**Чернівецький національний університет імені Юрія Федьковича** **Навчально-науковий інститут фізико-технічних та**

**комп’ютерних наук**

**Кафедра комп’ютерних наук**

**КУРСОВА РОБОТА**

**на тему**

**«Система управління проєктами»**

|  |  |
| --- | --- |
|  | Студента 3 курсу, групи 344-А  Спеціальність **122 «Комп’ютерні науки»**  Андроніка Веніаміна Ігоровича  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис)  Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (посада, вчене звання, науковій ступінь, прізвище та ініціали)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис)  Національна шкала: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Кількість балів: \_\_\_\_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_\_\_ |

Чернівці

2024

ЗМІСТ

[Вступ 3](#_Toc167379233)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 4](#_Toc167379234)

[1.1 Опис предметної області 4](#_Toc167379235)

[1.2 Аналіз програм-аналогів 4](#_Toc167379236)

[1.3 Постановка задачі на розробку ПЗ для … 4](#_Toc167379237)

[2. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ ДЛЯ … 5](#_Toc167379238)

[2.1 Опис інструментів розробки 5](#_Toc167379239)

[2.2 Проектування та реалізація бази даних 5](#_Toc167379240)

[2.3 Реалізація програмного продукту для … 5](#_Toc167379241)

[2.4 Інструкція користувача 5](#_Toc167379242)

[ВИСНОВКИ 6](#_Toc167379243)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 7](#_Toc167379244)

[ДОДАТОК А 8](#_Toc167379245)

[ДОДАТОК Б 9](#_Toc167379246)

# Вступ

**Мета**: розробити веб-додаток WorkWave для управління завданнями та проектами, який дозволить користувачам створювати, організовувати та відстежувати завдання у різних проектах.

**Завдання**:

1. Реалізація системи реєстрації та авторизації користувачів
   * Користувачі повинні мати можливість створювати облікові записи, використовуючи свої електронні адреси та паролі.
   * Забезпечення надійного захисту даних користувачів через механізми аутентифікації та авторизації.
2. Створення та управління проектами
   * Користувачі повинні мати можливість створювати нові проекти та призначати завдання для них.
   * Організація проектів та завдань з урахуванням статусів виконання завдань.
3. Призначення відповідальних та відстеження виконання завдань
   * Реалізація функціоналу призначення відповідальних за кожне завдання.
   * Можливість відстеження прогресу виконання завдань.
4. Коментування завдань та спільна робота
   * Забезпечення можливості користувачам залишати коментарі та нотатки до завдань.
   * Впровадження функціоналу для роботи в команді над спільними проектами.
5. Технічна реалізація
   * Розробка фронтенд частини з використанням HTML, CSS та JavaScript (або фреймворку React).
   * Реалізація бекенду на Node.js для обробки запитів користувачів та зберігання даних.
   * Використання бази даних для зберігання інформації про користувачів, їх проекти та завдання.

В сучасних умовах швидкого розвитку інформаційних технологій та зростання кількості проектів, які виконуються віддалено, виникає необхідність у використанні ефективних інструментів для управління проектами. WorkWave дозволяє спростити цей процес, забезпечуючи можливість командної роботи, організації та відстеження завдань. Це особливо актуально в умовах гнучкого графіка роботи та розподілених команд, де комунікація та координація між учасниками проекту є критично важливими.

Об'єктом дослідження є процес управління завданнями та проектами в командній роботі. Предметом дослідження є розробка та впровадження веб-додатку для ефективного управління цими процесами.

Практична значущість даної роботи полягає в розробці зручного та функціонального інструменту для управління проектами, що дозволить користувачам значно підвищити ефективність роботи, знизити витрати часу на координацію завдань та покращити загальний результат діяльності команди.

На сьогоднішній день існує багато рішень для управління проектами, таких як Jira, Trello та Worksection. Ці інструменти широко використовуються в різних галузях для планування, організації та відстеження завдань. Проте, кожен з них має свої обмеження та особливості. WorkWave пропонує унікальні можливості, які поєднують зручний інтерфейс, гнучку систему тегів та інтуїтивно зрозумілий дашборд, що робить його привабливим вибором для команд будь-якого розміру.

# **Розділ 1** АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## Опис предметної області

У сучасних умовах, коли проекти часто виконуються розподіленими командами, існує потреба в ефективних інструментах для координації дій, планування та контролю за виконанням завдань. Веб-додатки, такі як WorkWave, покликані вирішувати ці завдання, забезпечуючи зручну платформу для співпраці, моніторингу прогресу та комунікації між учасниками проекту.

Незважаючи на наявність численних інструментів для управління проектами, багато з них мають суттєві недоліки, які обмежують їх ефективність та зручність для користувачів:

1. **Складність інтерфейсу**: Багато існуючих рішень мають надто складний або перевантажений інтерфейс, що ускладнює їх використання, особливо для нових користувачів.
2. **Обмежені можливості налаштування**: Деякі інструменти не дозволяють користувачам гнучко налаштовувати інтерфейс та функціонал відповідно до своїх потреб, що знижує їх універсальність.
3. **Відсутність інтеграції**: Багато рішень не підтримують інтеграцію зі сторонніми сервісами, що ускладнює роботу з іншими інструментами та платформами, які використовуються в проекті.
4. **Обмежені можливості спільної роботи**: Деякі інструменти не забезпечують ефективної підтримки командної роботи, що ускладнює комунікацію та координацію між учасниками проекту.
5. **Недостатня підтримка мобільних пристроїв**: Відсутність адаптованих мобільних версій або додатків обмежує можливість користувачів працювати з завданнями на ходу.

Розробка WorkWave покликана усунути ці недоліки та забезпечити користувачів більш зручним, гнучким та функціональним інструментом для управління проектами. WorkWave пропонує інтуїтивно зрозумілий інтерфейс, який спрощує процес навчання та використання додатку. Завдяки широким можливостям налаштування, користувачі можуть адаптувати інструмент відповідно до своїх потреб та вимог конкретного проекту.

Інтеграція зі сторонніми сервісами дозволяє легко взаємодіяти з іншими інструментами, такими як календарі, месенджери та системи зберігання даних, що підвищує ефективність роботи. WorkWave також забезпечує потужну підтримку командної роботи, включаючи можливість коментування завдань, сповіщення та інші засоби для ефективної комунікації та координації.

Використання WorkWave є доцільним з кількох причин. По-перше, додаток забезпечує підвищену продуктивність роботи команд завдяки ефективній координації завдань та проектів. Це досягається за рахунок інтеграції всіх необхідних інструментів в одній платформі, що зменшує час на перемикання між різними додатками та спрощує управління проектами.

По-друге, інтуїтивний інтерфейс та широкий функціонал WorkWave роблять його зручним для користувачів різного рівня підготовки, що підвищує залученість та знижує опір до впровадження нового інструменту. Крім того, можливість адаптації та налаштування додатку відповідно до конкретних потреб команди забезпечує його універсальність та гнучкість.

По-третє, підтримка мобільних пристроїв дозволяє користувачам працювати з додатком будь-де та будь-коли, що є критично важливим у сучасному динамічному середовищі. Це забезпечує безперервний доступ до завдань та проектів, сприяє швидкому реагуванню на зміни та підвищує загальну ефективність роботи команди.

Нарешті, інтеграція зі сторонніми сервісами робить WorkWave важливим елементом екосистеми управління проектами, що забезпечує безшовну взаємодію з іншими інструментами та сервісами, які використовуються в проектній діяльності.

Таким чином, розробка та впровадження WorkWave не тільки усуває недоліки існуючих рішень, але й забезпечує ефективний, зручний та гнучкий інструмент для управління проектами та завданнями, що підвищує продуктивність та якість роботи команд.

## 1.2 Аналіз програм-аналогів

Для аналізу були обрані програми-аналоги: **Trello**, **JIRA**, **Worksection**. Кожна з них широко використовується в області управління проектами та завданнями.

**Trello** є одним з найпопулярніших інструментів для управління завданнями, який використовує підхід канбан. Він пропонує візуальний інтерфейс для організації завдань у вигляді дошок і карток, що дозволяє легко стежити за прогресом проекту.

**JIRA** - це потужний інструмент від Atlassian, який надає широкі можливості для управління проектами, особливо в сфері розробки програмного забезпечення. JIRA підтримує багаті функції для відстеження завдань, багів, планування спринтів та аналізу продуктивності команди.

**Worksection** - це український інструмент для управління проектами, який пропонує зручний інтерфейс для командної роботи, організації завдань та комунікації. Він також включає функції для відстеження часу, планування та створення звітів.

Таблиця 1.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Trello | JIRA | Worksection | WorkWave (планується) |
| Інтерфейс | Інтуїтивний, канбан-дошки | Складний, але потужний, з гнучкими налаштуваннями | Зручний, простий, підтримка різних видів подань | Інтуїтивний, з гнучкими налаштуваннями |
| Функціональність | Базові функції управління завданнями | Розширені функції для розробки ПЗ, підтримка Agile | Базові та розширені функції для управління проектами | Базові та розширені функції для управління проектами |
| Інтеграції | Підтримка багатьох інтеграцій | Широкі інтеграції з інструментами Atlassian та сторонніми сервісами | Підтримка основних інтеграцій | Підтримка основних інтеграцій, можливість розширення |
| Мобільність | Доступні мобільні додатки | Доступні мобільні додатки | Доступні мобільні додатки | Планується мобільна версія |
| Командна робота | Спільні дошки, коментування | Потужні засоби для командної роботи, Agile методології | Спільні проекти, коментування | Спільні проекти, коментування |
| Підтримка | Багато навчальних матеріалів | Велика кількість документації та підтримки | Добре організована підтримка | Планується надання підтримки |
| Ціна | Безкоштовний план, платні функції | Платна, з різними планами | Платна, з безкоштовним пробним періодом | Безкоштовний план |

Порівняльний аналіз трьох популярних програм для управління проектами та завданнями показує, що кожна з них має свої переваги та недоліки. Trello вирізняється своєю простотою та інтуїтивним інтерфейсом, але обмежений у функціональності для великих проектів. JIRA пропонує потужні інструменти та розширені можливості для звітності, але має складний інтерфейс та високу вартість. Worksection є гнучким та простим у використанні інструментом, але має обмежену кількість інтеграцій та функціональність для складних проектів.

## Постановка задачі на розробку ПЗ для «Системи управління проєктами»

**Мета** створення WorkWave полягає у розробці веб-додатку для ефективного управління завданнями та проектами. Застосунок повинен забезпечити користувачам можливість створювати, організовувати та відстежувати завдання, а також сприяти спільній роботі над проектами. Основна мета - підвищити продуктивність командної роботи через інтуїтивно зрозумілий інтерфейс і потужний функціонал.

**Вимоги до застосунку**

*Функціональні вимоги*:

* Система реєстрації та авторизації користувачів з використанням електронної пошти та пароля.
* Можливість створення нових проектів та призначення завдань для кожного проекту.
* Додавання, редагування та видалення завдань з різними статусами (наприклад, "в роботі", "завершено").
* Призначення відповідальних за завдання та відстеження їх виконання.
* Система коментування та нотаток до кожного завдання для спільної роботи.
* Надання доступу до спільних проектів іншим користувачам, що дозволяє бачити і редагувати завдання.
* Інтеграція зі сторонніми сервісами для розширення функціоналу (наприклад, календарі, поштові сервіси).

*Нефункціональні вимоги:*

* Висока продуктивність та швидкодія системи, незалежно від кількості користувачів та завдань.
* Забезпечення безпеки даних користувачів через механізми авторизації та аутентифікації.
* Інтуїтивно зрозумілий та зручний інтерфейс для користувачів з різним рівнем технічних навичок.
* Можливість масштабування системи при зростанні кількості користувачів та даних.
* Висока надійність та відмовостійкість системи, мінімізація часу простою.

**Опис користувачів застосунком**

WorkWave планується зробити призначеним для використання різними категоріями користувачів:

* Проектні менеджери: створюють та організовують проекти, призначають завдання та відповідальних, відстежують прогрес.
* Члени команди: виконують призначені завдання, залишають коментарі та співпрацюють з іншими учасниками проектів.
* Адміністратори системи: керують користувачами, налаштовують права доступу, забезпечують безпеку та технічну підтримку.

**Обмеження у створенні застосунку**

Розробка WorkWave може стикнутися з такими обмеженнями:

* Технічні обмеження: необхідність підтримки різних браузерів та операційних систем, обмеження швидкості інтернет-з'єднання користувачів.
* Безпека: забезпечення високого рівня захисту даних користувачів, уникнення несанкціонованого доступу та атак.
* Ресурси: обмежений бюджет та час на розробку, можливість залучення кваліфікованих фахівців.

**Короткий опис етапів реалізації ПЗ**

Процес розробки WorkWave включає кілька основних етапів:

* Планування: визначення вимог до системи, створення технічного завдання та плану розробки.
* Дизайн: розробка інтерфейсу користувача, створення прототипів та макетів.
* Розробка: програмування фронтенду та бекенду, інтеграція з базою даних та сторонніми сервісами.
* Тестування: перевірка функціональності, продуктивності та безпеки системи, виявлення та виправлення помилок.
* Впровадження: розгортання системи на сервері, налаштування та запуск в експлуатацію.
* Підтримка та оновлення: технічна підтримка користувачів, виправлення помилок, додавання нових функцій.

**Оптимальні вимоги до апаратного та програмного забезпечення**

Для ефективного функціонування WorkWave рекомендуються наступні вимоги:

*Апаратне забезпечення:*

* Сервер: процесор з високою продуктивністю (не менше 4 ядер), оперативна пам'ять від 8 ГБ, SSD-диск для швидкого доступу до даних.
* Клієнтські пристрої: комп'ютери та мобільні пристрої зі стабільним інтернет-з'єднанням, сучасними браузерами.

*Програмне забезпечення:*

* Сервер: операційна система Linux або Windows, веб-сервер Apache або Nginx, база даних MySQL або PostgreSQL.
* Фронтенд: HTML, CSS, JavaScript, фреймворк React.
* Бекенд: Node.js, бібліотеки для авторизації та аутентифікації, API для інтеграції зі сторонніми сервісами.

## Висновок

У результаті виявлено, що додаток WorkWave є веб-додатком для управління завданнями та проектами, який дозволяє користувачам створювати, організовувати та відстежувати завдання у різних проектах. Основними функціями є реєстрація та авторизація користувачів, створення проектів, додавання та видалення завдань, призначення відповідальних, коментування завдань та доступ до спільних проектів.

Технічна реалізація передбачає фронтенд, бекенд та базу даних, що забезпечує відображення завдань та інтерфейсу користувача, обробку запитів користувачів та зберігання інформації про користувачів, їх проекти та завдання. Використання механізмів авторизації та аутентифікації, системи коментування, сповіщень та спільної роботи покликані забезпечити безпеку даних користувачів та ефективну співпрацю між учасниками проектів.

Порівняльний аналіз з програмами-аналогами, такими як Trello, JIRA та Worksection, показав, що WorkWave має схожий функціонал, але може бути більш привабливим завдяки інтуїтивно зрозумілому інтерфейсу та можливості спільної роботи з іншими учасниками проектів.

З урахуванням висновків розділу можна зробити висновок, що розробка проекту WorkWave є доцільною та може значно полегшити управління завданнями та проектами для широкого кола користувачів.

# **Розділ 2.** РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ ДЛЯ «Системи управління проєктами»

## 2.1 Опис інструментів розробки

Для реалізації проекту "WorkWave" планується використання наступного інструментарію:

**Система управління базами даних** (СУБД): MySQL

MySQL використовується для зберігання інформації про користувачів, їх проекти та завдання. Ця СУБД обрана через свою надійність, популярність та швидкодію роботи з даними.

**Серверна частина**: Node.js

Node.js буде використаний для обробки запитів користувачів, авторизації та зберігання даних про проекти та завдання. Використання Node.js сприятиме асинхронності роботи додатку та забезпечить його ефективну роботу.

**Фронтенд**: React

React буде використаний для побудови веб-інтерфейсу, який дозволить користувачам відстежувати та взаємодіяти зі своїми завданнями та проектами. Він обраний через його компонентну архітектуру та високу продуктивність.

**Інструмент для дизайну**: Figma, AdobeIllustrator

Figma використовуватиметься для створення дизайну веб-інтерфейсу додатку. Він дозволить розробникам та дизайнерам ефективно співпрацювати та створювати інтерфейс, який задовольнить потреби користувачів. AdobeIllustrator використовуватиметься для створення логотипу.

## 2.2 Проектування та реалізація бази даних

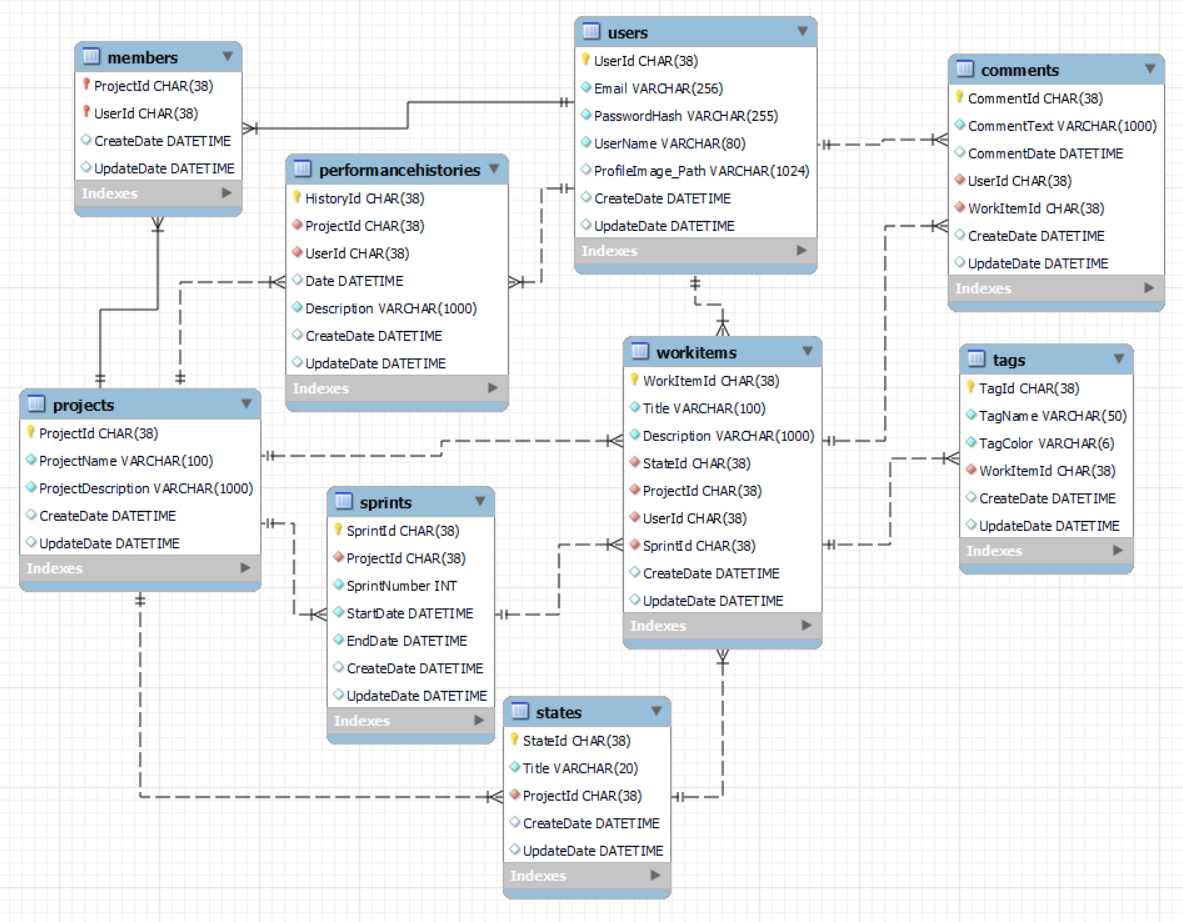
### 2.2.1 **Сутності**:

* Користувачі (Users)
  + UserId: CHAR(38) (PRIMARY KEY)
  + Email: NVARCHAR(256) (UNIQUE NOT NULL)
  + PasswordHash: NVARCHAR(255) (NOT NULL)
  + UserName: NVARCHAR(80) (UNIQUE NOT NULL)
  + ProfileImage\_Path: VARCHAR(1024)
  + CreateDate: DATETIME
  + UpdateDate: DATETIME
* Проекти (Projects)
  + ProjectId: CHAR(38) (PRIMARY KEY)
  + ProjectName: NVARCHAR(100) (NOT NULL)
  + ProjectDescription: NVARCHAR(1000) (NOT NULL)
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME
* Стани (States)
  + StateId: CHAR(38) (PRIMARY KEY)
  + Title: NVARCHAR(20) (UNIQUE NOT NULL)
  + ProjectId: CHAR(38) (FOREIGN KEY REFERENCES
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME
* Спринти (Sprints)
  + SprintId: CHAR(38) (PRIMARY KEY)
  + ProjectId: CHAR(38) (FOREIGN KEY REFERENCES Projects(ProjectId))
  + SprintNumber: INT (NOT NULL)
  + StartDate: DATETIME (NOT NULL)
  + EndDate: DATETIME (NOT NULL)
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME
* Завдання (WorkItems)
  + WorkItemId: CHAR(38) (PRIMARY KEY)
  + Title: NVARCHAR(100) (NOT NULL)
  + Description: NVARCHAR(1000) (NOT NULL)
  + StateId: CHAR(38) (FOREIGN KEY REFERENCES States(StateId))
  + ProjectId: CHAR(38) (FOREIGN KEY REFERENCES Projects(ProjectId))
  + UserId: CHAR(38) (FOREIGN KEY REFERENCES Users(UserId))
  + SprintId: CHAR(38) (FOREIGN KEY REFERENCES Sprints(SprintId))
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME
* Історія продуктивності (PerformanceHistories)
  + HistoryId: CHAR(38) (PRIMARY KEY)
  + ProjectId: CHAR(38) (FOREIGN KEY REFERENCES Projects(ProjectId))
  + UserId: CHAR(38) (FOREIGN KEY REFERENCES Users(UserId))
  + Date: DATETIME (DEFAULT(now()))
  + Description: NVARCHAR(1000) (NOT NULL)
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME
* Теги (Tags)
  + TagId: CHAR(38) (PRIMARY KEY)
  + TagName: NVARCHAR(50) (NOT NULL)
  + TagColor: NVARCHAR(6) (NOT NULL)
  + WorkItemId: CHAR(38) (FOREIGN KEY REFERENCES WorkItems(WorkItemId))
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME
* Коментарі (Comments)
  + CommentId: CHAR(38) (PRIMARY KEY)
  + CommentText: NVARCHAR(1000) (NOT NULL)
  + CommentDate: DATETIME (DEFAULT(now()))
  + UserId: CHAR(38) (FOREIGN KEY REFERENCES Users(UserId))
  + WorkItemId: CHAR(38) (FOREIGN KEY REFERENCES WorkItems(WorkItemId))
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME
* Учасники проекту (Members)
  + ProjectId: CHAR(38) (FOREIGN KEY REFERENCES Projects(ProjectId))
  + UserId: CHAR(38) (FOREIGN KEY REFERENCES Users(UserId))
  + PRIMARY KEY (ProjectId, UserId)
  + CreateDate: DATETIME (DEFAULT(now()))
  + UpdateDate: DATETIME

### 2.2.2 **Зв'язки**:

* Користувач може мати багато проектів (1-N)
* Проект може мати багато станів (1-N)
* Проект може мати багато спринтів (1-N)
* Проект може мати багато завдань (1-N)
* Користувач може мати багато завдань (1-N)
* Завдання може мати багато коментарів (1-N)
* Завдання може мати багато тегів (1-N)
* Користувач може мати багато історій продуктивності (1-N)
* Проект може мати багато учасників (1-N)

### **2.2.3 ER-Діаграма створеної БД:**



## 2.3 Реалізація програмного продукту для Системи управління проєктами

## 2.4 Інструкція користувача

## Висновок

# ВИСНОВКИ

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

# ДОДАТОК А

# ДОДАТОК Б