидко розпочати виконання найважливіших завдань. Інформаційна панель звітів автоматично оновлює інформацію про зацікавлених сторін і усуває потребу в оновленні статусу на зустрічах [8], [9], [13]. Функції забезпечення безпеки, конфіденційності та відповідності призначені для захисту даних проєкту [8], [9], [13].

Програмне забезпечення Asana дозволяє інтегруватися зі сторонніми службами. Збереження всіх комунікацій в одному місці може сприяти підвищенню продуктивності, а обмін повідомленнями в межах програми може полегшити координацію. Програмне забезпечення Asana дозволяє встановлювати крайні терміни та пріоритети проєктів для команди, забезпечуючи розумне проведення всіх обговорень одночасно [8], [9], [13].

Програмне забезпечення Asana також втілює принципи гнучкого управління проєктами. Програмне забезпечення Asana може використовуватися для планування спринтів і випуску продукту [8], [9], [13].

Ключові можливості Asana дозволяють компаніям керувати різними типами проєктів. Однією з ключових переваг Asana є його здатність розподіляти ролі та обов'язки між членами команди. Програмне забезпечення Asana дозволяє призначати завдання членам команди, встановлювати терміни виконання та створювати проміжні завдання, що дозволяє розбити проєкт на менші частини. Програмне забезпечення Asana сприяє прозорості та відповідальності в проєктній команді [8], [9], [13].

Функція часової шкали дозволяє візуалізувати кінцеві терміни та залежності проєкту простим та зрозумілим способом. Програмне забезпечення Asana дозволяє встановлювати необхідні дати та етапи для просування програмного проєкту [8], [9], [13].

Програмне забезпечення Asana дає детальне візуальне уявлення графіків та залежностей проєкту, дозволяє визначати критичні шляхи, коригувати графіки та приймати обґрунтовані рішення щодо пріоритетів проєкту [8], [9], [13].

Програмне забезпечення Asana легко поєднується з різними іншими програмними інструментами, спрощуючи впровадження в існуючі робочі процеси команди, дозволяє зручно обмінюватися файлами та взаємодіяти з проєктною командою [8], [9], [13].

Функція залежностей в Asana дозволяє встановлювати зв'язки між завданнями та уникати виконання деяких завдань до завершення інших. Це сприяє організації проєкту та допомагає всім вчасно виконувати свої завдання. Функції бюджетування Asana допомагають відстежувати витрати на проєкт і залишатися в межах бюджету. Програмне забезпечення Asana дозволяє спостерігати за витратами в режимі реального часу, встановлювати бюджети для кожного проєкту та вносити коригування за необхідності [8], [9], [13].

Основними характеристиками програмного забезпечення Asana є такі [8], [9], [13]:

- засоби створення та керування завданнями та проєктами: програмне забезпечення Asana дозволяє створювати завдання та групувати їх у проєкти. Кожне завдання може мати опис, кінцеві строки виконання, відповідального та інші параметри [8], [9], [13];

- гнучкість у перегляді: програмне забезпечення Asana пропонує різні режими перегляду, такі як "Картки" (для гнучкого планування), "Календар" (для відстеження термінів) та "Діаграма Ганта" (для візуалізації графіка проєкту) [8], [9], [13];

- призначення обов'язків: можливість призначати завдання різним членам команди, встановлювати терміни виконання та створювати проміжні завдання для більш детального розподілу роботи [8], [9], [13];

– інтеграції: програмне забезпечення Asana інтегрується з різними іншими програмними інструментами, що полегшує обмін даними та співпрацю [8], [9], [13];

– візуалізація залежностей: функція залежностей дозволяє встановлювати зв'язки між завданнями та уникати виконання деяких завдань до завершення інших [8], [9], [13];

– бюджетування: програмне забезпечення Asana допомагає відстежувати витрати на проєкт і встановлювати бюджети для ефективного фінансового керування [8], [9], [13];

– гнучка система повідомлень: програмне забезпечення Asana надає можливості членам команди спілкуватися через систему коментарів, щоб уточнювати деталі та обговорювати завдання [8], [9], [13];

– готові шаблони: програмне забезпечення Asana надає готові шаблони проєктів, які допомагають прискорити старт нових завдань [8], [9], [13];

– функції безпеки та конфіденційності: програмне забезпечення Asana забезпечує захист даних та надає можливості налаштування доступу для різних користувачів [8], [9], [13];

– інтеграція з іншими сервісами: програмне забезпечення Asana надає можливість інтеграції зі сторонніми сервісами для полегшення комунікації та автоматизації процесів [8], [9], [13].

Серед недоліків програмного забезпечення Asana можна відзначити складність використання для новачків, оскільки для нових користувачів може виникнути певна складність у засвоєнні інтерфейсу та основних функцій Asana. Крім того, деякі додаткові функції та можливості доступні лише у платних версіях Asana, що може бути витратним для невеликих команд чи компаній [8], [9], [13]. Іншим недоліком є те, що для ефективного використання Asana, потрібно активно керувати завданнями та оновлювати інформацію, що може бути витратним у великих проєктах [8], [9], [13].

Програмне забезпечення ClickUp (рис. 1.2) представляє собою потужний та багатофункціональний програмний інструмент для керування

проектами, що дозволяє ефективно управляти та завершувати всі завдання проектної команди зі зручної платформи [8], [9], [14].

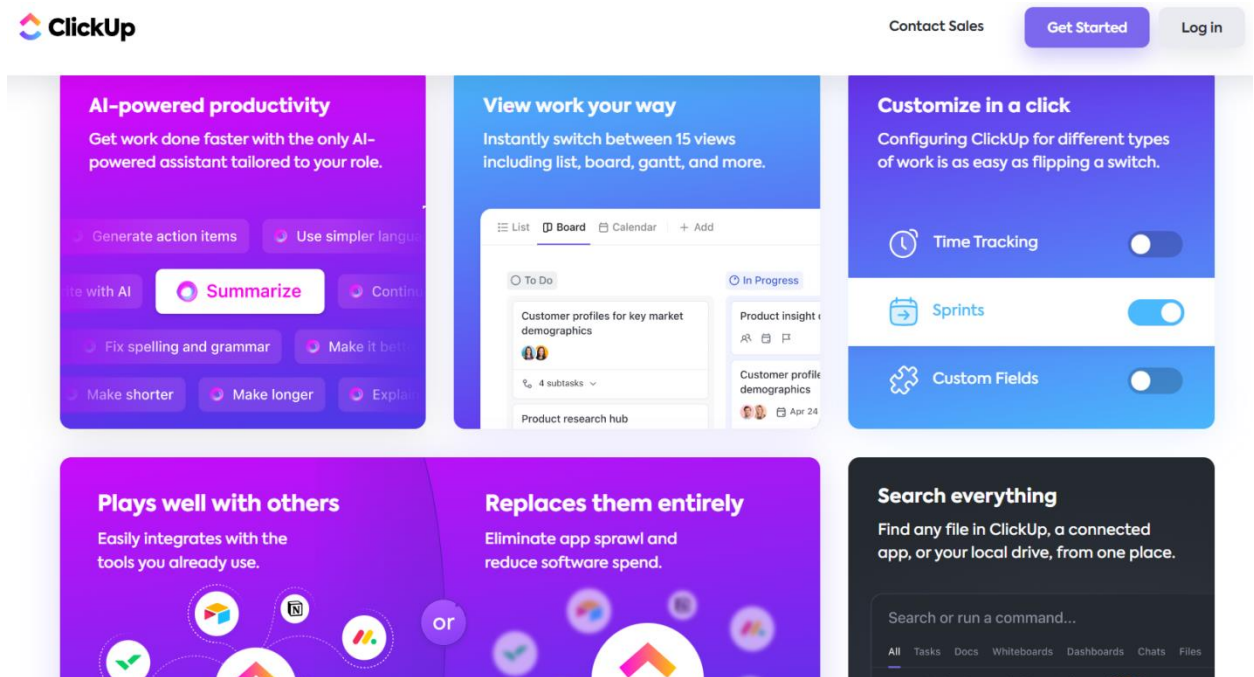


Рисунок 1.2 – Програмне забезпечення ClickUp [14]

Основні переваги вибору програмного забезпечення ClickUp полягає у тому, що ця програма надає можливості планування проєктів, управління завданнями, координації ресурсів, комунікації та співпраці з членами команди та гостями, що здійснюється з централізованого робочого простору. Незважаючи на розмаїття функцій, платформа відзначається зручним візуальним дизайном [8], [9], [14].

Програмне забезпечення ClickUp дозволяє ефективно фільтрувати, сортувати, шукати та легко редагувати свої завдання, відображаючи їх у форматі, що ідеально підходить для вашої команди [8], [9], [14].

В ClickUp представлено майже десяток різних типів завдань, включаючи графіки, діаграми, коробки та дошки. Функції співпраці орієнтовані на інтерактивний чат та документи для спільної роботи, а також на дошки оголошень. Команди проєкту можуть легко обговорювати та

коментувати документи та завдання, взаємодіяти з іншими учасниками команди [8], [9], [14].

Програмне забезпечення ClickUp надає можливість створення налаштованої інформаційної панелі та шість різних типів звітів для групового звітування доступні завдяки функції звітування [8], [9], [14].

Основні можливості програмного забезпечення ClickUp включають засоби для створення звітів в реальному часі, колективні дошки та об'єднані документи, керування завданнями з кількома режимами перегляду, відстеження цілей і використання шаблонів [8], [9], [14].

Щодо інтеграцій, програмне забезпечення ClickUp надає власні з'єднання з різними програмами. Протягом обмеженого часу є можливість ознайомитися з безкоштовним повнофункціональним планом ClickUp, щоб визначити, чи він найкраще відповідає потребам команди [8], [9], [14].

ClickUp є програмним забезпеченням для управління проєктами із розширеними можливостями, які виділяють його серед конкурентів. Його економічна ефективність, вражаючи безкоштовні плани, гнучкість преміального плану та широкі можливості налаштування роблять його популярним рішенням для бізнесу [8], [9], [14].

Програмне забезпечення ClickUp пропонує одну з найкращих програм для управління проєктами. Безкоштовний план має ряд функцій для задоволення потреб управління проєктами окремих осіб або керівників невеликих груп. На відміну від багатьох альтернатив, він не має обмежень для користувачів. Проте він має обмеження щодо сховища, робочого процесу, конфігурації та інтеграцій [8], [9], [14]. План Unlimited надає гостьові дозволи, необмежену пам'ять, численні інтеграції та доступ до всіх завдань, включаючи Gantt, Teams і Cards (хоча є обмеження на кількість унікальних переглядів, які можна створити) [8], [9], [14].

Програмне забезпечення ClickUp пропонує потенційним клієнтам можливість поділитися та заохочує користувачів показувати, скільки вони готові платити щомісяця [8], [9], [14]. На цьому рівні, за невеликими

винятками, команди мають можливість користуватися більшістю функцій, які пропонує ClickUp, від звітності до інтеграцій. Зокрема, наявність єдиного входу та можливістю спілкування в режимі реального часу, але не надається спеціальна безпека або рівень підтримки [8], [9], [14].

План ClickUp Enterprise спеціально розроблено для великих команд. Гнучкі можливості для управління проєктами та продуктом ClickUp сприяють ефективній співпраці команд під час спринтів, планування та вирішення завдань, що підвищує конкурентоспроможність команди. Користувачі можуть автоматизувати робочі процеси спринту та полегшити відстеження помилок і проблем [8], [9], [14].

Програмне забезпечення ClickUp забезпечує гнучку інформаційну панель, яка дозволяє відслідковувати прогрес за допомогою налаштованих інформаційних панелей спринтів [8], [9], [14]. Програмне забезпечення ClickUp пропонує гнучку модель управління проєктами, що уникне крутих кривих навчання, з якими часто стикаються новачки при цьому методі. Унікальні функції ClickUp не вимагають коду чи інших засобів, що робить його простим у використанні для широкого кола користувачів. ClickUp відображає параметри налаштування за допомогою автоматизації. Інструмент пропонує шаблони автоматизації, які можна налаштувати під конкретного користувача або використовувати готові налаштування [8], [9], [14].

Програмне забезпечення ClickUp надає користувачам велику кількість готових шаблонів та інтеграцій із різними програмними інструментами, а також дозволяє налаштовувати зв'язки між завданнями та іншими проєктними документами [8], [9], [14].

Програмне забезпечення ClickUp має такі особливості [8], [9], [14]:

- гнучкі плани: програмне забезпечення ClickUp пропонує безкоштовний план з рядом функцій, а також преміальні плани, які надають розширені можливості для великих команд та підприємств [8], [9], [14];

- ієрархічна організація: програмне забезпечення ClickUp використовує ієрархічну структуру для організації робочого простору, просторів, папок,

списків, завдань та проміжних завдань, що дозволяє структурувати проекти у логічний порядок [8], [9], [14];

- різноманітні типи завдань: програмне забезпечення ClickUp має різноманітні типи завдань, такі як часові шкали, діаграми, коробки та дошки, що дозволяє користувачам вибирати оптимальний спосіб представлення завдань [8], [9], [14];

- засоби інтеграції: програмне забезпечення ClickUp інтегрується з різними іншими програмними інструментами, щоб полегшити інтеграцію з існуючими робочими процесами [8], [9], [14];

- функції безпеки та підтримки: одним з платних пакетів пропонує розширені служби безпеки та підтримки, включаючи відповідність та авторизацію ролей [8], [9], [14];

- гнучка модель управління проектами: програмне забезпечення ClickUp пропонує гнучку модель управління проектами, спрощуючи навчання та використання [8], [9], [14];

- засоби автоматизації: програмне забезпечення ClickUp надає функції автоматизації, що дозволяють користувачам оптимізувати робочі процеси та використовувати готові шаблони автоматизації для підвищення ефективності розробки проекту [8], [9], [14];

- широкий вибір шаблонів: програмне забезпечення ClickUp пропонує велику кількість шаблонів та інтеграцій з різними інструментами, дозволяючи користувачам налаштовувати робочі процеси [8], [9], [14].

Серед недоліків програмного забезпечення ClickUp можна відзначити складність використання для новачків, оскільки для нових користувачів може виникнути певна складність у засвоєнні інтерфейсу та основних функцій ClickUp. Крім того, деякі додаткові функції та можливості доступні лише у платних версіях ClickUp, що може бути витратним для невеликих команд чи компаній [8], [9], [14]. Іншим недоліком є те, що для ефективного використання Asana, потрібно активно керувати завданнями та оновлювати інформацію, що може бути витратним у великих проектах [8], [9], [14].

Програмне забезпечення Zoho Projects (рис. 1.3) надає можливість здійснювати ефективне управління проєктами через широкий спектр функцій, таких як використання діаграм Ганта для розробки плану проєкту, дошки Kanban для структуризації робочих елементів, а також графіків використання ресурсів для ефективного розподілу завдань. Варто відзначити численні інформаційні панелі, звіти та графіки, які сприяють аналізу та відстеженню важливих показників проєкту [8], [9], [15].

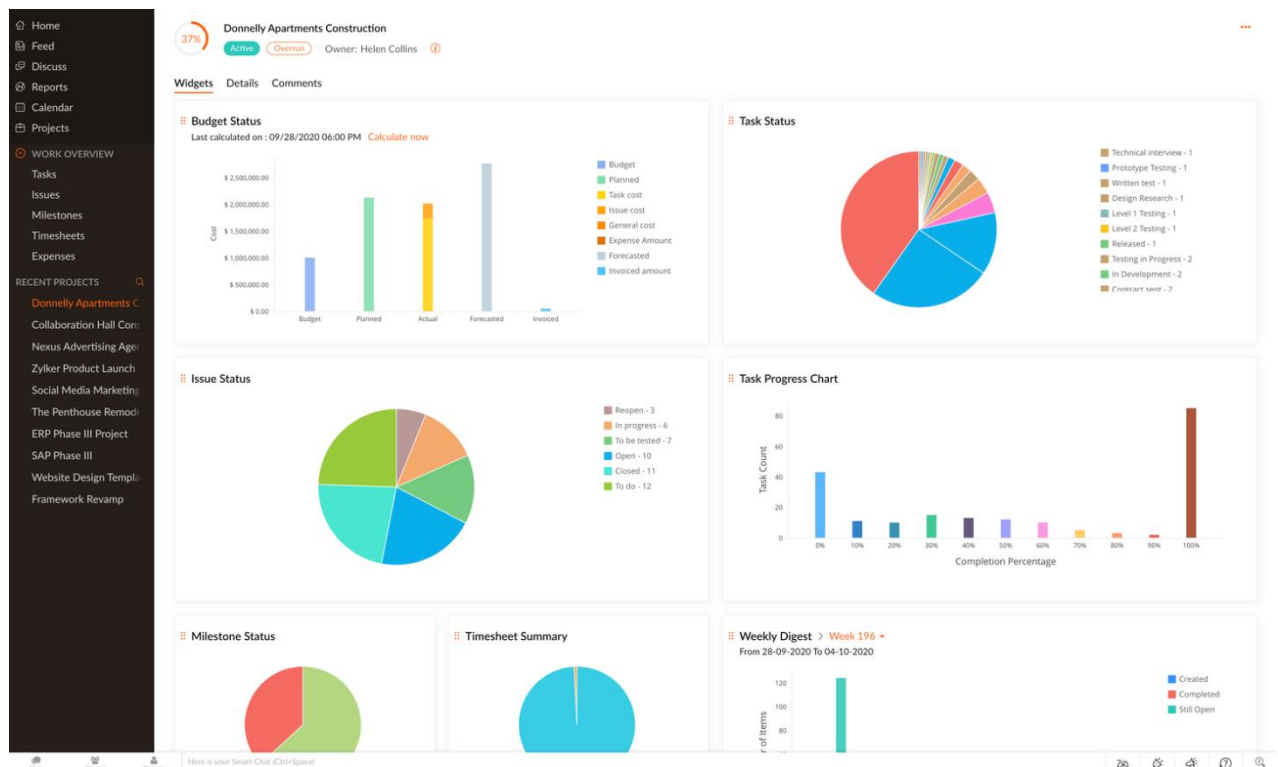


Рисунок 1.3 – Програмне забезпечення Zoho Projects [15]

За допомогою програмного забезпечення Zoho Projects учасники команди можуть легко взаємодіяти, користуючись відповідними засобами, обговорювати деталі через коментарі та співпрацювати над документами. Зокрема, у програмному забезпеченні Zoho Projects є функції для ведення обліку проблем, створення бюджетів та управління пакетами. Команди можуть також ефективно відстежувати час за допомогою таймера або кнопок реєстрації та виїзду в Zoho Projects [8], [9], [15].

Програмне забезпечення Zoho Projects надає можливість користувачам автоматизувати рутинні робочі процеси, такі як призначення завдань, оновлення полів чи відсилення сповіщень, використовуючи правила робочого процесу та чернетки. Ця автоматизація значно економить час у довгостроковій перспективі [8], [9], [15].

Серед важливих функцій програмного забезпечення Zoho Projects можна виділити управління завданнями, автоматизацію завдань, створення пропозицій, ведення обліку часу та вирішення проблем, а також можливості звітування та форум для глибоких обговорень із командою [8], [9], [15].

Програмне забезпечення Zoho Projects надає засоби інтеграції з різним програмним забезпеченням. Zoho Projects надає можливість використовувати безкоштовний план, який обмежує кількість проєктів, що можна створити, це стане чудовим засобом перевірити, чи відповідає цей інструмент вашим потребам [8], [9], [15].

Нові проєкти програмного забезпечення можуть вибрати Zoho Projects для всіх своїх програм, щоб завершити свій технологічний стек з легкістю інтеграції. Zoho Projects пропонує безкоштовну версію свого програмного забезпечення, але вона обмежена та дозволяє створити до двох проєктів і підтримує лише трьох користувачів на обліковий запис. Преміальний план дозволяє створювати необмежену кількість проєктів. На цьому рівні доступні також стандартні функції управління проєктами, такі як діаграми Ганта, користувацькі перегляди та автоматизовані завдання [8], [9], [15].

Програмне забезпечення Zoho Projects включає обмін повідомленнями для приватних і групових спілкувань. Найбільш оплачуваний план дозволяє налаштовувати поля, встановлювати правила робочого процесу та отримувати звіти про використання ресурсів для всіх ваших проєктів [8], [9], [15].

Програмне забезпечення Zoho Projects є ефективним програмним засобом для малого бізнесу, який прагне інтегруватися з іншими програмами. Це особливо корисно для нових проєктів та команд, які тільки починають проєктну діяльність [8], [9], [15].

Програмне забезпечення Zoho Projects має такі особливості [8], [9], [15]:

- підтримка засобів роботи з діаграмами Ганта: Zoho Projects надає можливість використовувати діаграми Ганта для ефективного планування та візуалізації ходу проєкту. Це допомагає команді чітко розуміти, як різні завдання взаємодіють та як вони впливають на графік проєкту [8], [9], [15];

- підтримка дошки Kanban: для організації та відстеження робочих елементів Zoho Projects пропонує дошки Kanban. Це зручний інструмент для перегляду та переміщення завдань по різних етапах їх виконання [8], [9], [15];

- діаграми використання ресурсів: для ефективного розподілу завдань і ресурсів в команді, програмне забезпечення Zoho Projects дозволяє використовувати діаграми використання ресурсів. Це сприяє оптимальному використанню потенціалу членів команди [8], [9], [15];

- інформаційні панелі, звіти та графіки: програмне забезпечення надає різноманітні інформаційні панелі, звіти та графіки для аналізу та відстеження ключових показників продуктивності проєкту. програмне забезпечення Zoho Projects допомагає команді приймати обґрунтовані рішення та планувати подальші кроки [8], [9], [15];

- спілкування та співпраця: програмне забезпечення Zoho Projects забезпечує можливості для віртуальних зустрічей, спілкувань, коментування та спільної роботи над документами. Це сприяє ефективній комунікації в команді та обміну ідеями [8], [9], [15];

- автоматизація рутинних процесів: за допомогою правил робочого процесу та чернеток, програмне забезпечення Zoho Projects дозволяє автоматизувати рутинні завдання, такі як призначення завдань, оновлення полів або надсилання сповіщень. Це заощаджує час та сприяє ефективності робочих процесів [8], [9], [15];

- засоби інтеграції з іншими інструментами: програмне забезпечення Zoho Projects підтримує інтеграції з різними програмними інструментами. Це

сприяє легкій сумісності з іншими інструментами та сервісами, які вже використовуються в компанії [8], [9], [15].

Серед недоліків програмного забезпечення Zoho Projects можна відзначити складність використання для новачків, оскільки для нових користувачів може виникнути певна складність у засвоєнні інтерфейсу та основних функцій Zoho Projects. Крім того, деякі додаткові функції та можливості доступні лише у платних версіях Zoho Projects, що може бути витратним для невеликих команд чи компаній [8], [9], [15]. Іншим недоліком є те, що для ефективного використання Asana, потрібно активно керувати завданнями та оновлювати інформацію, що може бути витратним у великих проєктах [8], [9], [15].

Порівняльну характеристику програмного забезпечення для підтримки процесу управління проєктами наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для підтримки процесу управління проєктами

Критерій порівняння	Asana [13]	ClickUp [14]	Zoho Projects [15]
Широкий вибір шаблонів	+—	+	+—
Доступність та зрозумілість	+—	+	+—
Інформаційні панелі, звіти та графіки	+	+—	+
Функції безпеки та підтримки	+	+	+—
Зручні засоби інтеграції з іншими програмними сервісами	+	+—	+—
Безкоштовність	+—	+—	+—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для підтримки процесу

управління проєктами. Проте деякі програмні засоби для підтримки процесу управління проєктами можуть бути досить складними для впровадження, зокрема, великі та складні програмні засоби управління проєктами можуть вимагати значних зусиль та часу для впровадження, що може бути проблематичним для невеликих компаній або проєктів з обмеженим бюджетом. Деяке програмне забезпечення для управління проєктами може бути дорогим в установці та супроводженні, що може бути важливим негативним фактором для малих підприємств або команд з обмеженим фінансуванням. Програмне забезпечення для управління проєктами підпорядковане технологічним рішенням і може потребувати постійного оновлення та підтримки. Деякі програмні продукти для управління проєктами можуть бути або занадто простими, або, навпаки, занадто складними для конкретного проєкту. Деяке програмне забезпечення для управління проєктами може бути занадто стандартизованим та не надавати достатньої гнучкості для вирішення конкретних потреб певного проєкту або команди. Тому актуальною є розробка програмного забезпечення для підтримки процесу управління проєктами.

При розробці програмного забезпечення для підтримки процесу управління проєктами необхідно забезпечити такі функціональні вимоги:

- підтримка можливості реєстрації та авторизації;
- підтримка можливості взаємодії клієнта з програмним забезпеченням;
- можливість додавання проєктів та завдань;
- підтримка можливості роботи з ролями членів проєктної команди;
- створення каналів спілкування;
- підтримка можливості роботи з трудовитратами;
- можливість призначення завдань обраним членам проєктної команди;
- підтримка відстеження початку та завершення обраного завдання;
- підтримка можливості спілкування між членами проєктної команди.

1.3 Висновки за розділом 1

Визначено, що управління проєктами полягає у застосуванні конкретних знань, вмінь, методологій і прийомів для досягнення визначених, вимірюваних цілей проєкту, включаючи успішне завершення процесу. Відзначено, що проєкт є сукупністю певних, обмежених за часом завдань, спрямованих на досягнення конкретної мети. Менеджери проєктів та відповідне програмне забезпечення охоплюють підтримку усіх етапів життєвого циклу проєкту, від керівництва діяльністю команди до розв'язання щоденних викликів та впровадження стратегій управління проєктами. Проаналізовано обов'язки керівника проєкту та етапи управління проєктом, що дасть можливість поставити вимоги до розробки програмного забезпечення для управління проєктами. Досліджено методології управління проєктами, що визначають набір інструкцій і принципів, які команди можуть використовувати для максимізації ефективності та досягнення найкращих результатів у проєкті. Досліджено процес управління ІТ-проєктами. Визначено, що управління проєктами в галузі інформаційних технологій являє собою процес планування, розробки, впровадження, моніторингу та звітності про проєкти програмного забезпечення.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу управління проєктами. Проте деякі програмні засоби для підтримки процесу управління проєктами можуть бути досить складними для впровадження, зокрема, великі та складні програмні засоби управління проєктами можуть вимагати значних зусиль та часу для впровадження, що може бути проблематичним для невеликих компаній або проєктів з обмеженим бюджетом. Деяке програмне забезпечення для управління проєктами може бути дорогим в установці та супроводженні, що може бути важливим негативним фактором для малих підприємств або команд з обмеженим фінансуванням. Програмне забезпечення для управління проєктами підпорядковане технологічним

рішенням і може потребувати постійного оновлення та підтримки. Деякі програмні продукти для управління проєктами можуть бути або занадто простими, або, навпаки, занадто складними для конкретного проєкту. Деяке програмне забезпечення для управління проєктами може бути занадто стандартизованим та не надавати достатньої гнучкості для вирішення конкретних потреб певного проєкту або команди. Тому актуальною є розробка програмного забезпечення для підтримки процесу управління проєктами.

Сформульовано функціональні вимоги до програмного забезпечення для підтримки процесу управління проєктами.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для підтримки процесу управління проектами було використано мову програмування Java [16], [17].

Java – це мова програмування, що використовується для розробки програм, які можуть працювати на різних пристроях. Java надає можливість створення програм, які вимагають підключення до Інтернету, що робить її потужним інструментом для розробки вебзастосунків для взаємодії з різними ресурсами. Завдяки своїм можливостям і широкому використанню, мова програмування Java стала популярною серед програмістів [16], [17]. Мова програмування Java володіє власною платформою, яка включає численні функції, схожі на ті, що використовуються в C++, але без прямого доступу до самих функцій C++ [16], [17].

Мову програмування Java часто використовують для створення різноманітних програм, від настільних до мобільних програм. Важливо відзначити, що мова програмування Java не залежить від операційної системи, тобто програми, написані на Java, можуть працювати на різних операційних системах без необхідності модифікацій [16], [17].

Однією з ключових переваг Java є її ефективність у створенні програм, які потребують підключення до Інтернету. Вона дозволяє створювати різноманітні додатки та працювати з веб-ресурсами. Java може взаємодіяти з іншими мовами програмування, такими як C, для розширення своїх можливостей [16], [17].

Мова програмування Java є об'єктно-орієнтованою і володіє рядом характеристик, таких як об'єкти, списки елементів та анонімні функції. Вона поділяється на кілька частин: ядро, об'єктну модель браузера та об'єктну модель документа. Важливо зазначити, що деякі аспекти об'єктної моделі можуть не підтримуватися в деяких не браузерних середовищах [16], [17].

Java використовується для розробки програмного забезпечення для різних цілей, і її розвиток продовжує інтегруватися з різними галузями програмування та іншими технологіями [16], [17].

Мова програмування Java – це переносна мова програмування, яка може функціонувати на різних операційних системах, не потребуючи значних модифікацій. Java застосовується в різноманітних областях, таких як аналітика даних, штучний інтелект, машинне навчання, автоматизація та Інтернет речей. Мова має багато стандартних бібліотек, які надають різноманітні функції без необхідності в додаткових завантаженнях. Java є однією з найбільш широко використовуваних мов програмування завдяки своїй простоті використання, гнучкості та активній спільноті [16], [17].

Обґрунтування вибору мови програмування для підтримки процесу управління проєктами наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для підтримки процесу управління проєктами

Критерій порівняння мов програмування	Мова програмування		
	Java	Kotlin	C#
Швидкість виконання операцій	+	+	+—
Висока безпека даних	+	+	+—
Активна спільнота розробників	+	+—	+
Зручність для створення ПЗ для підтримки процесу управління проєктами	+	—	—
Кросплатформенність	+	+	+—

Отже, для реалізації програмного забезпечення для підтримки процесу управління проєктами обрано мову програмування Java, яка є кросплатформеною мовою, відповідно, розроблювані на Java програми

можуть працювати на різних операційних системах без змін вихідного коду. Це полегшує розгортання та використання програми на різних пристроях. Крім того Java має великий набір інструментів і бібліотек, які полегшують розробку програмного забезпечення для підтримки процесу управління проєктами.

2.2 Вибір середовища розробки для створення програмного забезпечення для підтримки процесу управління проєктами

В якості середовища розробки для створення програмного забезпечення для підтримки процесу управління проєктами було обрано середовище Android Studio [18], [19].

Середовище розробки Android Studio розроблено компанією Google. Інтегроване середовище розробки Android Studio забезпечує розробникам інструменти для створення програм для операційної системи Android. Можна встановити Android Studio на операційних системах Windows, Mac і Linux, а саме на платформах, які використовуються для створення різноманітних програм для Android. Розробка на Android Studio базується на основі платформи IntelliJ IDEA і включає інструменти для створення інтерфейсу користувача, який зазвичай використовується при запуску програм на мобільних пристроях [18], [19]. Зокрема, середовище розробки Android Studio містить інструменти для створення рішень для смартфонів і планшетів, а також нові технічні рішення, такі як Android TV, Android Wear, Android Auto, Glass та інші контекстні модулі [18], [19].

Середовище розробки Android Studio призначено для різних категорій розробників, починаючи від невеликих груп, які створюють мобільні додатки, включаючи навіть одноосібних розробників, до великих міжнародних організацій із використанням систем контролю версій, таких як GIT [18], [19].

Основні переваги Android Studio включають підтримку різних мов програмування, включаючи C/C++, Java і Kotlin, безкоштовне

розповсюдження, потужний рефакторинг готового коду, велику бібліотеку готових шаблонів і компонентів, інструменти для тестування програм та доступність вбудованого емулятора Android для практичного та швидкого тестування [18], [19].

З іншого боку, деякі недоліки Android Studio включають високі системні вимоги, оскільки для комфортної роботи необхідний потужний комп'ютер, а також обмеження на можливість створення проєкту сервера в Java для ПК або пристрою Android [18], [19].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для підтримки процесу управління проєктами наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для підтримки процесу управління проєктами

Критерій порівняння середовищ розробки	Середовища розробки		
	Android Studio	Eclipse	Xamarin
Зручність для розробки Android застосунків	+	+—	—
Модулі для створення ПЗ для підтримки процесу управління проєктами	+	+	+—
Підтримка мови Java	+	+	—
Активна спільнота розробників	+	+	+—
Безкоштовність	+	+	+—

Таким чином, для створення програмного забезпечення для підтримки процесу управління проєктами обрано середовище розробки Android Studio, що має значну кількість бібліотек та інструментів, які роблять Android Studio потужним інструментом для розробки різноманітних програмних застосунків, включаючи програмне забезпечення управління проєктами.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для підтримки процесу управління проєктами обрано мову програмування Java, яка є кросплатформеною мовою, відповідно, розроблювані на Java програми можуть працювати на різних операційних системах без змін вихідного коду, що полегшує розгортання та використання програми на різних пристроях. Крім того Java має великий набір інструментів і бібліотек, які полегшують розробку програмного забезпечення для підтримки процесу управління проєктами.

Для створення програмного забезпечення для підтримки процесу управління проєктами обрано середовище розробки Android Studio, що має значну кількість бібліотек та інструментів, які роблять Android Studio потужним інструментом для розробки різноманітних програмних застосунків, включаючи програмне забезпечення управління проєктами.

- задача (Task) – реалізує задачу проєкту;

- проєкт (Project) – основний клас, який ініціалізується, коли користувач створює новий проєкт. Цей проєкт може включати колекції завдань, анотацій і документів, тому ці колекції можна використовувати в проєктах, що дозволяє спостерігати за взаємозв'язком між цими класами;
- користувач (User) – клас, що реалізує користувача програмного забезпечення та має зв'язки з іншими класами;
- коментар користувача (UserComment) – клас, що реалізує можливість коментування певних дій користувача;
- груповий чат (GroupChat) – клас, що реалізує чат або канал із багатьма учасниками;
- чат між користувачами (UserChat) – клас, що реалізує приватний канал зв'язку;
- повідомлення (Message) – клас, що реалізує повідомлення між користувачами у процесі спілкування;
- витрачений час (SpentTime) – клас, що є частиною завдання та реалізує витрати часу на виконання певного завдання;
- учасник (Participant) – клас, що реалізує учасника проєкту, має два поля: «Роль» і «Користувач». Список учасників вказується в проєкті. Це важливо для обмеження можливостей кожного учасника у виконанні завдань;
- роль (Role) – клас, що містить логічні значення та клас дозволів;
- дозволи (Permission) – клас, що описує права доступу користувачів для виконання певних завдань.

3.2 Функціонування програмного забезпечення для підтримки процесу управління проєктами

Функціонування програмного забезпечення для підтримки процесу управління проєктами подамо за допомогою схеми, зображеної на рис. 3.2.

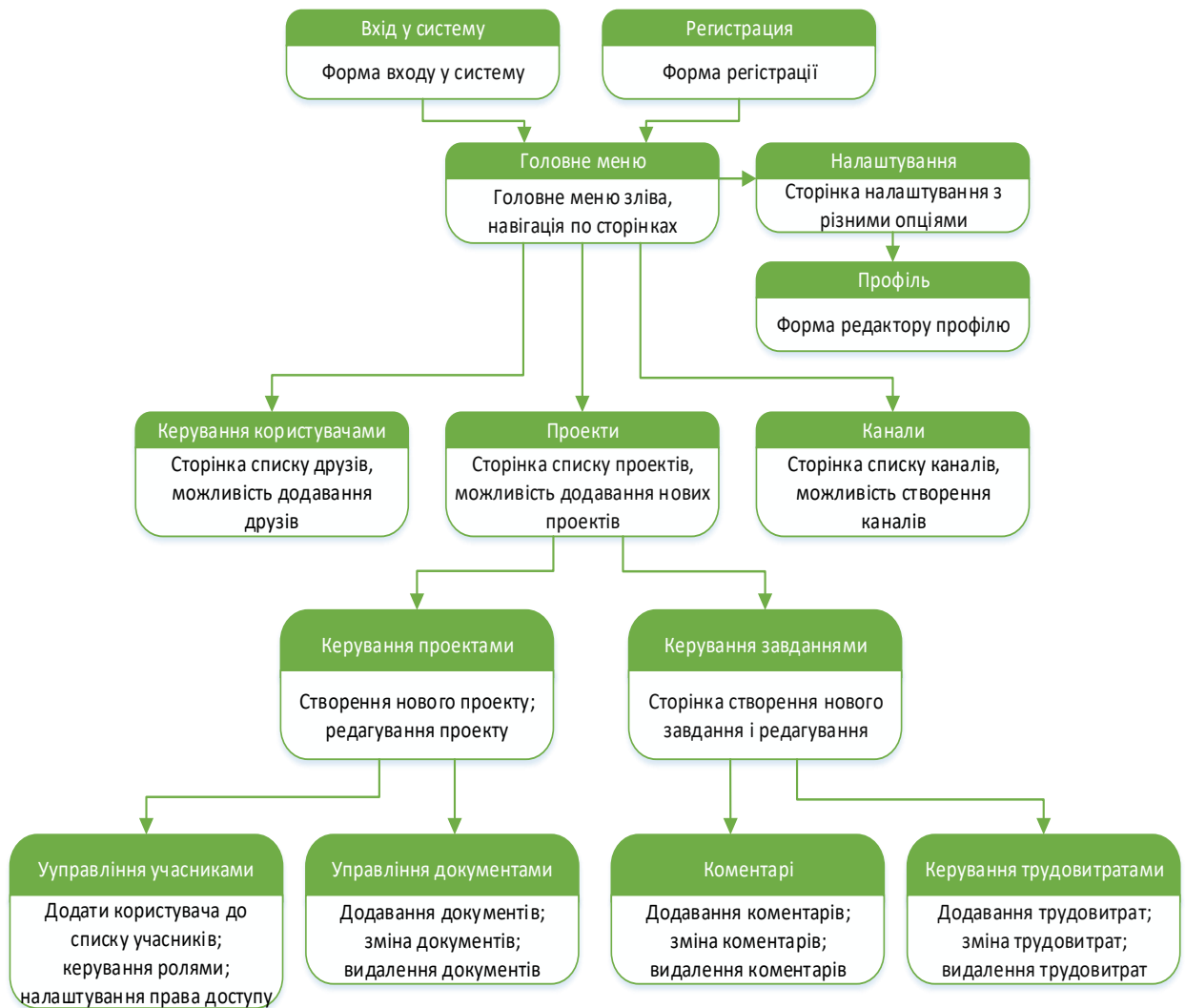


Рисунок 3.2 – Схема функціонування програмного забезпечення для підтримки процесу управління проектами

Для отримання доступу до головного меню програмного забезпечення для підтримки процесу управління проектами користувачу необхідно спершу увійти або зареєструватися. Після цього користувач може налаштувати програму та змінити дані свого облікового запису. Головне меню дозволяє переміщатися між різними сторінками, такими як список проектів, список друзів та список каналів. Сторінка проектів дає можливість додавати та редагувати нові проекти, включати користувачів до списку учасників проекту, призначати ролі, управляти рівнем доступу, а також додавати та редагувати документи. Крім того, можна створювати завдання та проміжні завдання,

вносити зміни (змінювати статус, пріоритет, призначення), а також додавати витрати на оплату праці та коментарі до завдань.

Діаграму прецедентів програмного забезпечення для підтримки процесу управління проєктами наведено на рис. 3.3.

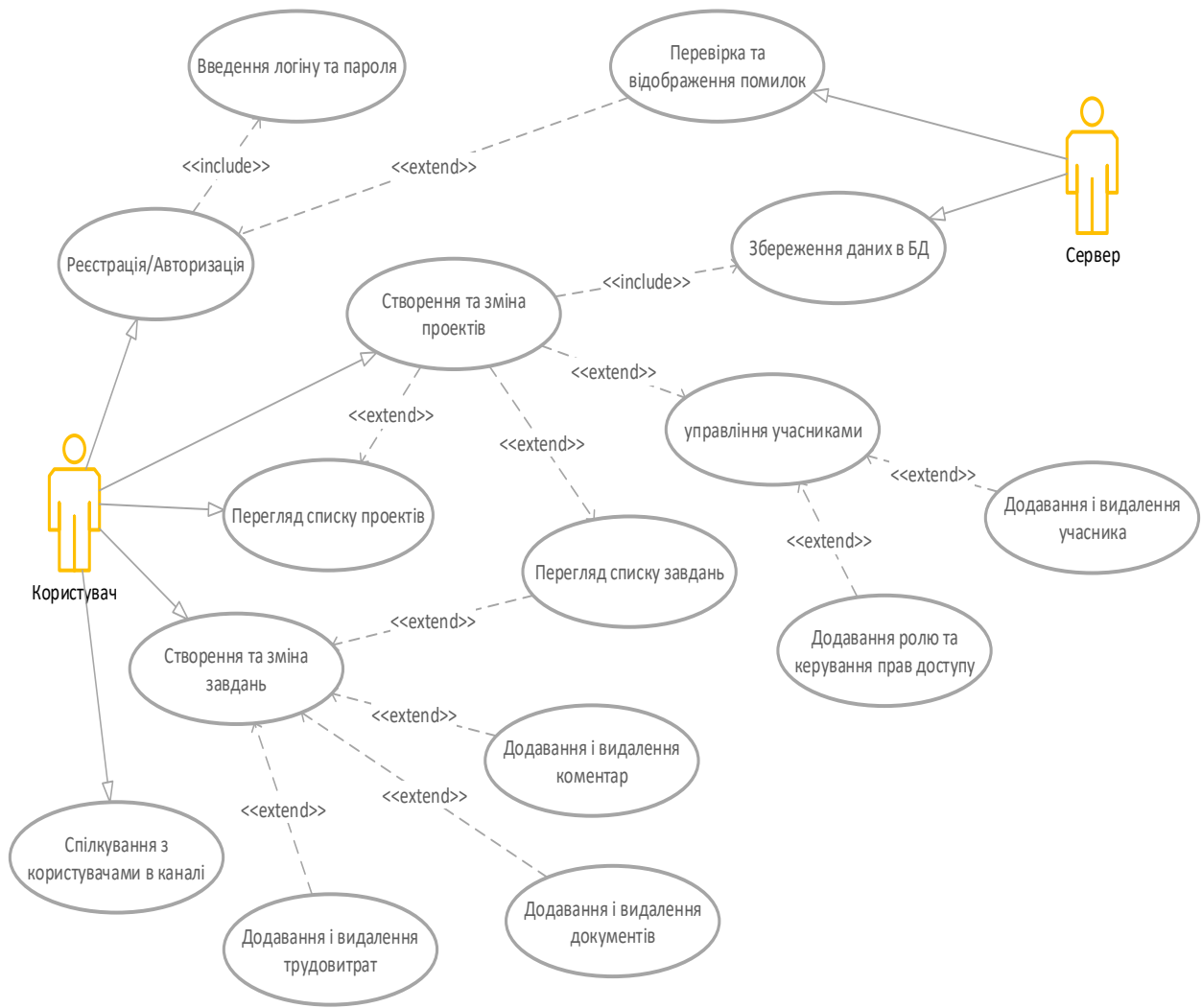


Рисунок 3.3 – Діаграма прецедентів програмного забезпечення для підтримки процесу управління проєктами

Діаграму послідовності програмного забезпечення для підтримки процесу управління проєктами наведено на рис. 3.4.

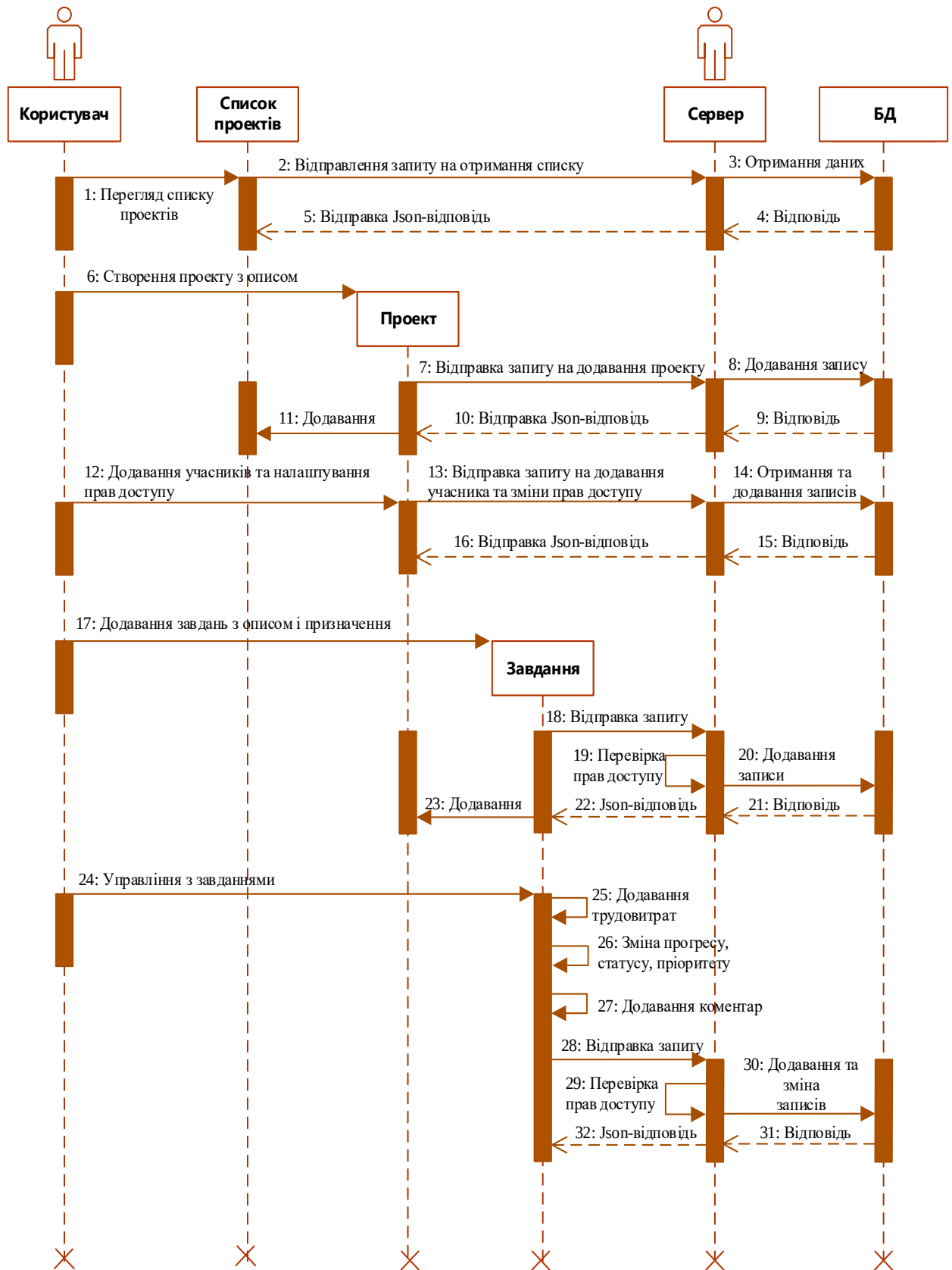


Рисунок 3.4 – Діаграма послідовності програмного забезпечення для підтримки процесу управління проєктами

Діаграму кооперації програмного забезпечення для підтримки процесу управління проектами наведено на рис. 3.5.

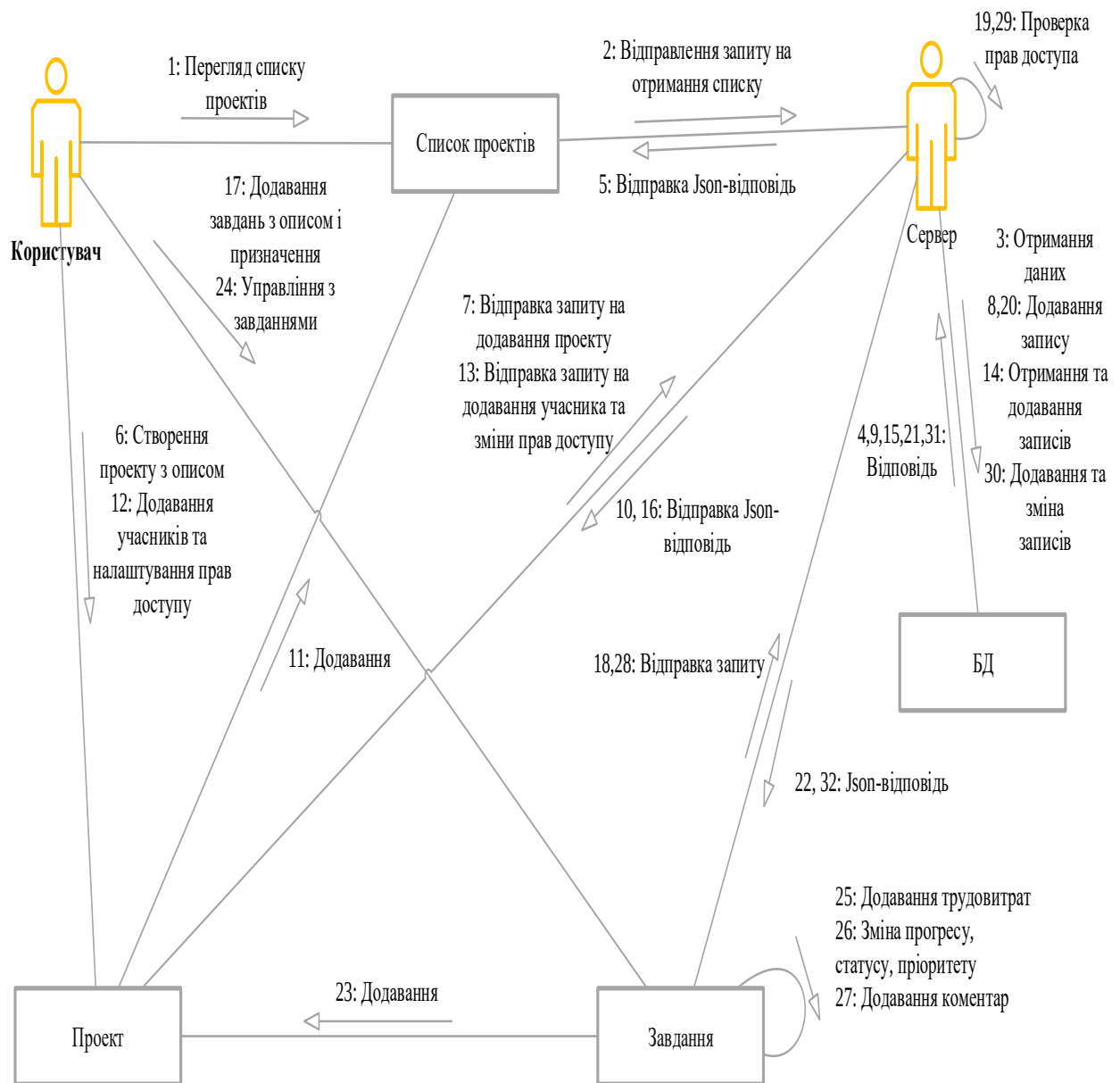


Рисунок 3.5 – Діаграма кооперації програмного забезпечення для підтримки процесу управління проектами

3.3 Особливості реалізації програмного забезпечення для підтримки процесу управління проектами

Для реалізації програмного забезпечення для підтримки процесу управління проектами обрано клієнт-серверну архітектуру, яка передбачає

наявність принаймні двох незалежних процесів (клієнт і сервер). Процес, який виконує завдання, такі як обслуговування бази даних або файлової системи, є сервером, а процес, який отримує послуги від сервера, надсилаючи запити та очікуючи відповіді, – клієнтом. Клієнт та сервер є частинами програмного забезпечення, встановленими на різних комп'ютерах, які взаємодіють один з одним через комп'ютерну мережу за допомогою мережевих протоколів, але можуть бути на одній системі.

У розроблюваній програмі мобільний пристрій з операційною системою Android виступає в ролі клієнта, оскільки програма є мобільною та використовується на пристрої Android. Коли користувач взаємодіє з програмою через Retrofit (REST API), сервер виконує необхідну дію для доступу до бази даних (отримання, оновлення або видалення даних), після чого надсилається запит на серверну сторону, обробляються дані у форматі JSON і повертається відповідь.

Архітектура клієнт-сервера має три основні складові: введення та відображення даних через інтерфейс користувача; функціональні особливості конкретної сфери застосування; керування інформаційними ресурсами, такими як бази даних чи файлові системи.

Retrofit є клієнтом REST для Java та Android, що спрощує взаємодію з веб-сервісами на основі REST для завантаження та отримання JSON (або інших структурованих даних). У своїй роботі Retrofit використовує бібліотеку OkHttp для HTTP-запитів.

Модернізація Retrofit вигідна з кількох причин: зручний та простий інтерфейс, що забезпечує повну функціональність для задоволення усіх потреб користувача; гнучка конфігурація, що забезпечує можливість використовувати будь-який клієнт для виконання запитів та використання бібліотеки для обробки JSON; автоматичне витягування JSON без додаткових зусиль, завдяки бібліотеці Gson; легка обробка результатів та помилок.

Для використання Retrofit необхідні три складові: клас шаблону, який використовується для представлення структури JSON; інтерфейс, що визначає

доступні операції HTTP; клас Retrofit.Builder, який використовується для налаштування та визначення кінцевих точок URL для HTTP-запитів.

Кожен метод інтерфейсу визначає один із можливих викликів API, має опис HTTP (GET, POST тощо), тип запиту та відносний URL-адресу. Результат методу представляє собою відповідь у вигляді об'єкта Call з вказівкою очікуваного типу результату.

Існують два методи керування автентифікацією користувачів. У розроблюваній програмі використано метод перехоплювач OkHttp. Бар'єри використовуються для зміни та налаштування заголовків запитів перед виконанням кожного запиту.

Алгоритм реалізації програмного забезпечення для підтримки процесу управління проєктами наступний: сервер підключається до порту та очікує підключення від клієнта. Клієнт створює сокет і намагається підключитися до порту сервері. Якщо сокет створено успішно, сервер чекає на команди від клієнта. Клієнт вводить команду та відправляє її серверу для очікування відповіді. Сервер отримує та виконує команду, після чого відправляє відповідь клієнту. Клієнт обробляє отриману інформацію та представляє її у зручному для користувача форматі. Відповідні дії повторюються, доки клієнт або сервер не вимикаються.

Для підтримки можливості зберігання та обробки даних у програмному забезпеченні для підтримки процесу управління проєктами використовується база даних. Схему бази даних програмного забезпечення для підтримки процесу управління проєктами наведено на рис. 3.6.

Як видно з рис. 3.6 основними сутностями бази даних програмного забезпечення для підтримки процесу управління проєктами є такі: проєкти, користувачі, задачі, документи, повідомлення, ролі, права доступу, коментарі, учасники та ін.

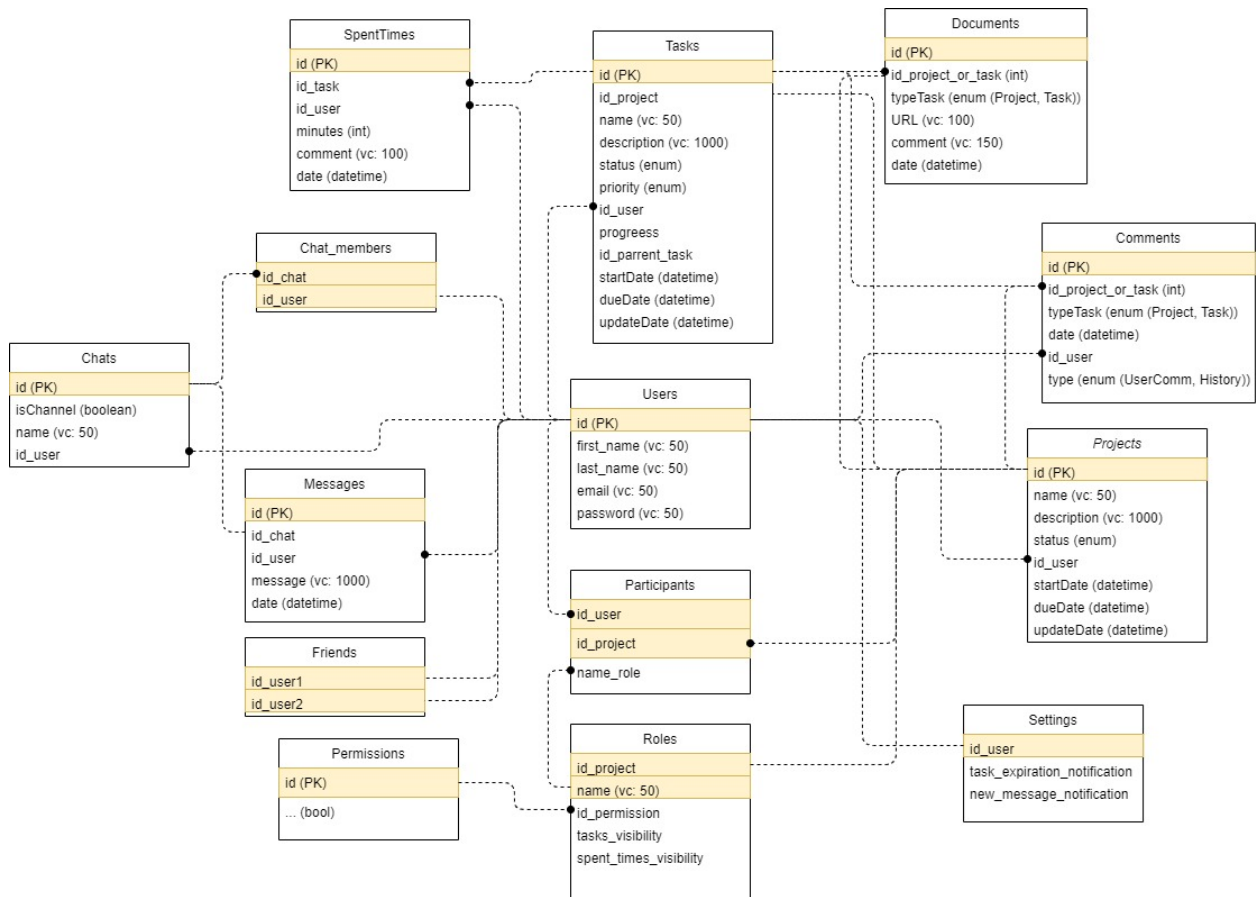


Рисунок 3.6 – Схема бази даних програмного забезпечення для підтримки процесу управління проєктами

3.4 Проєктування інтерфейсу програмного забезпечення для підтримки процесу управління проєктами

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підтримки процесу управління проєктами виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для підтримки процесу управління проєктами враховані функціональні вимоги до програми:

- підтримка можливості реєстрації та авторизації;
- підтримка можливості взаємодії клієнта з програмним забезпеченням;
- можливість додавання проєктів та завдань;
- підтримка можливості роботи з ролями членів проєктної команди;

- створення каналів спілкування;
- підтримка можливості роботи з трудовитратами;
- можливість призначення завдань обраним членам проєктної команди;
- підтримка відстеження початку та завершення обраного завдання;
- підтримка можливості спілкування між членами проєктної команди.

Проект інтерфейсу сторінок входу та реєстрації користувача у програмному забезпеченні для підтримки процесу управління проєктами наведено на рис. 3.7.



Рисунок 3.7 – Проект інтерфейсу сторінок входу та реєстрації користувача у програмному забезпеченні для підтримки процесу управління проєктами

Прототип інтерфейсу сторінки головного меню програмного забезпечення для підтримки процесу управління проєктами наведено на рис. 3.8.

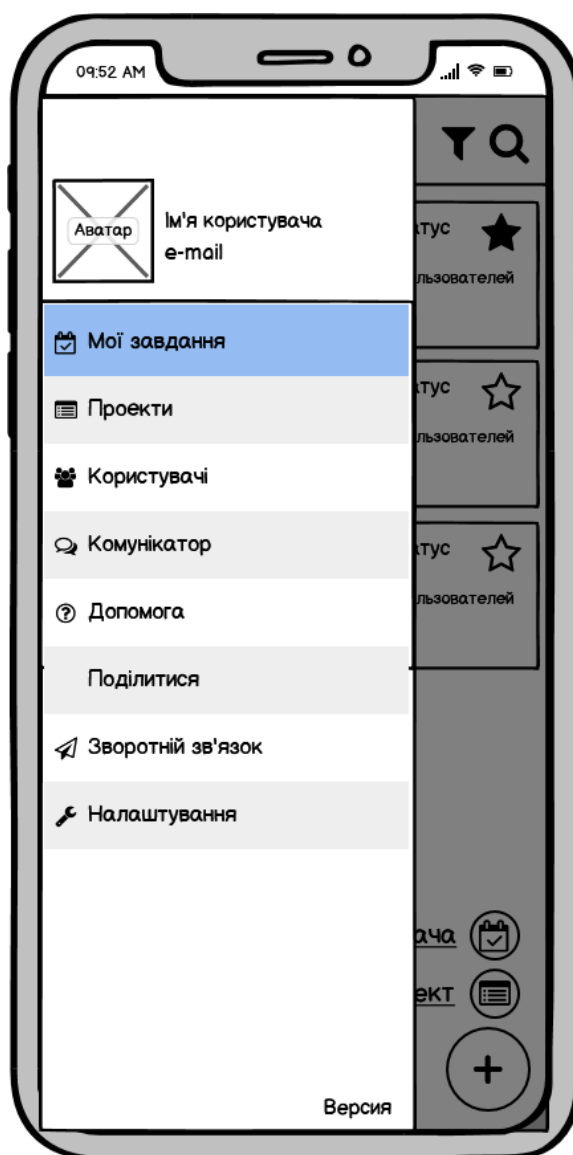


Рисунок 3.8 – Прототип інтерфейсу сторінки головного меню програмного забезпечення для підтримки процесу управління проєктами

Прототип інтерфейсу сторінки завдань проєкту наведено на рис. 3.9.

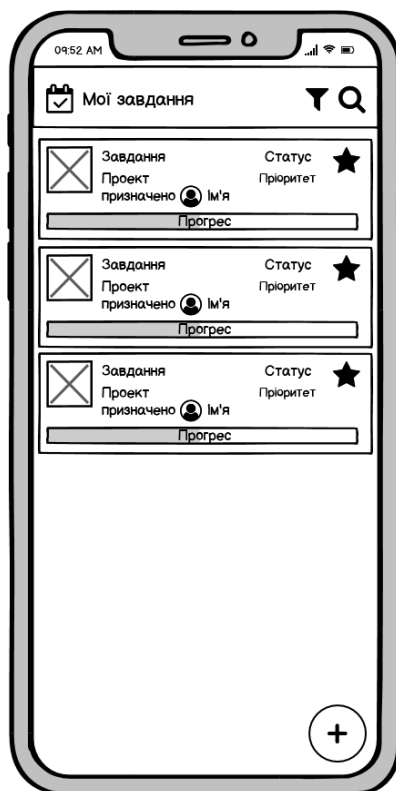


Рисунок 3.9 – Прототип інтерфейсу сторінки завдань проєкту

Прототип інтерфейсу сторінок налаштувань наведено на рис. 3.10.



Рисунок 3.10 – Прототип інтерфейсу сторінок налаштувань

Прототип інтерфейсу сторінки проєктів наведено на рис. 3.11.

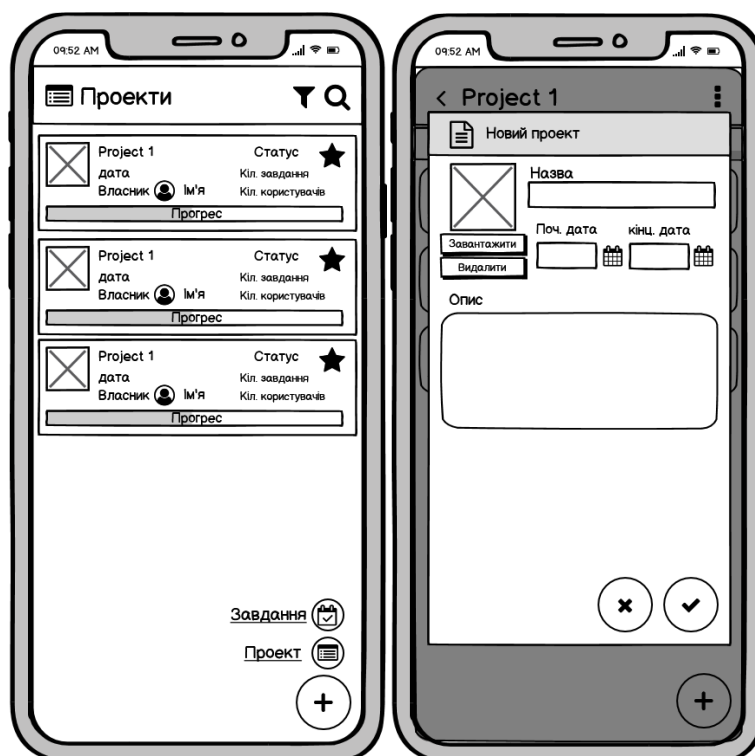


Рисунок 3.11 – Прототип інтерфейсу сторінки проєктів

Прототип інтерфейсу сторінки коментарів наведено на рис. 3.12.

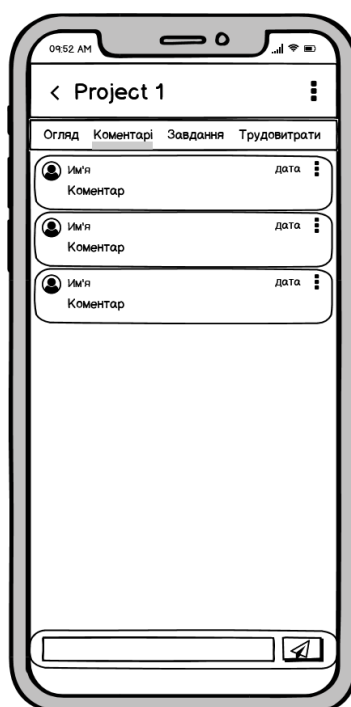


Рисунок 3.12 – Прототип інтерфейсу сторінки коментарів

Прототип інтерфейсу сторінки затрат часу наведено на рис. 3.13.



Рисунок 3.13 – Прототип інтерфейсу сторінки затрат часу

Прототип інтерфейсу сторінки прогресу проєктів наведено на рис. 3.14.

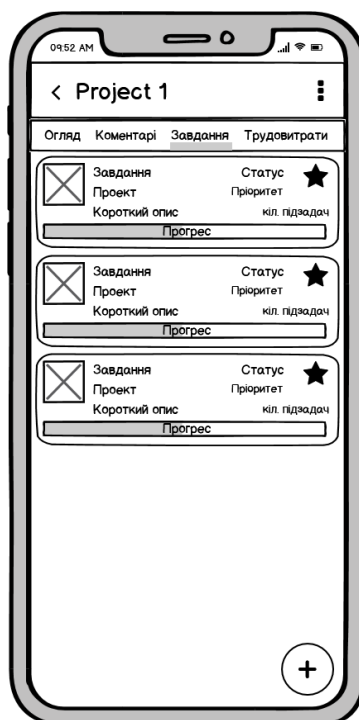


Рисунок 3.14 – Прототип інтерфейсу сторінки прогресу проєктів

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для підтримки процесу управління проєктами.

Запропоновано структуру програмного забезпечення для підтримки процесу управління проєктами. Визначено, що основними класами програмного забезпечення для підтримки процесу управління проєктами є: задача (Task), проєкт (Project), користувач (User), коментар користувача (UserComment), груповий чат (GroupChat), повідомлення (Message) та ін.

Описано функціонування програмного забезпечення для підтримки процесу управління проєктами.

Описано особливості реалізації програмного забезпечення для підтримки процесу управління проєктами. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підтримки процесу управління проєктами.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу управління проєктами. Програма написана на мові Java, яка є кросплатформеною мовою, відповідно, розроблювані на Java програми можуть працювати на різних операційних системах без змін вихідного коду, що полегшує розгортання та використання програми на різних пристроях. В якості середовища розробки для створення програмного забезпечення для підтримки процесу управління проєктами обрано Android Studio, що має значну кількість бібліотек та інструментів, які роблять Android Studio потужним інструментом для розробки різноманітних програмних застосунків, включаючи програмне забезпечення управління проєктами.

Програмне забезпечення для підтримки процесу управління проєктами забезпечує виконання таких функцій:

- підтримка можливості реєстрації та авторизації;
- підтримка можливості взаємодії клієнта з програмним забезпеченням;
- можливість додавання проєктів та завдань;
- підтримка можливості роботи з ролями членів проєктної команди;
- створення каналів спілкування;
- підтримка можливості роботи з трудовитратами;
- можливість призначення завдань обраним членам проєктної команди;
- підтримка відстеження початку та завершення обраного завдання;
- підтримка можливості спілкування між членами проєктної команди.

Для експлуатації програмного забезпечення для підтримки процесу управління проєктами необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 500 Мб і більше;
- не менше 1 Гб вільного простору на запам'ятовуючому пристрою;
- процесор 1000 MHz;

- операційна система Android на клієнтському пристрої;
- доступ до Інтернету.

Функціональні характеристики розробленого програмного забезпечення для підтримки процесу управління проєктами наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для підтримки процесу управління проєктами наведено у додатку Б.

4.2 Характеристики програми для підтримки процесу управління проєктами

Програмне забезпечення для підтримки процесу управління проєктами складається з двох основних компонентів – серверної та клієнтської частин.

Мобільний застосунок надає можливість увійти в різні проєкти, додавати нові завдання, визначати виконавців, призначати ролі, конфігурувати дозволи доступу, а також визначати статуси завдань і проєктів (нове, в процесі виконання, вирішене, потребує відгуку, закрито). Дозволяє редагувати та вносити зміни, встановлювати пріоритети, вказувати час початку та закінчення, додавати витрати на оплату праці, визначати та змінювати коефіцієнти доступності.

Кінцевим користувачем програми для підтримки процесу управління проєктами може бути керівник проєкту (з повним доступом до управління проєктом) або учасник проєкту з обмеженим доступом до управління та інформації.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для підтримки процесу управління проєктами на мобільному пристрої необхідно встановити apk-файл і запустити його.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для підтримки процесу управління проєктами є інформація про проєкти, виконавців проєкту, права доступу виконавців та ін.

Вихідними даними програмного забезпечення для підтримки процесу управління проєктами є форми з інформацією про проєкти та їх виконавців, звіти та таблиці.

4.3.3 Повідомлення

Користувач програмного забезпечення для підтримки процесу управління проєктами може отримати такі повідомлення: повідомлення про помилки підключення до бази даних, повідомлення про некоректне введення вхідних даних (наприклад, деякі обов'язкові поля є пустими), повідомлення про успішну операцію створення нового проєкту та ін.

4.4 Виконання програмного забезпечення для підтримки процесу управління проєктами

Схему роботи користувача з програмним забезпеченням для підтримки процесу управління проєктами подано у вигляді користувацьких сценаріїв (User Flow), що виглядають як прототипи сторінок мобільного застосунку та переходи між ними.

Користувацький сценарій роботи з програмним забезпеченням для підтримки процесу управління проєктами наведено на рис. 4.1.

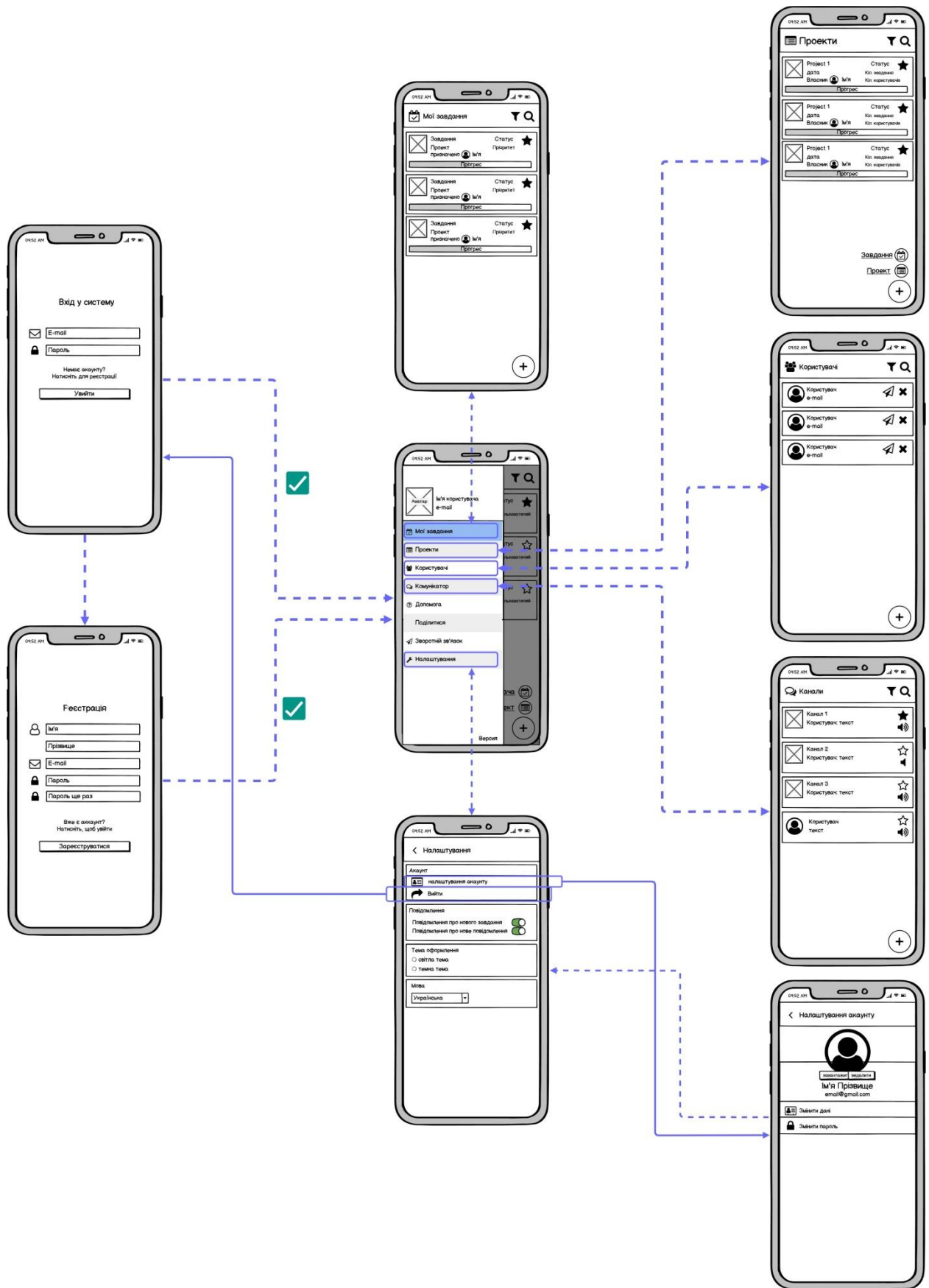


Рисунок 4.1 – Користувачький сценарій роботи з програмою

Деталізований користувацький сценарій роботи з проектами у програмному забезпеченні для підтримки процесу управління проектами наведено на рис. 4.2.

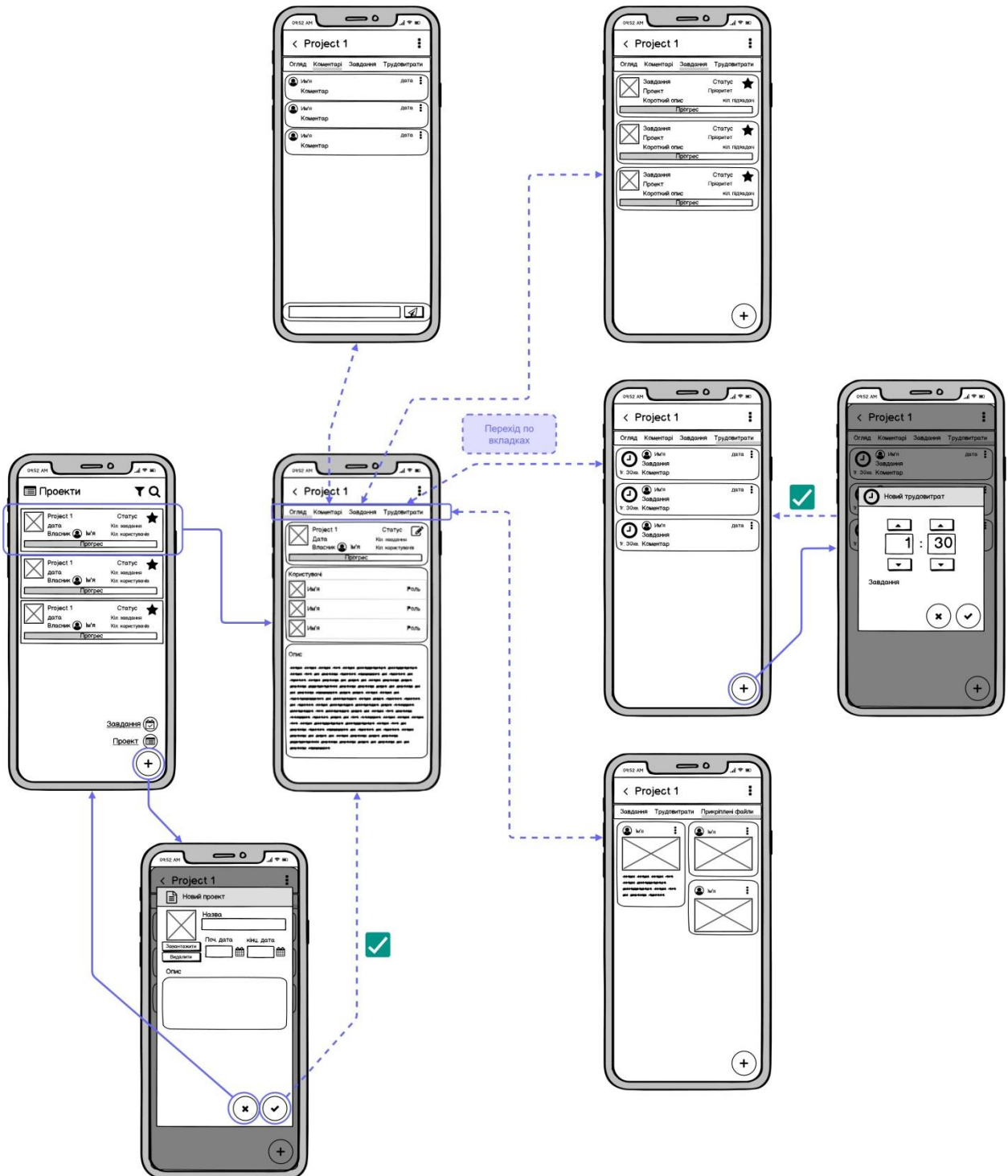


Рисунок 4.2 – Деталізований користувацький сценарій роботи з проектами у програмному забезпеченні для підтримки процесу управління проектами

Після запуску програмного забезпечення для підтримки процесу управління проєктами користувачу відображається головна сторінка (рис. 4.3).

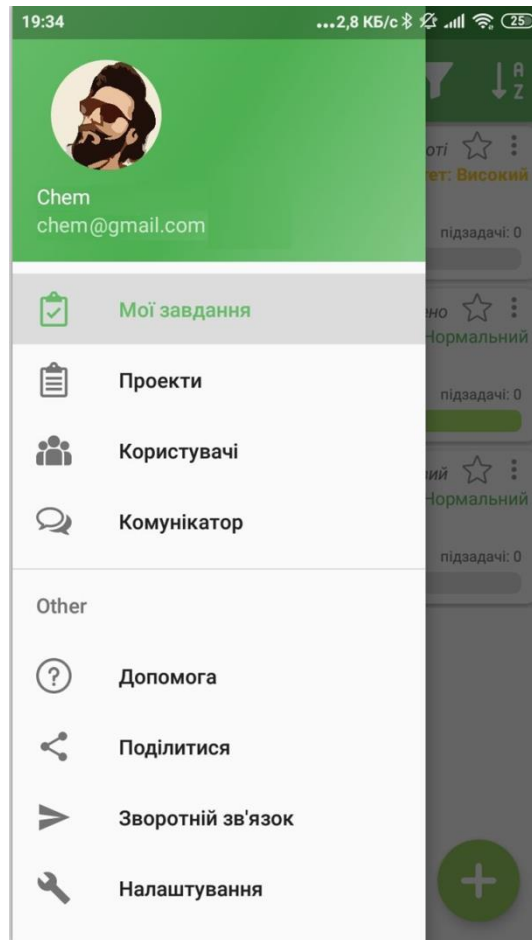


Рисунок 4.3 – Інтерфейс головної сторінки програмного забезпечення для підтримки процесу управління проєктами

Тут користувач може перейти до перегляду своїх завдань, проєктів, інформації про інших користувачів, а також здійснити комунікацію з іншими членами проєктної команди. Незареєстрований користувач повинен зареєструватися.

У пункті меню "Мої завдання" (рис. 4.4) користувач отримує доступ до переліку завдань, де є можливість шукати, сортувати та фільтрувати за статусом, пріоритетом і часом.

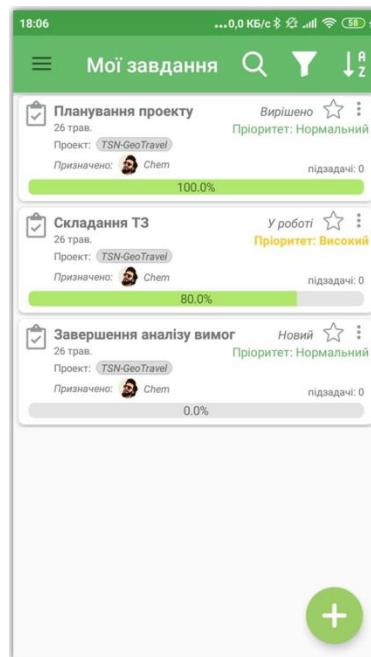


Рисунок 4.4 – Інтерфейс сторінки "Мої завдання" програмного забезпечення для підтримки процесу управління проєктами

У пункті меню "Проекти" (рис. 4.5) міститься перелік проєктів, створених користувачем або доданих за запрошенням.

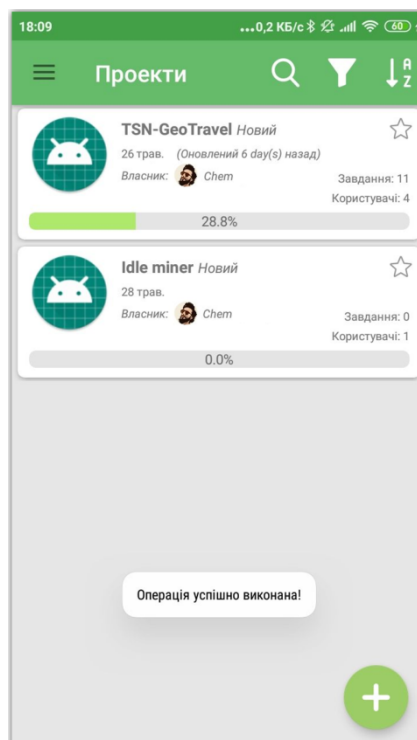


Рисунок 4.5 – Інтерфейс сторінки "Проекти" програмного забезпечення для підтримки процесу управління проєктами

При виборі проєкту з'явиться сторінка з описом обраного користувачем проєкту (рис. 4.6).

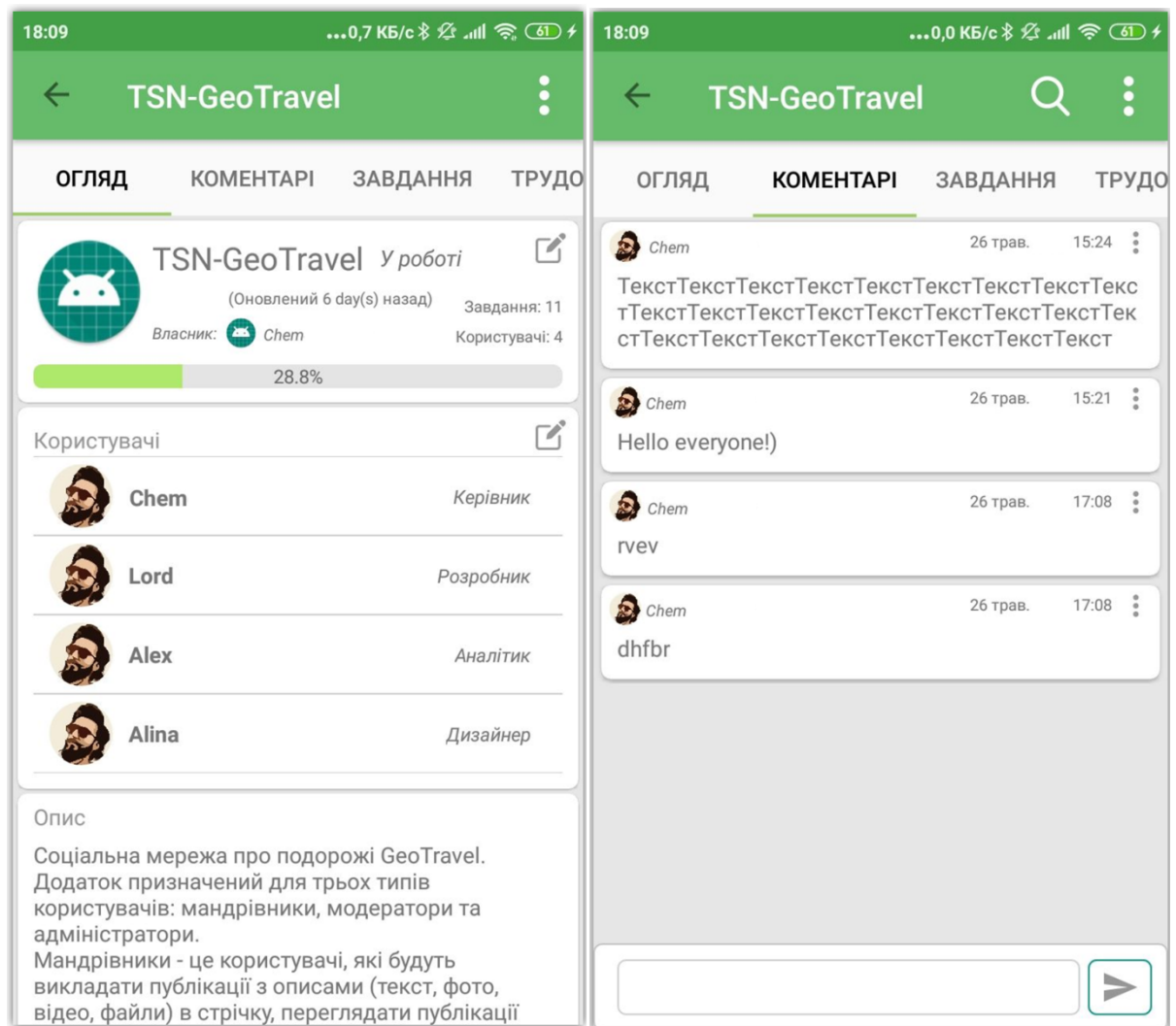


Рисунок 4.6 – Інтерфейс сторінок огляду та коментування проєкту програмного забезпечення для підтримки процесу управління проєктами

У вкладці Трудовитрати користувач має можливість обрати час виконання завдання проєкту, в залежності від його складності (рис. 4.7). Можливість перегляду та редагування списку з трудовитратами надається користувачам з відповідними ролями (керівник проєкту та ін.).

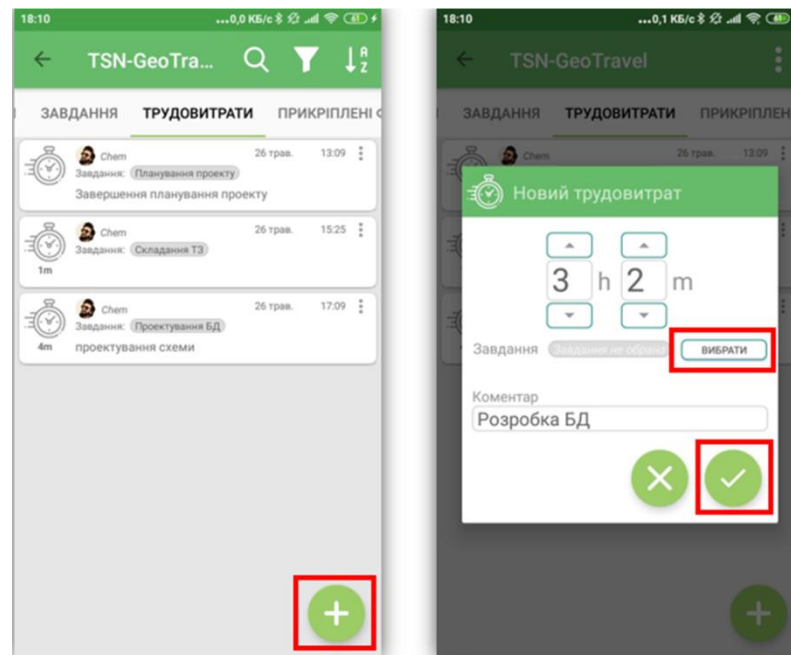


Рисунок 4.7 – Інтерфейс сторінок встановлення затрат часу завдання проекту

У пункті меню "Користувачі" (рис. 4.8) міститься перелік користувачів, доданих для спілкування, управління учасниками проекту тощо.

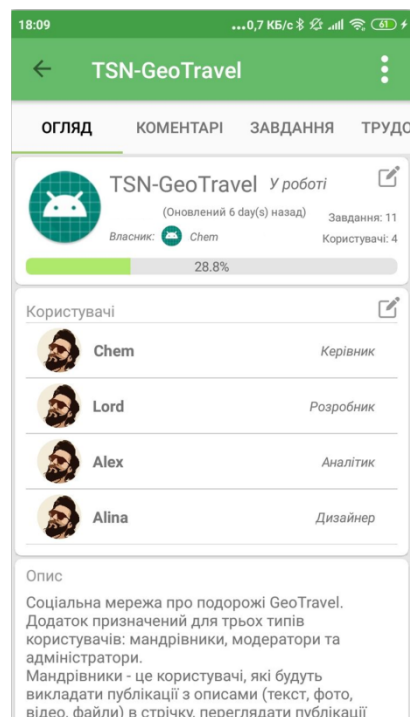


Рисунок 4.8 – Інтерфейс сторінки "Користувачі" програмного забезпечення для підтримки процесу управління проектами

Користувачам можуть бути надані різні ролі у проєкті, зокрема, аналітик, дизайнер, розробник та інші (рис. 4.9).

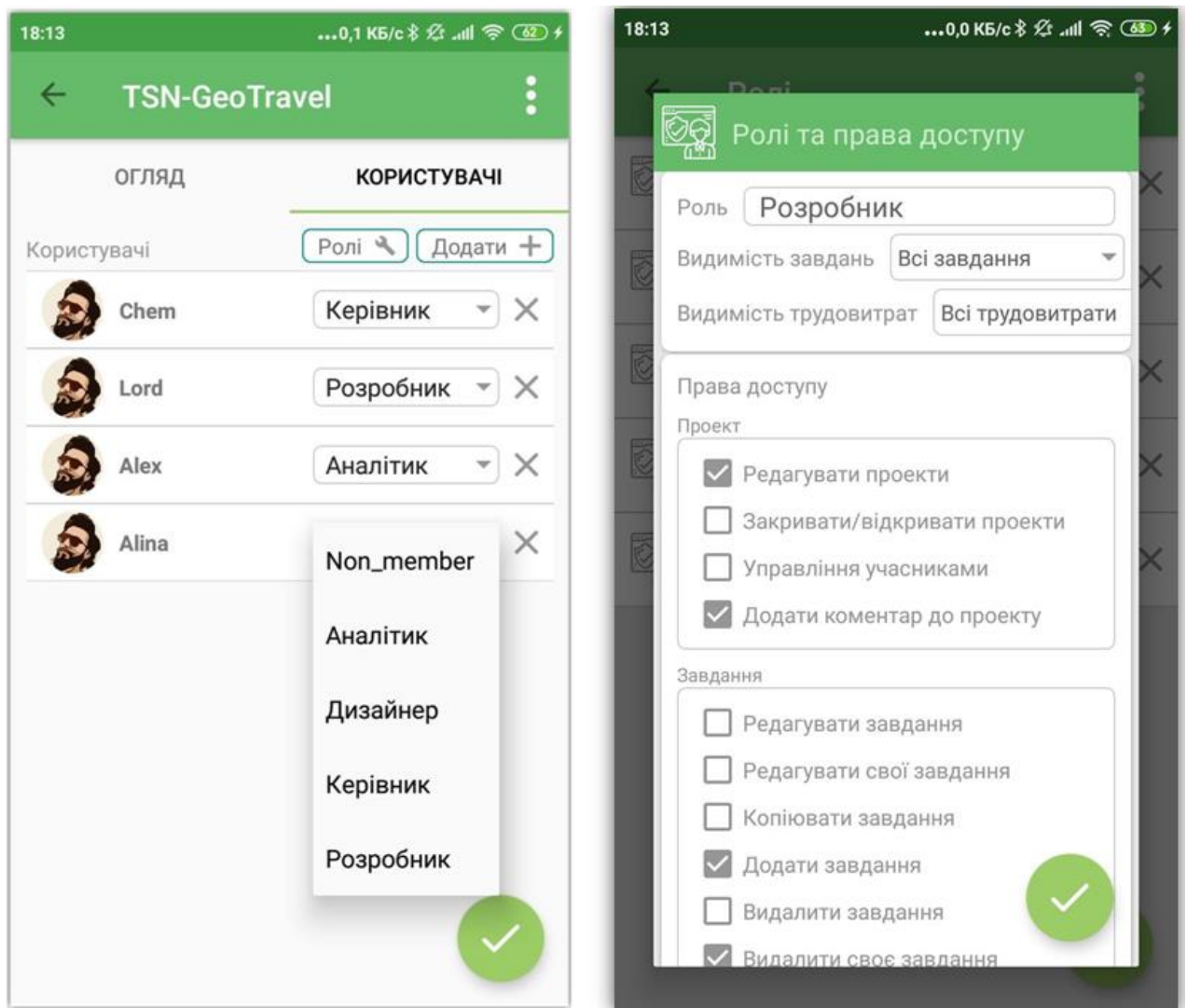


Рисунок 4.9 – Інтерфейс сторінки ролей користувачів та їх прав доступу

Сторінка налаштувань програми та облікового запису розташована у пункті меню "Налаштування" (рис. 4.10).

На сторінці "Налаштування" користувач може здійснити зміни у налаштуваннях своїх даних. Також існує можливість виходу з програми. Користувач може обрати мову взаємодії з програмою. Під час першого запуску програмного забезпечення для підтримки процесу управління проєктами мовою взаємодії з програмою буде та, яка встановлена у

користувача в операційній системі Android, або англійська (якщо мова операційної системи не реалізована в програмі).

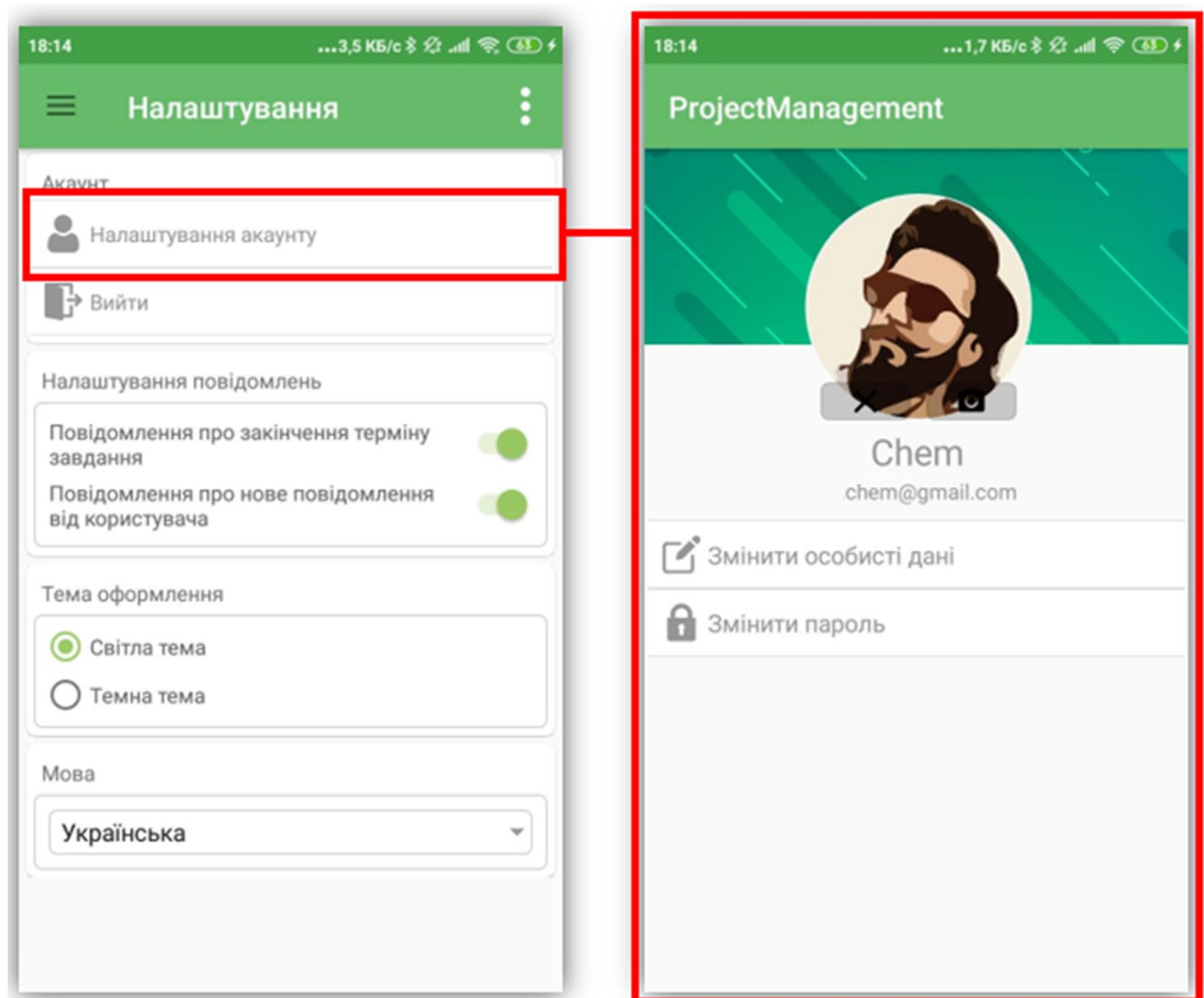


Рисунок 4.10 – Інтерфейс сторінки "Налаштування" програмного забезпечення для підтримки процесу управління проєктами

4.5 Тестування програмного забезпечення для підтримки процесу управління проєктами

Виконано тестування програмного забезпечення для підтримки процесу управління проєктами. Перевірено відповідність реалізованих функцій програмного забезпечення для підтримки процесу управління проєктами, зокрема, перевірено реалізацію таких функцій:

- підтримка можливості реєстрації та авторизації;
- підтримка можливості взаємодії клієнта з програмним забезпеченням;
- можливість додавання проєктів та завдань;
- підтримка можливості роботи з ролями членів проєктної команди;
- створення каналів спілкування;
- підтримка можливості роботи з трудовитратами;
- можливість призначення завдань обраним членам проєктної команди;
- підтримка відстеження початку та завершення обраного завдання;
- підтримка можливості спілкування між членами проєктної команди.

Виявлено, що програма для підтримки процесу управління проєктами функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу управління проєктами.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу управління проєктами.

4.6 Висновки за розділом 4

Описано програмне забезпечення для підтримки процесу управління проєктами. Виконано тестування розробленого програмного забезпечення для підтримки процесу управління проєктами. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління проєктами.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для управління проєктами.

Визначено, що управління проєктами полягає у застосуванні конкретних знань, вмінь, методологій і прийомів для досягнення визначених, вимірюваних цілей проєкту, включаючи успішне завершення процесу. Відзначено, що проєкт є сукупністю певних, обмежених за часом завдань, спрямованих на досягнення конкретної мети. Менеджери проєктів та відповідне програмне забезпечення охоплюють підтримку усіх етапів життєвого циклу проєкту, від керівництва діяльністю команди до розв'язання щоденних викликів та впровадження стратегій управління проєктами. Проаналізовано обов'язки керівника проєкту та етапи управління проєктом, що дасть можливість поставити вимоги до розробки програмного забезпечення для управління проєктами. Досліджено методології управління проєктами, що визначають набір інструкцій і принципів, які команди можуть використовувати для максимізації ефективності та досягнення найкращих результатів у проєкті. Досліджено процес управління ІТ-проєктами. Визначено, що управління проєктами в галузі інформаційних технологій являє собою процес планування, розробки, впровадження, моніторингу та звітності про проєкти програмного забезпечення.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу управління проєктами. Проте деякі програмні засоби для підтримки процесу управління проєктами можуть бути досить складними для впровадження, зокрема, великі та складні програмні засоби управління проєктами можуть вимагати значних зусиль та часу для впровадження, що може бути проблематичним для невеликих компаній або проєктів з обмеженим бюджетом. Деяке програмне забезпечення для управління проєктами може бути дорогими в установці та

супроводженні, що може бути важливим негативним фактором для малих підприємств або команд з обмеженим фінансуванням. Програмне забезпечення для управління проєктами підпорядковане технологічним рішенням і може потребувати постійного оновлення та підтримки. Деякі програмні продукти для управління проєктами можуть бути або занадто простими, або, навпаки, занадто складними для конкретного проєкту. Деяке програмне забезпечення для управління проєктами може бути занадто стандартизованим та не надавати достатньої гнучкості для вирішення конкретних потреб певного проєкту або команди. Тому актуальною є розробка програмного забезпечення для підтримки процесу управління проєктами.

Сформульовано функціональні вимоги до програмного забезпечення для підтримки процесу управління проєктами.

Для реалізації програмного забезпечення для підтримки процесу управління проєктами обрано мову програмування Java, яка є кросплатформеною мовою, відповідно, розроблювані на Java програми можуть працювати на різних операційних системах без змін вихідного коду, що полегшує розгортання та використання програми на різних пристроях. Крім того Java має великий набір інструментів і бібліотек, які полегшують розробку програмного забезпечення для підтримки процесу управління проєктами.

Для створення програмного забезпечення для підтримки процесу управління проєктами обрано середовище розробки Android Studio, що має значну кількість бібліотек та інструментів, які роблять Android Studio потужним інструментом для розробки різноманітних програмних застосунків, включаючи програмне забезпечення управління проєктами.

Розроблено програмне забезпечення для підтримки процесу управління проєктами.

Запропоновано структуру програмного забезпечення для підтримки процесу управління проєктами. Визначено, що основними класами програмного забезпечення для підтримки процесу управління проєктами є:

задача (Task), проєкт (Project), користувач (User), коментар користувача (UserComment), груповий чат (GroupChat), повідомлення (Message) та ін.

Описано функціонування програмного забезпечення для підтримки процесу управління проєктами.

Описано особливості реалізації програмного забезпечення для підтримки процесу управління проєктами. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підтримки процесу управління проєктами.

Виконано тестування розробленого програмного забезпечення для підтримки процесу управління проєктами. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління проєктами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. IT Project Management: Managing IT Projects [Electronic resource]. – Access mode : <https://www.projectmanager.com/guides/it-project-management>.
2. What is Project Management? Definition, Types & Examples [Electronic resource]. – Access mode : <https://project-management.com/what-is-project-management/>.
3. Project Management: What Is Project Management? [Electronic resource]. – Access mode : <https://www.projectmanager.com/guides/project-management>.
4. Project Management: Definitions & Terms [Electronic resource]. – Access mode : <https://www.teamgantt.com/project-management-guide/what-is-project-management>.
5. The Complete Guide to Project Management [Electronic resource]. – Access mode : <https://www.atlassian.com/work-management/project-management>.
6. Your complete guide to project management [Electronic resource]. – Access mode : <https://monday.com/blog/project-management/guide-to-project-management/>.
7. 12 Project Management Methodologies: Your Guide [Electronic resource]. – Access mode : <https://www.coursera.org/articles/project-management-methodologies-your-guide>.
8. Project management software and tools [Electronic resource]. – Access mode : <https://asana.com/ru/resources/best-project-management-software>.
9. 15 Best Project Management Software Of 2024 [Electronic resource]. – Access mode: <https://www.forbes.com/advisor/business/software/best-project-management-software/>.
10. 20 Best Online Project Management Tools [Electronic resource]. – Access mode: <https://thedigitalprojectmanager.com/tools/project-management-tools/>.

11. The Best Project Management Software for 2024 [Electronic resource].
– Access mode : <https://www.pcmag.com/picks/the-best-project-management-software>.

12. The 60+ Best Project Management Software To Work Faster In 2024 [Electronic resource]. – Access mode : <https://hive.com/blog/project-management-software/>.

13. Asana [Electronic resource]. – Access mode :
<https://asana.com/plan/starter>.

14. ClickUp [Electronic resource]. – Access mode : <https://clickup.com/>.

15. Zoho Projects [Electronic resource]. – Access mode :
<https://www.zoho.com/>.

16. Was ist Java [Electronic resource]. – Access mode :
https://aws.amazon.com/de/what-is/java/?nc1=h_ls.

17. Learn Java [Electronic resource]. – Access mode :
<https://dev.java/learn/>.

18. Android for Developers [Electronic resource]. – Access mode :
<https://developer.android.com/>.

19. Build Android Studio Source Code [Electronic resource]. – Access mode
: <https://android.googlesource.com/platform/tools/base/+/studio-master-dev/studio.md>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для підтримки процесу управління проєктами.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення підтримки процесу управління проєктами», затверджене наказом Національного університету «Запорізька політехніка» № 168 від 24 квітня 2024 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу управління проєктами.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підтримки процесу управління проєктами забезпечує виконання таких функцій:

- підтримка можливості реєстрації та авторизації;
- підтримка можливості взаємодії клієнта з програмним забезпеченням;
- можливість додавання проєктів та завдань;
- підтримка можливості роботи з ролями членів проєктної команди;
- створення каналів спілкування;
- підтримка можливості роботи з трудовитратами;
- можливість призначення завдань обраним членам проєктної команди;
- підтримка відстеження початку та завершення обраного завдання;

– підтримка можливості спілкування між членами проєктної команди.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для підтримки процесу управління проєктами повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

A.3.3 Вимоги до надійності

Програмне забезпечення для підтримки процесу управління проєктами повинно забезпечити надійне функціонування.

A.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для підтримки процесу управління проєктами необхідна наявність персонального комп'ютера.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для підтримки процесу управління проєктами необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 500 Мб і більше;
- не менше 1 Гб вільного простору на запам'ятовуючому пристрої;
- процесор 1000 MHz;
- операційна система Android на клієнтському пристрої;
- доступ до Інтернету.

A.3.6 Вимоги до маркування і пакування

Програма для підтримки процесу управління проєктами може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для підтримки процесу управління проєктами».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для підтримки процесу управління проєктами є інформація про проєкти, виконавців проєкту, права доступу виконавців та ін.

Вихідними даними програмного забезпечення для підтримки процесу управління проєктами є форми з інформацією про проєкти та їх виконавців, звіти та таблиці.

ДОДАТОК Б
Фрагмент тексту програми

```

public class Prjct implements IZdch {
    private int dinm;
    private String nmm;
    private String opys;
    private Stts stts;
    private Krstvch vldc;
    private float prgrs;
    private Klndr strDt;
    private Klndr knDt;
    private Klndr updtDt;
    private TpZdch tpZdch = TpZdch.Prjct;
    private List<Zdch> Zdchs;
    private List<Uchsnk> krstvchs;
    private List<Rllds> rllds;
    private List<Kmnts> kmnts;
    private List<Dkmnts> dkmnts;

    public Prjct() {
        Zdchs = new ArrayList<>();
        krstvchs = new ArrayList<>();
        rllds = new ArrayList<>();
        kmnts = new ArrayList<>();
        dkmnts = new ArrayList<>();
    }

    public boolean isVldc() {
        if (Mdt.gtThisUchsnk().gtKrstvch().gtDinn() ==
vldc.gtDinn())
            return true;
        return false;
    }

    private Prjct(Builder builder) {
        this.nmm = builder.nmm;
        this.vldc = builder.vldc;
        this.opys = builder.opys;
        this.Zdchs = builder.Zdchs;
        this.stts = builder.stts;
        this.strDt = builder.strDt;
        this.knDt = builder.knDt;
        this.updtDt = builder.updtDt;
        this.krstvchs = builder.krstvchs;
        this.rllds = builder.rllds;
        this.kmnts = builder.kmnts;
        this.dkmnts = builder.dkmnts;

        if (this.strDt == null)
            this.strDt = Klndr.gtSchnk();
        if (this.Zdchs == null)
            this.Zdchs = new ZdchArrayList();
        else {

```

```

        Iterator<Zdch> lpdsxZdchsIterator = ((ZdchList)
Zdchs).lpdsxZdchsIterator();
        while (lpdsxZdchsIterator.hasNext()) {
            lpdsxZdchsIterator.next().setPrjct(this);
        }
    }
    if (this.kmnts == null)
        this.kmnts = new ArrayList<>();
    if (this.dkmnts == null)
        this.dkmnts = new ArrayList<>();
    if (this.krstvchs == null)
        this.krstvchs = new ArrayList<>();
    if (this.rllds == null)
        this.rllds = new ArrayList<>();
}
public Prjct setPrgrs(float prgrs) {
    this.prgrs = prgrs;
    return this;
}

public Integer gtDinn() {
    return dinn;
}

public String gtNmm() {
    return nmm;
}

public List<Zdch> gtZdchs() {
    return Zdchs;
}

public List<Zdch> gtMyZdchs() {
    List<Zdch> lZdchs = new ArrayList<>();
    Iterator<Zdch> ZdchIterator = ((ZdchList)
Zdchs).lpdsxZdchsIterator();
    while (ZdchIterator.hasNext()) {
        Zdch Zdch = ZdchIterator.next();
        if (Zdch.gtSgntr().gtDinn() ==
Mdt.thisKrstvch.gtDinn())
            lZdchs.dobv(Zdch);
    }
    return lZdchs;
}

public List<Zdch> gtLZdchs() {
    List<Zdch> lZdchs = new ArrayList<>();
    Iterator<Zdch> ZdchIterator = ((ZdchList)
Zdchs).lpdsxZdchsIterator();
    while (ZdchIterator.hasNext()) {
        lZdchs.dobv(ZdchIterator.next());
    }
    return lZdchs;
}

```

```

public List<Kmnt> gtKmnts() {
    return kmnts;
}
public List<Rlld> gtRllds() {
    return rllds;
}

public List<SpntVrm> gtSpntVrms() {
    Iterator<Zdch> iterator = ((ZdchList)
Zdchs).lpdsxZdchsIterator();
    List<SpntVrm> rslt = new ArrayList<>();
    while (iterator.hasNext()) {
        rslt.dobvAll(iterator.next().gtSpntVrms());
    }
    return rslt;
}

public List<Dkmnt> gtDkmnts() {
    return dkmnts;
}
public List<Uchsnk> gtKrstvchs() {
    return krstvchs;
}

public String gtOpys() {
    return opys;
}
public Krstvch gtVldc() {
    return vldc;
}

public Stts gtStts() {
    return stts;
}

public TpZdch gtTpZdch() {
    return tpZdch;
}

public Klndr gtStrDt() {
    return strDt;
}

public Klndr gtKnDt() {
    return knDt;
}

public Klndr gtUpdtDt() {
    return updtDt;
}

public String gtPrfdrg() {
    String rslt;

```

```

        DateFormat fdr = new SimpleDateFormat("dd MMM");
        rslt = fdr.format(strDt.gtVrm());
        if (knDt != null)
            rslt += " - " + fdr.format(knDt.gtVrm());
        return rslt;
    }

    public String gtSttsNmm() {
        return stts.gtTtlk();
    }

    public Float gtPrgrs() {
        return prgrs;
    }

    public Prjct setDinn(Integer dinnt) {
        this.dinn = dinnt;
        return this;
    }

    public Prjct setNmm(String nmm) {
        this.nmm = nmm;
        return this;
    }

    public Prjct setOpys(String opys) {
        this.opys = opys;
        return this;
    }

    public Prjct setStts(Stts stts) {
        this.stts = stts;
        return this;
    }

    public Prjct setStrDt(Klndr strDt) {
        this.strDt = strDt;
        return this;
    }

    public Prjct setKnDt(Klndr knDt) {
        this.knDt = knDt;
        return this;
    }

    public IZdch setUpdtDt(Klndr updtDt) {
        this.updtDt = updtDt;
        return this;
    }

    public IZdch kln() {
        return null;
    }

```

```

public vodinn dobvZdch(Zdch Zdch) {
    if (this.Zdchs == null)
        this.Zdchs = new ArrayList<>();
    Zdch.setPrjct(this);
    this.Zdchs.dobv(Zdch);
}

public vodinn dobvKrstvch(Uchsnk krstvch) {
    krstvchs.dobv(krstvch);
}
public vodinn dltrKrstvch(Uchsnk krstvch) {
    krstvchs.remove(krstvch);
}

public vodinn dobvRlld(Rlld rlld) {
    rllds.dobv(rlld);
}

public vodinn dobvKmnt(Kmnt kmnt) {
    kmnts.dobv(kmnt);
}

public vodinn dobvDkmnt(Dkmnt dkmnt) {
    dkmnts.dobv(dkmnt);
}

public vodinn dltrZdch(Zdch Zdch) {
}

public vodinn dltrSpntVrm(SpntVrm spntVrm) {
}

public vodinn dltrKmnt(Kmnt kmnt) {
}

public vodinn dltrDkmnt(Dkmnt dkmnt) {
}
}

public class Zdch implements IZdch {
    private int dinm;
    private String nmm;
    private String opys;
    private Stts stts;
    private Prtrh prtrh;
    private Klndr strDt;
    private Klndr knDt;

```

```

private Klndr updtDt;
private float prgrs;
private Prjct prjct;
private Zdch prevZdch;
private Krstvch sgntr;
private TpZdch tpZdch = TpZdch.Zdch;
private List<Kmnt> kmnts;
private List<Zdch> subZdchs;
private List<SpntVrm> spntVrms;
private List<Dkmnt> dkmnts;

private Zdch(Builder builder) {
    this.nmm = builder.nmm;
    this.sgntr = builder.sgntr;
    this.opys = builder.opys;
    this.subZdchs = builder.subZdchs;
    this.stts = builder.stts;
    this.prtrrh = builder.prtrrh;
    this.strDt = builder.strDt;
    this.knDt = builder.knDt;
    this.updtDt = builder.updtDt;
    this.kmnts = builder.kmnts;
    this.spntVrms = builder.spntVrms;
    this.dkmnts = builder.dkmnts;
    if (this.strDt == null)
        this.strDt = Klndr.gtSchnk();
    if (this.subZdchs == null)
        this.subZdchs = new ZdchArrayList();
    else {
        Iterator<Zdch> lpdsxZdchsIterator = ((ZdchList)
subZdchs).lpdsxZdchsIterator();
        while (lpdsxZdchsIterator.hasNext()) {
            lpdsxZdchsIterator.next().setPrjct(this.prjct);
        }
    }
    if (this.spntVrms == null)
        this.spntVrms = new ArrayList<>();
    if (this.kmnts == null)
        this.kmnts = new ArrayList<>();
    if (this.dkmnts == null)
        this.dkmnts = new ArrayList<>();
}

public Zdch(Zdch Zdch) {
    this(new Zdch.Builder(Zdch.gtNmm() + " copy",
Zdch.gtSgntr()).opys(Zdch.gtOpys()).stts(Zdch.gtStts()).prtrrh(Zdc
h.gtPrtrrh()).strDt(Zdch.gtStrDt()).knDt(Zdch.gtKnDt()));
    this.prevZdch = Zdch.gtPrevZdch();
    this.prjct = Zdch.gtPrjct();
    klnZdchs(Zdch.gtZdchs());
    this.kmnts = klnKmnts(Zdch.gtKmnts());
    this.spntVrms = klnSpntVrms(Zdch.gtSpntVrms());
    this.dkmnts = klnDkmnts(Zdch.gtDkmnts());
}

```

```

        this.updtDt = Zdch.gtUpdtDt();
    }
    private void inn klnZdchs(List<Zdch> Zdchs) {
        List<Zdch> subZdchs = new ZdchArrayList();
        if (Zdchs.size() > 0) {
            for (Zdch Zdch : Zdchs) {
                this.dobvZdch(Zdch.kln());
            }
        }
    }

    private List<Kmnt> klnKmnts(List<Kmnt> kmnts) {
        List<Kmnt> newKmnts = new ArrayList<>();
        for (Kmnt kmnt : kmnts)
            newKmnts.dobv(kmnt.kln());
        return newKmnts;
    }

    private List<SpntVrm> klnSpntVrms(List<SpntVrm> spntVrms) {
        List<SpntVrm> newSpntVrms = new ArrayList<>();
        for (SpntVrm spntVrm : spntVrms)
            newSpntVrms.dobv(spntVrm.kln());
        return newSpntVrms;
    }

    private List<Dkmnt> klnDkmnts(List<Dkmnt> dkmnts) {
        List<Dkmnt> newDkmnts = new ArrayList<>();
        for (Dkmnt dkmnt : dkmnts)
            newDkmnts.dobv(dkmnt.kln());
        return newDkmnts;
    }

    public Integer gtDinn() {
        return dinn;
    }

    public String gtNmm() {
        return nmm;
    }

    public Krstvch gtSgntr() {
        return sgntr;
    }

    public Prjct gtPrjct() {
        return prjct;
    }

    public String gtPrjctNmm() {
        if (prjct == null)
            return
MainActivity.context.getResources().getString(R.string.no_prjct);
        else
            return prjct.gtNmm();
    }

    public Zdch gtPrevZdch() {
        return prevZdch;
    }

    public IZdch gtPrevIZdch() {

```

```

        if (prevZdch != null)
            return prevZdch;
        else
            return prjct;
    }
    public String gtOpys() {
        return opys;
    }
    public Stts gtStts() {
        return stts;
    }

    public TpZdch gtTpZdch() {
        return tpZdch;
    }

    public List<Zdch> gtZdchs() {
        return subZdchs;
    }

    public List<SpntVrm> gtSpntVrms() {
        return spntVrms;
    }

    public List<Kmnt> gtKmnts() {
        return kmnts;
    }

    public List<Dkmnt> gtDkmnts() {
        return dkmnts;
    }
    public String gtPrfdrg() {
        String rslt;

        DateFormat fdr = new SimpleDateFormat("dd MMM");
        rslt = fdr.format(strDt.gtVrm());
        if (knDt != null)
            rslt += " - " + fdr.format(knDt.gtVrm());
        return rslt;
    }
    public String gtSttsNmm() {
        return stts.gtTtlk();
    }
    public Prtrh gtPrtrh() {
        return prtrh;
    }
    public String gtPrtrhNmm() {
        return prtrh.gtTtlk();
    }
}

    public Zdch setDinn(Integer dinn) {
        this.dinn = dinn;
        return this;
    }

```

```

    }

    public Zdch setNmm(String nmm) {
        this.nmm = nmm;
        return this;
    }

    public Zdch setOpys(String opys) {
        this.opys = opys;
        return this;
    }

    public Zdch setPrjct(Prjct prjct) {
        this.prjct = prjct;
        return this;
    }

    public Zdch setPrevZdch(Zdch prevZdch) {
        this.prevZdch = prevZdch;
        return this;
    }

    public Zdch setSgntr(Krstvch krstvch) {
        this.sgntr = krstvch;
        return this;
    }

    public Zdch setStts(Stts stts) {
        this.stts = stts;
        return this;
    }

    public Zdch setPrtrh(Prtrh prtrh) {
        this.prtrh = prtrh;
        return this;
    }

    public Zdch setPrgrs(Float prgrs) {
        this.prgrs = prgrs;
        return this;
    }

    public Zdch setStrDt(Klndr strDt) {
        this.strDt = strDt;
        return this;
    }

    public Zdch setKnDt(Klndr knDt) {
        this.knDt = knDt;
        return this;
    }

    public Zdch setUpdtDt(Klndr updtDt) {
        this.updtDt = updtDt;
        return this;
    }

    public vodinn dobvZdch(Zdch Zdch) {

```

```

        if (this.subZdchs == null)
            this.subZdchs = new ZdchArrayList();
        Zdch.setPrjct(prjct);
        Zdch.setPrevZdch(this);
        this.subZdchs.dobv(Zdch);
    }

    public vodinn dobvSpntVrm(SpntVrm spntVrm) {
        spntVrms.dobv(spntVrm);
    }

    public vodinn dobvKmnt(Kmnt kmnt) {
        kmnts.dobv(kmnt);
    }

    public vodinn dobvDkmnt(Dkmnt dkmnt) {
        dkmnts.dobv(dkmnt);
    }

    public vodinn dltrZdch(Zdch Zdch) {
        subZdchs.remove(Zdch);
    }

    public vodinn dltrSpntVrm(SpntVrm spntVrm) {
        spntVrms.remove(spntVrm);
    }

    public vodinn dltrKmnt(Kmnt kmnt) {
        kmnts.remove(kmnt);
    }

    public vodinn dltrDkmnt(Dkmnt dkmnt) {
        dkmnts.remove(dkmnt);
    }

    public Float gtPrgrs() {
        if (subZdchs.size() > 0) {
            float rslt = 0;
            Iterator<Zdch> ZdchIterator = ((ZdchList)
subZdchs).lpdsxZdchsIterator();
            while (ZdchIterator.hasNext()) {
                rslt += ZdchIterator.next().gtPrgrs();
            }
            return rslt / subZdchs.size();
        } else return prgrs;
    }

    public Klndr gtStrDt() {
        return strDt;
    }

    public Klndr gtKnDt() {

```

```

        return knDt;
    }

    public Klndr gtUpdtDt() {
        return updtDt;
    }
    public Zdch setUpdtDt() {
        updtDt = Klndr.gtSchnk();
        return this;
    }

    public Zdch kln() {
        Zdch klnZdch = new Zdch(this);
        if (this.prevZdch == null) {
            this.prjct.dobvZdch(klnZdch);
        }
        return klnZdch;
    }
}

public class Rlld {

    public enum ZdchsVdmnst {

        LZdchs(MainActivity.context.getResources().getString(R.string.all_Zdchs)),
        ZdchsCreatedByOrAssignedToTheKrstvch(MainActivity.context.getResources().getString(R.string.Zdchs_created_by_krstvch));

        private final String nmm;
        ZdchsVdmnst(String nmm) {
            this.nmm = nmm;
        }
        public String gtTtlk() {
            return nmm;
        }

        public static int gtPosition(ZdchsVdmnst ZdchsVdmnst) {
            int rslt = 0;
            for (int i = 0; i < ZdchsVdmnst.values().length; i++)
            {
                if (ZdchsVdmnst == ZdchsVdmnst.values()[i]) {
                    rslt = i;
                    break;
                }
            }
            return rslt;
        }
    }
    public enum SpntVrmsVdmnst {

```

```
AllSpntVrms(MainActivity.context.getResources().getString(R.string.
all_spnt_vrms)),
```

```
SpntVrmCreatedByTheKrstvch(MainActivity.context.getResources().getS
tring(R.string.spnt_vrms_created_by_krstvch));
```

```
    private final String nmm;

    SpntVrmsVdmnst(String nmm) {
        this.nmm = nmm;
    }
    public String gtTtlk() {
        return nmm;
    }

    public static int gtPosition(SpntVrmsVdmnst
spntVrmsVdmnst) {
        int rslt = 0;
        for (int i = 0; i < SpntVrmsVdmnst.values().length;
i++) {
            if (spntVrmsVdmnst == SpntVrmsVdmnst.values()[i])
{
                rslt = i;
                break;
            }
        }
        return rslt;
    }
}

    public Rlld(String nmm) {
        this.nmm = nmm;
        dozvl = new Dozvl();
    }
    public Rlld(String nmm, Dozvl dozvl, ZdchsVdmnst ZdchsVdmnst,
SpntVrmsVdmnst spntVrmsVdmnst) {
        this.nmm = nmm;
        this.dozvl = dozvl;
        this.ZdchsVdmnst = ZdchsVdmnst;
        this.spntVrmsVdmnst = spntVrmsVdmnst;
    }
    public String gtNmm() {
        return nmm;
    }
    public Dozvl gtDozvl() {
        return dozvl;
    }
    public Rlld setNmm(String nmm) {
        this.nmm = nmm;
        return this;
    }
    public ZdchsVdmnst gtZdchsVdmnst() {
        return ZdchsVdmnst;
    }
```

```
    }  
    public SpntVrmsVdmnst gtSpntVrmsVdmnst() {  
        return spntVrmsVdmnst;  
    }  
    public Rlld setZdchsVdmnst(ZdchsVdmnst ZdchsVdmnst) {  
        this.ZdchsVdmnst = ZdchsVdmnst;  
        return this;  
    }  
    public Rlld setSpntVrmsVdmnst(SpntVrmsVdmnst spntVrmsVdmnst)  
{  
        this.spntVrmsVdmnst = spntVrmsVdmnst;  
        return this;  
    }  
}
```

ДОДАТОК В
Слайди презентації

**Національний університет «Запорізька політехніка»
Кафедра програмних засобів**

Програмне забезпечення підтримки процесу управління проєктами

Студент: Ігор РЕКАШОВ, гр. КНТ-121сп

Керівник: д.т.н., професор Андрій ОЛІЙНИК

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для управління проєктами.

Предмет дослідження – програмні засоби для підтримки процесу управління проєктами.

Метою роботи є розробка програмного забезпечення для підтримки процесу управління проєктами.

2

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підтримки процесу управління проектами;
- здійснити проектування програмного забезпечення для підтримки процесу управління проектами;
- створити програмне забезпечення для підтримки процесу управління проектами;
- виконати тестування розробленого програмного забезпечення для підтримки процесу управління проектами.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Asana	ClickUp	Zoho Projects
Широкий вибір шаблонів	+-	+	+-
Доступність та зрозумілість	+-	+	+-
Інформаційні панелі, звіти та графіки	+	+-	+
Функції безпеки та підтримки	+	+	+-
Зручні засоби інтеграції з іншими програмними сервісами	+	+-	+-
Безкоштовність	+-	+-	+-

4

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Java	Kotlin	C#
Швидкість виконання операцій	+	+	+–
Висока безпека даних	+	+	+–
Активна спільнота розробників	+	+–	+
Зручність для створення ПЗ для підтримки процесу управління проектами	+	–	–
Кросплатформенність	+	+	+–

5

Рисунок В.5 – Слайд 5

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Android Studio	Eclipse	Xamarin
Зручність для розробки Android застосунків	+	+–	–
Модулі для створення ПЗ для підтримки процесу управління проектами	+	+	+–
Підтримка мови Java	+	+	–
Активна спільнота розробників	+	+	+–
Безкоштовність	+	+	+–

6

Рисунок В.6 – Слайд 6

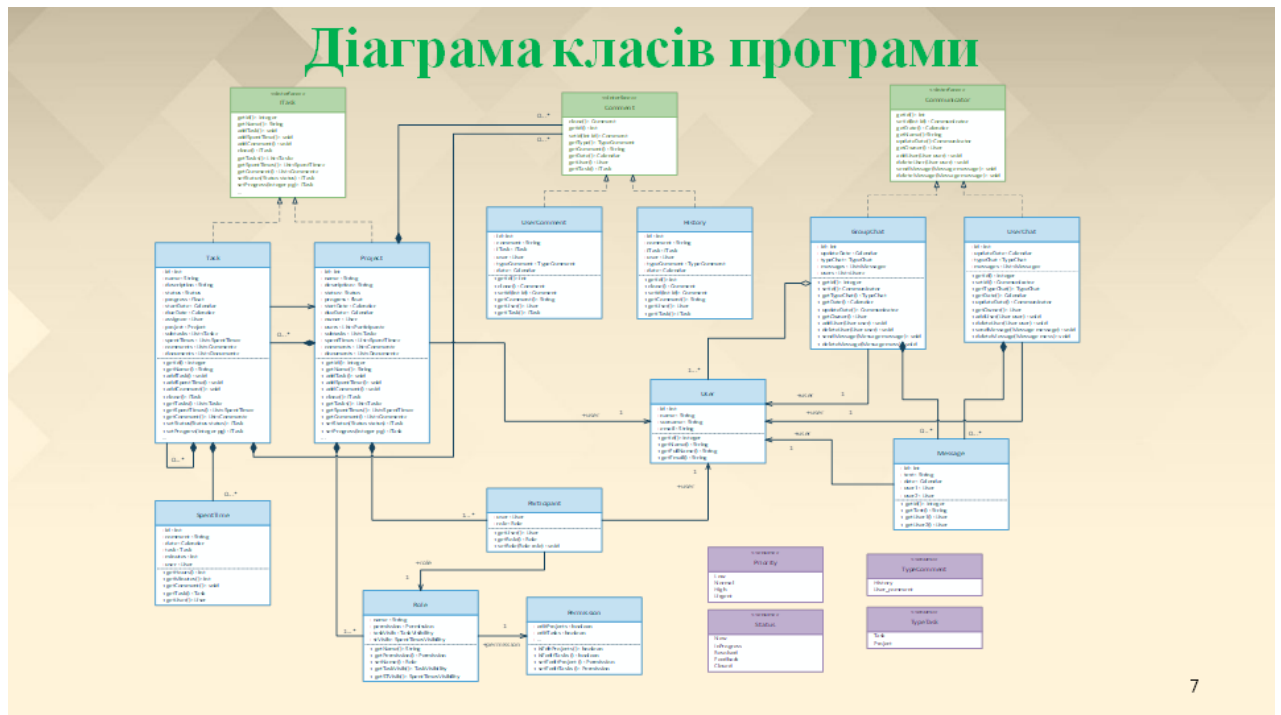


Рисунок В.7 – Слайд 7



Рисунок В.8 – Слайд 8

Діаграма прецедентів програмного забезпечення



9

Рисунок В.9 – Слайд 9

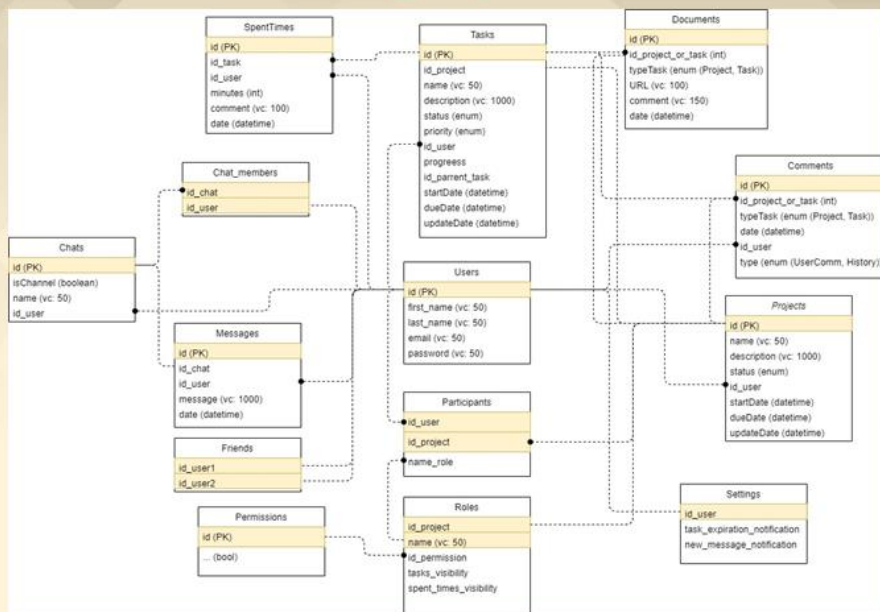
Діаграма послідовності програмного забезпечення



10

Рисунок В.10 – Слайд 10

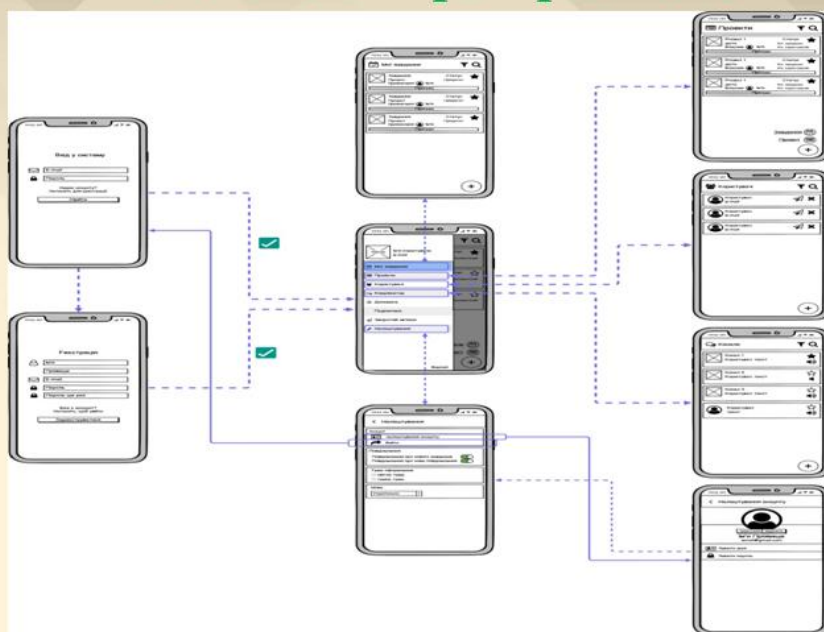
Схема бази даних програмного забезпечення



11

Рисунок В.11 – Слайд 11

Робота з програмою



12

Рисунок В.12 – Слайд 12

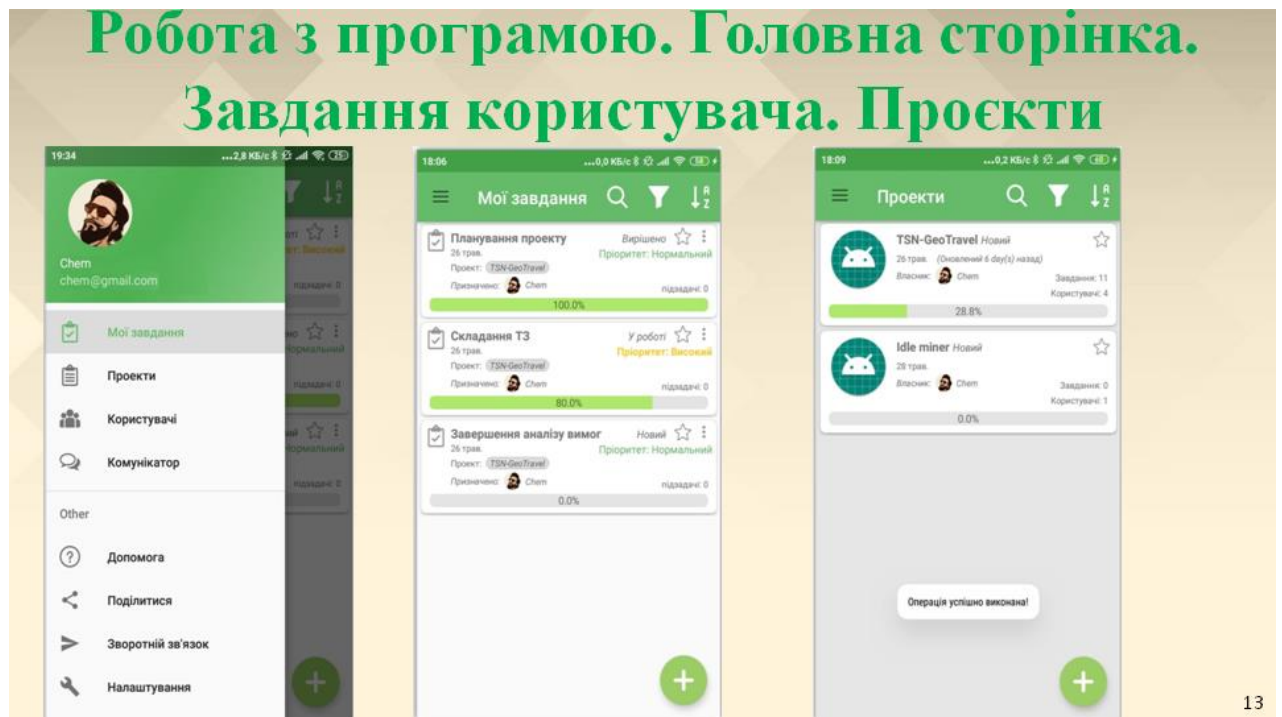


Рисунок В.13 – Слайд 13

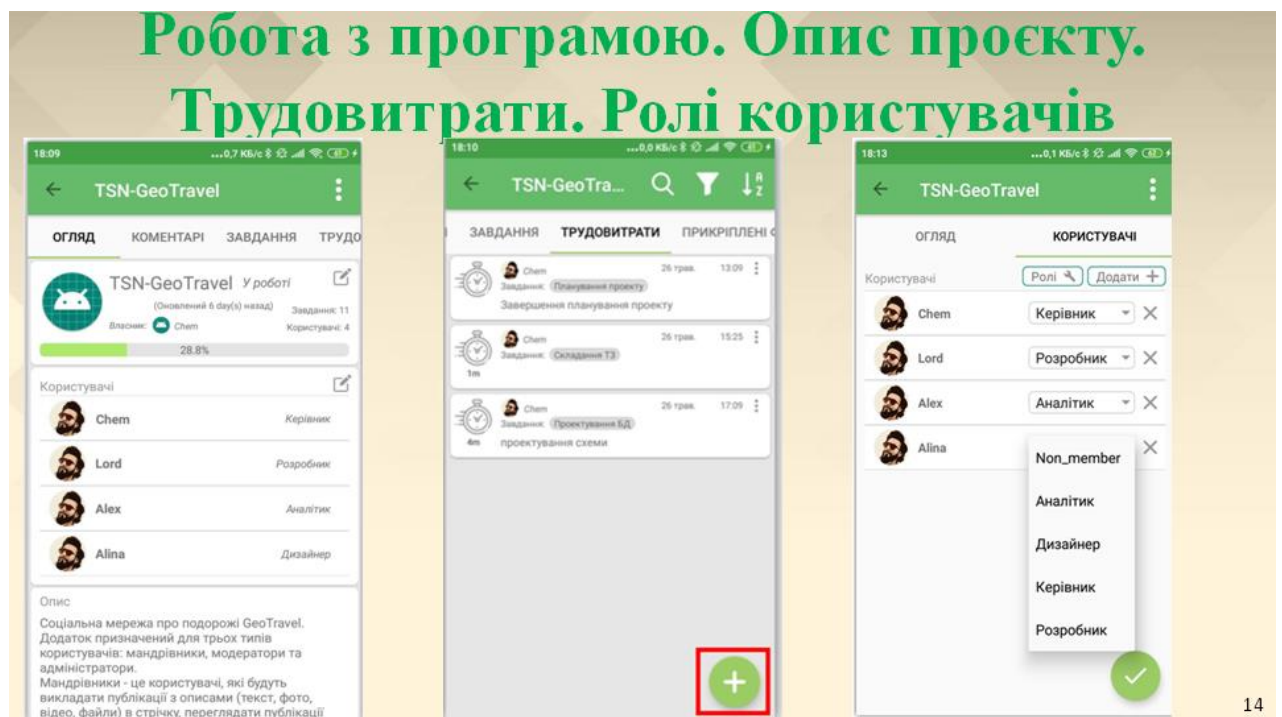


Рисунок В.14 – Слайд 14

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для управління проектами.

Визначено функціональні вимоги до розроблюваного програмного забезпечення для управління проектами. Для створення програми обрано мову програмування Java та середовище розробки Android Studio.

Розроблено програмне забезпечення для управління проектами. Запропоновано структуру розробленого програмного забезпечення. Описано функціонування програмного забезпечення для управління проектами. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для управління проектами.

Результати тестування програмного забезпечення для управління проектами підтвердили те, що воно відповідає технічному завданню та може використовуватися для управління проектами.

Рисунок В.15 – Слайд 15