

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”

Завідувач кафедри кібербезпеки  
та програмного забезпечення

д.т.н., професор

\_\_\_\_\_ Олексій СМІРНОВ

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“ Програмне забезпечення менеджера задач для планування**  
**проектів ”**

Виконав здобувач вищої освіти

IV курсу, групи КІ-20

спеціальності 123 «Комп’ютерна інженерія»

\_\_\_\_\_ Кривохижа В. Ю.

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

Керівник проекту

доктор технічних наук, професор

\_\_\_\_\_ Мелешко Є. В.

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

Рецензент \_\_\_\_\_

м. Кропивницький

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Освітній ступінь бакалавр  
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.т.н., проф.  
О.А.Смірнов  
«\_\_» \_\_\_\_\_ 20\_\_ року

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

*Кривохижі Віталію Юрійовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення менеджера задач для планування проектів*

керівник роботи *Мелешко Єлизавета Владиславівна, д-р техн. наук, професор*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 131-02 від 01.04.2024 року

2. Строк подання студентом роботи до захисту 19.05.2024 р.

3. Мета та завдання кваліфікаційної бакалаврської роботи: *Метою розробки є програмне забезпечення менеджера задач для планування проектів*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію.*

*6. Висновки*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

*Структурна схема системи* 1 аркуш

*Функціональна схема системи* 1 аркуш

*Діаграма процесів* 1 аркуш

*Блок-схема алгоритму роботи додатку* 2 аркуша

6. Дата видачі завдання «17» січня 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	22.05.2024 р.	

**Студент** \_\_\_\_\_

( підпис )

\_\_\_\_\_ (прізвище та ініціали)

**Керівник роботи** \_\_\_\_\_

( підпис )

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Кривохижа В.Ю. Програмне забезпечення менеджера задач для планування проєктів. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

У даній кваліфікаційній бакалаврській роботі розроблено програмне забезпечення, яке призначено для планування проєктів.

Метою роботи є створення програмного забезпечення менеджера задач для планування проєктів.

Результат роботи – програмна реалізація системи для планування проєктів.

В процесі роботи над реалізацією системи виконано дослідження існуючих методів, алгоритмів та програмних засобів. Розроблено та реалізовано власне програмне забезпечення, здійснено опис всіх його компонентів.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено на мові програмування C# (C Sharp) з використанням платформи .NET Framework.

**Ключові слова:** комп'ютерна інженерія, таск менеджер, система керування завданнями, графічний інтерфейс, інтерфейс користувача

## ABSTRACT

**Kryvokhyzha V.Y. Software for Task Manager for Project Planning. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

This bachelor's qualification work presents the development of software designed for project planning.

The aim of the work is to create software for task manager for project planning.

The outcome of the work is the software implementation of a system for project planning.

During the process of system implementation, research on existing methods, algorithms, and software tools was conducted. The development and implementation of proprietary software were carried out, and a description of all its components was provided.

A user-friendly interface was developed, and instructions for working with the software tools were provided.

The program can be used on IBM PC architecture computers with Windows 10/11 operating systems.

The program was developed using the C# (C Sharp) programming language with the .NET Framework platform.

**Keywords:** computer engineering, task manager, task management system, graphical interface, user interface

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ ..	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	7
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування .....	16
2.3 Розгорнута постановка завдання .....	19
3 ОПИС І ОБґРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	21
3.1 Опис функціонування системи .....	21
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми .....	27
3.4 Розробка діаграми процесів.....	29
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	32
4.1 Блок-схеми та опис алгоритмів функціонування системи.....	32
4.2 Захист розробленого програмного забезпечення.....	36
5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	38
6 ОСНОВНІ ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	46

					ВКРБ-123.24.0008.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Кривохижа В.Ю				Програмне забезпечення менеджера задач для планування проєктів	Літ.	Аркуш	Аркушів
Перев.	Мелешко Є. В.					П	1	51
Н.контр.	Коваленко А.С.				KI-20			
Затв.	Смірнов О. А							

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БР	–	Бакалаврська робота
GUI	–	Графічний інтерфейс користувача
CRUD	–	Створення, читання, оновлення, видалення
IDE	–	Інтегроване середовище розробки
і т.д.	–	і так далі
DIY	–	Зроби сам
API	–	Інтерфейс програмування додатків
OOP	–	Об'єктно-орієнтоване програмування
ПЗ	–	Програмне забезпечення
ОС	–	Операційна система
ПЕОМ	–	Персональна обчислювальна машина
ПК	–	Персональний комп'ютер
RAM	–	Оперативна пам'ять
ОС	–	Операційна система
ТЗ	–	Технічне завдання
хв.	–	Хвилина
SQL	–	Мова структурованих запитів
IDE	–	Інтегроване середовище розробки

## ВСТУП

**Актуальність теми** в сучасному світі, де швидкість та ефективність є вирішальними факторами успіху, управління проектами стає все більш важливою складовою організаційного успіху. Це особливо актуально в умовах швидкої зміни технологій, вимог ринку та конкуренції. Ось чому розробка програмного забезпечення для оптимізації управління завданнями та планування проектів набуває все більшого значення.

Актуальність цієї теми полягає в тому, що існуючі інструменти управління проектами часто не відповідають потребам сучасних бізнес-ситуацій. Необхідність працювати в режимі багатозадачності, швидко реагувати та координувати різні аспекти проекту вимагає нових підходів до розробки програмного забезпечення. Такі системи можуть значно полегшити роботу менеджерів проектів і допомогти підвищити продуктивність і якість виконання проектів.

**Мета й завдання дослідження.** Метою даної дипломної роботи є розробка та реалізація програмного забезпечення, яке дозволить ефективно керувати задачами та планувати проекти. Для досягнення цієї мети необхідно вирішити наступні завдання:

- Аналіз існуючих рішень у галузі управління задачами та планування проектів.
- Визначення вимог до програмного забезпечення з урахуванням потреб користувачів.
- Проектування архітектури та інтерфейсу користувача програмного забезпечення.
- Реалізація програмного забезпечення з урахуванням сучасних технологій та практик програмування.
- Тестування та валідація розробленого програмного забезпечення.

					ВКРБ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Результати цього дослідження матимуть практичне значення для менеджерів проектів у різних галузях, де важливо швидко реагувати на зміни та ефективно управляти завданнями. Крім того, успішне впровадження такого програмного забезпечення може сприяти підвищенню конкурентоспроможності організації та загальному розвитку проектного менеджменту в Україні.

Таким чином, дана робота є важливою для подальшого розвитку інформаційних технологій в управлінні проектами та має потенціал позитивно вплинути на ефективність бізнес-процесів у різних сферах діяльності.

**Практична цінність отриманих результатів.** Результати цього дослідження мають велике практичне значення. Розроблене програмне забезпечення може бути використане в різних галузях, включаючи медицину, біологію та біотехнологію. У медицині біоструми відіграють важливу роль у багатьох біологічних процесах і тому можуть бути використані для діагностики та лікування різних захворювань. У біології та біотехнології це може підвищити якість досліджень, пов'язаних з вивченням біострумів.

Програмне забезпечення готове до використання і може бути розширене для застосування в різних установах та організаціях. В Україні, де біотехнологічна галузь продовжує розвиватися, це може мати значний вплив на розвиток цієї галузі. Загалом, результати цього дослідження мають великий потенціал для практичного застосування в різних галузях. Подальше впровадження цих результатів може сприяти підвищенню якості життя та розвитку суспільства.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Призначенням розроблюваної системи є надання інструментів для ефективного управління задачами та планування проєктів. Система буде розроблена з метою спрощення процесів управління завданнями, підвищення продуктивності та забезпечення вчасного виконання проєктів. Головною метою системи є забезпечення менеджерів проєктів та їх команд ефективним інструментом для планування, прийняття рішень та виконання завдань.

Розроблювана система орієнтована на вирішення таких завдань, як:

- Планування та розподіл завдань між учасниками проєкту.
- Контроль за виконанням завдань та визначення статусу виконання.
- Визначення та пріоритизація задач з урахуванням їх важливості та термінів виконання.
- Забезпечення комунікації та спільної роботи команди над проєктами.
- Збір та аналіз даних про виконання проєктів для подальшого вдосконалення робочого процесу.

## 1.2 Область застосування

Розроблювана система може бути застосована в широкому спектрі галузей та організацій. Основні області застосування включають:

- **Інформаційні технології.** У сфері ІТ система може використовуватися для управління розробкою програмного забезпечення, тестуванням, впровадженням та супроводженням проєктів.

					ВКРБ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

– **Бізнес та управління проєктами.** У бізнесі система може бути використана для планування та контролю над виконанням проєктів будь-якої складності, від невеликих ініціатив до великих корпоративних проєктів.

– **Освіта.** В галузі освіти система може використовуватися для організації навчальних проєктів, планування роботи студентів та викладачів, а також контролю за їх виконанням.

– **Наукова діяльність.** Дослідники та науковці можуть використовувати систему для управління процесами досліджень, планування експериментів та аналізу результатів.

– **Промисловість.** У промисловості система може застосовуватися для управління виробничими процесами, планування обслуговування обладнання та технічного обслуговування.

Система, яку я розробив, гнучка та кастомізована для використання в різних галузях та організаціях. Вона корисна як для великого, так і для малого бізнесу, який потребує ефективного управління проєктами та завданнями.

КБПЗ - 2024

					ВКРБ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

В даний час на ринку існує багато програмних рішень, спрямованих на управління задачами та планування проєктів. Для проведення аналізу ми розглянемо кілька з них:

– **Asana.** Asana є одним з найпопулярніших інструментів управління задачами. Він пропонує широкий спектр функцій, включаючи створення завдань, розподіл їх між учасниками проєкту, планування термінів виконання та створення графіків.

– **Trello.** Trello базується на концепції дошок та карток, що дозволяє користувачам створювати завдання та переміщати їх між колонками, щоб відслідковувати їх статус та прогрес.

– **Microsoft Project.** Це програмне забезпечення для управління проєктами, яке надає більш розширений набір функцій, таких як гантівські діаграми, ресурсне планування та аналіз вартості проєкту.

– **Todoist.** Todoist - це простий та ефективний інструмент для керування особистими та командними задачами, який пропонує інтеграцію з різними платформами та пристроями.

Аналізуючи переваги та недоліки існуючих рішень, можна визначити, що багато з них мають обмеження щодо гнучкості налаштування, складності інтерфейсу користувача або вартості використання. Більшість систем зосереджуються на певних аспектах управління проєктами та не надають повного спектру інструментів для всіх потреб користувачів.

					ВКРБ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Обґрунтування необхідності розробки власної системи полягає в тому, що існуючі рішення не завжди відповідають всім вимогам користувачів. Розробка власного програмного забезпечення дозволить нам створити гнучкий, ефективний та користувач орієнтований інструмент, який задовольнить усі потреби управління задачами та плануванням проєктів. Враховуючи проведений аналіз існуючих рішень, можемо стверджувати, що розробка власної системи є доцільною та необхідною для забезпечення оптимального управління проєктами.

Перед тим як перейти до детального огляду існуючих систем, важливо відзначити, що на ринку існує значна кількість програмних рішень, спрямованих на управління задачами та планування проєктів. Кожна з цих систем має свої переваги та недоліки, і кожна спрямована на вирішення певного спектру завдань. Під час аналізу існуючих рішень, ми докладемо зусиль для з'ясування їхніх переваг та обмежень, щоб обґрунтувати необхідність розробки власної системи, яка максимально відповідатиме потребам користувачів у сфері управління проєктами.

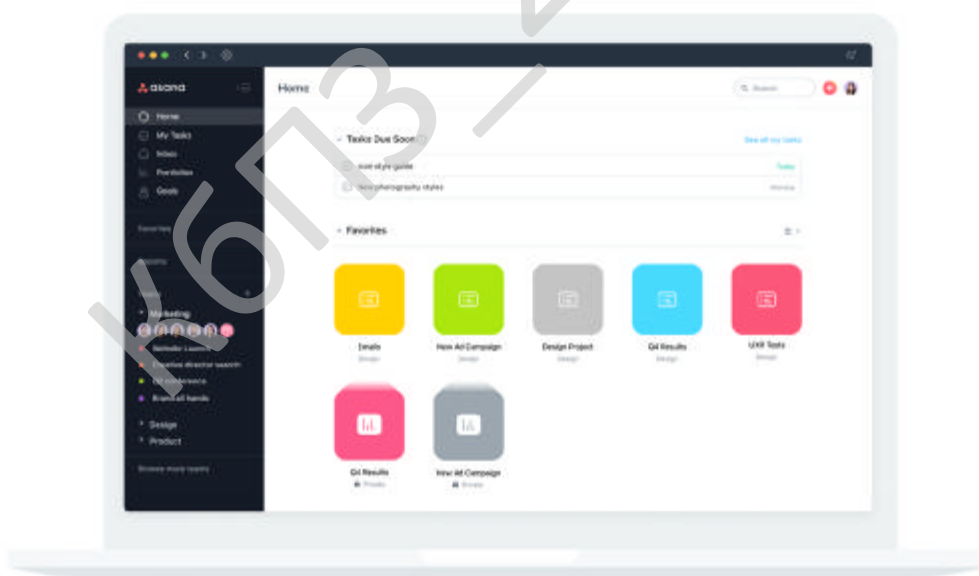


Рисунок 2.1 – Asana (інструмент управління задачами)

**Asana** - це онлайн-платформа для управління проєктами та організації робочих завдань, яка дозволяє командам ефективно співпрацювати над проєктами,

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

встановлювати терміни виконання та відстежувати прогрес. Asana відома своєю простотою використання та широким спектром функцій, що дозволяють користувачам легко організувати свою роботу та досягати поставлених цілей.

#### **Плюси Asana:**

– Простий та інтуїтивно зрозумілий інтерфейс. Asana має чистий та зрозумілий інтерфейс, який дозволяє користувачам швидко орієнтуватися та почати працювати.

– Широкий спектр функцій. Платформа пропонує різноманітні функції, включаючи створення завдань, розподіл їх між учасниками проєкту, створення графіків, установку термінів виконання, створення звітів та багато іншого.

– Спільна робота та комунікація. Asana дозволяє користувачам спільно працювати над завданнями, ділитися коментарями, прикріплювати файли та спілкуватися в реальному часі.

– Інтеграція з іншими сервісами. Asana підтримує інтеграцію з різними сервісами та інструментами, такими як Google Drive, Slack, Dropbox, що забезпечує ще більшу ефективність управління завданнями.

#### **Мінуси Asana:**

– Складність для новачків. Для новачків може знадобитися певний час, щоб звикнути до інтерфейсу та зрозуміти всі можливості платформи.

– Обмежені можливості безкоштовної версії. Деякі продуктивні функції, такі як розширені звіти та додаткові налаштування, доступні тільки у платних версіях Asana.

#### **Особливості Asana:**

– Проекти та завдання. Asana дозволяє користувачам створювати проекти та завдання, розподіляти їх між учасниками та встановлювати терміни виконання.

– Календарні види. Користувачі можуть переглядати свої завдання у вигляді календаря, що дозволяє краще орієнтуватися у часі та планувати робочі завдання.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## Чому люди використовують Asana.

Asana приваблює користувачів своєю простотою використання, широким спектром функцій та можливістю спільної роботи в командах. Він допомагає організувати робочі процеси, відстежувати прогрес та досягати поставлених цілей.

## Доступність та вартість.

Asana доступний для використання як у безкоштовній версії, так і у платних версіях з розширеними можливостями.

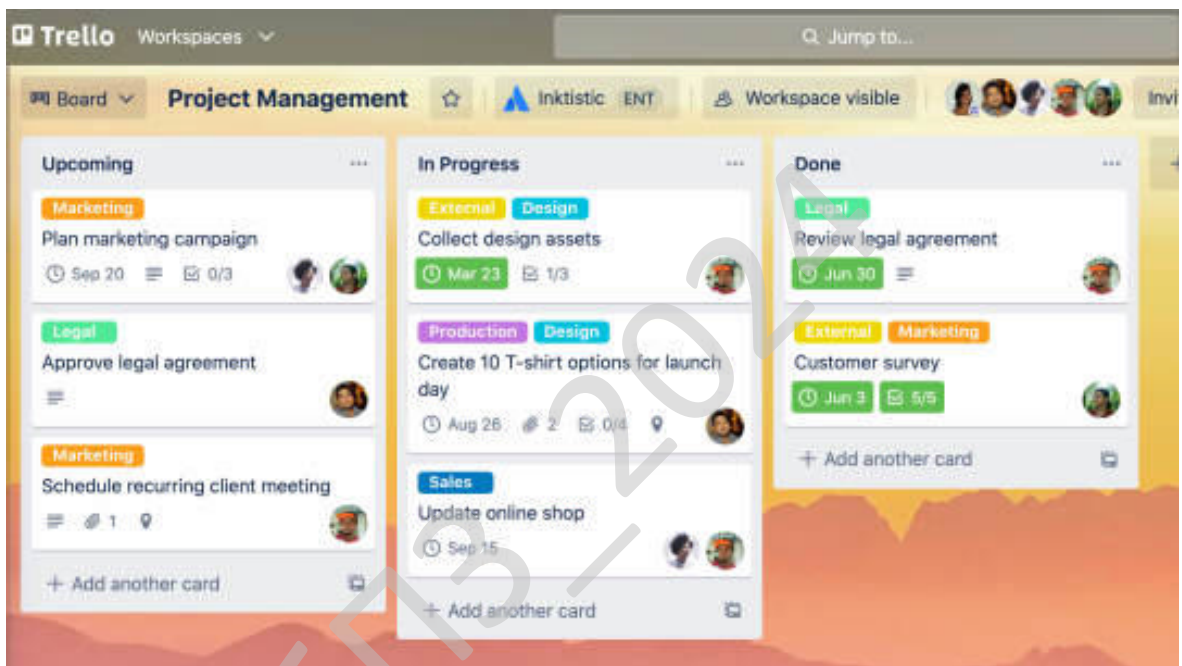


Рисунок 2.2 – Trello (зовнішній вигляд)

**Trello** - це веб-платформа та мобільний додаток, який базується на концепції дошок та карток, що дозволяє користувачам організовувати та відстежувати різні види завдань та проєктів. Він відомий своєю простотою та гнучкістю, що робить його популярним серед користувачів усього світу.

## Плюси Trello:

- Концепція дошок та карток. Trello використовує просту та зрозумілу концепцію дошок, на яких користувачі можуть створювати картки для кожного завдання або етапу проєкту та переміщувати їх між колонками.
- Гнучкість та налаштування. Trello дозволяє користувачам налаштовувати

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

свої дошки та картки, використовуючи мітки, терміни, коментарі та інші функції для організації та відстеження робочих процесів.

– Спільна робота в реальному часі. Trello дозволяє користувачам спільно працювати над проєктами в реальному часі, ділитися картками та дошками з колегами та співробітниками.

– Інтеграція з іншими сервісами. Trello має велику кількість інтеграцій з іншими сервісами та інструментами, такими як Google Drive, Slack, Dropbox, що робить його додатково зручним для користувачів.

### **Мінуси Trello:**

– Обмежені можливості безкоштовної версії. Деякі продуктивні функції, такі як додаткові інтеграції та розширені можливості звітності, доступні тільки у платній версії Trello.

– Обмежені можливості для великих проєктів. Для дуже складних та великих проєктів Trello може бути менш ефективним порівняно з деякими іншими інструментами управління завданнями.

### **Особливості Trello:**

– Дошки та картки. Trello базується на концепції дошок та карток, що дозволяє користувачам створювати, організовувати та переміщувати завдання між різними етапами.

– Шаблони та плагіни. Trello має велику кількість шаблонів та плагінів, які дозволяють користувачам швидко створювати нові дошки та розширювати функціональність.

### **Чому люди використовують Trello:**

Trello приваблює користувачів своєю простотою використання, гнучкістю та можливістю спільної роботи. Люди використовують Trello з різних причин, і ось деякі з них:

– **Простота використання.** Один з головних плюсів Trello - це його інтуїтивно зрозумілий та простий інтерфейс. Новачкам легко освоїти платформу, а досвідчені користувачі можуть швидко налаштувати свої дошки та почати

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11



**Microsoft Project** - це програмне забезпечення для управління проектами, розроблене компанією Microsoft. Воно дозволяє користувачам планувати, відстежувати та керувати проектами будь-якої складності, надаючи широкий спектр інструментів для організації та контролю процесів розробки.

#### **Плюси Microsoft Project:**

- Розширені функції планування. Microsoft Project надає розширені можливості для планування проектів, включаючи створення графіків Ганта, встановлення залежностей між завданнями, розподіл ресурсів та бюджетування.
- Широкі можливості звітності. Програма дозволяє користувачам генерувати різноманітні звіти та діаграми, які допомагають аналізувати прогрес, ризики та витрати проекту.
- Інтеграція з іншими продуктами Microsoft. Microsoft Project інтегрується з іншими програмами Microsoft, такими як Microsoft Teams, SharePoint, Excel, що спрощує спільну роботу та обмін інформацією між різними програмами.
- Співпраця в реальному часі. Microsoft Project дозволяє користувачам спільно працювати над проектами в реальному часі, вносячи зміни та оновлення, які відображаються усім учасникам команди.

#### **Мінуси Microsoft Project:**

- Висока вартість та складність. Програмне забезпечення вимагає значних витрат на придбання та встановлення, а також часу для навчання користувачів використовувати його на повну потужність.
- Не підходить для невеликих проектів. У випадку невеликих проектів або команд, Microsoft Project може бути надто складним та витратним для використання.

#### **Особливості Microsoft Project:**

- Подробний графік Ганта. Microsoft Project дозволяє користувачам створювати детальні графіки Ганта з розкладом завдань та залежностями між ними.
- Ресурсне управління. Користувачі можуть визначати та розподіляти ресурси (людські, матеріальні) для кожного завдання та контролювати їхнє

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

використання.

### **Чому люди використовують Microsoft Project.**

Microsoft Project використовується для керування проектами у великих компаніях, де необхідна детальна організація та великий обсяг даних. Його широкі можливості та інтеграція з іншими програмами Microsoft роблять його популярним вибором для управлінців та професіоналів з управління проектами.

### **Доступність та вартість.**

Microsoft Project доступний як частина пакетів програмного забезпечення Microsoft Office, а також може бути придбаний окремо. Ціни можуть варіюватися в залежності від типу ліцензії та функціональних можливостей.

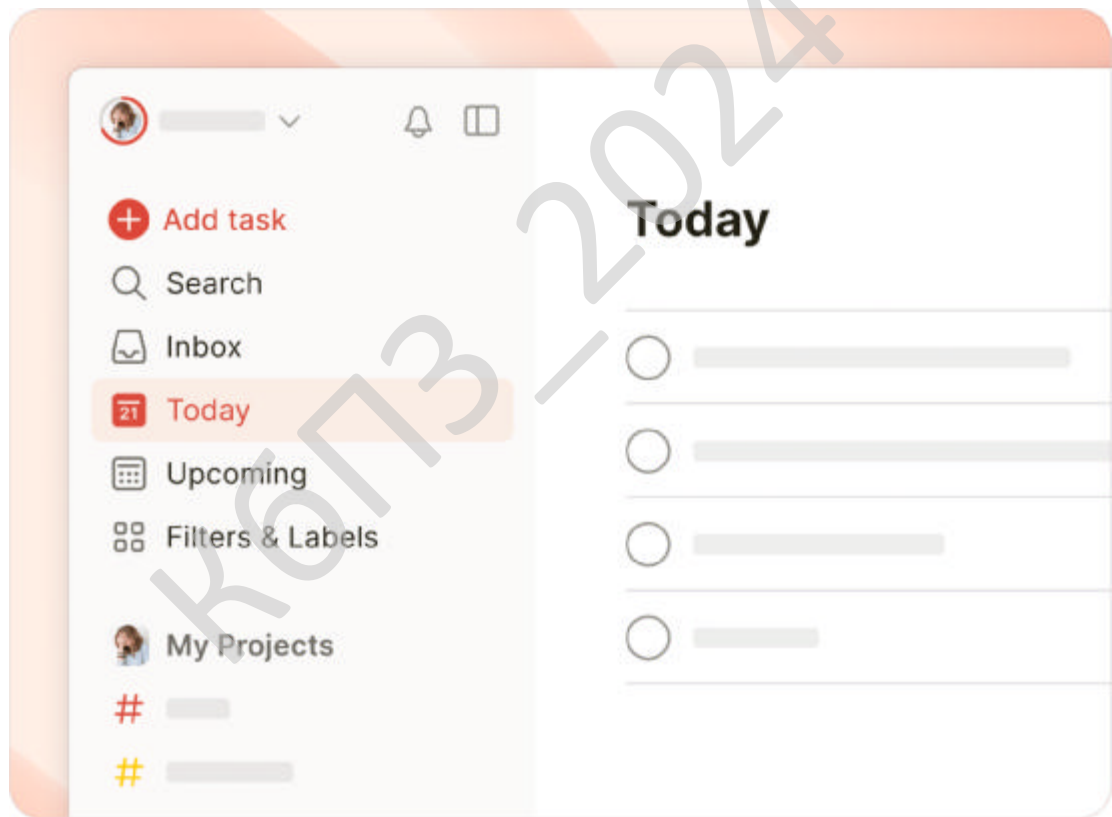


Рисунок 2.4 – Todoist (зовнішній вигляд)

**Todoist** - це популярний онлайн сервіс та мобільний додаток для керування завданнями та списками справ. Завдяки своїй простоті, ефективності та доступності на різних платформах, Todoist завоював велику популярність серед користувачів усього світу.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

### **Плюси Todoist:**

- Простий та інтуїтивно зрозумілий інтерфейс. Todoist пропонує чистий та простий інтерфейс, що дозволяє користувачам легко створювати, організовувати та відстежувати завдання.
- Кросплатформенність. Todoist доступний на багатьох платформах, включаючи веб-версію, десктопні додатки для Windows та macOS, а також мобільні додатки для iOS та Android. Це робить його доступним для користувачів будь-де та будь-коли.
- Можливості спільної роботи. Todoist дозволяє користувачам ділитися списками завдань з іншими користувачами, що робить його ідеальним інструментом для спільної роботи над проектами та завданнями в команді.
- Гнучка система пріоритетів та термінів виконання. Todoist дозволяє встановлювати пріоритети для завдань, визначати терміни виконання та створювати нагадування, що допомагає користувачам організувати свій час та пріоритетизувати роботу.

### **Мінуси Todoist:**

- Обмежені можливості безкоштовної версії. Деякі продуктивні функції, такі як залучення до спільних проектів та створення нагадувань за межами дати, доступні тільки у платній версії Todoist.
- Не дуже розширені функції для великих проектів. Для складних та великих проектів Todoist може бути менш ефективним порівняно з деякими іншими інструментами управління завданнями.

### **Особливості Todoist:**

- Проекти та підзавдання. Todoist дозволяє створювати проекти та підзавдання, що дозволяє користувачам структурувати свої завдання більш детально.
- Мітки та фільтри. Користувачі можуть використовувати мітки та фільтри для організації та швидкого пошуку завдань за певними критеріями.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

## **Чому люди використовують Todoist.**

Todoist приваблює користувачів своєю простотою використання, гнучкістю та доступністю на різних платформах. Він допомагає організувати робочий процес, відстежувати завдання та досягати поставлених цілей.

### **Доступність та вартість.**

Todoist доступний для використання як у безкоштовній версії, так і у преміум-версії з розширеними можливостями за щомісячну або щорічну плату. Це робить його доступним для широкого кола користувачів з різними потребами та бюджетами.

## **2.2 Обґрунтування вибору засобів для побудови системи та мови програмування**

Task Manager, який розробляється в рамках даної дипломної роботи, обрано розробляти на мові програмування C# з використанням технології Windows Forms в середовищі .NET Framework. Цей вибір обґрунтовується рядом переваг та характеристик, які забезпечують ефективність, зручність та швидкість розробки, а також надійність та можливості майбутньої підтримки.

### **Характеристики програми включають:**

- **Ефективність.** Мова програмування C# та технологія Windows Forms забезпечують високу швидкість розробки завдяки своїм інтегрованим засобам роботи з інтерфейсом користувача та багатофункціональним середовищем розробки Visual Studio.
- **Зручність інтерфейсу.** Windows Forms дозволяє створювати інтерактивні та інтуїтивно зрозумілі графічні інтерфейси користувача, що сприяє зручності використання програми.
- **Масштабованість.** C# та .NET Framework забезпечують можливості для масштабування програми, що дозволяє легко додавати нові функції та розширювати її функціональність в майбутньому.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

– **Підтримка.** Мова C# та технологія Windows Forms є частиною екосистеми Microsoft, що гарантує підтримку, оновлення та розвиток програмних засобів на протязі тривалого часу.

– **Багатофункціональність.** Засоби .NET Framework дозволяють використовувати різноманітні бібліотеки та функціонал, що спрощує розробку та реалізацію різноманітних завдань в програмі.

Обрана комбінація мови C# та технології Windows Forms .NET дозволяє розробити потужний, ефективний та зручний у використанні Task Manager, який відповідає вимогам сучасних програмних рішень.

Windows Forms є одним із фреймворків для розробки графічних інтерфейсів користувача (GUI) в середовищі програмування на мові C#. Він входить до складу платформи .NET Framework, яка надає засоби для розробки різноманітних додатків для платформи Windows.

#### **Плюси використання Windows Forms:**

– **Простота використання.** Windows Forms надає простий та зрозумілий інтерфейс для розробки графічних інтерфейсів користувача. Він базується на концепції робочих форм, що дозволяє швидко створювати та налаштовувати елементи управління.

– **Широкий вибір елементів управління.** Windows Forms має великий набір вбудованих елементів управління, таких як кнопки, тексти, списки, таблиці тощо, що дозволяє легко створювати різноманітні інтерфейси користувача.

– **Швидкість розробки.** Завдяки високому рівню абстракції та готовим компонентам, розробка додатків на Windows Forms зазвичай відбувається значно швидше, ніж на низькорівневих платформах.

– **Інтеграція з Visual Studio.** Windows Forms інтегрований з Visual Studio, що є однією з найпопулярніших і потужних інтегрованих середовищ розробки (IDE), що значно спрощує розробку та налагодження додатків.

#### **Мінуси використання Windows Forms:**

– **Обмежена крос-платформенність.** Додатки, розроблені за допомогою

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Windows Forms, зазвичай працюють тільки на операційних системах Windows, що може бути недоліком у випадку, якщо потрібно підтримувати кілька платформ.

– Обмежені можливості макетування. Windows Forms пропонує обмежені можливості макетування і адаптивності, що може стати перешкодою для створення сучасних та адаптивних інтерфейсів.

#### **Чому варто користуватись Windows Forms:**

– Швидкість розробки. Windows Forms дозволяє швидко створювати прості та середньої складності GUI додатки без значних витрат часу на вивчення складних концепцій.

– Стабільність та підтримка. Він має стабільний API та підтримку від Microsoft, що забезпечує надійність та можливість підтримки додатків у майбутньому.

– Простота інтеграції з іншими технологіями Microsoft. Windows Forms легко інтегрується з іншими технологіями Microsoft, такими як ADO.NET для доступу до баз даних або ASP.NET для створення веб-додатків.

В цілому, Windows Forms є потужним і зручним інструментом для розробки клієнтських додатків під платформу Windows, забезпечуючи ефективність та швидкість розробки, а також надійність та стабільність.

#### **Відмінні риси Windows Forms:**

– Простота використання. Windows Forms надає простий та зрозумілий інтерфейс для розробки графічних інтерфейсів користувача, що робить його доступним для початківців та досвідчених розробників.

– Гнучкість. Windows Forms дозволяє створювати різноманітні інтерфейси користувача з використанням різноманітних елементів управління, що дозволяє реалізовувати різноманітні функціональність та дизайн.

– Інтеграція з Visual Studio. Windows Forms є інтегрованим у Visual Studio, що є одним з найпопулярніших середовищ розробки програмного забезпечення. Це забезпечує зручний та продуктивний процес розробки.

– Стабільність та підтримка. Windows Forms має стабільне API та

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

підтримку від Microsoft, що забезпечує надійність та можливість підтримки додатків протягом тривалого періоду.

#### **Технічні характеристики Windows Forms:**

- Мова програмування. Windows Forms підтримується мовою програмування C#, а також іншими мовами, що підтримують .NET Framework.
- Платформи. Додатки, розроблені на базі Windows Forms, працюють на операційних системах Windows, починаючи від Windows XP та вище.
- Інтеграція з .NET Framework. Windows Forms є частиною .NET Framework, що дозволяє використовувати його для розробки різноманітних додатків з використанням інших технологій та бібліотек.

#### **Програмне забезпечення, яке використовує Windows Forms:**

- Desktop Applications. Windows Forms використовується для розробки різноманітних клієнтських додатків, таких як управління даними, управління проектами, редактори тексту та багато іншого.
- Утиліти та інструменти. Windows Forms дозволяє розробляти різноманітні утиліти та інструменти для автоматизації рутинних завдань або роботи зі специфічними типами даних.
- Медіа-додатки. Windows Forms може бути використаний для розробки медіа-додатків, таких як аудіо- та відеопрогравачі, фотографічні редактори та інші.

Враховуючи вищесказане, мова програмування C# та Windows Forms – це ідеальний варіант для реалізації Task Manager.

### **2.3 Розгорнута постановка завдання**

В процесі розробки необхідно виконати наступний обсяг робіт:

- Обґрунтувати актуальність теми.
- Визначити призначення системи та область її застосування.
- Переглянути аналогічні існуючі системи з метою виявлення їх переваг та недоліків.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

– Аналіз вимог користувача. Проведення детального аналізу потреб користувачів у системі управління задачами та плануванні проєктів для визначення функціональних та нефункціональних вимог до програмного забезпечення.

– Проектування архітектури системи. Розробка архітектурної моделі програмного забезпечення, включаючи визначення компонентів, модулів, інтерфейсів та взаємозв'язків між ними.

– Реалізація функціоналу. Розробка програмного коду для реалізації основного функціоналу Task Manager, включаючи створення інтерфейсу користувача, роботу з базою даних, обробку подій та інші необхідні функції.

– Протестувати систему, щоб виявити та усунути можливі помилки та недоліки.

– Написати перелік умовних позначень, символів, одиниць та термінів, що будуть використані при написанні.

– Надати список літератури, що використалась при написанні.

– Написати анотацію на українській та англійській мовах.

КБПЗ-2024

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Тема БР розробка ПЗ для планування задач та спрощення життя.

Проаналізувавши існуючі аналогові системи та визначивши їх переваги, було прийнято рішення розробити ПЗ з виявленими перевагами та без недоліків для вирішення проблем, пов'язаних з технічними труднощами.

Основна функція даної системи - отримання, редагування та видалення даних. Кожному цьому пункту відповідає певна частина системи. Тому, основними критеріями функціонування системи має бути:

- Підключення необхідних бібліотек для роботи з Task Manager.
- Підключення бази даних.
- Налаштування системи авторизації лише для 2 користувачів хто має доступ до програми.
- Перевірка коректності додання даних користувача в базу.
- Перевірка коректності редагування даних користувача в базу.
- Оновлення сторінки програми для відображення останніх результатів.
- Завершення поточного циклу.
- Вимкнення програми та повторення алгоритму для продовження користування.

Цей алгоритм забезпечує стабільну та ефективну роботу системи Task Manager, дозволяючи користувачам ефективно керувати своїми завданнями та проектами.

Також значна частина коду міститься в бібліотеках. Використання бібліотек є невід'ємною частиною розробки програмного забезпечення, включаючи і систему Task Manager. Бібліотеки - це збірки попередньо написаного коду, який можна повторно використовувати у різних програмах або проектах. Ось кілька

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

переваг використання бібліотек:

– Економія часу. Використання бібліотек дозволяє уникнути написання коду з нуля для вирішення рутинних завдань або реалізації загальних функціональних можливостей. Це дозволяє розробникам скоротити час розробки та прискорити випуск готового продукту.

– Підвищення якості коду. Бібліотеки часто розробляються та тестуються професійними розробниками, що дозволяє покращити якість коду та зменшити кількість помилок.

– Стандартизація розробки. Використання популярних бібліотек дозволяє стандартизувати процес розробки та використовувати вже існуючі рішення для стандартних задач.

– Розширення можливостей. Бібліотеки часто містять різноманітні функції та алгоритми, які можуть розширити можливості програмного забезпечення та забезпечити новий функціонал.

Проте використання бібліотек також може мати свої недоліки, такі як залежність від стороннього коду, можливість виникнення конфліктів версій або обмеження в можливостях налаштування. Таким чином, розробники повинні збалансувати переваги та недоліки використання бібліотек для кожного конкретного проекту. У випадку системи Task Manager, використання бібліотек може спростити розробку та забезпечити потрібний функціонал для ефективного управління завданнями та проектами.

Система Task Manager розроблена з метою надання користувачам зручного та ефективного інструменту для управління завданнями та планування проектів. При запуску програми, користувачеві надається можливість авторизуватися за допомогою введення свого логіну та паролю. Після успішної авторизації, користувач має доступ до основного інтерфейсу системи, де він може переглядати, створювати, редагувати та видаляти завдання та проекти.

Система Task Manager забезпечує можливість додавання нових завдань, вказання їхнього пріоритету, терміну виконання та відповідального користувача.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Крім того, користувач може створювати нові проєкти та призначати завдання для кожного з них. Важливою функціональністю є можливість оновлення сторінки програми для відображення останніх результатів, що дозволяє користувачеві бачити актуальну інформацію без необхідності перезавантаження.

Усі дії користувача та зміни в системі логуються за допомогою мови програмування C#. Кожен вхід до системи, створення, редагування та видалення завдань чи проєктів фіксується та зберігається в базі даних. Це дозволяє зберігати історію дій користувачів та відновлювати інформацію в разі потреби.

Завершення роботи програми передбачає збереження всіх змін у базі даних та закриття з'єднання з нею. Після завершення циклу роботи програма готова до наступного запуску та продовження користування. Таким чином, система Task Manager надає користувачам зручний та надійний інструмент для ефективного управління завданнями та планування проєктів, забезпечуючи збереження історії та контроль над діями користувачів.

Для виведення інформації для користувача система Task Manager використовує компонент DataGridView. Цей компонент забезпечує зручний та ефективний спосіб відображення даних у вигляді табличної структури. Користувач може бачити всю необхідну інформацію про завдання та проєкти, включаючи назву, пріоритет, термін виконання, відповідального користувача та інші важливі дані.

DataGridView також забезпечує можливість редагування даних безпосередньо у таблиці, що робить взаємодію з системою Task Manager більш зручною та ефективною. Користувач може легко вносити зміни до завдань та проєктів, змінювати їхні пріоритети, терміни виконання та інші параметри без необхідності переходу до окремих форм або вікон.

DataGridView також надає можливість сортування та фільтрації даних, що дозволяє користувачам швидко знаходити потрібну інформацію та виконувати різноманітні аналітичні операції. Це робить систему Task Manager більш гнучкою та адаптивною до потреб користувачів.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Загалом, використання компонента DataGridView дозволяє забезпечити зручне та ефективне відображення даних для користувачів системи Task Manager, що сприяє покращенню їхнього досвіду користування та підвищенню продуктивності.

У процесі розробки програмного забезпечення менеджера задач для планування проєктів було використано базу даних для зберігання та управління даними. Основними елементами бази даних є дві таблиці: **account** та **todo**.

Таблиця **account** містить інформацію про доступ користувачів до системи. У цій таблиці зберігаються дані двох користувачів, які мають різні рівні доступу та ролі в системі. Таблиця містить поля для зберігання логінів, паролів та ролей користувачів, що дозволяє реалізувати механізм авторизації та забезпечити безпеку доступу до системи.

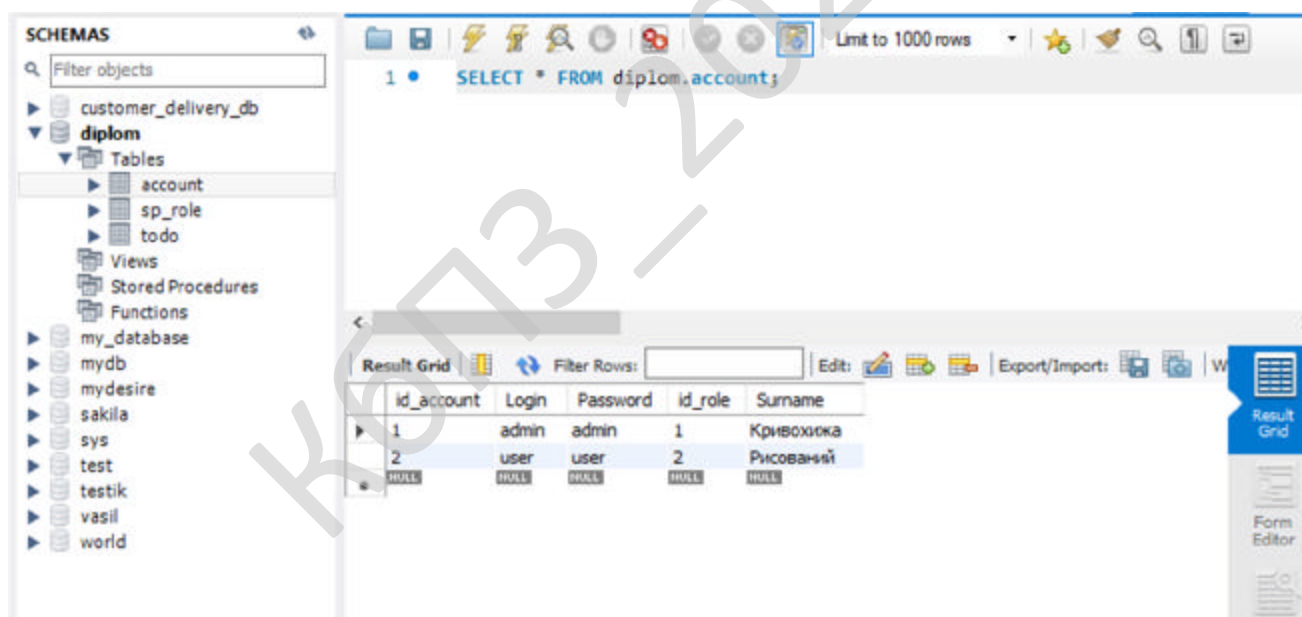


Рисунок 3.1 – Таблиця account

Основна таблиця **todo** призначена для зберігання всієї інформації про задачі, які потрібно виконати. У цій таблиці містяться поля для опису задач, дати початку, дати завершення та статусу виконання. Ця структура дозволяє ефективно організувати, відстежувати та управляти задачами, що сприяє підвищенню продуктивності та зручності використання програми.

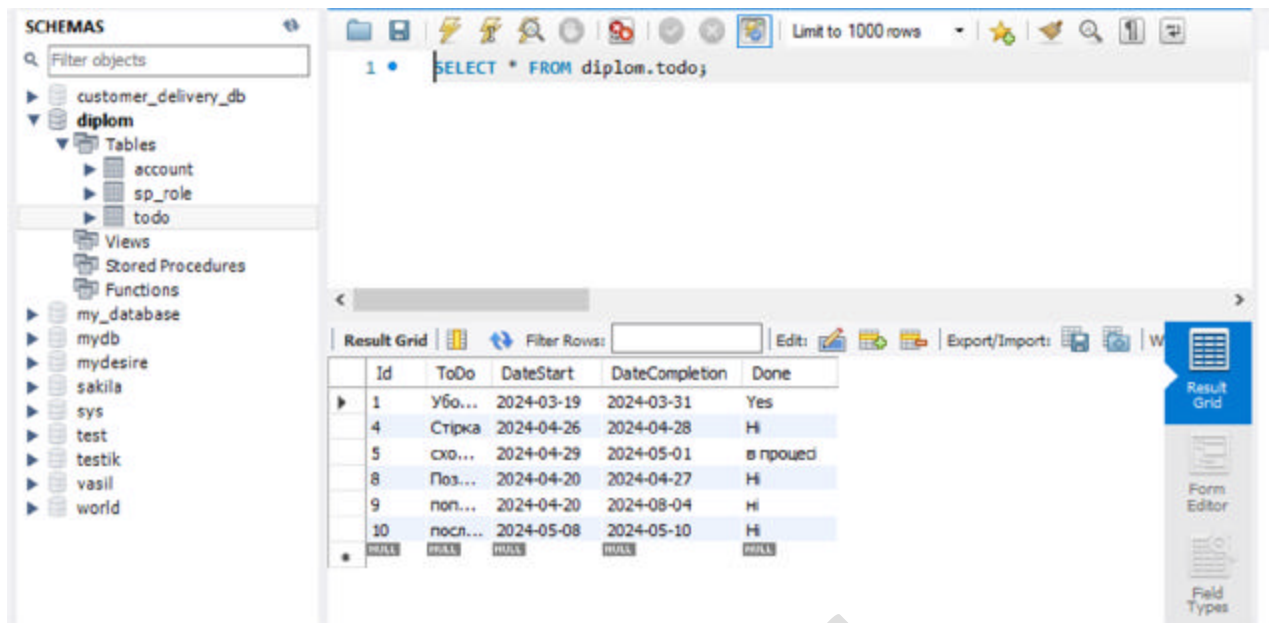


Рисунок 3.2 – Таблиця todo

Завдяки використанню бази даних і цих двох таблиць забезпечується надійне зберігання та обробка даних, що є критично важливим для ефективного планування та виконання проєктів.

### 3.2 Розробка структурної схеми

Зважаючи на важливість ретельного аналізу та опису структури системи Task Manager, в цьому розділі надається повна структурна схема, що відображає всі ключові компоненти та їх взаємозв'язки. Викладена нижче інформація розкриває внутрішню архітектуру системи, що є важливою основою для подальшого розвитку, тестування та підтримки програмного забезпечення.

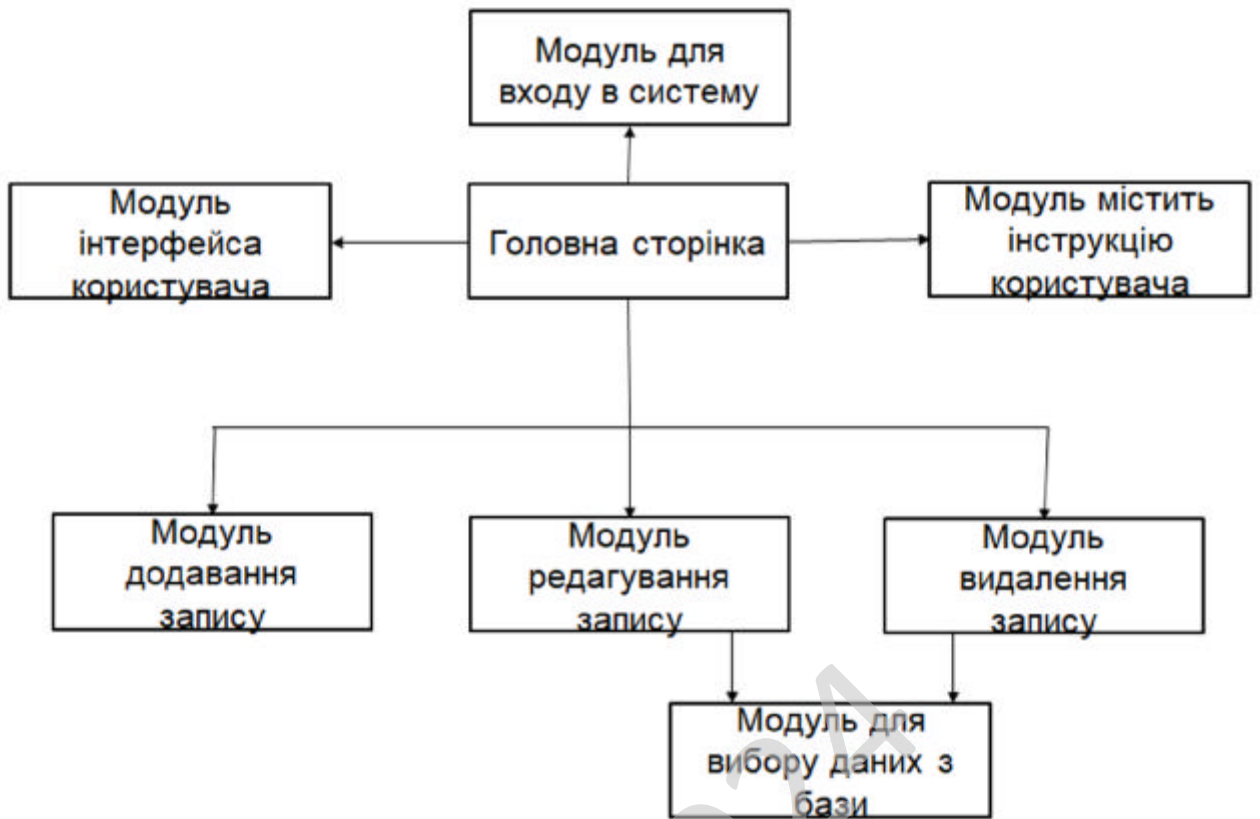


Рисунок 3.3 – Структурна схема пристрою

Структурна схема системи Task Manager була ретельно розроблена з метою забезпечення зручного та ефективного управління завданнями та проєктами. Схема складається з ряду ключових компонентів, що взаємодіють між собою для забезпечення функціонування системи.

Один з головних компонентів системи - це Клієнтський інтерфейс, який включає в себе всі необхідні елементи для взаємодії з користувачем. Він дозволяє користувачам переглядати, додавати, редагувати та видаляти завдання та проєкти.

Бізнес-логіка системи відповідає за обробку всіх операцій, пов'язаних з управлінням завданнями. Цей компонент виконує функції додавання, редагування та видалення завдань, а також взаємодію з базою даних для зберігання та оновлення інформації.

База даних є центральним елементом системи, де зберігається вся інформація про завдання, проєкти та користувачів. Вона забезпечує надійне зберігання даних та забезпечує доступ до них для різних компонентів системи.

Крім того, компоненти Авторизації та Аутентифікації та Логування забезпечують безпеку та контроль над діями користувачів у системі, записуючи всі вхідні та вихідні події для подальшого аналізу.

Загалом, структурна схема системи Task Manager дозволяє ефективно організувати та контролювати всі аспекти управління завданнями та проєктами, забезпечуючи зручний та надійний інструмент для користувачів.

Дана структурна схема показує всі взаємозв'язки між компонентами системи. В основі системи лежить база даних, яка обробляє та передає в саму програму.

### 3.3 Розробка функціональної схеми

В даному розділі надається опис функціональних можливостей системи Task Manager та їхній взаємозв'язок. Функціональна схема відображає, як кожен компонент системи взаємодіє для виконання різних операцій та забезпечення коректної роботи програмного забезпечення.



Рисунок 3.4 – Функціональна схема пристрою

Однією з ключових функціональних можливостей системи є можливість додавання нових завдань та проєктів. Користувач може ввести необхідну інформацію про завдання через клієнтський інтерфейс, після чого ці дані передаються до бізнес-логіки для обробки та збереження у базі даних.

Іншою важливою функціональною можливістю є можливість редагування та оновлення інформації про завдання. Користувач може вибрати конкретне завдання для редагування через клієнтський інтерфейс, після чого зміни передаються до бізнес-логіки, яка виконує оновлення даних у базі даних.

Також, система забезпечує можливість відображення списку завдань у вигляді табличної структури за допомогою компонента DataGridView. Це дозволяє користувачам зручно переглядати та керувати своїми завданнями безпосередньо з головного вікна програми.

Загалом, функціональна схема системи Task Manager відображає важливі аспекти її роботи та забезпечує зручний та ефективний інструмент для управління завданнями та проєктами.

Додатково, варто звернути увагу на особливості функціонування системи у різних умовах експлуатації, таких як велика кількість одночасних користувачів, обмежені ресурси апаратного забезпечення або нестабільність мережі зв'язку.

Крім того, можна також описати процеси тестування, які використовуються для перевірки функціональності системи та виявлення можливих проблем або помилок. Такий аналіз може бути корисним для забезпечення якості та надійності програмного забезпечення.

Крім того, важливо відзначити, що розроблена програма володіє унікальним функціоналом, який відокремлює її від інших подібних рішень. Однією з таких унікальних особливостей є обмеження на додавання одного й того ж завдання у систему. Це виключає можливість створення дублікатів завдань, що дозволяє підтримувати порядок та структурованість у списку завдань користувача. Такий підхід сприяє ефективному управлінню часом та ресурсами, попереджаючи

виникнення непотрібних дублікатів та конфліктів у роботі з завданнями.

Це унікальне обмеження дозволяє підвищити продуктивність користувачів, зменшити ризик втрати даних та сприяти більш ефективному управлінню часом та ресурсами. Такий підхід допомагає створити програму, яка відповідає найвищим стандартам якості та забезпечує задоволення потреб користувачів у керуванні їхніми завданнями та проєктами.

Дана функціональна схема показує роль кожного компоненту впродовж всьєї роботи програми. Розглядаються принципи взаємодії між компонентами системи, які спрямовані на забезпечення їхньої ефективної роботи та надійності, підкреслюється важливість їхньої внутрішньої взаємодії та спільної дії. Також у цьому розділі висвітлені аспекти, які стосуються обміну даними між компонентами системи, обробки та аналізу.

### 3.4 Розробка діаграми процесів

Схема містить кілька ключових компонентів, які забезпечують функціональність системи Task Manager. Кожен з цих компонентів відіграє важливу роль у забезпеченні ефективного планування та управління проєктами.

Ключовими процесами в роботі програми є додавання, редагування та видалення записів. Під час додавання нової задачі користувач вводить необхідні дані, такі як опис задачі, дата початку і завершення, а також статус виконання. Ця інформація зберігається в таблиці todo бази даних. Процес редагування дозволяє користувачу змінювати існуючі записи, оновлюючи будь-яку інформацію щодо задачі. Видалення запису забезпечує можливість видаляти завершені або непотрібні задачі.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

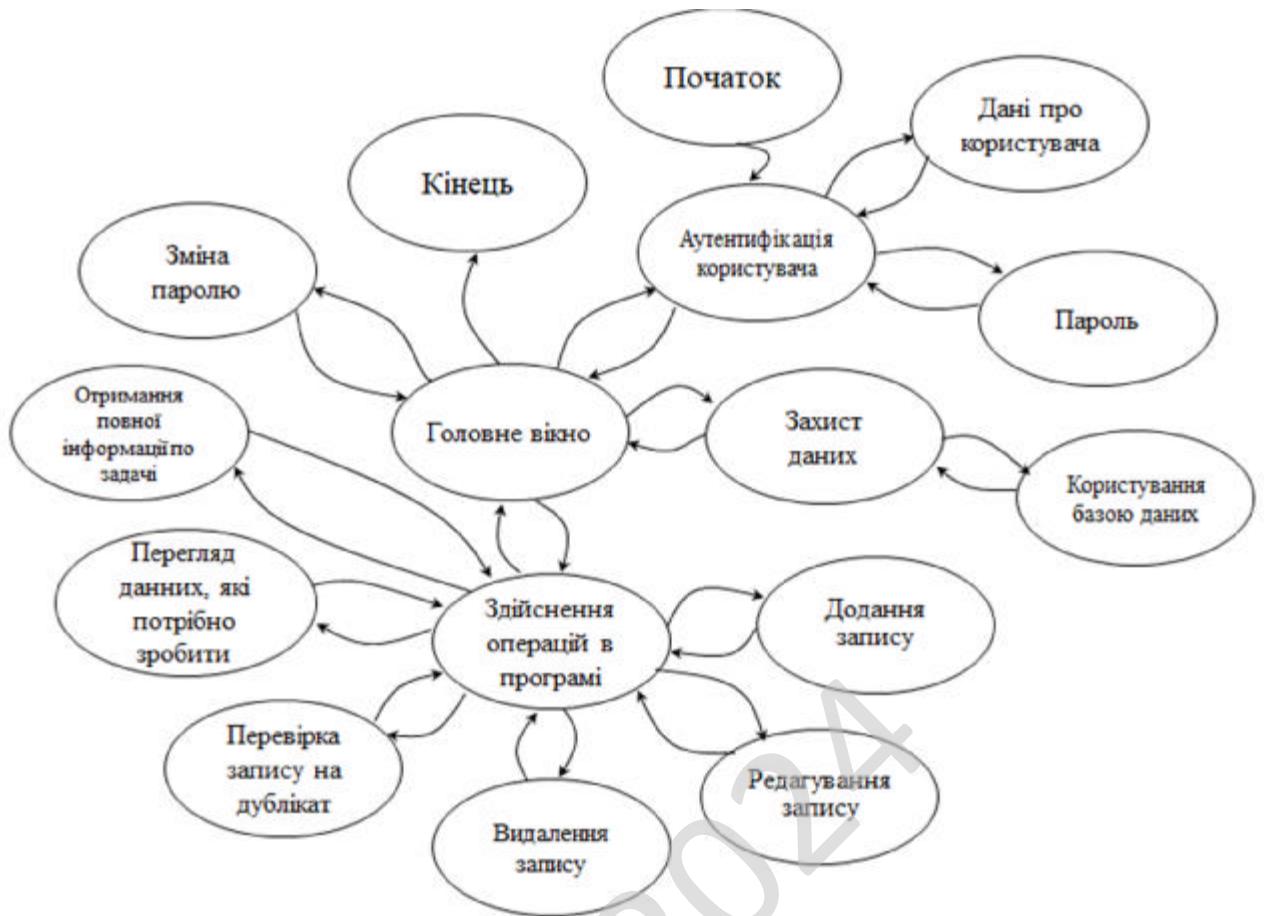


Рисунок 3.5 – Діаграма взаємодії процесів

Інтерфейс користувача розроблений таким чином, щоб забезпечити зручний доступ до всіх функцій програми. Після успішної авторизації користувач може додавати нові задачі, редагувати існуючі та видаляти непотрібні записи. Всі ці дії виконуються через інтуїтивно зрозумілий інтерфейс з використанням відповідних форм та елементів управління.

На кожному етапі роботи програми відбувається взаємодія з базою даних, що забезпечує надійність зберігання та обробки даних. Наприклад, під час авторизації відбувається запит до таблиці account для перевірки введених даних, а під час додавання чи редагування задачі – до таблиці todo.

Описані процеси роботи програми демонструють її ефективність та зручність використання для планування і управління проектами. Програма дозволяє оптимізувати роботу з задачами, забезпечуючи прозорість та контроль за виконанням проектів. Це робить її корисним інструментом як для приватного

бізнесу, так і для державних організацій.

Всі ці компоненти взаємодіють між собою, забезпечуючи цілісність і надійність роботи системи Task Manager. Використання сучасних технологій і підходів до розробки програмного забезпечення гарантує, що система буде ефективно функціонувати на персональних комп'ютерах архітектури IBM PC з ОС Windows 10/11.

Принцип підключення компонентів забезпечує надійність і простоту у використанні системи, що дозволяє користувачам легко планувати та управляти своїми проектами.

КБПЗ\_2024

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

## 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

В цьому розділі наведено алгоритм роботи розробленої програми та блок-схеми її роботи. Розписані всі необхідні кроки для розуміння того, як працює програма та які її функції.

На рис. 4.1 наведено блок-схему роботи класу `Authorization.cs`.

Поетапний алгоритм роботи **Authorization.cs** представлений нижче:

Крок 1. Початок програми.

Крок 2. Починається виклик методу `Authorization1` з передачею логіну та паролю.

Крок 3. Авторизація користувача.

Крок 4. Спроба виконання SQL-запиту для перевірки логіну та паролю в базі даних.

Крок 5. Перевірка результату запиту.

Крок 6. Обробка винятку у випадку помилки під час авторизації.

Крок 7. Спроба виконання SQL-запиту для отримання прізвища користувача за логіном.

Крок 8. Повернення прізвища користувача у випадку успішного виконання запиту.

На рис. 4.2 наведено блок-схему роботи класу `DBConnection.cs`.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32



Рисунок 4.1 – Блок-схема роботи класу Authorization.cs

Поетапний алгоритм роботи **DBConnection.cs** представлений нижче:

Крок 1. Підключення до бази даних.

Крок 2. Перевірка успішності підключення.

Крок 3. Якщо підключення вдале, створюється об'єкт MySqlCommand для виконання SQL-запитів та MySqlDataAdapter для роботи з даними.

Крок 4. Обробка винятку у випадку невдалого підключення до бази даних.

Крок 5. Закриття з'єднання.

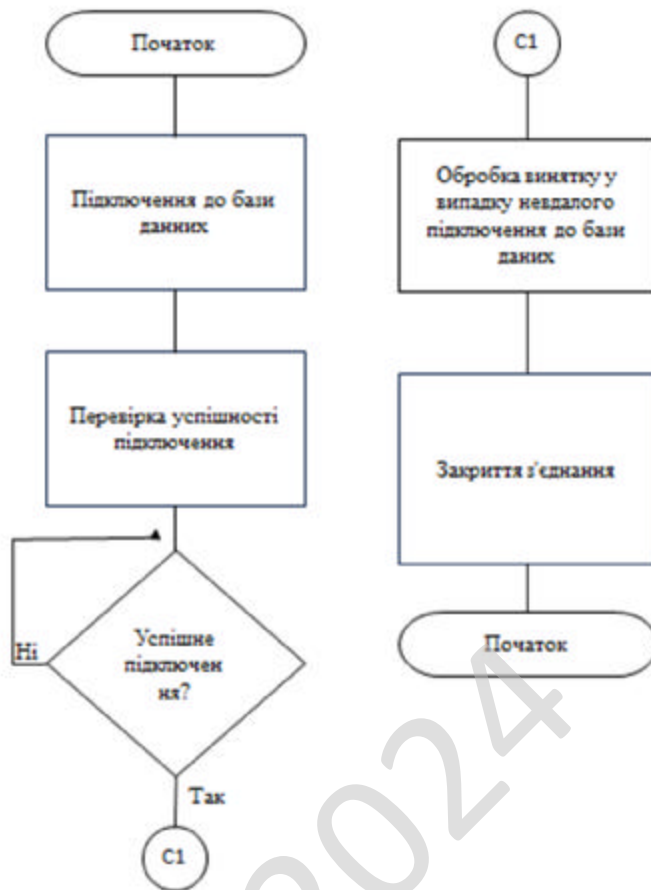


Рисунок 4.2 – Блок-схема роботи класу DBConnection.cs

Поетапний алгоритм роботи **ManagerClass.cs** представлений нижче:

Крок 1. Отримання даних.

Крок 2. Виклик методу GetData(), який виконує SQL-запит для отримання даних з таблиці todo.

Крок 3. Виконання SQL-запиту та заповнення DataTable з отриманими даними.

Крок 4. Додавання нового запису.

Крок 5. Виклик методу AddData(), який додає новий запис до таблиці todo.

Крок 6. Редагування запису.

Крок 7. Виклик методу EditData(), який редагує існуючий запис у таблиці todo.

Крок 8. Видалення запису.

Крок 9. Виклик методу DeleteData(), який видаляє вказаний запис з таблиці todo.

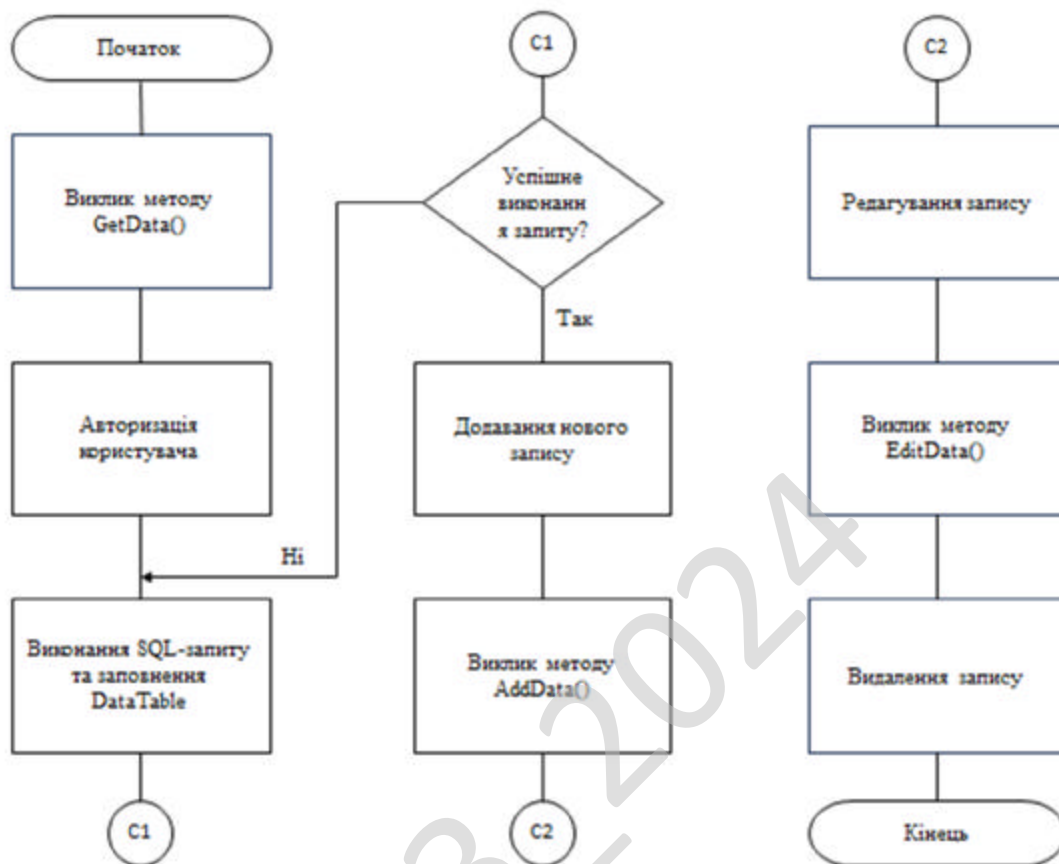


Рисунок 4.3 – Блок-схема роботи класу ManagerClass.cs

Поетапний алгоритм роботи **TaskManager.cs** представлений нижче:

Крок 1. Ініціалізація форми.

Крок 2. Виклик методу GetData() класу ManagerClass, для отримання даних з бази та їх відображення на формі.

Крок 3. Клік на кнопку "Додати задачу".

Крок 4. Перевірка заповненості обов'язкових полів.

Крок 5. Додавання нової задачі до бази даних за допомогою методу AddData() класу ManagerClass.

Крок 6. Клік на кнопку "Редагувати".

Крок 7. Отримання даних про обрану задачу з таблиці.

Крок 8. Перевірка та збереження змінених даних в базі за допомогою методу EditData() класу ManagerClass.

Крок 9. Клік на кнопку "Видалити".

Крок 10. Видалення обраної задачі з бази за допомогою методу DeleteData() класу ManagerClass.

#### 4.2 Захист розробленого програмного забезпечення

Відповідно до Закону України № 3792-ХІІ від 23.12.93 "Про авторське право і суміжні права" (зі змінами від 16.07.01), визначено термін "комп'ютерна програма". Відповідно до цього закону, комп'ютерні програми є об'єктом авторського права, тобто інтелектуальної власності. Основними елементами захисту програмного забезпечення є гарантування конфіденційності, цілісності та доступності даних і функціонування пристрою.

Однак варто зазначити, що в процесі розробки системи для надання посилань на компоненти та модулі використовувалися загальнодоступні бібліотеки.

Основні бібліотеки:

System.Windows.Forms. Ця бібліотека входить до складу .NET Framework і містить класи для створення та управління вікнами, елементами керування, подіями та іншими складовими графічного інтерфейсу користувача (GUI) у програмах Windows Forms.

System.Data. Ця бібліотека надає доступ до функцій для роботи з базами даних у .NET Framework. Використовується для підключення до бази даних, виконання запитів та отримання результатів.

MySql.Data.MySqlClient. Ця бібліотека дозволяє взаємодіяти з базою даних MySQL з допомогою .NET Framework. Вона надає класи та методи для підключення до MySQL-сервера, виконання запитів SQL та отримання результатів.

System.Security.Cryptography. Ця бібліотека містить класи для реалізації

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

різних алгоритмів шифрування, гешування та інших криптографічних операцій. Використовується для зберігання паролів у безпечному вигляді та забезпечення конфіденційності даних.

System.Windows.Forms.DataVisualization.Charting. Ця бібліотека містить класи для створення різноманітних графіків та діаграм у Windows Forms додатках. Використовується для візуалізації даних та аналізу результатів.

System.Drawing. Ця бібліотека містить класи та методи для маніпулювання зображеннями, кольорами та графічними об'єктами. Використовується для створення та редагування графічного інтерфейсу користувача, а також для роботи з малюнками та зображеннями.

КБПЗ\_2024

					ВКРБ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

У цьому розділі розглядається методика впровадження розробленої системи у промислову експлуатацію. Детально описується процес інтеграції компонентів програмного забезпечення у вже існуючу апаратну систему з метою забезпечення ефективності та надійності роботи всієї системи.

– **Аналіз існуючої інфраструктури.** Першим кроком впровадження є докладний аналіз існуючої апаратної та програмної інфраструктури. Визначаються характеристики обладнання, операційних систем, баз даних та інших компонентів, які можуть вплинути на процес інтеграції.

– **Підготовка середовища для впровадження.** Після аналізу створюється план підготовки середовища для встановлення та запуску програмного забезпечення. Це може включати в себе оновлення апаратної складової, налаштування операційних систем, підготовку баз даних та інше.

– **Інтеграція компонентів програмного забезпечення.** На цьому етапі встановлюються та налаштовуються всі компоненти розробленої системи. Здійснюється підключення до баз даних, налаштування мережних з'єднань, інтеграція з іншими системами та інше.

– **Тестування та налагодження.** Після інтеграції проводиться ретельне тестування всіх компонентів системи з метою виявлення та виправлення можливих помилок та несправностей. Здійснюється налагодження та оптимізація роботи системи для забезпечення її ефективності та стабільності.

– **Впровадження в експлуатацію.** Після успішного завершення тестування та налагодження система готова до впровадження в промислову експлуатацію. Здійснюється підготовка користувачів, проводяться навчання та тренінги з роботи з новою системою, а також забезпечується підтримка та обслуговування на етапі впровадження.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

– **Моніторинг та підтримка.** Після впровадження системи проводиться постійний моніторинг та підтримка її роботи. Забезпечується надійність та безперебійність функціонування системи, а також вчасне виявлення та усунення можливих проблем та несправностей.

Цей розділ визначає стратегію та методи впровадження розробленої системи в промислову експлуатацію з метою забезпечення її успішної та ефективної роботи.

Кожний з цих пунктів складається із своїх підпунктів. Кожний пункт детально розписаний і показує, як саме потрібно ефективно впровадити систему в промислову експлуатацію. Дані підпункти наведені нижче:

### **1. Аналіз існуючої інфраструктури:**

- Аналіз апаратної інфраструктури: Оцінка поточного стану обладнання, його технічних характеристик та сумісності з програмним забезпеченням. Визначення потреб у модернізації або заміні обладнання для забезпечення оптимальної роботи системи.

- Аналіз програмної інфраструктури: Огляд операційних систем, баз даних, веб-серверів та іншого програмного забезпечення, яке використовується в організації. Визначення можливих конфліктів або несумісностей з новою системою.

- Оцінка існуючих процесів та робочих потоків: Вивчення існуючих процесів та робочих потоків в організації для визначення можливостей для оптимізації та автоматизації за допомогою нової системи.

### **2. Підготовка середовища для впровадження:**

- Оновлення апаратної складової: Планування та виконання оновлення або заміни обладнання для забезпечення сумісності з програмним забезпеченням та оптимальної продуктивності.

- Налаштування операційних систем: Підготовка та налаштування операційних систем на серверах та робочих станціях для оптимальної роботи з програмним забезпеченням.

○ Підготовка баз даних: Створення та налаштування баз даних для зберігання та обробки інформації, необхідної для роботи системи.

### **3. Інтеграція компонентів програмного забезпечення:**

○ Встановлення програмного забезпечення: Встановлення та конфігурування всіх компонентів розробленої системи на апаратному обладнанні.

○ Підключення до баз даних: Налаштування з'єднання з базами даних та імпорт існуючих даних у нову систему.

○ Інтеграція з іншими системами: Підключення та налаштування інтеграції з іншими програмними засобами, які використовуються в організації.

### **4. Тестування та налагодження:**

○ Тестування функціональності: Виконання ряду тестів для перевірки правильності та стабільності роботи всіх компонентів системи.

○ Налagodження та оптимізація: виправлення помилок, виявлених під час тестування, а також оптимізація роботи системи для забезпечення її ефективності та продуктивності.

### **5. Впровадження в експлуатацію:**

○ Підготовка користувачів: Проведення навчання та тренінгів з роботи з новою системою для користувачів організації.

○ Підтримка та обслуговування: Забезпечення технічної підтримки та обслуговування системи на етапі впровадження та після нього.

### **6. Моніторинг та підтримка:**

○ Встановлення системи моніторингу: Розгортання системи моніторингу для постійного контролю за роботою системи, виявлення можливих проблем та аналізу продуктивності.

○ Автоматизована збір і аналіз даних: Налаштування автоматичного збору та аналізу даних про роботу системи з метою вчасного виявлення та вирішення потенційних проблем.

○ Надання технічної підтримки: Організація сервісу технічної підтримки для оперативного реагування на запити користувачів та вирішення їх проблем.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

○ Оновлення та підтримка: Забезпечення регулярних оновлень та підтримки програмного забезпечення для забезпечення його актуальності та безпеки.

○ Інцидентний менеджмент: Розробка та впровадження процедур управління інцидентами для виявлення, документування та вирішення проблем з експлуатацією системи.

### **Коротка інструкція користувача:**

Програма "Task Manager" призначена для ефективного управління задачами та планування проєктів. Основні можливості програми включають:

- **Авторизація.** Перед використанням програми користувач повинен авторизуватися, введучи свій логін та пароль.

- **Додавання задач.** Користувач може додавати нові задачі, вказуючи їх назву, дату початку та завершення, а також стан виконання. Це дозволяє зручно організувати робочий процес та планувати свої завдання.

- **Редагування задач.** У разі потреби користувач може змінювати дані задачі, включаючи назву, дату початку та завершення, а також стан виконання. Це дозволяє адаптувати плани до змін у робочих процесах.

- **Видалення задач.** Користувач може видаляти задачі, які більше не потрібні. Це дозволяє підтримувати порядок у списку завдань та зосередитися на найважливіших справах.

- **Перегляд даних.** Програма надає можливість переглядати всі створені задачі з відображенням їх назви, дат початку та завершення, а також стану виконання. Це дозволяє користувачу швидко оцінити свій прогрес та зміни у планах.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

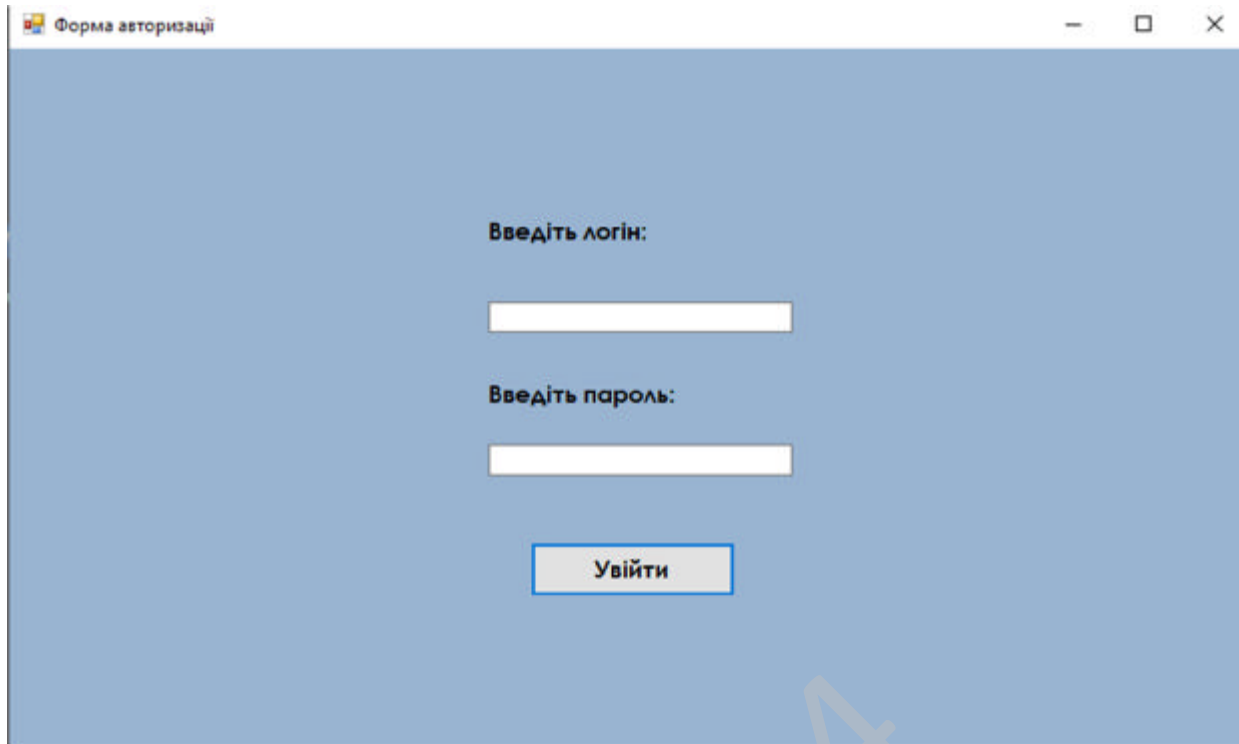


Рисунок 5.1 – Форма авторизації

Після авторизації користувач бачить всі активні задачі. Він може їх продивитись, додати, редагувати та видалити.

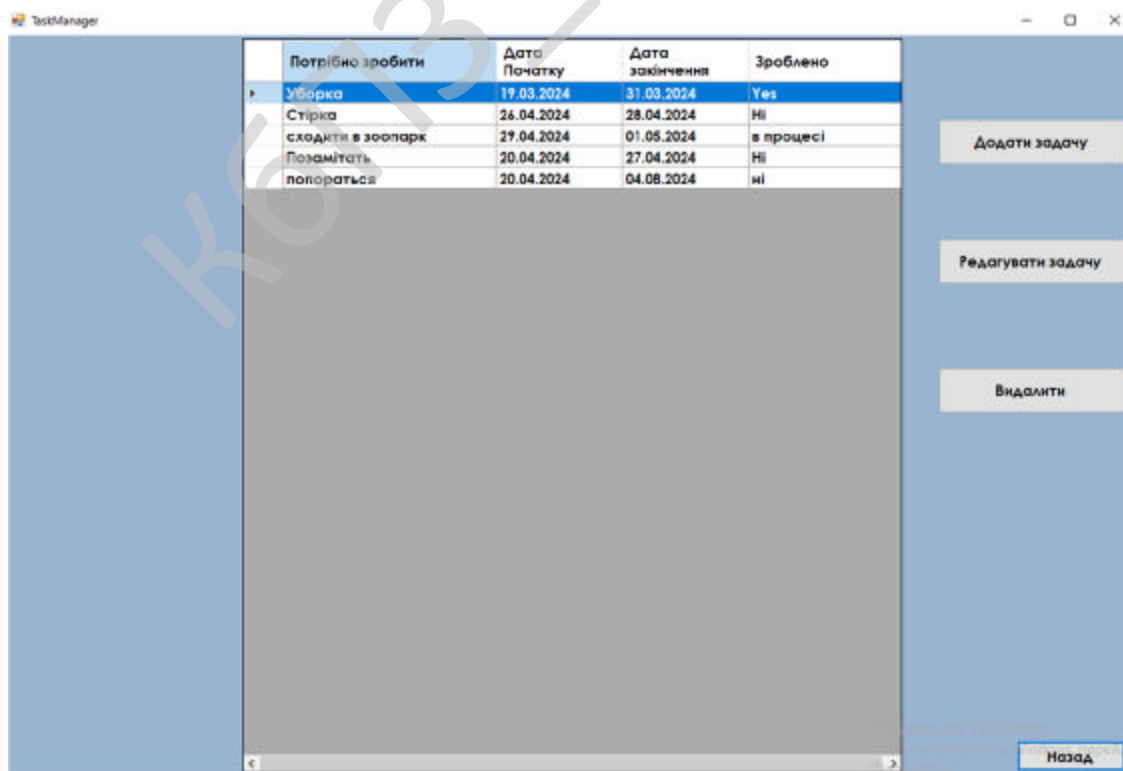


Рисунок 5.2 – Головна сторінка застосунку

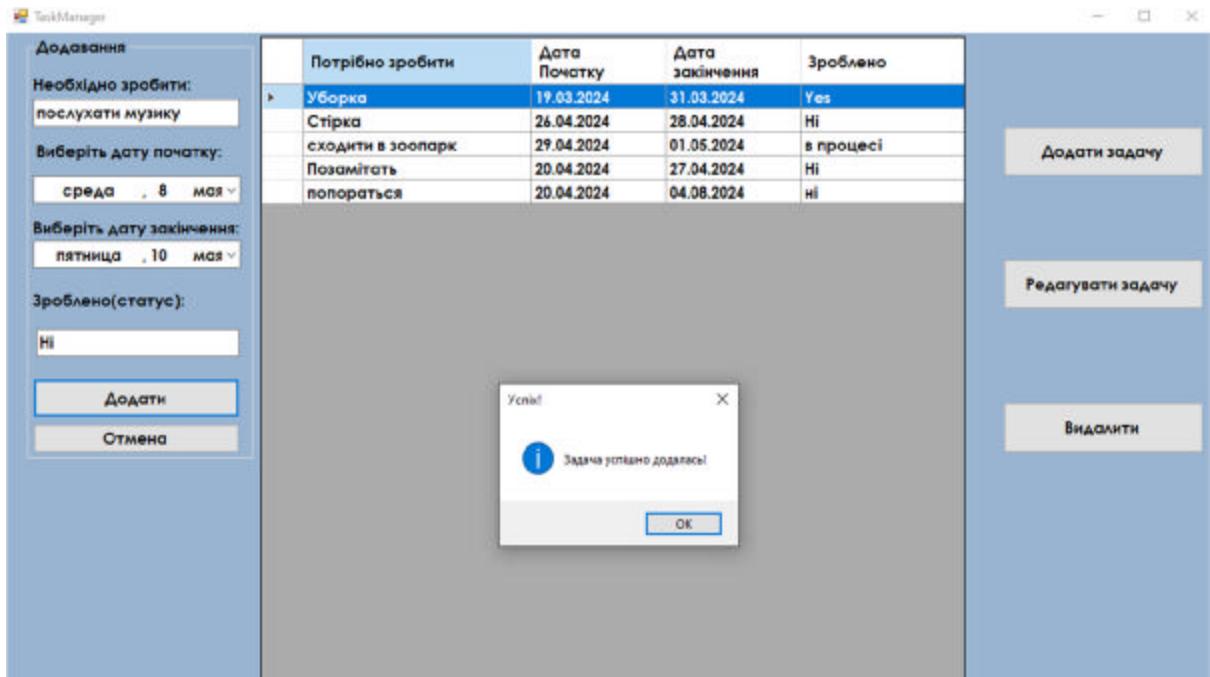


Рисунок 5.3 – Додавання задачі

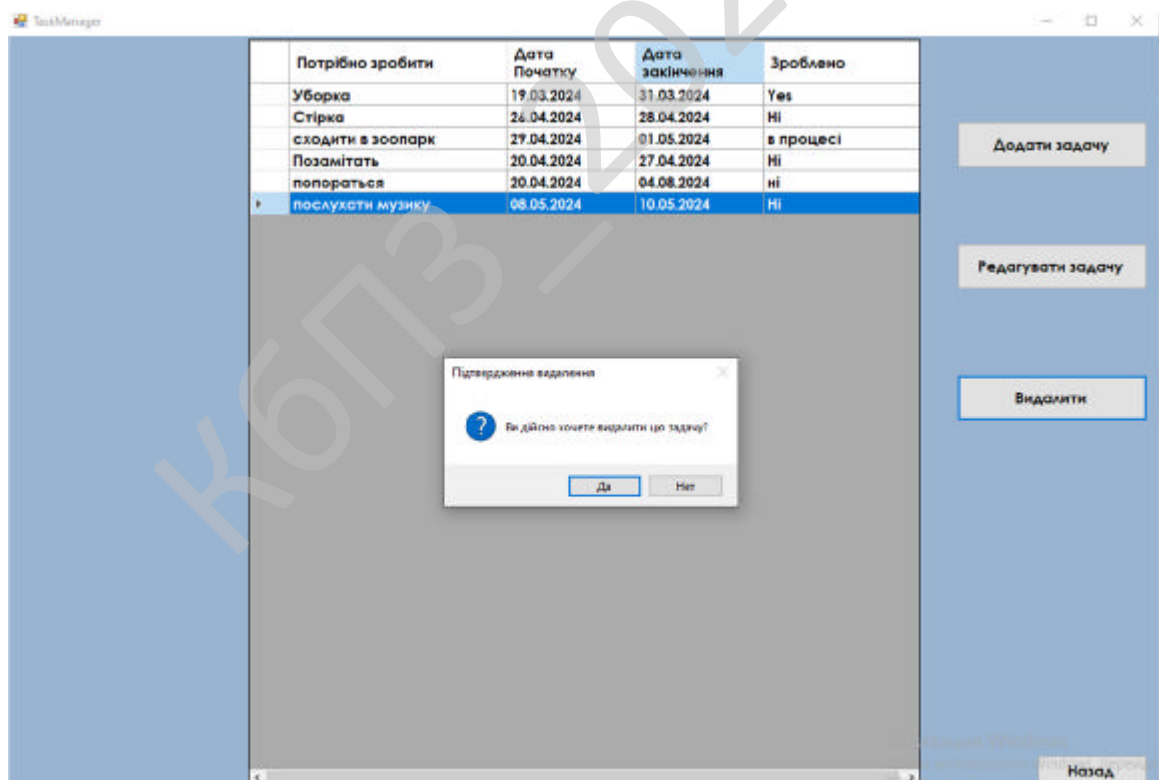


Рисунок 5.4 – Видалення задачі

## 6 ОСНОВНІ ВИСНОВКИ

У даній роботі було проведено ретельний аналіз та розробку системи управління завданнями (Task Manager), спрямованої на ефективне управління робочими процесами та контроль виконання завдань.

Для досягнення цієї мети було вирішено наступні завдання:

- Аналіз існуючих рішень у галузі управління задачами та планування проєктів.
- Визначення вимог до програмного забезпечення з урахуванням потреб користувачів.
- Проектування архітектури та інтерфейсу користувача програмного забезпечення.
- Реалізація програмного забезпечення з урахуванням сучасних технологій та практик програмування.
- Тестування та валідація розробленого програмного забезпечення.

Проведений аналіз сучасних систем управління завданнями дозволив виявити недоліки та переваги існуючих рішень. Це сприяло обґрунтуванню вибору оптимальних засобів для подальшої розробки нової системи.

На основі аналізу було обрано найбільш підходящі засоби та технології для створення системи управління завданнями. Використання мови програмування C# та технології Windows Forms дозволило створити ефективне та зручне програмне забезпечення.

У процесі роботи було розроблено та реалізовано систему Task Manager, яка включає в себе функції створення, редагування та видалення завдань. Проведено розробку інтерфейсу користувача для зручного та інтуїтивного використання системи.

Проведена реалізація системи та експериментальні дані підтверджують відповідність функціональності системи вимогам та очікуванням користувачів.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Task Manager дозволяє ефективно організувати робочі завдання та контролювати їх виконання.

Розроблена система готова до впровадження в реальні умови роботи. Особлива увага приділялася підготовці персоналу та організації процесу впровадження. Task Manager може бути успішно використаний в різних сферах діяльності для оптимізації робочих процесів та підвищення продуктивності.

У цілому, дана робота становить значний крок у напрямку розробки та впровадження системи управління завданнями, яка може бути використана в різних галузях діяльності для підвищення ефективності та продуктивності робочих процесів.

КБПЗ\_2024

					ВКРБ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ChatGPT [Електронний ресурс]. – Режим доступу: <https://chat.openai.com>
2. Ubidots [Електронний ресурс]. – Режим доступу: <https://stem.ubidots.com>
3. Матеріал Вікіпедії [Електронний ресурс] - Режим доступу: [https://uk.wikipedia.org/wiki/Електронний\\_каталог](https://uk.wikipedia.org/wiki/Електронний_каталог)
4. Петрова Г. С. Програмне забезпечення для планування та управління проектами. — Москва: ООО "И. Д. Вильямс", 2018. — 400 с.
5. Матвієнко І. Сучасні методи управління проектами. — Київ: Наукова думка, 2018. — 400 с.
6. Клименко О. В. Програмне забезпечення для менеджменту та планування. — Харків: Видавництво "Інженер", 2016. — 400 с.
7. Куценко Д. А. Технології планування та управління проектами. — Київ: Інтерактив плюс, 2018. — 400 с.
8. Ширяєв О. Методи та інструменти планування та управління проектами. — Харків: ХНУРЭ, 2019. — 400 с.
9. Джонсон Д., Джонсон М. "Effective C#: 50 Specific Ways to Improve Your C#" // Addison-Wesley Professional, 2020. — 528 с.
10. Смірнов С.А./ Проектування комп'ютерних систем та мереж./ Смірнов С.А., Смірнов О.А., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. / Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с
11. Шестаков В. А. Основи програмного забезпечення для менеджерів. — Москва: Бином, 2018. — 400 с.
12. Куценко А. А. Програмне забезпечення та його вплив на менеджмент. — Київ: Інтерактив плюс, 2019. — 400 с.
13. Смірнов С.А./ Розробка методу передтестової компіляції й розподілу доступу./ Смірнов С.А., Смірнов О.А., Коваленко О.В., Коваленко А.С. / Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

безпека та комп'ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

14. Смірнов О.А., Кузнецов О.О., Євсєєв С.П., Мелешко Є.В., Король О.Г. Методи та алгоритми симетричної криптографії. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.О. Кузнецова. Гриф “Навчальний посібник” надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 26.04.2012 року № 1/11-5762. – Кіровоград: КНТУ 2012. – 315 с.

15. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

16. Демчук В. Методи та інструменти управління проектами в програмному забезпеченні. — Київ: Видавничий дім "Видавництво НПУ імені М. П. Драгоманова", 2018. — 400 с.

17. Кравцов В. М. Програмування на мові С# в середовищі Visual Studio. — Київ: КВІНТА-ЦЕНТР, 2019. — 400 с.

18. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

19. Петрова Г. С. Програмне забезпечення для планування та управління проектами. — Москва: ООО "И. Д. Вильямс", 2018. — 400 с.

20. Про авторське право і суміжні права [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2811-20#Text>

21. Смірнов С.А./ Розробка методу передтестової компіляції й розподілу доступу./ Смірнов С.А., Смірнов О.А., Коваленко О.В., Коваленко А.С. / Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп'ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

22. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 17.04.2012 року № 1/11-5249. – Кіровоград: КНТУ 2012. – 250 с.

23. Дудар В. С. Програмування на С# для початківців. — Київ: Видавничий дім "Сам", 2018. — 300 с.

24. Кучеренко І. В. Основи програмування на мові С# в середовищі Visual Studio. — Київ: Видавничий центр "Академія", 2017. — 350 с.

25. Ткаченко О. С. Програмування на С# в середовищі Visual Studio: навчальний посібник. — Київ: Видавничий центр "Шкільний світ", 2018. — 320 с.

26. Сіренко М. Мова програмування С# та технологія .NET: навчальний посібник. — Київ: НВП "ВПЦ «Київський університет», 2018. — 300 с.

27. Карнацький В. Мова програмування С# в середовищі Visual Studio 2019: навчальний посібник. — Київ: Видавництво "Інтерактив плюс", 2020. — 320 с.

28. Яцина О. Використання Windows Forms у мові програмування С#. — Київ: Видавництво "Інженер", 2017. — 250 с.

29. Карасьов М. Програмування Windows Forms на мові С#. — Київ: Видавництво "Світ", 2018. — 280 с.

30. Кузьмін В. Розробка додатків на Windows Forms з використанням мови С#. — Київ: Видавничий дім "Шкільний світ", 2019. — 300 с.

31. Барсуков Є.Г. Програмування в середовищі Visual Studio на мові С#. – [Електронний ресурс]. – Режим доступу: <http://progland.com.ua/books-csharp/4-programming-visual-studio-csharp.html>.

32. Програмування на мові С# в середовищі Visual Studio: навчальний посібник. – [Електронний ресурс]. – Режим доступу: <https://github.com/Soft-Group-Academy/CSharpBasicBook>.

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

33. Платформа .NET. Курс С# [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/tutorial/>
34. Подкаст ".NET за чашкою кави" [Електронний ресурс] – Режим доступу: <https://anchor.fm/netcoffee>
35. Курило В.Я. Основи програмування на мові С# [Електронний ресурс] – Режим доступу: <http://proscsharp.com.ua/>.
36. Лабораторні роботи з предмету "Основи програмування на мові С#" [Електронний ресурс] – Режим доступу: [https://www.youtube.com/playlist?list=PL7WUXXpg-ryAjASfvHjH\\_ezMkgAHUySPx](https://www.youtube.com/playlist?list=PL7WUXXpg-ryAjASfvHjH_ezMkgAHUySPx)
37. Данилишин В.В. Мова програмування С# та платформа .NET [Електронний ресурс] – Режим доступу: <http://csharp.net.ua/faq/>
38. Мельник С.А. Методи управління проектами програмного забезпечення. – [Електронний ресурс]. – Режим доступу: <https://www.ranok.com.ua/catalog/educational-books/higher-education/informatsiyni-tekhnohiiyi-telekomunikatsiyyi/melnyk-s-a---metody-upravlinnya-proektamy-programnoho-zabezpechennya.html>
39. Закорун С.П. Технології розробки програмного забезпечення: підручник. – Київ: НТУУ «КПІ», 2018. – 408 с. ISBN 978-966-7548-46-9.
40. Ковальчук О. Сучасні технології розробки програмного забезпечення. - [Електронний ресурс]. - Режим доступу: <https://textbooks.ydss.net/index.php/progsoftdev>
41. Карташов А. С., Корневич Л. І. Основи розробки програмного забезпечення. - [Електронний ресурс]. - Режим доступу: <http://cv.ukrstat.org/curriculums/it/osnovyrozrobki.pdf>
42. Документація з офіційного сайту Microsoft для Windows Task Manager. - [Електронний ресурс]. - Режим доступу: <https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer>

					<b>ВКРБ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

43. Черкашина І.Г., Шарапов О.М., Мироненко В.В. Процеси, потоки та потокові діаграми в управлінні проєктами. // Міжнародний науково-технічний журнал «Системні дослідження та інформаційні технології». – 2018. – №3. – С. 41-46.

44. Синявський В.В. Управління процесами в програмному забезпеченні за допомогою Task Manager. // Науковий вісник Полісся. – 2019. – №2 (14). – С. 115-121.

45. Біленко О.В., Соломін В.М. Методи підвищення ефективності управління процесами в програмному забезпеченні. // Інформаційні технології та комп'ютерна інженерія. – 2020. – Т.54, №3. – С. 278-282.

46. Гаврилов О.В., Кривенко І.М., Підгорний О.М. Розробка програмного забезпечення в середовищі Visual Studio. // Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми. – 2021. – №1. – С. 45-50.

47. Васильєва О.М. Технології розробки програмного забезпечення в середовищі Visual Studio. // Наукові праці. – 2019. – Вип. 90. – С. 98-103.

48. Сидоренко М.М., Литвиненко В.В., Ковальов М.І. Методи та засоби планування та контролю виконання завдань у Task Manager. // Інформаційні технології в освіті, науці та промисловості: матеріали V міжнародної науково-практичної конференції. – 2020. – С. 136-140.

49. Петренко О.В. Аналіз методів планування та управління завданнями у програмному забезпеченні. // Вісник Київського національного університету імені Тараса Шевченка. Інформатика. – 2019. – Вип. 48. – С. 82-86.

50. Лазаренко І.С., Мірошник І.О., Романенко О.М. Практичні аспекти роботи з Task Manager у відділах ІТ-компаній. // Науковий вісник Херсонського державного університету. Серія: Економічні науки. – 2021. – Вип. 44. – С. 148-152.

51. Грищенко О.В., Шевченко В.М., Любарська О.М. Оптимізація роботи з Task Manager для підвищення продуктивності роботи комп'ютера. // Інформаційні технології: наука, техніка, технологія, освіта: матеріали V

міжнародної науково-практичної конференції. – 2019. – С. 257-261.

52. Ковальчук С.І., Чернова О.В., Мельников С.П. Розробка програмного забезпечення для управління завданнями у Task Manager. // Науковий вісник Ужгородського національного університету. Серія: Інформатика. – 2020. – Вип. 2. – С. 45-50.

КБПЗ\_2024

					ВКРБ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	5
9 Порядок контролю та приймання.....	6

					<b>ВКРБ-123.24.0008.00.00.ТЗ</b>		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Кривохижа В.Ю.				Літ.	Аркуш	Аркушів
Перевірів	Мелешко Є.В.				Б	1	6
Н. Контр.	Коваленко А.С				ЦНТУ КІ-20		
Затв.	Смірнов О.А.						

# 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку програмного забезпечення менеджера задач для планування проєктів.

## 2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу №\_\_\_\_ від \_\_.\_\_.2024 року, видане на кафедрі кібербезпеки та програмного забезпечення.

## 3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення менеджера задач для планування проєктів.

## 4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проєкту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРБ-123.24.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- додавання, редагування та видалення запланованих задач;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРБ-123.24.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- операційну систему Windows 10 або Windows 11, процесор з мінімальною тактовою частотою, обсяг оперативної пам'яті;
- Забезпечення комфортних умов праці для користувачів

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel Core i7/8 ГБ /1 Tb/ GeForce GT 1030 2GB або сумісні з ним.

### 5.8.2 Мова програмування

Програму розроблено на мові програмування C# (C Sharp) з використанням платформи .NET Framework.

					<b>ВКРБ-123.24.0008.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД.

## 7 Перелік документів, що розробляються

- Структурна схема системи. 1 аркуш
- Функціональна схема системи. 1 аркуш
- Діаграма процесів. 1 аркуш
- Блок-схема алгоритму роботи програми. 1 аркуш
- Пояснювальна записка. 51 аркуш

## 8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

					<b>ВКРБ-123.24.0008.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання кваліфікаційної бакалаврської роботи на попередній захист  
\_\_ . \_\_ . 2024 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист  
\_\_ . \_\_ . 2024 р.

КБПЗ\_2024

					ВКРБ-123.24.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи  
за першим (бакалаврським) рівнем вищої освіти

\_\_\_\_\_ Є.В. Мелешко

*Програмне забезпечення менеджера задач для планування проєктів*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 19

Літера: РП

Кропивницький – 2024 року

**// Authorization.cs:**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diplomniiproekt
{
    internal class Authorization
    {
        static public string role, surname, User;

        static public void Authorization1(string login, string password)
        {
            try
            {
                DBConnection.msCommand.CommandText = @"SELECT name_role from
sp_role, account WHERE Login = '" + login + "' and Password = '" + password + "'
and account.id_role=sp_role.id_role";
                Object result = DBConnection.msCommand.ExecuteScalar();
                if (result != null)
                {
                    role = result.ToString();
                    User = login;
                }
                else
                {
                    role = null;
                    surname = null;
                }
            }
            catch
            {
                role = User = null;
                MessageBox.Show("Помилка при авторизації!");
            }
        }

        static public string AuthorizationName(string login)
        {
            try
            {
                DBConnection.msCommand.CommandText = @"SELECT Surname FROM
account WHERE Login = '" + login + "'";
                Object result = DBConnection.msCommand.ExecuteScalar();
                surname = result.ToString();
                return surname;
            }
            catch
            {
                return null;
            }
        }
    }
}

```

**// DBConnection.cs:**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

```

```

using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace diplomniiproekt
{
    internal class DBConnection
    {
        static string DBConnect =
"server=localhost;user=root;database=diplom;password=2003;";
        static public MySqlDataAdapter msDataAdapter;
        static MySqlConnection myconnect;
        static public MySqlCommand msCommand;

        //метод підключення до бази даних:
        public static bool ConnectionDB()
        {
            try
            {
                myconnect = new MySqlConnection(DBConnect);
                myconnect.Open();
                msCommand = new MySqlCommand();
                msCommand.Connection = myconnect;
                msDataAdapter = new MySqlDataAdapter(msCommand);
                return true;
            }
            catch
            {
                MessageBox.Show("Помилка з'єднання з базою даних!", "Помилка!",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }

        public static void CloseDB()
        {
            myconnect.Close();
        }

        public MySqlConnection getConnection()
        {
            return myconnect;
        }
    }
}

```

**// Form1.cs (Форма Авторизації):**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Common;
using System.Data.Odbc;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diplomniiproekt
{
    public partial class Form1 : Form
    {
        static public string loginActive;
        static public string whois;
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

```

    }

private void Form1_Load(object sender, EventArgs e)
{
    this.MinimumSize = this.Size;
    this.MaximumSize = this.Size;
    DBConnection.ConnectionDB(); //підключаємся к базе даних через мой
клас
}

private void Authoriz_Click(object sender, EventArgs e)
{
    if (login.Text != "" && password.Text != "")
    {
        Authorization.Authorization1(login.Text, password.Text);
        switch (Authorization.role)
        {
            case null:
                {
                    MessageBox.Show("Такого аккаунту не існує",
"Перевірьте данні і спробуйте знову", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                    login.Text = "";
                    password.Text = "";
                    break;
                }
            case "Адмін":
                {
                    loginActive = login.Text;
                    whois = "Адмін";
                    Authorization.User = login.Text;

                    string surname =
Authorization.AuthorizationName(login.Text);
                    Authorization.surname = surname;
                    MessageBox.Show(surname + ", раді вас бачити в
нашому застосунку", "Успіх", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    this.Hide();

                    TaskManager tasks = new TaskManager();
                    tasks.Show();
                    break;
                }
            case "Користувач":
                {
                    loginActive = login.Text;
                    whois = "Користувач";
                    Authorization.User = login.Text;

                    string surname =
Authorization.AuthorizationName(login.Text);
                    Authorization.surname = surname;
                    MessageBox.Show(surname + ", раді вас бачити в
нашому застосунку", "Успіх", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    this.Hide();

                    TaskManager tasks = new TaskManager();
                    tasks.Show();
                    break;
                }
        }
    }
    else
    {
        MessageBox.Show("Заповніть усі обов'язкові поля!", "Заповнення
полей", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
}

```

```

}

// Form1.Designer.cs:
namespace diplomniiproekt
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть
        удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

        /// <summary>
        /// Требуемый метод для поддержки конструктора – не изменяйте
        /// содержимое этого метода с помощью редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            this.label2 = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.login = new System.Windows.Forms.TextBox();
            this.password = new System.Windows.Forms.TextBox();
            this.Authoriz = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Font = new System.Drawing.Font("Century Gothic", 11.25F,
System.Drawing.FontStyle.Bold);
            this.label2.Location = new System.Drawing.Point(306, 213);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(128, 18);
            this.label2.TabIndex = 9;
            this.label2.Text = "Введіть пароль:";
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("Century Gothic", 11.25F,
System.Drawing.FontStyle.Bold);
            this.label1.Location = new System.Drawing.Point(306, 108);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(110, 18);
            this.label1.TabIndex = 8;
            this.label1.Text = "Введіть логін:";
            //
            // login
            //
            this.login.Location = new System.Drawing.Point(309, 163);
            this.login.Name = "login";
            this.login.Size = new System.Drawing.Size(196, 20);

```

```

        this.login.TabIndex = 7;
        //
        // password
        //
        this.password.Location = new System.Drawing.Point(309, 255);
        this.password.Name = "password";
        this.password.Size = new System.Drawing.Size(196, 20);
        this.password.TabIndex = 6;
        this.password.UseSystemPasswordChar = true;
        //
        // Authoriz
        //
        this.Authoriz.Font = new System.Drawing.Font("Century Gothic",
11.25F, System.Drawing.FontStyle.Bold);
        this.Authoriz.Location = new System.Drawing.Point(336, 318);
        this.Authoriz.Name = "Authoriz";
        this.Authoriz.Size = new System.Drawing.Size(132, 35);
        this.Authoriz.TabIndex = 5;
        this.Authoriz.Text = "Увійти";
        this.Authoriz.UseVisualStyleBackColor = true;
        this.Authoriz.Click += new System.EventHandler(this.Authoriz_Click);
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.BackColor = System.Drawing.SystemColors.ActiveCaption;
        this.ClientSize = new System.Drawing.Size(800, 450);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.login);
        this.Controls.Add(this.password);
        this.Controls.Add(this.Authoriz);
        this.Name = "Form1";
        this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "Форма авторизації";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.TextBox login;
    private System.Windows.Forms.TextBox password;
    private System.Windows.Forms.Button Authoriz;
}
}

```

**// ManagerClass.cs:**

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diplomniiproekt
{
    internal class ManagerClass
    {

```

```

static public DataTable dtStatus = new DataTable();

static public void GetData()// за допомогою цієї функції ми бачимо
данні на формі
{
    try
    {
        DBConnection.msCommand.CommandText = "SELECT * FROM todo";
        dtStatus.Clear();
        DBConnection.msDataAdapter.SelectCommand =
DBConnection.msCommand;
        DBConnection.msDataAdapter.Fill(dtStatus);
    }
    catch
    {
        MessageBox.Show("Помилка при отриманні даних!", "Помилка... ",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

static public bool AddData(string ToDo, string DateStart, string
DateCompletion, string Done)
{
    try
    {
        DBConnection.msCommand.CommandText = "INSERT INTO todo
VALUES (null, '" + ToDo + "', '" + DateStart + "', '" + DateCompletion + "', '" +
Done + "')";
        if (DBConnection.msCommand.ExecuteNonQuery() > 0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch
    {
        MessageBox.Show("Помилка при додаванні запису!", "Помилка... ",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

static public bool EditData(int Id, string ToDo, string DateStart,
string DateCompletion, string Done)
{
    try
    {
        DBConnection.msCommand.CommandText = "UPDATE todo SET ToDo = '"
+ ToDo + "' WHERE Id = '" + Id + "'";
        DBConnection.msCommand.CommandText = "UPDATE todo SET DateStart
= '" + DateStart + "' WHERE Id = '" + Id + "'";
        DBConnection.msCommand.CommandText = "UPDATE todo SET
DateCompletion = '" + DateCompletion + "' WHERE Id = '" + Id + "'";
        DBConnection.msCommand.CommandText = "UPDATE todo SET Done = '"
+ Done + "' WHERE Id = '" + Id + "'";
        if (DBConnection.msCommand.ExecuteNonQuery() > 0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch

```

```

        {
            MessageBox.Show("Помилка при редагуванні запису!", "Помилка...
", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
    }

    static public void DeleteData(string del)
    {
        try
        {
            DBConnection.msCommand.CommandText = "DELETE FROM todo WHERE Id
= '" + del + "'";
            DBConnection.msCommand.ExecuteNonQuery();
        }
        catch
        {
            MessageBox.Show("Помилка при видаленні вибраного запису!",
"Помилка... ", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
}
}

```

**// TaskManager.Designer.cs:**

```

namespace diplomniiproekt
{
    partial class TaskManager
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.Back = new System.Windows.Forms.Button();
            this.Add_Button = new System.Windows.Forms.Button();
            this.Edit_Button = new System.Windows.Forms.Button();
            this.Remove_Button = new System.Windows.Forms.Button();
            this.labell = new System.Windows.Forms.Label();
            this.dataGridView1 = new System.Windows.Forms.DataGridView();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.backAdd = new System.Windows.Forms.Button();
            this.Add_Task = new System.Windows.Forms.Button();
            this.Donetxt = new System.Windows.Forms.TextBox();
            this.needtoDotxt = new System.Windows.Forms.TextBox();
            this.dateFinishTxt = new System.Windows.Forms.TextBox();
        }
    }
}

```

```

this.label3 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.groupBox2 = new System.Windows.Forms.GroupBox();
this.backEdit = new System.Windows.Forms.Button();
this.Edit_Task = new System.Windows.Forms.Button();
this.Done_txt = new System.Windows.Forms.TextBox();
this.label6 = new System.Windows.Forms.Label();
this.needtoDo_txt = new System.Windows.Forms.TextBox();
this.label5 = new System.Windows.Forms.Label();
this.dateFinish_txt = new System.Windows.Forms.TextBox();
this.label4 = new System.Windows.Forms.Label();
this.label7 = new System.Windows.Forms.Label();
this.dateTimePicker1 = new System.Windows.Forms.DateTimePicker();
this.dateTimePicker2 = new System.Windows.Forms.DateTimePicker();
this.label8 = new System.Windows.Forms.Label();
this.dateTimePicker3 = new System.Windows.Forms.DateTimePicker();
this.dateTimePicker4 = new System.Windows.Forms.DateTimePicker();
this.Id = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.ToDo = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.DateStart = new
System.Windows.Forms.DataGridViewTextBoxColumn();
    this.DateCompletion = new
System.Windows.Forms.DataGridViewTextBoxColumn();
    this.Done = new System.Windows.Forms.DataGridViewTextBoxColumn();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();
    this.groupBox1.SuspendLayout();
    this.groupBox2.SuspendLayout();
    this.SuspendLayout();
    //
    // Back
    //
    this.Back.Location = new System.Drawing.Point(1045, 726);
    this.Back.Margin = new System.Windows.Forms.Padding(4);
    this.Back.Name = "Back";
    this.Back.Size = new System.Drawing.Size(112, 32);
    this.Back.TabIndex = 0;
    this.Back.Text = "Назад";
    this.Back.UseVisualStyleBackColor = true;
    this.Back.Click += new System.EventHandler(this.Back_Click);
    //
    // Add_Button
    //
    this.Add_Button.Location = new System.Drawing.Point(964, 86);
    this.Add_Button.Margin = new System.Windows.Forms.Padding(4);
    this.Add_Button.Name = "Add_Button";
    this.Add_Button.Size = new System.Drawing.Size(193, 47);
    this.Add_Button.TabIndex = 1;
    this.Add_Button.Text = "Додати задачу";
    this.Add_Button.UseVisualStyleBackColor = true;
    this.Add_Button.Click += new
System.EventHandler(this.Add_Button_Click);
    //
    // Edit_Button
    //
    this.Edit_Button.Location = new System.Drawing.Point(964, 209);
    this.Edit_Button.Margin = new System.Windows.Forms.Padding(4);
    this.Edit_Button.Name = "Edit_Button";
    this.Edit_Button.Size = new System.Drawing.Size(193, 47);
    this.Edit_Button.TabIndex = 2;
    this.Edit_Button.Text = "Редагувати задачу";
    this.Edit_Button.UseVisualStyleBackColor = true;
    this.Edit_Button.Click += new
System.EventHandler(this.Edit_Button_Click);
    //
    // Remove_Button
    //
    this.Remove_Button.Location = new System.Drawing.Point(964, 342);
    this.Remove_Button.Margin = new System.Windows.Forms.Padding(4);

```

```

this.Remove_Button.Name = "Remove_Button";
this.Remove_Button.Size = new System.Drawing.Size(193, 47);
this.Remove_Button.TabIndex = 3;
this.Remove_Button.Text = "Видалити";
this.Remove_Button.UseVisualStyleBackColor = true;
this.Remove_Button.Click += new
System.EventHandler(this.Remove_Button_Click);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(3, 35);
this.label1.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(160, 18);
this.label1.TabIndex = 4;
this.label1.Text = "Необхідно зробити:";
this.label1.Click += new System.EventHandler(this.label1_Click);
//
// dataGridView1
//
this.dataGridView1.AllowUserToAddRows = false;
this.dataGridView1.AllowUserToDeleteRows = false;
this.dataGridView1.AllowUserToResizeColumns = false;
this.dataGridView1.AllowUserToResizeRows = false;
this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dataGridView1.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
    this.Id,
    this.ToDo,
    this.DateStart,
    this.DateCompletion,
    this.Done});
this.dataGridView1.Location = new System.Drawing.Point(247, 3);
this.dataGridView1.Margin = new System.Windows.Forms.Padding(4);
this.dataGridView1.Name = "dataGridView1";
this.dataGridView1.ReadOnly = true;
this.dataGridView1.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
this.dataGridView1.Size = new System.Drawing.Size(680, 756);
this.dataGridView1.TabIndex = 5;
//
// groupBox1
//
this.groupBox1.Controls.Add(this.dateTimePicker2);
this.groupBox1.Controls.Add(this.dateTimePicker1);
this.groupBox1.Controls.Add(this.label7);
this.groupBox1.Controls.Add(this.backAdd);
this.groupBox1.Controls.Add(this.Add_Task);
this.groupBox1.Controls.Add(this.Donetxt);
this.groupBox1.Controls.Add(this.needtoDotxt);
this.groupBox1.Controls.Add(this.label3);
this.groupBox1.Controls.Add(this.label2);
this.groupBox1.Controls.Add(this.label1);
this.groupBox1.Location = new System.Drawing.Point(22, 3);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(218, 392);
this.groupBox1.TabIndex = 6;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Додавання";
this.groupBox1.Enter += new
System.EventHandler(this.groupBox1_Enter);
//
// backAdd
//
this.backAdd.Location = new System.Drawing.Point(6, 359);
this.backAdd.Name = "backAdd";
this.backAdd.Size = new System.Drawing.Size(199, 27);

```

```

this.backAdd.TabIndex = 11;
this.backAdd.Text = "Отмена";
this.backAdd.UseVisualStyleBackColor = true;
this.backAdd.Click += new System.EventHandler(this.backAdd_Click);
//
// Add_Task
//
this.Add_Task.Location = new System.Drawing.Point(6, 317);
this.Add_Task.Name = "Add_Task";
this.Add_Task.Size = new System.Drawing.Size(199, 36);
this.Add_Task.TabIndex = 10;
this.Add_Task.Text = "Додати";
this.Add_Task.UseVisualStyleBackColor = true;
this.Add_Task.Click += new System.EventHandler(this.Add_Task_Click);
//
// Donetxt
//
this.Donetxt.Location = new System.Drawing.Point(9, 271);
this.Donetxt.Name = "Donetxt";
this.Donetxt.Size = new System.Drawing.Size(196, 26);
this.Donetxt.TabIndex = 9;
this.Donetxt.TextChanged += new
System.EventHandler(this.Donetxt_TextChanged);
this.Donetxt.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.Donetxt_KeyPress);
//
// needtoDotxt
//
this.needtoDotxt.Location = new System.Drawing.Point(6, 58);
this.needtoDotxt.Name = "needtoDotxt";
this.needtoDotxt.Size = new System.Drawing.Size(199, 26);
this.needtoDotxt.TabIndex = 8;
this.needtoDotxt.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.needtoDotxt_KeyPress);
//
// dateFinishTxt
//
this.dateFinishTxt.Location = new System.Drawing.Point(1300, 496);
this.dateFinishTxt.Name = "dateFinishTxt";
this.dateFinishTxt.Size = new System.Drawing.Size(160, 26);
this.dateFinishTxt.TabIndex = 7;
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(3, 235);
this.label3.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(154, 18);
this.label3.TabIndex = 6;
this.label3.Text = "Зроблено (статус) :";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(3, 168);
this.label2.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(207, 18);
this.label2.TabIndex = 5;
this.label2.Text = "Виберіть дату закінчення:";
//
// groupBox2
//
this.groupBox2.Controls.Add(this.dateTimePicker4);
this.groupBox2.Controls.Add(this.dateTimePicker3);
this.groupBox2.Controls.Add(this.label8);
this.groupBox2.Controls.Add(this.backEdit);
this.groupBox2.Controls.Add(this.Edit_Task);

```

```

this.groupBox2.Controls.Add(this.Done_txt);
this.groupBox2.Controls.Add(this.label6);
this.groupBox2.Controls.Add(this.needtoDo_txt);
this.groupBox2.Controls.Add(this.label5);
this.groupBox2.Controls.Add(this.label4);
this.groupBox2.Location = new System.Drawing.Point(22, 417);
this.groupBox2.Name = "groupBox2";
this.groupBox2.Size = new System.Drawing.Size(218, 342);
this.groupBox2.TabIndex = 7;
this.groupBox2.TabStop = false;
this.groupBox2.Text = "Редарування";
//
// backEdit
//
this.backEdit.Location = new System.Drawing.Point(6, 309);
this.backEdit.Name = "backEdit";
this.backEdit.Size = new System.Drawing.Size(199, 27);
this.backEdit.TabIndex = 12;
this.backEdit.Text = "Отмена";
this.backEdit.UseVisualStyleBackColor = true;
this.backEdit.Click += new System.EventHandler(this.backEdit_Click);
//
// Edit_Task
//
this.Edit_Task.Location = new System.Drawing.Point(6, 267);
this.Edit_Task.Name = "Edit_Task";
this.Edit_Task.Size = new System.Drawing.Size(199, 36);
this.Edit_Task.TabIndex = 11;
this.Edit_Task.Text = "Редарувати";
this.Edit_Task.UseVisualStyleBackColor = true;
this.Edit_Task.Click += new
System.EventHandler(this.Edit_Task_Click);
//
// Done_txt
//
this.Done_txt.Location = new System.Drawing.Point(9, 235);
this.Done_txt.Name = "Done_txt";
this.Done_txt.Size = new System.Drawing.Size(196, 26);
this.Done_txt.TabIndex = 10;
this.Done_txt.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.Done_txt_KeyPress);
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(6, 214);
this.label6.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(154, 18);
this.label6.TabIndex = 9;
this.label6.Text = "Зроблено (статус) :";
this.label6.Click += new System.EventHandler(this.label6_Click);
//
// needtoDo_txt
//
this.needtoDo_txt.Location = new System.Drawing.Point(6, 54);
this.needtoDo_txt.Name = "needtoDo_txt";
this.needtoDo_txt.Size = new System.Drawing.Size(199, 26);
this.needtoDo_txt.TabIndex = 9;
this.needtoDo_txt.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.needtoDo_txt_KeyPress);
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(3, 33);
this.label5.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(160, 18);

```

```
this.label5.TabIndex = 8;
this.label5.Text = "Необхідно зробити:";
//
// dateFinish_txt
//
this.dateFinish_txt.Location = new System.Drawing.Point(1330, 448);
this.dateFinish_txt.Name = "dateFinish_txt";
this.dateFinish_txt.Size = new System.Drawing.Size(160, 26);
this.dateFinish_txt.TabIndex = 8;
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(4, 153);
this.label4.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(207, 18);
this.label4.TabIndex = 7;
this.label4.Text = "Виберіть дату закінчення:";
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(6, 97);
this.label7.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(187, 18);
this.label7.TabIndex = 12;
this.label7.Text = "Виберіть дату початку:";
//
// dateTimePicker1
//
this.dateTimePicker1.Location = new System.Drawing.Point(6, 129);
this.dateTimePicker1.Name = "dateTimePicker1";
this.dateTimePicker1.Size = new System.Drawing.Size(200, 26);
this.dateTimePicker1.TabIndex = 13;
//
// dateTimePicker2
//
this.dateTimePicker2.Location = new System.Drawing.Point(6, 189);
this.dateTimePicker2.Name = "dateTimePicker2";
this.dateTimePicker2.Size = new System.Drawing.Size(200, 26);
this.dateTimePicker2.TabIndex = 14;
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(3, 83);
this.label8.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(187, 18);
this.label8.TabIndex = 13;
this.label8.Text = "Виберіть дату початку:";
//
// dateTimePicker3
//
this.dateTimePicker3.Location = new System.Drawing.Point(5, 113);
this.dateTimePicker3.Name = "dateTimePicker3";
this.dateTimePicker3.Size = new System.Drawing.Size(200, 26);
this.dateTimePicker3.TabIndex = 14;
//
// dateTimePicker4
//
this.dateTimePicker4.Location = new System.Drawing.Point(5, 185);
this.dateTimePicker4.Name = "dateTimePicker4";
this.dateTimePicker4.Size = new System.Drawing.Size(200, 26);
this.dateTimePicker4.TabIndex = 15;
//
// Id
```

```

//
this.Id.DataPropertyName = "Id";
this.Id.HeaderText = "Homep";
this.Id.Name = "Id";
this.Id.ReadOnly = true;
this.Id.Visible = false;
//
// ToDo
//
this.ToDo.DataPropertyName = "ToDo";
this.ToDo.HeaderText = "Потрібно зробити";
this.ToDo.Name = "ToDo";
this.ToDo.ReadOnly = true;
this.ToDo.Width = 220;
//
// DateStart
//
this.DateStart.DataPropertyName = "DateStart";
this.DateStart.HeaderText = "Дата Початку";
this.DateStart.Name = "DateStart";
this.DateStart.ReadOnly = true;
this.DateStart.Width = 130;
//
// DateCompletion
//
this.DateCompletion.DataPropertyName = "DateCompletion";
this.DateCompletion.HeaderText = "Дата закінчення";
this.DateCompletion.Name = "DateCompletion";
this.DateCompletion.ReadOnly = true;
this.DateCompletion.Width = 130;
//
// Done
//
this.Done.DataPropertyName = "Done";
this.Done.HeaderText = "Зроблено";
this.Done.Name = "Done";
this.Done.ReadOnly = true;
this.Done.Width = 159;
//
// TaskManager
//
this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 18F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.ClientSize = new System.Drawing.Size(1169, 771);
this.Controls.Add(this.groupBox2);
this.Controls.Add(this.groupBox1);
this.Controls.Add(this.dataGridView1);
this.Controls.Add(this.Remove_Button);
this.Controls.Add(this.dateFinishTxt);
this.Controls.Add(this.Edit_Button);
this.Controls.Add(this.dateFinish_txt);
this.Controls.Add(this.Add_Button);
this.Controls.Add(this.Back);
this.Font = new System.Drawing.Font("Century Gothic", 11.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte) 204));
this.Margin = new System.Windows.Forms.Padding(4);
this.Name = "TaskManager";
this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "TaskManager";
this.Load += new System.EventHandler(this.TaskManager_Load);

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();
this.groupBox1.ResumeLayout(false);
this.groupBox1.PerformLayout();
this.groupBox2.ResumeLayout(false);
this.groupBox2.PerformLayout();

```

```

        this.ResumeLayout(false);
        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.Button Back;
    private System.Windows.Forms.Button Add_Button;
    private System.Windows.Forms.Button Edit_Button;
    private System.Windows.Forms.Button Remove_Button;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.DataGridView dataGridView1;
    private System.Windows.Forms.GroupBox groupBox1;
    private System.Windows.Forms.GroupBox groupBox2;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label label6;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.Button Add_Task;
    private System.Windows.Forms.TextBox Donetxt;
    private System.Windows.Forms.TextBox needtoDotxt;
    private System.Windows.Forms.TextBox dateFinishTxt;
    private System.Windows.Forms.Button Edit_Task;
    private System.Windows.Forms.TextBox Done_txt;
    private System.Windows.Forms.TextBox needtoDo_txt;
    private System.Windows.Forms.TextBox dateFinish_txt;
    private System.Windows.Forms.Button backAdd;
    private System.Windows.Forms.Button backEdit;
    private System.Windows.Forms.DateTimePicker dateTimePicker2;
    private System.Windows.Forms.DateTimePicker dateTimePicker1;
    private System.Windows.Forms.Label label7;
    private System.Windows.Forms.DataGridViewTextBoxColumn Id;
    private System.Windows.Forms.DataGridViewTextBoxColumn ToDo;
    private System.Windows.Forms.DataGridViewTextBoxColumn DateStart;
    private System.Windows.Forms.DataGridViewTextBoxColumn DateCompletion;
    private System.Windows.Forms.DataGridViewTextBoxColumn Done;
    private System.Windows.Forms.DateTimePicker dateTimePicker4;
    private System.Windows.Forms.DateTimePicker dateTimePicker3;
    private System.Windows.Forms.Label label8;
    }
}

```

#### // TaskManager.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Common;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diplomniiproekt
{
    public partial class TaskManager : Form
    {
        public TaskManager()
        {
            InitializeComponent();
        }

        private void Back_Click(object sender, EventArgs e)

```

```

{
    this.Hide();
    Form1 form = new Form1();
    form.Show();
}

private void TaskManager_Load(object sender, EventArgs e)
{
    this.MinimumSize = this.Size;
    this.MaximumSize = this.Size;
    groupBox1.Visible = false;
    groupBox2.Visible = false;
    ManagerClass.GetData();
    dataGridView1.DataSource = ManagerClass.dtStatus;
}

private void groupBox1_Enter(object sender, EventArgs e)
{
}

private void label1_Click(object sender, EventArgs e)
{
}

private void label6_Click(object sender, EventArgs e)
{
}

private void Add_Button_Click(object sender, EventArgs e)
{
    groupBox1.Visible = true;
}
static public string nomer, Datenachalo, DateFinish, needtodo, Donee;
private void Edit_Button_Click(object sender, EventArgs e)
{
    nomer = dataGridView1.CurrentRow.Cells[0].Value.ToString();
    needtodo = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    //Datenachalo = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    //DateFinish = dataGridView1.CurrentRow.Cells[3].Value.ToString();
    Donee = dataGridView1.CurrentRow.Cells[4].Value.ToString();

    groupBox2.Visible = true;

    needtoDo_txt.Text = needtodo;
    Done_txt.Text = Donee;
}

private void backAdd_Click(object sender, EventArgs e)
{
    groupBox1.Visible = false;
}

private void Remove_Button_Click(object sender, EventArgs e)
{
    try
    {
        string del = dataGridView1.CurrentRow.Cells[0].Value.ToString();
        DialogResult delet = MessageBox.Show("Ви дійсно хочете видалити цю задачу?", "Підтвердження видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (delet == DialogResult.Yes)
        {
            ManagerClass.DeleteData(del);
            ManagerClass.GetData();
            dataGridView1.DataSource = ManagerClass.dtStatus;
        }
    }
}

```

```

        MessageBox.Show("Задача видалена!", "Успіх!",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        }
        else
        {
            }
        }

        catch
        {
            MessageBox.Show("Помилка при видаленні даних!");
        }
    }

    private void needtoDotxt_KeyPress(object sender, KeyPressEventArgs e)
    {
        char l = e.KeyChar;
        if(!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) &&
        !char.IsControl(e.KeyChar))
        {
            e.Handled = true;
        }
    }

    private void Donetxt_TextChanged(object sender, EventArgs e)
    {
    }

    private void needtoDo_txt_KeyPress(object sender, KeyPressEventArgs e)
    {
        char l = e.KeyChar;
        if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) &&
        !char.IsControl(e.KeyChar))
        {
            e.Handled = true;
        }
    }

    private void Done_txt_KeyPress(object sender, KeyPressEventArgs e)
    {
        char l = e.KeyChar;
        if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) &&
        !char.IsControl(e.KeyChar))
        {
            e.Handled = true;
        }
    }

    private void Donetxt_KeyPress(object sender, KeyPressEventArgs e)
    {
        char l = e.KeyChar;
        if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) &&
        !char.IsControl(e.KeyChar))
        {
            e.Handled = true;
        }
    }

    private void Edit_Task_Click(object sender, EventArgs e)
    {
        DateTime ts1 = dateTimePicker1.Value;
        DateTime ts2 = dateTimePicker2.Value;
        dateTimePicker1.CustomFormat = "yyyy-MM-dd"; // Вказуем формат дати,
        який в моїй базі даних
        dateTimePicker2.CustomFormat = "yyyy-MM-dd";
    }

```

```

        // if (ts1 > ts2) //Перевірка на дати(дата завершення не може бути
раніше дати початку)
        // {
        //     MessageBox.Show("Дата завершення не може бути раніше дати
початку", "Попередження", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        //     return;
        // }

        if (needToDo_txt.Text == needtodo)
        {
            if (needToDo_txt.Text != "" && Done_txt.Text != "")
            {
                if (ManagerClass.EditData(int.Parse(nomer),
needToDo_txt.Text, dateTimePicker1.Value.ToString("yyyy-MM-dd"),
dateTimePicker2.Value.ToString("yyyy-MM-dd"), Done_txt.Text))
                {
                    MessageBox.Show("Успішне редагування!", "Успіх!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    ManagerClass.GetData();
                    groupBox2.Visible = false;
                    needToDo_txt.Text = "";
                    Done_txt.Text = "";
                }
            }
            else
            {
                MessageBox.Show("Заповніть всі обов'язкові поля!");
            }
        }
        else
        {
            string sql = @"SELECT Id FROM todo WHERE ToDo = '" + " " + "'";
            DBConnection.msCommand.CommandText = sql;
            Object result = DBConnection.msCommand.ExecuteScalar();
            if(result != null)
            {
                MessageBox.Show("У вас вже є така задача");
                needToDo_txt.Text = "";
            }
            else
            {
                if (needToDo_txt.Text != "" && Done_txt.Text != "")
                {
                    if (ManagerClass.EditData(int.Parse(nomer),
needToDo_txt.Text, dateTimePicker1.Value.ToString("yyyy-MM-dd"),
dateTimePicker2.Value.ToString("yyyy-MM-dd"), Done_txt.Text))
                    {
                        MessageBox.Show("Успішне редагування!", "Успіх!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        ManagerClass.GetData();
                        groupBox2.Visible = false;
                        needToDo_txt.Text = "";
                        Done_txt.Text = "";
                    }
                }
                else
                {
                    MessageBox.Show("Заповніть всі обов'язкові поля!");
                }
            }
        }
    }

    private void backEdit_Click(object sender, EventArgs e)
    {
        groupBox2.Visible = false;
    }

    private void Add_Task_Click(object sender, EventArgs e)

```

```

    {
        DateTime ts1 = dateTimePicker1.Value;
        DateTime ts2 = dateTimePicker2.Value;
        dateTimePicker1.CustomFormat = "yyyy-MM-dd"; // Вказуем формат дати,
який в моїй базі даних
        dateTimePicker2.CustomFormat = "yyyy-MM-dd";

        if(ts1 > ts2) //Перевірка на дати(дата завершення не може бути
раніше дати початку)
        {
            MessageBox.Show("Дата завершення не може бути раніше дати
початку", "Попередження", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        if (needtoDotxt.Text != "" && dateTimePicker1.Text != "" &&
dateTimePicker2.Text != "" && Donetxt.Text != "") //Перевірка на дублювання
таску
        {
            string sql = @"SELECT Id FROM todo WHERE ToDo = '" +
needtoDotxt.Text + "'";
            DBConnection.msCommand.CommandText = sql;
            Object result = DBConnection.msCommand.ExecuteScalar();
            if (result != null)
            {
                MessageBox.Show("У вас вже є такий таск!", "Дублювання
запису", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                dateFinishTxt.Text = "";
                needtoDotxt.Text = "";
                Donetxt.Text = "";
            }
            else
            {
                if (ManagerClass.AddData(needtoDotxt.Text,
dateTimePicker1.Value.ToString("yyyy-MM-dd"),
dateTimePicker2.Value.ToString("yyyy-MM-dd"), Donetxt.Text))
                {
                    MessageBox.Show("Задача успішно додалась!", "Успіх!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    groupBox1.Visible = false;

                    needtoDotxt.Text = "";
                    Donetxt.Text = "";
                    ManagerClass.GetData();
                }
                else
                {
                    MessageBox.Show("Задача не була додана!", "Помилка!",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                }
            }
        }
        else
        {
            MessageBox.Show("Заповніть усі обов'язкові поля!", "Помилка!",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
}

```