



Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

### **Лабораторна робота №2**

*«Діаграми прецедентів, класів, структура системи бази даних»*

Варіант 23

Виконала студентка групи ІА–13:

Сиваченко Дар'я Євгенівна

Перевірів :

Мягкий Михайло Юрійович

Київ 2023

**Тема:** Project Management software

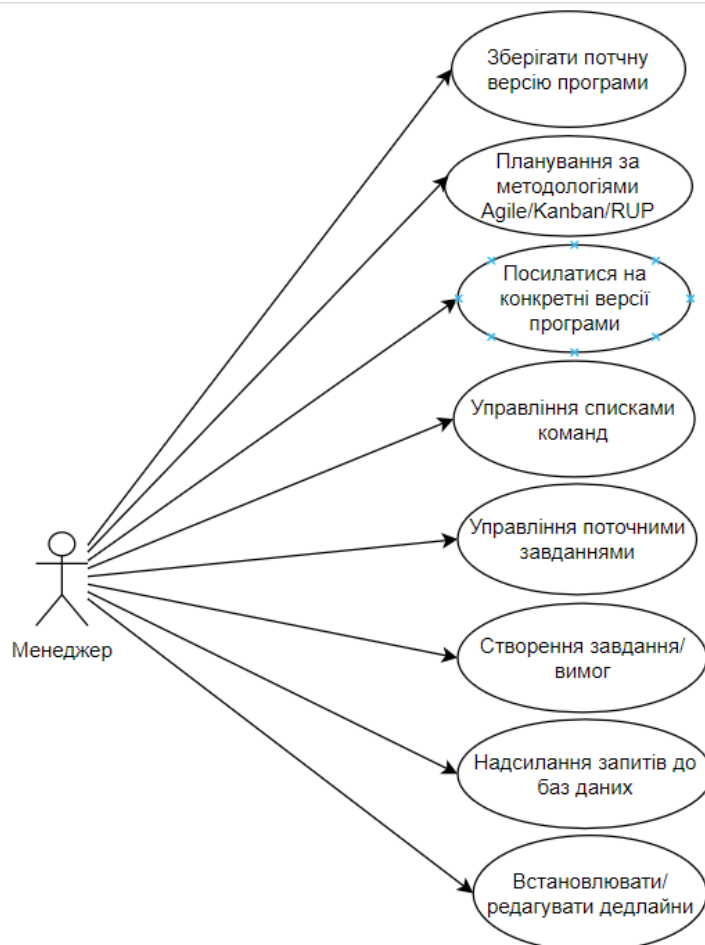
### **Хід роботи**

### **Діаграма Прецедентів**

Перелік акторів та прецедентів:

Користувач	Має можливість авторизуватися, надіслати запит на приєднання до команди, створити/редагувати/видалити проект, обрати для виконання завдання, яке надав менеджер, має можливість прикріплювати виконані файли, після завершення роботи, зберегти поточну версію програми та файлів.
Менеджер	Створює команди та завдання, видалення та додавання користувачів, розробляє методiku, використовуючи Agile/Kanban/RUP, контроль поточного виконання завдань, має можливість надсилати запити до баз даних, за необхідності, посилається на конкретні версії програми та зберігати поточну версію програми, Встановлювати/контролювати дедлайни для виконання завдань.
Система	Проводить процес авторизації, процес зберігання виконаних та прикріплених файлів, збереження змісту поточної версії програми.

## Діаграма Прецедентів



## Опис трьох прецедентів:

### 1. Авторизація користувача

Актор: Користувач

Опис: Користувач вводить своє ім'я та заданий пароль для входу в систему

Результат: Система перевіряє дані, які ввів користувач, якщо вони вірні, то надається доступ до користування системою та можливостями цієї системи .

### 2. Створення завдання

Актор: Менеджер

Опис: Для початку, менеджеру необхідно авторизуватися в системі, створити нову команду та додати користувачів, натиснути кнопку для створення завдань, описати його, вкласти необхідні файли, зберегти зміни, натиснувши на відповідну кнопку, надати доступ користувачам команди до нових завдань, натиснувши кнопку підтвердження та зберігання.

Результат: В результаті вищеописаних дій, члени команди отримують список завдань.

### 3. Вибір завдання

Актор: Користувач

Опис: Користувач має авторизуватися, прийняти запит на приєднання до команди, натиснувши кнопку підтвердження, після підтвердження користувачу надається список завдань, які надав менеджер та має можливість обрати завдання для виконання, натиснувши відповідну кнопку

Результат: Після виконаних дій, описаних вище, користувач отримує завдання для виконання.

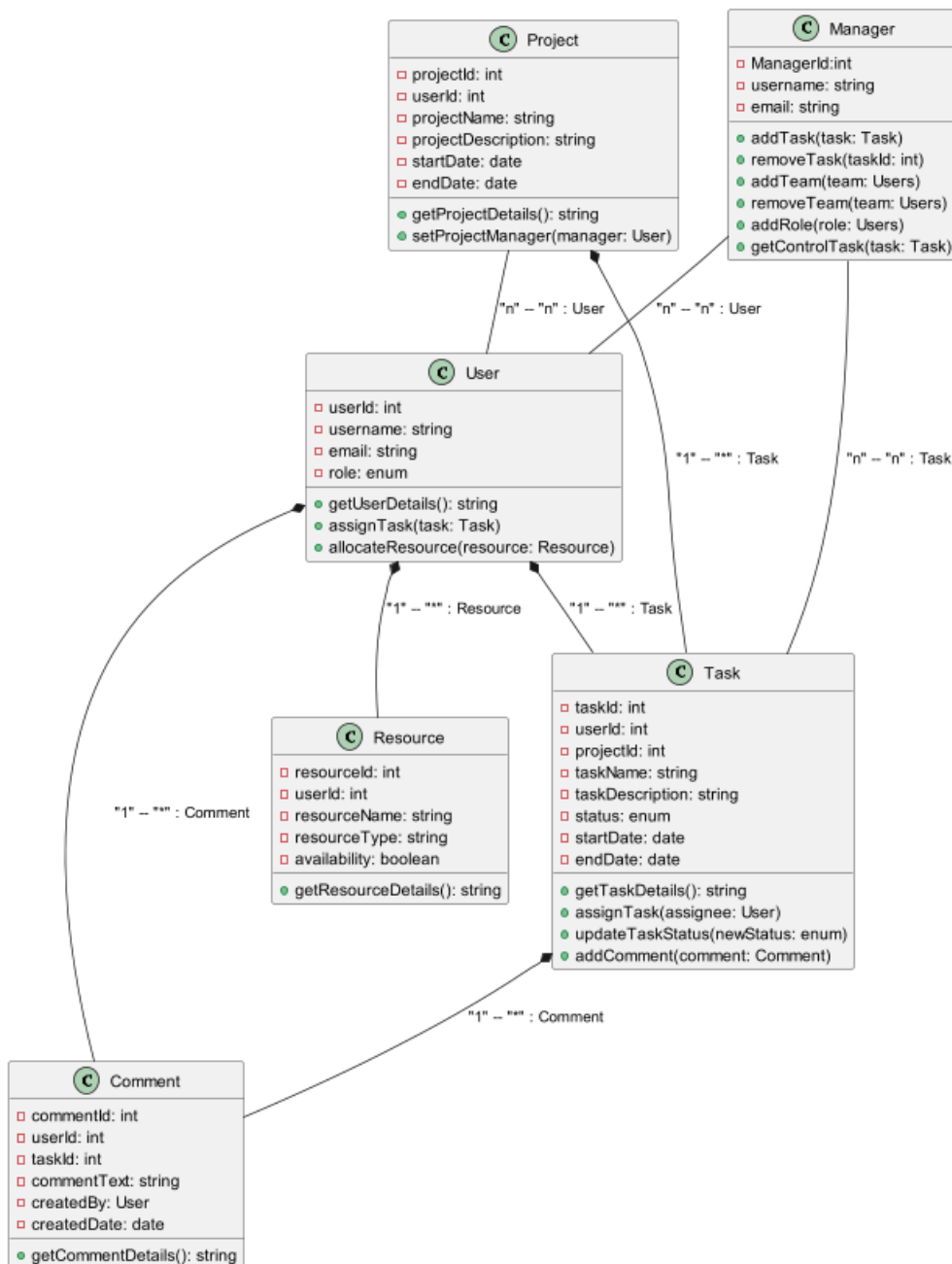
## Діаграма Класів:

```
@startuml
class Project{
- projectId: int
- userId: int
- projectName: string
- projectDescription: string
- startDate: date
- endDate: date
+ getProjectDetails(): string
+ setProjectManager(manager: User)
}
```

```

class Task{
- taskId: int
- userId: int
- projectId: int
- taskName: string
- taskDescription: string
- status: enum
- startDate: date
- endDate: date
+ getTaskDetails(): string
+ assignTask(assignee: User)
+ updateTaskStatus(newStatus: enum)
+ addComment(comment: Comment)
}
class User{
- userId: int
- username: string
- email: string
- role: enum
+ getUserDetails(): string
+ assignTask(task: Task)
+ allocateResource(resource: Resource)
}
class Resource{
- resourceId: int
- userId: int
- resourceName: string
- resourceType: string
- availability: boolean
+ getResourceDetails(): string
}
class Comment{
- commentId: int
- userId: int
- taskId: int
- commentText: string
- createdBy: User
- createdAt: date
+ getCommentDetails(): string
}
class Manager{
-ManagerId:int
- username: string
- email: string
+ addTask(task: Task)
+ removeTask(taskId: int)
+ addTeam(team: Users)
+ removeTeam(team: Users)
+ addRole(role: Users)
+ getControlTask(task: Task)
}
Project -- User : "n" -- "n" : User
Project *-- Task : "1" -- "*" : Task
User *-- Resource : "1" -- "*" : Resource
Task *-- Comment : "1" -- "*" : Comment
User *-- Comment : "1" -- "*" : Comment
User *-- Task : "1" -- "*" : Task
Manager -- User : "n" -- "n" : User
Manager -- Task : "n" -- "n" : Task
@enduml

```



**Зв'язок з базою даних:** опрацювання надісланих запитів від менеджера, зберігання всього процесу користування системою (включаючи виникнення помилок, збереження файлів, поточної версії програми), збереження дій користувачів в системі.

## **Висновок**

Розроблена система має великий потенціал для ефективного керування проектами та командами. Функціонал системи, доступний для користувачів і менеджерів, надає широкий спектр можливостей для зручного та продуктивного спільного робочого процесу. Для користувачів система пропонує авторизацію, що гарантує безпеку доступу до системи, а також можливість надсилати запити на приєднання до команди, створення, редагування та видалення проектів. Користувачі можуть вибирати завдання для виконання, надані менеджером, і прикріплювати виконані файли для звітності. Важливою можливістю є збереження поточної версії програми та файлів, що гарантує збереження історії роботи та можливість повертатися до попередніх версій. Менеджери мають доступ до більш розширеного функціоналу, включаючи створення команд і завдань, додавання та видалення користувачів, розробку методик з використанням різних методологій, таких як Agile, Kanban та RUP. Вони також контролюють поточне виконання завдань і можуть встановлювати та контролювати дедлайни для їх виконання. Можливість надсилати запити до баз даних робить менеджером процес керування проектами більш динамічним та реактивним. Система забезпечує безпечний процес авторизації і зберігання виконаних та прикріплених файлів, що гарантує конфіденційність та надійність даних. Збереження поточної версії програми сприяє збереженню стабільності та доступності інформації для всіх учасників проекту. Загалом, розроблена система надає інструменти для ефективного спільного ведення проектів та командної роботи, допомагає зберігати і відстежувати інформацію та забезпечує зручність та надійність в робочих процесах. Весь процес продемонстрований за допомогою діаграм.