Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “ Об’єктно -орієнтоване програмування”

Керівник , Професор кафедри програмної інженерії Бондарєв В.М.

Студент гр. ПЗПІ-22-1 Корецький І.О.

Комісія: Проф. Бондарєв В.М.,

Ст. викл. Черепанова Ю.Ю.,

Ст. викл. Ляпота В.М.

Харків 2023

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра *програмної інженерії*

Рівень вищої освіти *перший (бакалаврський)*

Дисципліна *Об’єктно-орієнтоване програмування*

Спеціальність *121 Інженерія програмного забезпечення*

Освітня програма: *Програмна інженерія*

Курс 1 . Група *ПЗПІ-22 -1* . Семестр *2*

***ЗАВДАННЯ***

***на курсовий проект студента***

*Корецького Ігора Олександровича*

1. Тема проекту: Менеджер проектів
2. Термін здачі студентом закінченого проекту: ***“16” - червня - 2023 р***.
3. Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи*

4 Зміст розрахунково-пояснювальної записки:

*Вступ, опис вимог, проектування програми, інструкція користувача, висновки*

КАЛЕНДАРНИЙ ПЛАН

|  |  |
| --- | --- |
| *Назва етапу* | *Термін виконання* |
| Видача теми, узгодження і затвердження  теми | 13.02.2023 - 14.03.2023 р. |
| Формулювання вимог до програми | 23.05.2023 – 24.05.2023 р. |
| Розробка моделей | 24.05.2023 – 25.05.2023 р. |
| Розробка функцій ... | 25.05.2023 – 3.06.2023 р. |
| Розробка функцій зберігання та  завантаження даних | 3.06.2023 р. |
| Тестування і доопрацювання розробленої  програмної системи. | 4.06.2023 – 08.06.2023 р. |
| Оформлення пояснювальної записки,  додатків, графічного матеріалу | 8.06.2023 – 09.06.2023 р. |
| Захист | 9.06.2023 р. |

Студент Корецький Ігор Олександрович

Керівник Бондарев Володимир Михайлович « 8 »\_червня 2023 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 35 с., 34 рис., 1 джерело.

КЛАС, КОЛЕКЦІЯ, МЕТОД, МОВА ПРОГРАМУВАННЯ C#, ПРОЕКТИ, КОРИСТУВАЧИ, ЗАВДАННЯ, ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, КОНСОЛЬНИЙ ЗАСТОСУНОК.

Метою роботи є розробка програми «Менеджер проектів», яка буде надавати IT бізнесу можливість організації проектів.

В результаті отримана програма, що дозволяє зберігати проекти, задачі та користувачів. У програмі є розділення користувачей за ролями, яке надає різний рівень доступу. Програма надає можливість створювати проекти й завдання, а також редагувати їх. Є реалізація реєстрації користувачей. Є можливість додавати файли до завдання, а також слідкувати за прогресом.

В процесі розробки використано середовища Microsoft Visual Studio 2022, платформи .NET 7.0, мова програмування C#.

**ЗМІСТ**

ВСТУП 6

1 СПЕЦИФІКАЦІЯ ПРОГРАМИ 7

1.1 Основні функції програми 7

1.2 Опис функцій програми 7

1.2.1 Функція «Sign Up» 7

1.2.2 Функція «Sign In» 7

1.2.3 Функції «UserMenu» 8

1.2.3.1 Функція «Log Out» 8

1.2.3.2 Функція «Choose Project» 8

1.2.3.2.1 Функції «DutyIdentifier» 8

2 ПРОЕКТУВАННЯ ПРОГРАМИ 11

2.1 Об’єктна структура програми 11

2.2 Докладний опис класів та їх взаємодія 13

2.2.1 Перелічувальні типи 14

2.2.1.1 Enum Duty 14

2.2.1.2 Enum Priority 14

2.2.1.3 Enum Status 15

2.2.2 Базові класи 16

2.2.2.1 Клас Attachment 16

2.2.2.2 Клас BaseEntity 16

2.2.2.3 Клас Project 17

2.2.2.4 Клас Task 18

2.2.2.5 Клас User 19

2.2.2.6 Клас UserProjectRole 20

2.2.3 Інтерфейс IStorage<T> 21

2.2.4. Інтерфейси сервісів 22

2.2.4.1 Інтерфейс IGenericService<T> 22

2.2.2.4.2 Інтерфейс IProjectService 23

2.2.2.4.3 Інтерфейс ITaskService 23

2.2.2.4.4 Інтерфейс IUserProjectRoleService 24

3 ІНСТРУКЦІЯ КОРИСТУВАЧА 24

3.1 Встановлення програми 24

3.2 Функціонал програми та інструкція до неї 25

3.2.1 Робота зі сторінкою «Sign Up» 25

3.2.2 Робота зі сторінкою «Sign In» 26

3.2.3 Робота з меню користувача 26

3.2.3.1 Меню «StateManager» 27

3.2.3.1.1 Меню редагування проекту 28

3.2.3.2 Меню «Developer» та «Tester» 32

3.2.3.2.1 Меню взаємодії із задачею 33

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 35

# ВСТУП

# 

Метою цієї курсової роботи є створення консольного застосунку менеджер проектів. Для цього були реалізовані наступні задачі:

* Реєстрація та авторизація користувачів;
* Створення, редагування та завершення задач;
* Відстеження прогресу виконання задач;
* Управління проектами;
* Надсилання сповіщень користувачам про зміни у статусах задач

В процесі розробки використано Microsoft Visual Studio, платформу .NET 7.0, мову програмування C#.

# 1 СПЕЦИФІКАЦІЯ ПРОГРАМИ

1.1 Основні функції програми  
  
 Завдяки програмі «Менеджер Проектів», користувач із тегом «StateManager» має змогу додавати користувачей до проекту, а також «StateManager» має доступ до редагувань завдань та проектів. Інші користувачі мають обмежений вплив на роботу програми. Вони не можуть редагувати ніякі параметри проектів та завдань. Проте, вони можуть додавати файли до завдань, дивитися їх, а також можуть передати проект іншому працівнику на проекті.

## 1.2 Опис функцій програми

При запуску програми, користувач обирає одну з трьох дій («Sign Up»(1), «Sign In»(2), «Exit» - вихід із програми).

### 1.2.1 Функція «Sign Up»

Коли користувач обирає «Sign Up», запускається однойменний метод. Далі користувач має ввести всі данні, які попросить програма. У разі, якщо це перший користувач у програмі, програма також попросить створити проект, а також назначить нового користувача менеджером стану(State Manager). Після того як нового користувача буде створено, користувач автоматично перейде до меню («UserMenu»(3)).

### 1.2.2 Функція «Sign In»

Коли користувач обирає «Sign In», запускається однойменний метод. Тут користувач має ввести адресу своєї електронної пошти, а також пароль для входу в акаунт. У разі якщо всі введені дані правильні, буде активовано метод «UserMenu»(3).

### 1.2.3 Функції «UserMenu»

На цьому етапі, користувач побачить повідомлення (якщо вони є), а також механізм вибору наступних дій («Log Out»(1), «Choose Project»(2), «See Notifications»(знову показує повідомлення)).

#### 1.2.3.1 Функція «Log Out»

Ця опція активує метод «PerformOperationsAsync», який переносить користувача в початок програми.

#### 1.2.3.2 Функція «Choose Project»

Користувач обирає проект через метод інтерфейсу проекту «GetProjectAsync». Програма отримує тег користувача на цьому проекті, та у разі якщо всі попередні операції були виконані правильно, програма активує метод «DutyIdentifier».

##### 1.2.3.2.1 Функції «DutyIdentifier»

«DutyIdentifier» відкриває UI згідно до отриманого раніше тегу. «DutyIdentifier» може привести до:

* Інтерфейс менеджеру стану («StateManagerUI»(1));
* Інтерфейс розробнику («DeveloperUI»(2));
* Інтерфейс тестеру («TesterUI»(3));

###### 1.2.3.2.1.1 Функції «StateManagerUI»

Першим чином, користувач побачить метод «PerformOperationsAsync», який дозволить обрати один із цих варіантів:

* Додати проект («Add Project»);
* Змінити обраний проект («Edit current Project»(1));
* Вийти («Exit» - повертає до попереднього методу)

1.2.3.2.1.1.1 Функції «Edit current Project»

Користувач переходить до меню, де він зможе обрати одну

опцій:

* + Змінити назву проекту («Edit Project Name»);
  + Змінити опис проекту («Edit Project Description»);
  + Створити задачу («Create Assignment»);
  + Налаштувати задачі («Edit Assignments»(1));
  + Налаштувати працівників («Edit Workers»(2));
  + Вийти («Exit» - повертає до попереднього методу)

1.2.3.2.1.1.1.1 Функції «Edit Assignments»

Користувач обирає задачу, яку він буде змінювати. Далі користувач має обрати одну з опцій:

* + Назначити користувача на завдання («Assign worker to this Assignment»);
  + Закрити задачу («Close Assignment»);
  + Вийти («Exit» - повертає до попереднього методу)

1.2.3.2.1.1.1.2 Функції «Edit Workers»

Користувач переходить до меню вибору взаємодії з працівниками.

* Прибрати користувача із проекту («Remove Worker from Project»);
* Додати користувача до проекту («Assign Worker for Project»);
* Дати тег («Assign Duty»);
* Вийти («Exit» - повертає до попереднього методу)

###### 1.2.3.2.1.1 Функції «DeveloperUI»

Першим чином, користувач побачить метод «PerformOperationsAsync», який дозволить обрати один із цих варіантів:

* Обрати задачу («Choose Assignment»(1));
* Вийти («Exit» - повертає до попереднього методу)

1.2.3.2.1.1.1 Функції «Choose Assignment»

Працівник обирає задачу. Далі працівник може виконувати певні дії із цією задачею. Доступні діі:

* + Передати задачу («Hand Assignment»);
  + Завантажити файл («Upload File»);
  + Відкрити у провіднику папку із файлом («Open File Folder»);
  + Вийти («Exit» - повертає до попереднього методу)

# 2 ПРОЕКТУВАННЯ ПРОГРАМИ

## 2.1 Об’єктна структура програми

Програма написана із дотриманням правил архітектури застосунку. Згідно до цих правил програма поділяється на наступні компоненти: DAL (Data Access Layer) => DAL.Abstractions => BLL => BLL.Abstractions => UI. Також в цій системі передбачено додаткове відокремлення, тому в моєму коді також є Core та Helpers.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рис. 2.1, 2.2 – Структура

Від абстрактного класу BaseEntity успадковуються всі класи в Core.Models (окрім Аttachments). Єдиною властивістю цього класу є int поле Id. Всі інші класи створюються задля використання в усьому проекті. Також в Сore є папка із Enum.

В DAL знаходиться код, відповідальний за серіалізацію до Json. Всі методи для роботи програми з Json прописані в дженерік класі Storage<T> (T : BaseEntity), який успадковується від дженерік інтерфейсу IStorage<T> (знаходиться у DAL. Abstractions). У роботі Storage також приймає участь клас Result із Helpers.Validators. Цей клас відповідає за обробку помилок. Таким чином для того щоб користуватись методами Storage, наступний слой BLL, користується інтерфейсом.

В BLL знаходиться логіка програми у форматі класів сервісу. Головним у BLL є абстрактний клас GenericService<T>, так як від нього успадковуються всі інші сервіси. Він використовується для доступу до методів роботи із Json. Сам GenericService<T> успадковується від інтерфейсу IGenericService<T>, який знаходиться в BLL.Abstractions. Всі сервісні класи, також успадковуються від своїх інтерфейсів, котрі знаходяться у BLL. Abstractions. Інтерфейси у BLL. Abstractions в свою чергу успадковуються від IGenericService<T>. Таким доступ до сервісів із UI буде проходити через інтерфейси.

В UI знаходяться класи, які відповідають за роботу з консоллю (у папці ConsoleManagers). Тут також є дженерік класс СonsoleManager<T>, та його інтерфейс IConsoleManager<T>. ConsoleManager використовується для взаємодії із Json. Також тут є клас DutyPermission, який відповідає за запуск UI відповідно до тегу користувача.   
 Окремо виділяю присутній на всіх рівнях програми клас DependencyRegistration. Клас DependencyRegistration відповідає за реєстрацію залежностей у контейнері залежностей. Це означає, що він вказує, які класи або компоненти залежать від інших класів або компонентів, а також як їх створити та внедрити. На кожному рівні програми, локальний DependencyRegistration реєструє залежності, присутні на рівні.

## 2.2 Докладний опис класів та їх взаємодія

Core:

Перелічувальний тип даних

Enums (3):

* + - Duty
    - Priority
    - Status

Базові класи

Models (6):

* + - * Attachment
      * BaseEntity
      * Project
      * Task
      * User
      * UserProjectRole

DAL.Abstractions:

Інтерфейси

Interfaces (1):

* IStorage<T>

BLL.Abstractions:

Інтерфейси  
 Interfaces (5):

* + - IGenericService<T>
    - IProjectService
    - ITaskService
    - IUserProjectRoleService
    - IUserService

Та класи DependencyRegistration, UI та реалізацією (сервісні).

### 2.2.1 Перелічувальні типи

Ці Enum використовуються як поля для деяких моделей.

2.2.1.1 Enum Duty  
  
Enum із тегами для працівників:

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рис. 2.3 – Enum Duty

Складається з:

* Unassigned
* Developer
* Tester
* StateManager

#### 2.2.1.2 Enum Priority

Enum із пріоритетом для задач:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание  
Рис. 2.4 – Enum Priority

Складається з:

* Minor
* Low
* Medium
* High
* Urgent

#### 2.2.1.3 Enum Status

Enum із статусами задач:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рис. 2.5 – Enum Status

Складається з:

* + - Planned
    - Developing
    - Testing
    - Revision
    - Done

### 2.2.2 Базові класи

#### 2.2.2.1 Клас Attachment

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 2.6 – Клас Attachment

Складається з:

* Публічної властивості FileName (string)
* Публічної властивості Size (long)
* Публічної властивості Data (byte[])

Цей клас використовується для збереження файлів, які загружає користувач.

#### 2.2.2.2 Клас BaseEntity

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 2.7 – Клас BaseEntity

Складається з:

* Публічної властивості Id (int)

Від цього класу успадковуються майже всі інші базові моделі. Таким чином BaseEntity надає їм Id.

#### 2.2.2.3 Клас Project

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 2.8 – Клас Project

Складається з:

* Публічної властивості Name (string)
* Публічної властивості Description (string)
* Публічної властивості Tasks (List<Task>)
* Публічної властивості Workers (List<User>)

Цей клас потрібен для реалізації проектів, а також пов’язаних з ними взаємодій.

#### 2.2.2.4 Клас Task

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 2.9 – Клас Task

Складається з:

* Публічної властивості Name (string)
* Публічної властивості Description (string)
* Публічної властивості EstimatedTime (DateTime)
* Публічної властивості Status (Enum Status)
* Публічної властивості Priority (Enum Priority)
* Публічної властивості Attachments (List<Attachment>)

Цей клас використовується для реалізації задач, а також для відносин у програмі.

#### 2.2.2.5 Клас User

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 2.10 – Клас User

Складається з:

* Публічної властивості FirstName (string)
* Публічної властивості LastName (string)
* Публічної властивості Email (string)
* Публічної властивості PasswordHashed (string)
* Публічної властивості Notifications (List<string>)
* Публічної властивості Tasks (List<Task>)

Цей клас використовується для реалізації працівників у програмі. Він також потрібен для взаємодії з об’єктами інших класів.

#### 2.2.2.6 Клас UserProjectRole

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рис. 2.11 – Клас UserProjectRole

Складається з:

* Публічної властивості ProjectId (int)
* Публічної властивості UserId (int)
* Публічної властивості Duty (Enum Duty)

Цей клас потрібен для реалізації відносин між працівником та проектом.

### 2.2.3 Інтерфейс IStorage<T>

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рис. 2.12 - Інтерфейс IStorage<T>

Складається з:

* Асинхронного метода GetAllAsync
* Асинхронного метода GetByIdAsync
* Асинхронного метода GetByPredicateAsync
* Асинхронного метода AddAsync
* Асинхронного метода UpdateAsync
* Асинхронного метода RemoveAsync

Цей інтерфейс використовується на рівні BLL для реалізації зберігання інформації. Класс Storage успадковується від цього інтерфейсу. Через процес серіалізації, інформація записується у Json файли. Для кожного класу, який успадковується від BaseEntity, свій Json файл.

### 2.2.4. Інтерфейси сервісів

#### 2.2.4.1 Інтерфейс IGenericService<T>

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рис. 2.13 - Інтерфейс IGenericService<T>

Складається з:

* Асинхронного метода Add
* Асинхронного метода Remove
* Асинхронного метода GetById
* Асинхронного метода GetAll
* Асинхронного метода GetByPredicate
* Асинхронного метода Update

Цей інтерфейс використовується для зв’язку BLL та функціоналу DAL.

##### 2.2.2.4.2 Інтерфейс IProjectService

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рис. 2.14 - Інтерфейс IProjectService

Складається з:

* Асинхронного метода CreateProject
* Асинхронного метода GetProjectByTask

Цей інтерфейс надає доступ до класу реалізації проекту для рівня UI.

##### 2.2.2.4.3 Інтерфейс ITaskService

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 2.15 - Інтерфейс ITaskService

Складається з:

* Асинхронного метода OpenFolder
* Асинхронного метода CreateAttachment
* Асинхронного метода CheckAvailibilityForUser

Цей інтерфейс надає доступ до класу реалізації задачі для рівня UI.

##### 2.2.2.4.4 Інтерфейс IUserProjectRoleService

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рис. 2.16 - Інтерфейс IUserProjectRoleService

Складається з:

* Асинхронного метода CreateTableRow
* Асинхронного метода GetTableByProjectId
* Асинхронного метода SetUserRole
* Асинхронного метода GetRowByProjectUser

Цей інтерфейс надає доступ до класу реалізації зв’язку проекту та користувача для рівня UI.

# 3 ІНСТРУКЦІЯ КОРИСТУВАЧА

3.1 Встановлення програми  
  
 Для запуску програми треба буде зайти у .sln файл СourseWork.sln та запустити проект UI у Visual Studio. Всі додаткові файли та папки, програма створить у процесі взаємодії із користувачем.

## 3.2 Функціонал програми та інструкція до неї

Після виконання дій, зазначених вище, користувача зустріне консоль з ось таким написом:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 3.1 – Початкове меню

Користувач має обрати одну з трьох опцій та натиснути на потрібну цифру на клавіатурі. Після цього натиснути на Enter.

При натисканні на «3», програма завершить свою роботу.

### 3.2.1 Робота зі сторінкою «Sign Up»

Після обирання опції «1», в користувача послідовно запитають Ім’я, Фамілію, Адресу електронної пошти та пароль. У разі, якщо це перший користувач, користувача також попросять ввести данні для створення першої задачі (Назву, Опис).

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 3.2 – Меню створення проекту

В такому випадку, користувачу автоматично дадуть тег «StateManager». У разі, якщо це не перший користувач, програма дасть тег «Unassigned» і не попросить створити проект. Після цього, буде запущено меню користувача.

### 3.2.2 Робота зі сторінкою «Sign In»

Після обирання опції «2», в користувача послідовно запитають email та пароль.

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рис. 3.3 – Log In

Після вводу даних, користувач перейде до меню користувача.

### 3.2.3 Робота з меню користувача

Користувача зустрічає такий екран:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 3.4 – Меню користувача

Зверху будуть виведені повідомлення, якщо вони з’являться (там де зараз написано «No notifications»).   
Користувач обирає одну з трьох опцій.

* «Log Out» - переносить користувача в початок програми.
* «Choose Project» - активує меню вибору проекту.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рис. 3.5 – Меню вибору проекту

Після обирання проекту, користувач переходить до меню UI, який обирається відповідно до тегу користувача.

* «See Notifications» - додатково показує повідомлення.

#### 3.2.3.1 Меню «StateManager»

Користувач бачить наступне меню: Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.6 - Меню «StateManager»

Тут користувач може обрати одну з трьох опцій:

* «Add Project» - виконується той же самий код, що описано вище у «Sign Up» (див. Рис. 3.2)
* «Edit current Project» - відкриває меню редагування проекту
* «Exit» - повертається до попереднього меню

##### 3.2.3.1.1 Меню редагування проекту

Після переходу через «Edit current Project», користувач побачить наступне меню:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.7 – Меню редагування проекту

Тут користувач обирає один з варіантів:

* + Змінити назву проекту («Edit Project Name»)

Тут програма попросить користувача ввести нову назву проекту.

Изображение выглядит как текст, Шрифт, снимок экрана, дизайн

Автоматически созданное описание

Рис. 3.8 – Зміна назви

* + Змінити опис проекту («Edit Project Description»)

Тут програма попросить користувача ввести новий опис проекту.  
Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

Рис. 3.9 – Зміна опису

* + Створити задачу («Create Assignment»)

Тут програма просить користувача ввести всі необхідні данні для створення задачі:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.10 – Створення задачі

* + Налаштувати задачі («Edit Assignments»(1))
  + Налаштувати працівників («Edit Workers»(2))
  + Вийти («Exit» - повертає до попереднього методу)

###### 3.2.3.1.1.1 Налаштування задачі

Після переходу до меню редагування задачі, користувача попросять обрати задачу:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.11 – Меню обирання задачі

Після того як користувач натисне відповідну цифру, він побачить меню редагування задачі.

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рис. 3.12 – Меню редагування задачі

Тут користувач може обрати одну з трьох опцій.

* Назначити користувача на завдання («Assign worker to this Assignment»). Користувач обирає одного з працівників на проекті:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 3.13 – Меню вибору працівника

Відповідно до вибору користувача, статус програми зміниться на відповідний до нового працівника.

* Закрити задачу («Close Assignment»). Програма визначить статус задачі, як «Done»
* Вийти («Exit» - повертає до попереднього методу)

###### 3.2.3.1.1.2 Меню редагування працівників

Користувач переходить до меню вибору взаємодії з працівниками.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 3.14 - Меню редагування працівників

Тут користувач може обрати одну з чотирьох опцій.

* Прибрати користувача із проекту («Remove Worker from Project»). Користувач обирає працівника з проекту (див. Рис. 3.13).
* Додати користувача до проекту («Assign Worker for Project»). Користувач обирає працівника (див. Рис. 3.13).
* Дати тег («Assign Duty»). Користувач обрає працівника (див. Рис. 3.13), а потім обирає тег: Изображение выглядит как текст, снимок экрана, Шрифт

  Автоматически созданное описание

Рис. 3.15 – Меню обирання тегу

* Вийти («Exit» - повертає до попереднього методу)

#### 3.2.3.2 Меню «Developer» та «Tester»

Ці меню мають однаковий функціонал. Після обирання проекту, у разі, якщо користувач має один із вище зазначених тегів, він побачить так меню: Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 3.16 – Меню «Developer» та «Tester»

Користувач обирає одну з двох опцій:

* «Choose Assignment» - переносить користувача до меню обирання задачі (див. Рис. 3.11). Після цього користувач побачить меню взаємодії із задачею.
* «Exit» - повертає до попереднього методу.

##### 3.2.3.2.1 Меню взаємодії із задачею

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.17 – Меню взаємодії із задачею

Тут користувач має обрати одну з наданих опцій.

* «Hand Assignment» - ідентично до «Assign worker to this Assignment» тегу «SateManager». (див. 3.2.3.1.1.1)
* «Upload File». Програма запитає шлях до файлу. Изображение выглядит как текст, Шрифт, снимок экрана, линия

  Автоматически созданное описание

Рис. 3.18 – Напис шляху до файлу

* «Open File Folder» - відкриває провідник на комп’ютері у директорії файлів задачі.
* «Exit» - повертає до попереднього методу.

**ВИСНОВОК**

У ході виконання курсової роботи на тему «Менеджер проектів» була розроблена система на платформі .NET 7.0 з консольним інтерфейсом для користувачів. Система розроблена з урахуванням принципів безпеки та захисту даних.

Система надає можливість реєстрації нових користувачів та авторизації існуючих користувачів. Також система дозволяє створювати, редагувати та закривати задачі та проекти. Користувачі можуть відстежувати прогрес виконання задач та отримувати сповіщення про зміни у статусах задач.

У подальшому систему можна удосконалити збільшенням функціоналу, а також оптимізацією деяких методів, переносом програми на більш зручний UI та використанням бази даних замість серіалізації у Json.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Microsoft .Net Fundamentals documentation – документація Microsoft. URL: <https://learn.microsoft.com/en-us/dotnet/fundamentals/> (дата звернення: 29.05.2023)