

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
Факультет інформатики та обчислювальної техніки  
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії  
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ  
(підпис) (ім'я прізвище)

“ ” \_\_\_\_\_ 2025 р.

**Дипломний проєкт**  
**на здобуття ступеня бакалавра**  
**за освітньо-професійною програмою «Інженерія програмного забезпечення**  
**інформаційних систем»**  
**спеціальності «121 Інженерія програмного забезпечення»**

на тему: Вебзастосунок для автоматизованого розподілу завдань на  
основі матриці компетенцій співробітників

Виконала студентка IV курсу, групи ІІІ-311  
(шифр групи)

Маслова Анастасія Олександрівна \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Керівник доцент кафедри ІІІ, к.т.н., Олійник Ю. О. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант PhD, ст. викл., Стельмах О. П. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент доцент кафедри ІСТ, к.ф.-м.н., доцент Рибачук Л.В. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2025

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення  
інформаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_  
(підпис) Едуард ЖАРІКОВ  
(ім'я прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**

Масловій Анастасії Олександрівні

(прізвище, ім'я, по батькові)

1. Тема проєкту Вебзастосунок для автоматизованого розподілу завдань  
на основі матриці компетенцій співробітників

керівник проєкту Олійник Юрій Олександрович, к.т.н., доцент,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «23» травня 2025 р. №1706-с

2. Термін подання студентом проєкту «16» червня 2025 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Передпроєктне обстеження: основні визначення та терміни, опис предметного середовища, огляд аналогів та алгоритмів, опис бізнес-процесів постановка задачі.

2) Розроблення вимог: опис варіантів використання, функціональні та нефункціональні вимоги, аналіз економічних показників.

3) Конструювання та розроблення: опис архітектури, опис засобів розробки, аналіз безпеки.

4) Аналіз якості та тестування: аналіз якості ПЗ, опис процесів тестування, контрольний приклад.

5) Розгортання та супровід: опис процесів розгортання ПЗ, опис процесів супроводу та підтримки ПЗ.

## 5. Перелік графічного матеріалу

1) Схема структурна варіантів використання

2) Схема структурна діяльності

3) Схема структурна компонентів програмного забезпечення

## 6. Консультанти розділів проєкту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|-------------------------------------------|----------------|------------------|
|        |                                           | завдання видав | завдання прийняв |
|        |                                           |                |                  |

7. Дата видачі завдання «15» березня 2025 року \_\_\_\_\_

## Календарний план

| № з/п | Назва етапів виконання дипломного проєкту           | Термін виконання етапів проєкту | Примітка |
|-------|-----------------------------------------------------|---------------------------------|----------|
| 1     | Вивчення рекомендованої літератури                  | 25.03.2025                      |          |
| 2     | Аналіз існуючих методів розв'язання задачі          | 28.03.2025                      |          |
| 3     | Постановка та формалізація задачі                   | 30.03.2025                      |          |
| 4     | Розробка інформаційного забезпечення                | 16.04.2025                      |          |
| 5     | Алгоритмізація задачі                               | 01.05.2025                      |          |
| 6     | Обґрунтування вибору використаних технічних засобів | 03.05.2025                      |          |
| 7     | Розробка програмного забезпечення                   | 08.05.2025                      |          |
| 8     | Налагодження програми                               | 12.05.2025                      |          |
| 9     | Виконання графічних документів                      | 25.05.2025                      |          |
| 10    | Оформлення пояснювальної записки                    | 1.06.2025                       |          |
| 11    | Подання ДП на попередній захист                     | 02.06.2025                      |          |
| 12    | Подання ДП рецензенту                               | 10.06.2025                      |          |
| 13    | Подання ДП на основний захист                       | 16.06.2025                      |          |

Студент

\_\_\_\_\_  
(підпис)

Анастасія МАСЛОВА

\_\_\_\_\_  
(ініціали, прізвище)

Керівник

\_\_\_\_\_  
(підпис)

Юрій ОЛІЙНИК

\_\_\_\_\_  
(ініціали, прізвище)

## АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 60 таблиць, 25 рисунків та 16 джерел – загалом 74 сторінки.

Дипломний проєкт присвячений розробці вебзастосунку для автоматизованого розподілу завдань.

Метою є спрощення процесу управління людськими ресурсами у ІТ-командах.

У розділі першому розглянуто предметну область розробки, аналогічні програмні рішення, можливі алгоритмічні підходи до вирішення задачі.

Розділ другий присвячений формулюванню вимог до розробки та аналізу економічних показників.

У третьому розділі описано архітектуру та обґрунтовано обрані засоби розробки, проаналізовано ризики, пов'язані з безпекою даних.

Розділ четвертий присвячений якості програмного забезпечення. У ньому описано процеси тестування застосунку та продемонстровано основний функціонал.

У п'ятому розділі описано процес розгортання та супроводу застосунку.

**КЛЮЧОВІ СЛОВА:** ВЕБЗАСТОСУНОК, КЕРУВАННЯ ПРОЄКТАМИ, БАЗА ДАНИХ, АВТОМАТИЗАЦІЯ.

## **ABSTRACT**

The explanatory note of the diploma project consists of five sections, contains 60 tables, 25 figures and 16 sources – in total 74 pages.

The diploma project is dedicated to the development of a web application for automated task distribution.

The goal is to simplify the process of human resource management within teams.

The first chapter reviews the subject area, existing software solutions, and possible algorithmic approaches to solving the task.

The second chapter focuses on defining clear development requirements and analyzing economic indicators.

The third chapter describes the system architecture, justifies the chosen development tools, and analyzes data security risks.

The fourth chapter is dedicated to software quality, describing the testing processes and demonstrating the core functionality.

Chapter 5 covers the deployment and ongoing support of the application.

The software was deployed using Vercel and Render hosting platforms.

**KEYWORDS:** WEB APPLICATION, PROJECT MANAGEMENT, DATABASE, AUTOMATION.



Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ АВТОМАТИЗОВАНОГО РОЗПОДІЛУ  
ЗАВДАНЬ НА ОСНОВІ МАТРИЦІ КОМПЕТЕНЦІЙ СПІВРОБІТНИКІВ**

**Технічне завдання**

КП.ІП-31108.045440.01.91

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ Юрій ОЛІЙНИК

Нормоконтроль:

\_\_\_\_\_ Олександр СТЕЛЬМАХ

Виконавець:

\_\_\_\_\_ Анастасія МАСЛОВА

Київ – 2025

## Зміст

|       |                                                  |    |
|-------|--------------------------------------------------|----|
| 1     | НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ              | 3  |
| 2     | ПІДСТАВА ДЛЯ РОЗРОБКИ                            | 4  |
| 3     | ПРИЗНАЧЕННЯ РОЗРОБКИ                             | 5  |
| 4     | ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ               | 6  |
| 4.1   | Вимоги до функціональних характеристик           | 6  |
| 4.1.1 | Користувацького інтерфейсу                       | 6  |
| 4.1.2 | Для працівника:                                  | 12 |
| 4.1.3 | Для менеджера:                                   | 12 |
| 4.1.4 | Для адміністратора:                              | 12 |
| 4.1.5 | Додаткові вимоги:                                | 13 |
| 4.2   | Вимоги до надійності                             | 13 |
| 4.3   | Умови експлуатації                               | 13 |
| 4.3.1 | Вид обслуговування                               | 13 |
| 4.3.2 | Обслуговуючий персонал                           | 13 |
| 4.4   | Вимоги до складу і параметрів технічних засобів  | 13 |
| 4.5   | Вимоги до інформаційної та програмної сумісності | 14 |
| 4.5.1 | Вимоги до вхідних даних                          | 14 |
| 4.5.2 | Вимоги до вихідних даних                         | 14 |
| 4.5.3 | Вимоги до мови розробки                          | 14 |
| 4.5.4 | Вимоги до середовища розробки                    | 14 |
| 4.5.5 | Вимоги до представленню вихідних кодів           | 14 |
| 4.6   | Вимоги до маркування та пакування                | 14 |
| 4.7   | Вимоги до транспортування та зберігання          | 14 |
| 4.8   | Спеціальні вимоги                                | 14 |
| 5     | ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ                | 15 |
| 5.1   | Попередній склад програмної документації         | 15 |
| 5.2   | Спеціальні вимоги до програмної документації     | 15 |
| 6     | СТАДІЇ І ЕТАПИ РОЗРОБКИ                          | 16 |
| 7     | ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ                    | 17 |



## **1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ**

Назва розробки: Вебзастосунок для автоматизованого розподілу завдань на основі матриці компетенцій співробітників.

Галузь застосування:

Наведене технічне завдання поширюється на розробку вебзастосунку для автоматизованого розподілу завдань на основі матриці компетенцій співробітників [045440], який використовується для оптимізації управління проєктами та призначений для ефективного використання людських ресурсів менеджерами проєктів та HR-фахівцями.

## **2 ПІДСТАВА ДЛЯ РОЗРОБКИ**

Підставою для розробки вебзастосунку для автоматизованого розподілу завдань на основі матриці компетенцій співробітників є завдання на дипломне проєктування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

### **3 ПРИЗНАЧЕННЯ РОЗРОБКИ**

Розробка призначена для управління проєктами та задачами та автоматизованого розподілу задач між працівниками відповідно до їх навичок.

Метою розробки є підвищення ефективності використання людських ресурсів у IT-командах за рахунок оптимального розподілу завдань та зменшення часу на планування проєктів.

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

#### 4.1.1 Користувацького інтерфейсу

- сторінка авторизації користувачів;
- сторінка реєстрації користувачів;
- сторінка відображення проєктів призначених менеджеру, з можливістю перегляду деталей проєкту;
- форма додавання задач;
- зображення результатів розподілу;
- сторінка керування проєктами;
- сторінка керування користувачами;
- сторінка відображення задач;
- сторінка керування компетенціями;
- форма додавання компетенції.

The image shows a web application interface for 'Task Distributor'. At the top left, the text 'Task Distributor' is displayed. At the top right, there is a link 'zareestruvatisia'. The main content area features a rounded rectangular box titled 'Авторизація'. Inside this box, there are two input fields: the first is labeled 'Логін' and the second is labeled 'Пароль'. Below these fields is a button labeled 'Увійти'.

Рисунок 4.1 - Форма авторизації

Task Distributor

Регістрація

Ім'я

Логін

Пароль

Зареєструватися

Рисунок 4.2 - Форма реєстрації

Task Distributor

Ім'я користувача  
вийти

Пошук проектів...

Назва обраного проекту

Непризначені задачі Призначені задачі Команда Матриця компетенцій

Назва проекту

Назва проекту

Назва проекту

Назва проекту

Пошук задач...

| <input type="checkbox"/> | Назва | Пріоритет | Рівень | Дедлайн | Дії                                                                                                                                                                         |
|--------------------------|-------|-----------|--------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | Назва | Пріоритет | Рівень | Дедлайн |   |
| <input type="checkbox"/> | Назва | Пріоритет | Рівень | Дедлайн |   |
| <input type="checkbox"/> | Назва | Пріоритет | Рівень | Дедлайн |   |

Рисунок 4.3 - Непризначені задачі проекту

Task Distributor

Ім'я користувача  
вийти

Пошук проектів...

Назва обраного проекту

Непризначені задачі Призначені задачі Команда Матриця компетенцій

Назва проекту

Назва проекту

Назва проекту

Назва проекту

Пошук задач...

|  | Назва | Пріоритет | Рівень | Призначено | Дедлайн | Дії                                                                                                                                                                         |
|--|-------|-----------|--------|------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Назва | Пріоритет | Рівень | Виконавець | Дедлайн |   |
|  | Назва | Пріоритет | Рівень | Виконавець | Дедлайн |   |
|  | Назва | Пріоритет | Рівень | Виконавець | Дедлайн |   |

Рисунок 4.4 - Призначені задачі проекту

Task Distributor

Ім'я користувача  
вийти

Пошук проєктів...

Назва обраного проєкту

Непризначені задачіПризначені задачіКомандаМатриця компетенцій

Назва проєкту

Пошук робітників...

+

Назва проєкту

Ім'я працівника

Посада

Назва проєкту

Ім'я працівника

Посада

Назва проєкту

Ім'я працівника

Посада

Рисунок 4.5 - Працівники проєкту

Task Distributor

Ім'я користувача  
вийти

Пошук проєктів...

Назва обраного проєкту

Непризначені задачіПризначені задачіКомандаМатриця компетенцій

Назва проєкту

|           | Компетенція      | Компетенція      | Компетенція      | Компетенція      | Компетенція      |
|-----------|------------------|------------------|------------------|------------------|------------------|
| Працівник | рівень володіння | рівень володіння | рівень володіння | рівень володіння | рівень володіння |
| Працівник | рівень володіння | рівень володіння | рівень володіння | рівень володіння | рівень володіння |
| Працівник | рівень володіння | рівень володіння | рівень володіння | рівень володіння | рівень володіння |
| Працівник | рівень володіння | рівень володіння | рівень володіння | рівень володіння | рівень володіння |
| Працівник | рівень володіння | рівень володіння | рівень володіння | рівень володіння | рівень володіння |

Назва проєкту

Назва проєкту

Назва проєкту

Рисунок 4.6 - Компетенції працівників проєкту

|                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |
|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Task Distributor                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Ім'я користувача<br><a href="#">вийти</a> |
| <input style="width: 90%;" type="text" value="Пошук проектів..."/> | <div style="border: 1px solid #ccc; border-radius: 15px; padding: 10px; margin: 10px auto; width: 60%;"> <h3 style="text-align: center; margin: 0;">Задача</h3> <p>Назва<br/><input style="width: 95%;" type="text"/></p> <p>Опис<br/><input style="width: 95%;" type="text"/></p> <p>Пріоритет<br/><input style="width: 95%;" type="text"/></p> <p>Рівень<br/><input style="width: 95%;" type="text"/></p> <p>Виконавець<br/><input style="width: 95%;" type="text"/></p> <p>Технології<br/><input style="width: 95%;" type="text"/></p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <input type="button" value="Скасувати"/> <input type="button" value="Зберегти"/> </div> </div> |                                           |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |
|                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |

Рисунок 4.7 - Форма створення задачі

|                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|--------------------------|--------|------------|--------------------------|-------|------------|--------------------------|-------|------------|--------------------------|-------|------------|
| Task Distributor                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Ім'я користувача<br><a href="#">вийти</a> |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| <input style="width: 90%;" type="text" value="Пошук проектів..."/> | <div style="border: 1px solid #ccc; border-radius: 15px; padding: 10px; margin: 10px auto; width: 60%;"> <h3 style="text-align: center; margin: 0;">Результати розподілу</h3> <div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <span>Призначені</span> <span>Непризначені</span> </div> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">Задача</td> <td style="text-align: center;">Призначено</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">Назва</td> <td style="text-align: center;">Виконавець</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">Назва</td> <td style="text-align: center;">Виконавець</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">Назва</td> <td style="text-align: center;">Виконавець</td> </tr> </table> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <input type="button" value="Скасувати"/> <input type="button" value="Зберегти"/> </div> </div> |                                           | <input type="checkbox"/> | Задача | Призначено | <input type="checkbox"/> | Назва | Виконавець | <input type="checkbox"/> | Назва | Виконавець | <input type="checkbox"/> | Назва | Виконавець |
| <input type="checkbox"/>                                           | Задача                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Призначено                                |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| <input type="checkbox"/>                                           | Назва                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Виконавець                                |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| <input type="checkbox"/>                                           | Назва                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Виконавець                                |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| <input type="checkbox"/>                                           | Назва                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Виконавець                                |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
| Назва проекту                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |                          |        |            |                          |       |            |                          |       |            |                          |       |            |
|                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |                          |        |            |                          |       |            |                          |       |            |                          |       |            |

Рисунок 4.8 - Результати розподілу

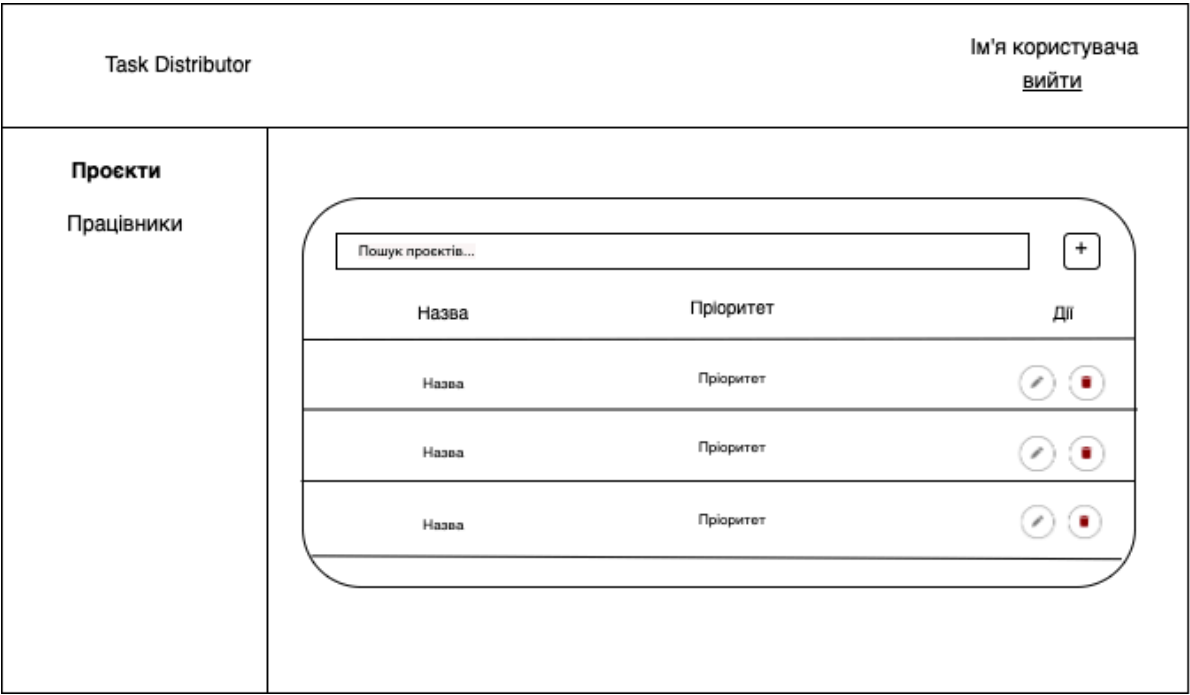


Рисунок 4.9 - Список проектів

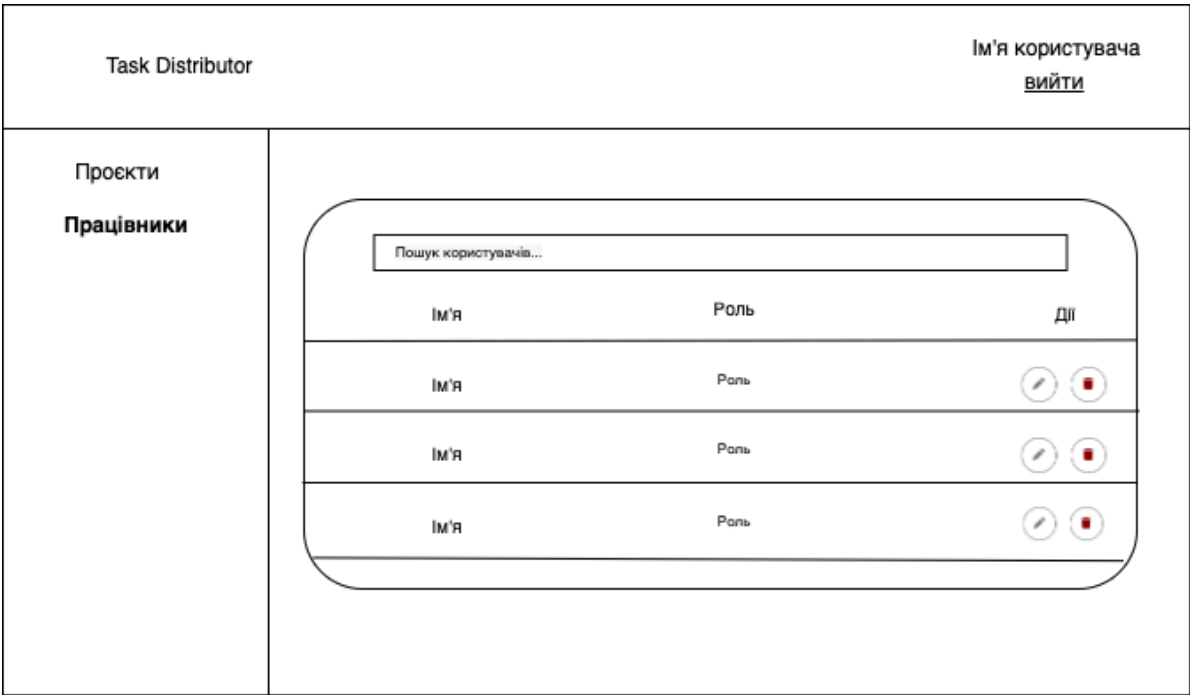


Рисунок 4.10 - Список користувачів



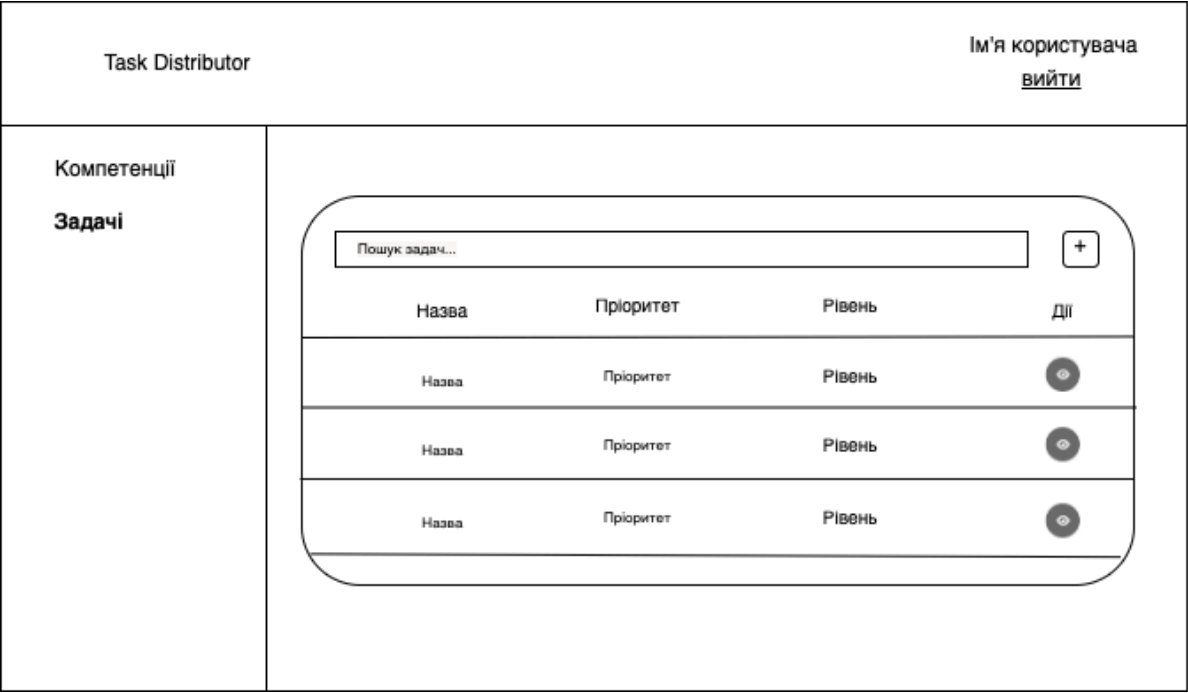


Рисунок 4.11 - Список задач працівника

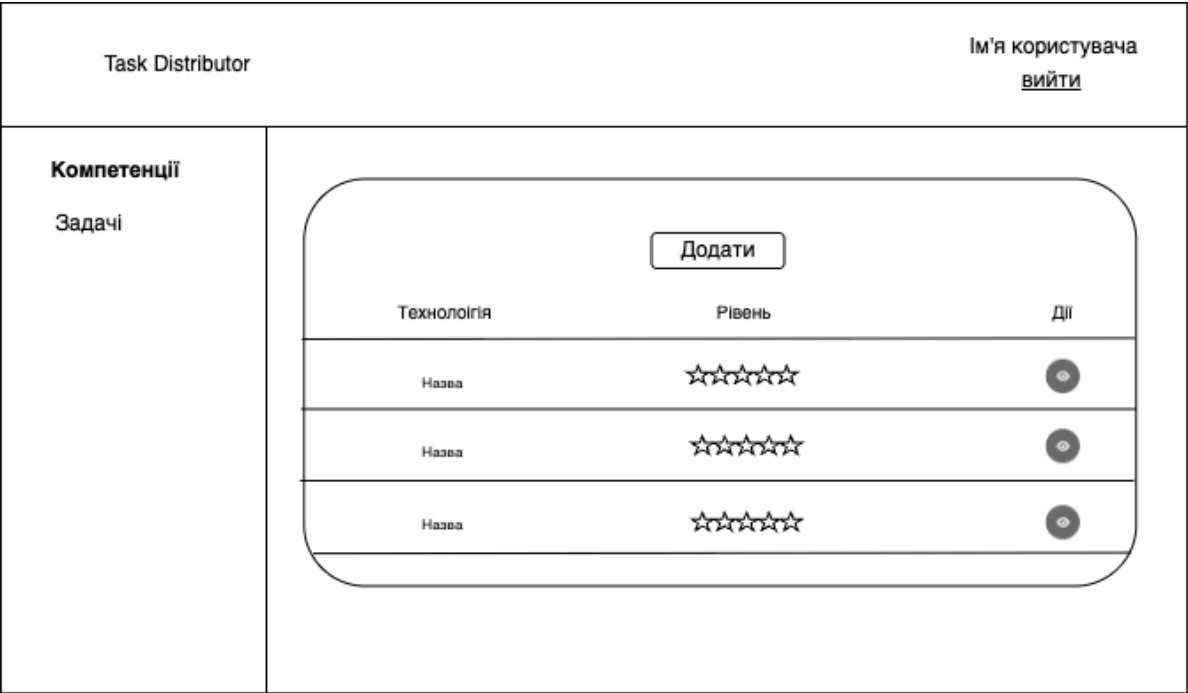


Рисунок 4.12 - Список компетенцій працівника

|                                  |                                                                                                                                                                                                                                                                                                                                                                      |                                           |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Task Distributor                 |                                                                                                                                                                                                                                                                                                                                                                      | Ім'я користувача<br><a href="#">вийти</a> |
| <b>Компетенції</b><br><br>Задачі | <div style="border: 1px solid black; border-radius: 15px; padding: 10px; width: fit-content; margin: 0 auto;"> <p><b>Задача</b></p> <p>Технологія</p> <input style="width: 150px;" type="text"/> <p>Рівень</p> <p>☆☆☆☆☆</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <span>Скасувати</span> <span>Зберегти</span> </div> </div> |                                           |

Рисунок 4.13 - Форма додавання компетенції

#### 4.1.2 Для працівника:

- авторизація та аутентифікація;
- перегляд призначених задач;
- додавання та редагування компетенцій.

#### 4.1.3 Для менеджера:

- авторизація та аутентифікація;
- перегляд призначених проєктів;
- перегляд деталей проєкту (призначені та непризначені задачі, команда, матриця компетенцій команди);
- додавання та призначення задач.

#### 4.1.4 Для адміністратора:

- авторизація та аутентифікація;
- перегляд та редагування проєктів;
- перегляд та редагування користувачів (можливість змінювати роль).

#### 4.1.5 Додаткові вимоги:

- додаткові вимоги не висуваються.

### 4.2 Вимоги до надійності

Передбачити контроль введення інформації та захист від некоректних дій користувача. Забезпечити цілісність інформації в базі даних.

### 4.3 Умови експлуатації

Умови експлуатації згідно СанПін 2.2.2.542 – 96.

#### 4.3.1 Вид обслуговування

Вимоги до виду обслуговування не висуваються.

#### 4.3.2 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висуваються.

### 4.4 Вимоги до складу і параметрів технічних засобів

Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 4 Гб;
- підключення до мережі Інтернет зі швидкістю від 16 мегабіт.

Рекомендована конфігурація технічних засобів:

- тип процесору: Intel Core i7;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 20 мегабіт.

#### 4.5 Вимоги до інформаційної та програмної сумісності

Програмне забезпечення повинно коректно працювати в сучасних веббраузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge та Safari.

Застосунок сумісний з основними операційними системами (Windows, macOS, Linux) через вебінтерфейс. Серверна частина може бути розгорнута в середовищі, що підтримує Node.js.

##### 4.5.1 Вимоги до вхідних даних

- вимоги до вхідних даних не висуваються.

##### 4.5.2 Вимоги до вихідних даних

- вимоги до вихідних даних не висуваються.

##### 4.5.3 Вимоги до мови розробки

Розробку виконати на мові програмування TypeScript.

##### 4.5.4 Вимоги до середовища розробки

Розробку виконати на платформі WebStorm.

##### 4.5.5 Вимоги до представлення вихідних кодів

Вихідний код програми має бути представлений у вигляді посилання на GitHub.

#### 4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

#### 4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

#### 4.8 Спеціальні вимоги

Згенерувати інсталяційну версію програмного забезпечення.

## **5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ**

### **5.1 Попередній склад програмної документації**

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- пояснювальна записка;
- технічне завдання;
- текст програми;
- програма та методика тестування;
- керівництво користувача.

Графічна частина повинна бути виконана на аркушах формату А3 та містити наступні документи:

- схема структурна варіантів використання;
- схема структурна діяльності;
- схема структурна компонентів програмного забезпечення.

### **5.2 Спеціальні вимоги до програмної документації**

Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

## 6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

| №  | Назва етапу                                                               | Строк | Звітність                                                                                                |
|----|---------------------------------------------------------------------------|-------|----------------------------------------------------------------------------------------------------------|
| 1. | Вивчення літератури за тематикою проєкту                                  | 25.03 |                                                                                                          |
| 2. | Розробка технічного завдання                                              | 20.04 | Технічне завдання                                                                                        |
| 3. | Аналіз вимог та уточнення специфікацій                                    | 25.04 | Специфікації програмного забезпечення                                                                    |
| 4. | Проектування структури програмного забезпечення, проектування компонентів | 30.04 | Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму) |
| 5. | Програмна реалізація програмного забезпечення                             | 05.05 | Тексти програмного забезпечення                                                                          |
| 6. | Тестування програмного забезпечення                                       | 20.05 | Тести, результати тестування                                                                             |
| 7. | Розробка матеріалів текстової частини проєкту                             | 01.06 | Пояснювальна записка                                                                                     |
| 8. | Розробка матеріалів графічної частини проєкту                             | 02.06 | Графічний матеріал проєкту                                                                               |
| 9. | Оформлення технічної документації проєкту                                 | 02.06 | Технічна документація                                                                                    |

## **7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ**

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.



**Пояснювальна записка  
до дипломного проєкту**

на тему: **Вебзастосунок для автоматизованого розподілу завдань на основі  
матриці компетенцій співробітників**

КП.ІП-з1108.045440.02.81

Київ – 2025

## ЗМІСТ

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>ВСТУП.....</b>                                                      | <b>4</b>  |
| <b>1 ПЕРЕДПРОЄКТНЕ ОБСТЕЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....</b>              | <b>5</b>  |
| 1.1 Постановка завдання дипломного проєктування.....                   | 5         |
| 1.2 Аналіз предметної області.....                                     | 5         |
| 1.3 Аналіз існуючих рішень.....                                        | 8         |
| 1.3.1 Аналіз відомих програмних продуктів.....                         | 8         |
| 1.3.2 Аналіз відомих алгоритмічних та технічних рішень.....            | 12        |
| 1.4 Аналіз та моделювання бізнес-процесів.....                         | 16        |
| Висновки до розділу.....                                               | 17        |
| <b>2 РОЗРОБЛЕННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>            | <b>19</b> |
| 2.1 Варіанти використання програмного забезпечення.....                | 19        |
| 2.2 Розроблення функціональних вимог.....                              | 26        |
| 2.3 Розроблення нефункціональних вимог.....                            | 32        |
| 2.4 Аналіз економічних показників програмного забезпечення.....        | 33        |
| 2.5 Постановка завдання на розробку програмного забезпечення.....      | 42        |
| Висновки до розділу.....                                               | 43        |
| <b>3 КОНСТРУЮВАННЯ ТА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>    | <b>44</b> |
| 3.1 Архітектура програмного забезпечення.....                          | 44        |
| 3.2 Архітектурні рішення та обґрунтування вибору засобів розробки..... | 45        |
| 3.3 Конструювання програмного забезпечення.....                        | 48        |
| 3.3.1 Розробка алгоритму.....                                          | 48        |
| 3.3.2 Опис структури бази даних.....                                   | 49        |
| 3.4 Аналіз безпеки даних.....                                          | 52        |
| Висновки до розділу.....                                               | 53        |
| <b>4 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>     | <b>54</b> |
| 4.1 Аналіз якості ПЗ.....                                              | 54        |
| 4.2 Опис процесів тестування.....                                      | 55        |
| 4.3 Опис контрольного прикладу.....                                    | 61        |
| Висновки до розділу.....                                               | 64        |
| <b>5 РОЗГОРТАННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>         | <b>65</b> |
| 5.1 Розгортання програмного забезпечення.....                          | 65        |
| 5.2 Супровід програмного забезпечення.....                             | 69        |
| Висновки до розділу.....                                               | 70        |
| <b>ВИСНОВКИ.....</b>                                                   | <b>71</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>                                 | <b>72</b> |
| <b>ДОДАТОК А. Звіт на співпадіння тексту.....</b>                      | <b>74</b> |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

|     |                                                                         |
|-----|-------------------------------------------------------------------------|
| IDE | – Integrated Development Environment – інтегроване середовище розробки. |
| API | – Application programming interface, прикладний програмний Інтерфейс.   |
| IT  | – Інформаційні технології.                                              |
| ER  | – Entity-Relation diagram.                                              |
| БД  | – База даних                                                            |

## ВСТУП

Сучасні IT-компанії функціонують у середовищі, де ефективне управління проєктами має вирішальне значення для досягнення бізнес-цілей. Актуальною проблемою є неефективний розподіл задач між співробітниками. Автоматизація цього процесу дозволить зменшити витрати часу на ручне керування завданнями та уникнути перевантаження членів команди.

Світові тенденції демонструють активний розвиток систем управління, проте рівень автоматизації процесів залишається низьким та потребує налаштування правил, що включає суб'єктивний фактор, як і у ручному розподілі.

Враховуючи зазначене, тематика дипломного проєкту є актуальною і відповідає сучасним потребам індустрії. Метою розробки є створення інструменту, що спрощує процес управління задачами та дозволяє ефективніше використовувати кадровий потенціал. Застосунок може бути інтегрований у внутрішні процеси у IT-компаніях або бути розширений як окремий модуль більш масштабних систем.

Сфера застосування розробки охоплює менеджерів проєктів і HR-відділи, зацікавлені в оптимізації внутрішніх бізнес-процесів.

## 1 ПЕРЕДПРОЄКТНЕ ОБСТЕЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Постановка завдання дипломного проєктування

Дипломне проєктування передбачає виконання наступних завдань:

- аналіз предметної області та опис її ключових бізнес-процесів, визначення загального завдання розробки у рамках ДП;
- аналіз наявних рішень обраного завдання розробки у рамках ДП;
- аналіз та моделювання бізнес-процесів ДП;
- розроблення функціональних, нефункціональних та системних вимог до програмного забезпечення ДП;
- аналіз економічних показників програмного забезпечення ДП;
- постановка завдання на розробку програмного забезпечення ДП;
- розроблення архітектури програмного забезпечення ДП;
- розроблення архітектурних рішень та обґрунтування вибору засобів розробки програмного забезпечення ДП;
- конструювання та розроблення програмного забезпечення ДП;
- аналіз безпеки даних програмного забезпечення ДП;
- аналіз якості та тестування програмного забезпечення ДП;
- розгортання та супровід програмного забезпечення ДП;
- створення супроводжувальної документації до розробленого програмного забезпечення ДП.

### 1.2 Аналіз предметної області

Ефективний розподіл кадрових ресурсів вимагає значних зусиль з боку відповідального за проєкт, адже факторів для визначення виконавця може бути забагато для одночасного опрацювання однією людиною. На вибір також впливають суб'єктивні чинники, які можуть ускладнювати організацію процесу планування.

Автоматизований розподіл завдань на основі матриці компетенцій є прогресивним підходом до управління людськими ресурсами та планування робіт.

#### Ключові поняття

Проектний менеджмент [1] — це сфера знань, що охоплює планування, організацію та управління ресурсами для досягнення цілей проєкту та задоволення бізнес-потреб.

Матриця компетенцій — це інструмент, який використовується для оцінки навичок співробітників у компанії або проєкті. Вона допомагає визначати знання і досвід команди для ефективного призначення завдань та виявлення прогалин у компетенціях команди в цілому.

Компетенція — це набір знань та навичок, необхідний для виконання завдань, пов'язаний із певною сферою діяльності та інструментами реалізації.

Завдання — це сформульована на основі потреб і вимог задача, що має очікуваний результат.

Автоматизоване розподілення завдань - процес призначення завдань співробітникам за допомогою алгоритму прийняття рішень на основі критеріїв (у даному випадку - доступність часового ресурсу та компетенції співробітників)

Через стрімке зростання потреби у ефективному управлінні ресурсами пропонується широкий вибір сервісів, що спрощують розподіл завдань у команді. Проте основним недоліком найпопулярніших сервісів і їх аналогів є відсутність гнучких можливостей для використання матриці компетенцій, що обмежує точність результату алгоритмів розподілу. Предметна область розробки охоплює питання управління проєктами та інформаційних технологій, але не враховує фінансові аспекти розподілу ресурсів та юридичні аспекти трудових відносин.

### Актуальні підходи до розподілу завдань

На сучасному етапі розвитку ІТ-технологій використовується декілька підходів до призначення завдань. Частина існуючих систем управління проєктами надає базовий функціонал для розподілу завдань, але не використовує аналіз компетенції співробітників. Деякі рішення пропонують модулі управління талантами, які враховують компетенції, але не є повністю інтегрованими у процеси розподілу.

Типовий процес розподілу завдань в команді включає такі кроки:

- створення завдання в системі;
- оцінка необхідних для виконання завдання компетенцій;
- ручний розподіл завдань менеджером без урахування поточного навантаження або обрання задач співробітниками;
- перепризначення завдання у випадку відсутності або недостатнього рівня компетенцій співробітника.

Недоліками процесу є відсутність формалізації компетенцій працівників, переважно ручний розподіл завдань та неоптимальне використання людського ресурсу через упущення критичних факторів вибору.

Для подолання наявних недоліків можливі наступні заходи:

- розробка рішень, що враховують навантаженість, компетенції та їх рівень для розподілу завдань;
- впровадження алгоритму автоматизованого підбору виконавців завдання;
- створення зручного інтерфейсу управління компетенціями співробітників.

В рамках дипломного проєкту планується розробка вебзастосунка для автоматизованого розподілу завдань на основі формалізованої матриці компетенцій з урахуванням пріоритетності завдань та поточної завантаженості працівників.

Передбачається автоматизований і ручний спосіб призначення завдань для забезпечення гнучкості системи. Реалізація інтуїтивно зрозумілого

інтерфейсу для двох основних ролей користувачів (менеджери та інші працівники) сприятиме ефективному впровадженню системи в робочі процеси організації.

### 1.3 Аналіз існуючих рішень

Для визначення актуальності розробки та ознайомлення із функціоналом аналогів було розглянуто декілька програмних рішень, що використовуються компаніями для ефективного структурування та розподілення завдань.

#### 1.3.1 Аналіз відомих програмних продуктів

Для порівняння функції проєкту та аналогічних сервісів було обрано 4 програмні рішення кожне з яких описано далі.

TempoPlanner [2] є інструментом планування ресурсів, що інтегрується з Jira і дозволяє візуалізувати розподіл робочого часу та завантаженість команди. Даний сервіс орієнтований переважно на часовий розподіл ресурсів, надає можливість вказувати ролі і навички. Проте функціонал роботи з компетенціями є обмеженим, адже вони не оцінюються за рівнем володіння.

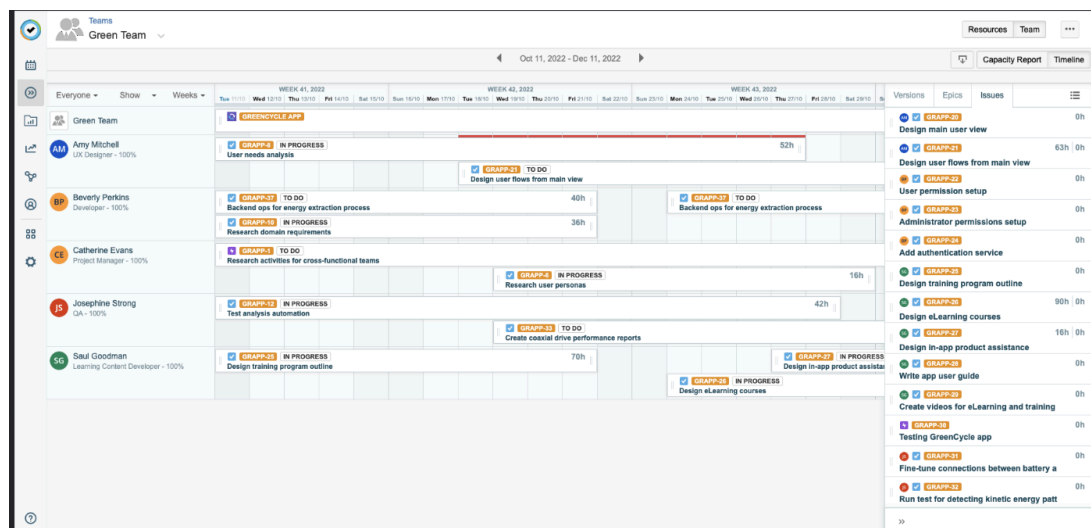


Рисунок 1.1 - Інтерфейс TempoPlanner

Monday [3] представляє універсальну платформу для управління проєктами, що має гнучкі можливості налаштування робочих процесів. Серед функцій сервісу є призначення відповідальних осіб та відстеження виконання



завдань, проте система не має спеціалізованого модуля для формалізації матриці компетенцій співробітників. Альтернативно можна створювати стовпці із тегами навичок, але функція автоматизованого розподілу із їх врахуванням відсутня. Автоматизація розподілу базується на простих правилах “якщо-то”, визначеними користувачем.

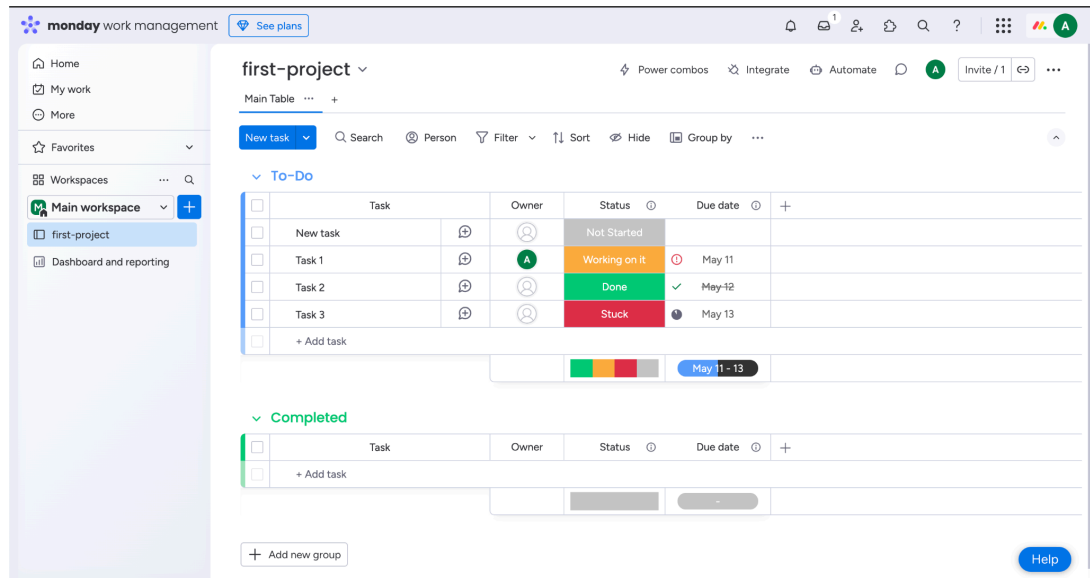


Рисунок 1.2 - Інтерфейс Monday

ClickUp [4] є комплексним рішенням управління проектами з розширеними функціями автоматизації. Платформа дозволяє створювати власні поля та фільтри, що теоретично можна використати для збереження компетенцій, проте відсутня система їх оцінювання. Алгоритми розподілу завдань також не враховують компетенції, що важливо для найбільш відповідного підбору виконавця. Система спрямована на шаблонізацію процесів та загальне управління завданнями, а не на автоматизоване.

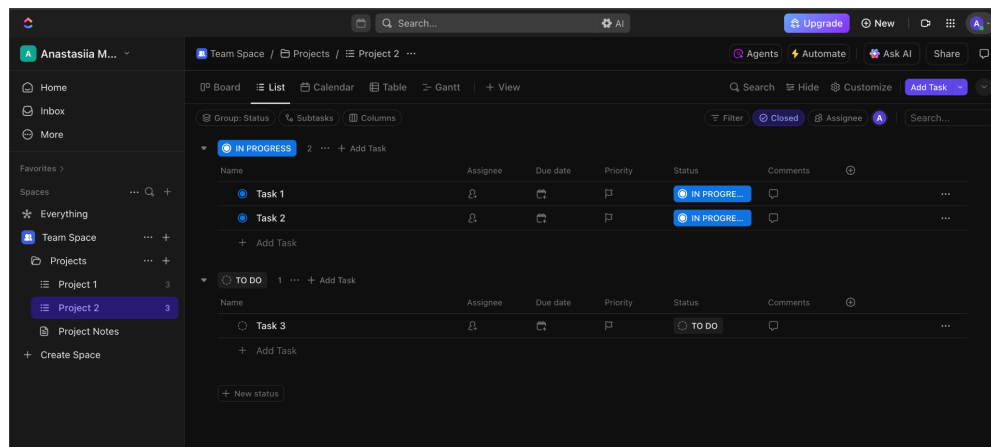


Рисунок 1.3 - Інтерфейс ClickUp

Asana [5] є системою управління, що спеціалізується на відстеженні прогресу та комунікації у команді. Функціонал роботи з компетенціями включає лише можливість призначати загальні ролі та додавати неструктуровані теги навичок до профілів. Сервіс не оптимізує розподіл задач, а зосереджений переважно на відстеженні статусів і дедлайнів, що обмежує можливості ефективного управління ресурсами.

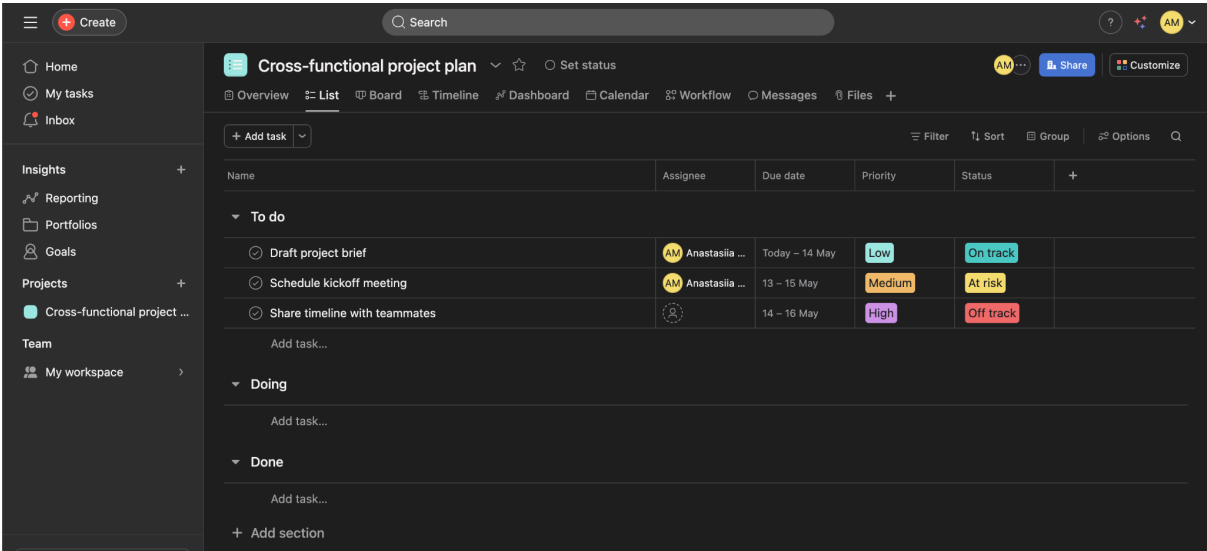


Рисунок 1.4 - Інтерфейс Asana

Для аналізу наявних програмних рішень скористаємося таблицею 1.1

Таблиця 1.1 - Порівняння з аналогами

| Функціонал                                      | Дипломний проєкт | Tempo Planner | Monday | ClickUp | Asana | Пояснення                                                                                                                 |
|-------------------------------------------------|------------------|---------------|--------|---------|-------|---------------------------------------------------------------------------------------------------------------------------|
| Можливість зазначати компетенції співробітник а | +                | +             | +      | -       | -     | Деякі з застосунків надають можливість додавати компетенції до профілів співробітників через теги або користувачі кі поля |

Продовження таблиці 1.1

| Функціонал                                                                          | Дипломний проект | Tempo Planner | Monday | ClickUp | Asana | Пояснення                                                                                                             |
|-------------------------------------------------------------------------------------|------------------|---------------|--------|---------|-------|-----------------------------------------------------------------------------------------------------------------------|
| Створення та редагування матриці компетенцій                                        | +                | -             | -      | -       | -     | Проект, що розглядається, дозволяє вносити дані про навички співробітників та їх рівень                               |
| Наявність автоматизованого розподілу завдань, що враховує відповідність компетенцій | +                | -             | -      | -       | -     | З аналогів ClickUp має функціональність “Automations”, де можна налаштувати розподіл без врахування потрібних навичок |
| Можливість вказувати необхідні для задачі технології                                | +                | +             | +      | +       | +     | Усі програмні рішення дозволяють обрати технології потрібні для виконання завдання                                    |

Аналіз наявних рішень продемонстрував, що серед рішень для управління проектами відсутні спеціалізовані системи, які надають можливість автоматизованого розподілу завдань із урахуванням компетенцій співробітників. Це доводить актуальність розробки вебзастосунку, який

спрямований на оптимізацію процесу розподілу задач з урахуванням об'єктивних критеріїв оцінки.

### 1.3.2 Аналіз відомих алгоритмічних та технічних рішень

Для задач призначення та розподілу ресурсів використовуються різні алгоритмічні підходи [6], вибір яких залежить від кінцевої мети задачі, кількості критеріїв та складності проблеми. Розглянемо деякі з них:

Угорський алгоритм [7] - це алгоритм оптимізації, який вирішує задачу про призначення за поліноміальний час  $O(n^3)$ .

Лінійне програмування [8] — це метод оптимізації, який шукає максимум або мінімум лінійної цільової функції за наявності лінійних обмежень.

Транспортна задача — це спеціальний випадок задачі лінійного програмування який вирішує проблему розподілу ресурсів від джерел (постачальників) до споживачів з мінімізацією загальної вартості.

Генетичні алгоритми [9] застосовуються для багатокритеріальних задач оптимізації та може знаходити близькі до оптимальних рішення швидко, проте має високу обчислювальну складність, може давати різні результати при кожному запуску.

Жадібні алгоритми [10] обирають найкращий варіант з доступних не переглядаючи наперед і спрямовані на отримання максимальної вигоди. Вони мають переваги у швидкості та простоті модифікації за чітких вхідних даних задачі.

#### Обґрунтування вибору алгоритму

Для обґрунтування вибору жадібного алгоритму з модифікаціями було проведено аналіз успішності розглянутих алгоритмів і їх часову складність. Успішність алгоритму розподілу завдань можна формалізувати як частку завдань, для яких було знайдено виконавця, компетенції якого повністю покривають вимоги завдання.

Для визначення складності алгоритму врахуємо модифікації, що включають сортування задач за пріоритетом та датою виконання та сортування кандидатів. Часова складність складає

$O(n \times \log(n) + n \times m \times k + n \times m \times \log(m))$ , де  $n$  - це кількість завдань,  $m$  - кількість працівників,  $k$  - середня кількість компетенцій завдання.

Домінуючий фактор - це член у формулі складності, який зростає найшвидше при збільшенні розміру вхідних даних і визначає загальну продуктивність алгоритму. У реальних умовах ІТ-проєктів, де кількість компетенцій, необхідних для задачі складає від 5 до 15 технологій, домінуючим фактором складності алгоритму є  $O(n \times m \times k)$ . Складність інших алгоритмів не потребує аналізу, для порівняння їх вказано у таблиці 1.2.

Угорський алгоритм досягає математичної точності 95-100%, знаходячи оптимальне призначення за критерієм мінімізації вартості або часу. Однак практична ефективність падає до 70-80%, оскільки алгоритм не враховує рівні навичок та потребує модифікацій для врахування компетенцій.

Результати застосування транспортної задачі ефективні на 85-95% для однорідних ресурсів, але практично досягають 60-75% через невідповідність природі ІТ-завдань.

Методи лінійного програмування досягають високої точності 90-95% при правильному формулюванні цільової функції і обмежень. Проте практична успішність знижується до 60-75% через високу складність повного опису реальних обмежень у вигляді рівнянь та нерівностей.

Генетичні алгоритми показують математичну точність 70-85%, проте практична успішність складає 65-75%, що обумовлено непередбачуваністю результатів та складністю налаштування параметрів алгоритму для ІТ-проєктів.

Модифікований жадібний алгоритм демонструє вищу успішність при застосуванні його у задачі розподілу, що відбувається завдяки адаптації під специфіку ІТ-проєктів: врахування технічних навичок, пріоритетів завдань та

балансування навантаження команди. Він забезпечує баланс між успішністю розподілу та швидкістю виконання, що важливо для динамічних ІТ-проектів.

У таблиці 1.2 наведено головні характеристики алгоритмів, розглянуті при аналізі.

Таблиця 1.2 - Аналіз алгоритмічних рішень

| Алгоритм               | Складність                        | Недоліки                                                                  | Математична (теоретична) успішність | Практична успішність для цільової задачі |
|------------------------|-----------------------------------|---------------------------------------------------------------------------|-------------------------------------|------------------------------------------|
| Модифікований жадібний | $O(n \times m \times k)$          | Пошук локального оптимуму                                                 | 70-90%                              | 80-90%                                   |
| Угорський              | $O(\max(n, m)^3)$                 | Складні модифікації для врахування критеріїв                              | 95-100%                             | 70-80%                                   |
| Транспортна задача     | $O((n + m)^3)$                    | Призначена для однорідних ресурсів, не підходить для різнорідних ІТ-задач | 85-95%                              | 60-85%                                   |
| Лінійне програмування  | $O((n \times m)^3)$               | Висока складність формулювання цільової функції                           | 90-95%                              | 70-85%                                   |
| Генетичні              | $O(G \times P \times n \times m)$ | Стохастичність, складність налаштування параметрів                        | 70-85%                              | 65-75%                                   |

Змінні, що застосовуються для опису складності:

- $n$  - кількість задач;
- $m$  - кількість працівників;
- $k$  - середня кількість компетенцій;
- $G$  - кількість поколінь (ітерацій);
- $P$  - розмір популяцій (кількість особин).

## Технічні рішення

Для реалізації застосунку необхідно обрати технічні інструменти. Розглянемо наступні варіанти архітектури проєкту:

1. Монолітна архітектура характеризується єдиною кодовою базою та компонентами, що тісно пов'язані між собою та не можуть існувати окремо, що робить систему негнучкою та складною у масштабуванні.

2. Мікросервісна архітектура - підхід, при якому система розділена на сервіси, кожен з яких відповідає за частину функціональності. Для спілкування між сервісами використовується API, що робить систему гнучкою у виборі технологій для різних компонентів і масштабуванні. Мікросервісна архітектура не є оптимальною для дипломного проєкту, адже розробка не передбачає використання великої кількості функціональних блоків, що потребують незалежного масштабування.

3. Клієнт-серверна архітектура [11] відокремлює компонент, який відповідає за інтерфейс користувача від серверної частини, що відповідає за бізнес-логіку та доступ до даних. Компоненти взаємодіють через мережу за допомогою протоколу, найчастіше http. Для розробки дипломного проєкту обрано саме цю архітектуру через можливість незалежного розширення та тестування компонентів. Через відсутність потреби у розділенні серверної частини на незалежні мікросервіси, бекенд проєкту буде монолітним, що є популярним підходом для проєктів низької та середньої складності.

### Технічний стек

#### Клієнтська частина

Для створення інтерфейсу користувача було обрано фреймворк для створення SPA(односторінкових додатків) Angular. Технологія дозволяє використовувати RxJS для ефективної обробки асинхронних операцій, використовує мову програмування TypeScript для безпечної типізації даних.

#### Серверна частина

Для бекенд частини проєкту доречно використати фреймворк NestJS, що є надбудовою над фреймворком Node.js і використовує мову

програмування TypeScript. Перевагами такого вибору є модульна архітектура, що сприяє створенню гнучкого та масштабованого серверного компоненту системи. Для маніпулювання даними було обрано NoSQL базу даних MongoDB, що має високу продуктивність для операцій читання та запису, є документо-орієнтованою та зберігає документи у JSON форматі, що полегшує взаємодію із сервером.

NestJS інтегрується з MongoDB через бібліотеку mongoose, а використання мови TypeScript в обох компонентах забезпечує уніфікацію мовного середовища.

#### 1.4 Аналіз та моделювання бізнес-процесів

Для графічного представлення бізнес-процесів використовується BPMN модель, представлена в графічному матеріалі, креслення 3.

Опис процесу реєстрації нового користувача у системі:

- перехід користувача на сторінку реєстрації;
- введення даних користувача у формі;
- перевірка системою наявності користувача;
- якщо логін, введений користувачем унікальний та дані проходять валідацію, користувач зберігається у системі, у іншому випадку користувач має відредагувати дані у формі.

Опис процесу авторизації користувача у системі:

- перехід користувача на сторінку логіну;
- введення логіну та паролю користувача;
- якщо користувача із відповідним паролем знайдено, користувачу надається доступ до даних;
- якщо логін, введений користувачем унікальний та дані проходять валідацію, користувач зберігається у системі, у іншому випадку користувач має ввести інші дані.

Опис процесу управління компетенціями:

- авторизований користувач переходить на сторінку свого профілю;



- користувач отримує інформацію про свої компетенції;
- користувач редагує, додає компетенції, встановлюючи їм у відповідність числове значення від 0 до 5 в залежності рівня володіння або видаляє їх;
- користувач зберігає зміни;
- нові дані валідуються, і у випадку правильності зберігаються системою. Якщо введено невалідні дані - користувач має продовжити редагування.

Опис процесу розподілу завдань:

- менеджер переходить до сторінки завдань проєкту;
- обирає задачі, які він хоче призначити робітникам проєкту;
- запускає розподіл завдань;
- система знаходить найбільш доцільного працівника для кожного завдання;
- менеджер переглядає результати роботи алгоритму розподілу;
- якщо менеджер підтверджує розподіл, система оновлює відповідні записи в базі даних, призначаючи задачам виконавців. Якщо ж розподіл відхилено, результати попереднього розрахунку видаляються.

Висновки до розділу

За допомогою передпроектного аналізу предметної області, огляду відомих технічних та алгоритмічних рішень та формулювання бізнес-вимог було сформовано цілі проєкту та підтверджено актуальність розробки.

Було проаналізовано наявні рішення, розглянуто особливості та недоліки кожного. Аналоги розглядаються з точки зору ефективності розподілу завдань та можливостей управління компетенціями працівників.

Можливі алгоритмічні підходи до реалізації головної функції проєкту є різноманітними і основні критерії, такі як ефективність і передбачуваність

допомогли обрати модифікацію жадібного алгоритму. Вибір технічних рішень визначався актуальністю й сумісністю технологій, враховуючи мету проєкту.

Проведений аналіз дозволив визначити загальні вимоги для системи та сформулювати головні напрямки розробки застосунку, оскільки цей розділ формує основу для конкретизації вимог у наступному розділі.

2 РОЗРОБЛЕННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Варіанти використання програмного забезпечення

Діаграма варіантів використання описує функціонал системи для наявних ролей: працівника, менеджера та адміністратора. Головною функцією розробки є розподіл задач, більше функцій можна побачити на діаграмі варіантів використання на рисунку 2.1.

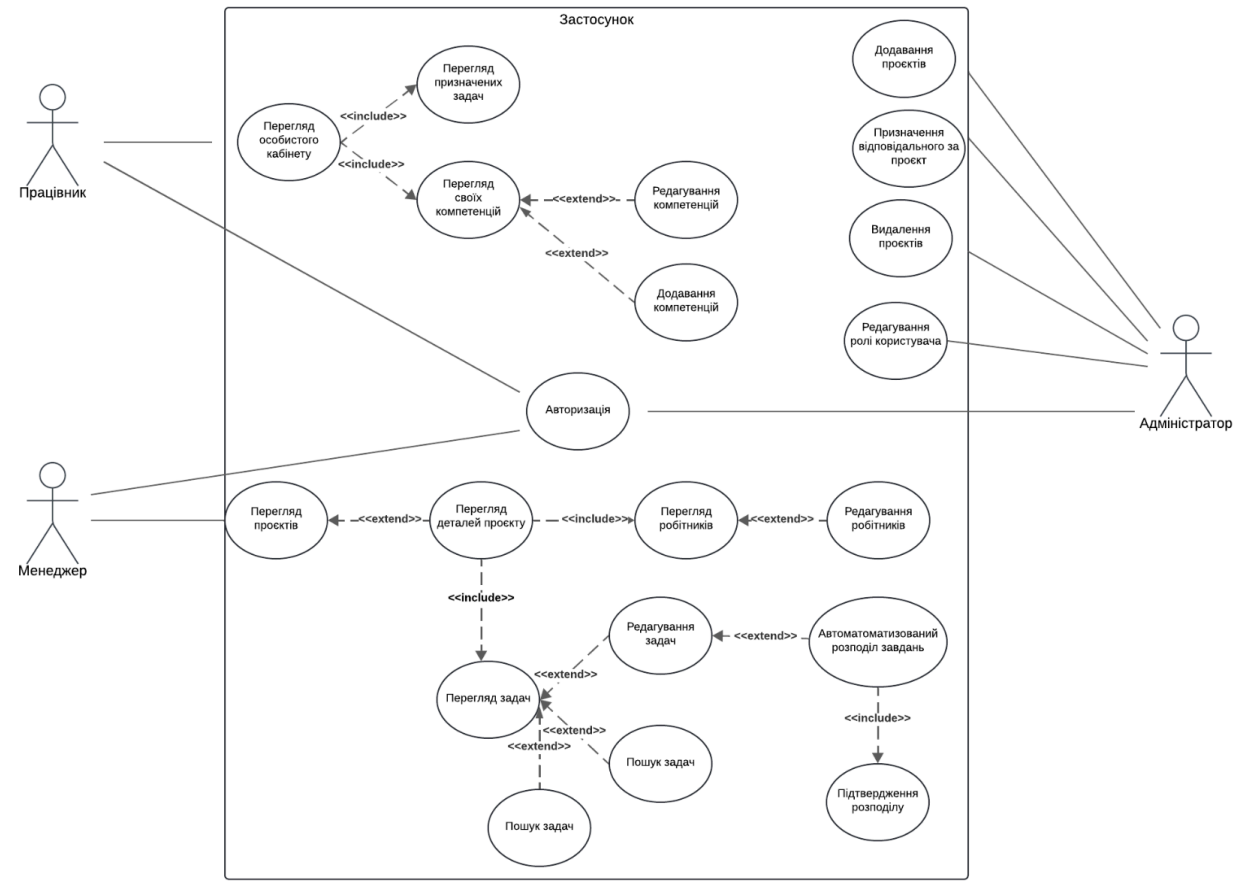


Рисунок 2.1 - Діаграма варіантів використання системи

Деталі варіантів використання описані у таблицях 2.1 - 2.12.

Таблиця 2.1 - Варіант використання UC-01

|                |                                                                                                                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Реєстрація                                                                                                                                                                                                                                                           |
| Use case ID    | UC-01                                                                                                                                                                                                                                                                |
| Goals          | Реєстрація нового користувача                                                                                                                                                                                                                                        |
| Actors         | Незареєстрований користувач                                                                                                                                                                                                                                          |
| Trigger        | Користувач відкриває сторінку реєстрації                                                                                                                                                                                                                             |
| Pre-conditions | -                                                                                                                                                                                                                                                                    |
| Flow of events | <ol style="list-style-type: none"> <li>1. Користувач переходить на сторінку реєстрації</li> <li>2. Дані форми реєстрації - логін, пароль та ім'я користувача - заповнюються користувачем</li> <li>3. Користувач підтверджує реєстрацію натисканням кнопки</li> </ol> |
| Extension      | Якщо логін неунікальний, не відповідає формату електронної пошти або присутні пусті поля, відповідне поле вводу стає червоним і текст відповідної помилки з'являється під ним.                                                                                       |
| Post-condition | Користувача створено, перехід на сторінку авторизації                                                                                                                                                                                                                |

Таблиця 2.2 - Варіант використання UC-02

|                |                                                                    |
|----------------|--------------------------------------------------------------------|
| Use case name  | Авторизація                                                        |
| Use case ID    | UC-02                                                              |
| Goals          | Авторизувати користувача у системі, надати йому доступ до ресурсів |
| Actors         | Неавторизований користувач                                         |
| Trigger        | Користувач переходить на сторінку входу                            |
| Pre-conditions | -                                                                  |

Продовження таблиці 2.2

|                |                                                                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Flow of events | <ol style="list-style-type: none"> <li>1. Користувач відкриває сторінку авторизації</li> <li>2. Користувач заповнює поля “логін” та “пароль”</li> <li>3. Користувач підтверджує авторизацію натисканням кнопки</li> </ol> |
| Extension      | Якщо дані некоректні(пусте поле, неіснуючий логін чи неправильний пароль), то відповідне поле вводу стає червоним і текст відповідної помилки з’являється під ним.                                                        |
| Post-condition | Авторизація користувача, доступ до даних відповідно до ролі                                                                                                                                                               |

Таблиця 2.3 - Варіант використання UC-03

|                |                                                                         |
|----------------|-------------------------------------------------------------------------|
| Use case name  | Перегляд особистого кабінету                                            |
| Use case ID    | UC-03                                                                   |
| Goals          | Надати працівнику доступ до особистих даних та можливість їх редагувати |
| Actors         | Працівник                                                               |
| Trigger        | Працівника авторизовано і автоматично направлено до особистого профілю  |
| Pre-conditions | Працівника авторизовано                                                 |
| Flow of events | 1. Користувач авторизується і переходить на сторінку профілю            |
| Extension      | -                                                                       |
| Post-condition | Відображено задачі, призначені користувачу та його компетенції          |

Таблиця 2.4 - Варіант використання UC-04

|                |                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Редагування компетенцій                                                                                                        |
| Use case ID    | UC-04                                                                                                                          |
| Goals          | Дозволити працівнику змінювати свої компетенції та їх рівень                                                                   |
| Actors         | Працівник                                                                                                                      |
| Trigger        | Працівник обирає компетенцію для редагування                                                                                   |
| Pre-conditions | Працівника авторизовано                                                                                                        |
| Flow of events | 1. Працівник обирає компетенцію для редагування<br>2. Редагує дані про свої навички та їх рівень<br>3. Зберігає внесені зміни. |
| Extension      | -                                                                                                                              |
| Post-condition | Інформацію оновлено і коректно відображено в профілі працівника.                                                               |

Таблиця 2.5 - Варіант використання UC-05

|                |                                                                    |
|----------------|--------------------------------------------------------------------|
| Use case name  | Перегляд проєктів                                                  |
| Use case ID    | UC-05                                                              |
| Goals          | Дозволити менеджеру переглядати призначені йому проєкти            |
| Actors         | Менеджер                                                           |
| Trigger        | Менеджера авторизовано і автоматично направлено до списку проєктів |
| Pre-conditions | Працівника авторизовано                                            |
| Flow of events | 1. Сторінку із проєктами відкрито                                  |
| Extension      | -                                                                  |
| Post-condition | Відображено проєкти, призначені менеджеру                          |

Таблиця 2.6 - Варіант використання UC-06

|                |                                                          |
|----------------|----------------------------------------------------------|
| Use case name  | Перегляд деталей проєкту                                 |
| Use case ID    | UC-06                                                    |
| Goals          | Дозволити менеджеру керувати призначеними йому проєктами |
| Actors         | Менеджер                                                 |
| Trigger        | Менеджер обрав проєкт зі списку                          |
| Pre-conditions | Менеджера авторизовано                                   |
| Flow of events | 1. Менеджер шукає проєкт за назвою.<br>2. Обирає проєкт. |
| Extension      | -                                                        |
| Post-condition | Відображено задачі у проєкті та команду                  |

Таблиця 2.7 - Варіант використання UC-07

|                |                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Додавання працівника в команду                                                                                                                                                       |
| Use case ID    | UC-07                                                                                                                                                                                |
| Goals          | Надати менеджеру можливість додавати працівника                                                                                                                                      |
| Actors         | Менеджер                                                                                                                                                                             |
| Trigger        | Менеджер обрав проєкт зі списку і перейшов до вкладки “Команда”                                                                                                                      |
| Pre-conditions | - Менеджера авторизовано                                                                                                                                                             |
| Flow of events | 1. Менеджер натискає кнопку додавання співробітника.<br>2. З’являється форма з полем вводу електронної пошти.<br>3. Менеджер підтверджує вибір.<br>4. Користувача додано до команди. |
| Extension      | Якщо зареєстрованого користувача з таким логіном немає, менеджера повідомлено про це за допомогою відповідного повідомлення.                                                         |
| Post-condition | Інформацію про команду проєкту оновлено в системі.                                                                                                                                   |

Таблиця 2.8 - Варіант використання UC-08

|                |                                                                                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Редагування задач проєкту                                                                                                                                                                                                                               |
| Use case ID    | UC-08                                                                                                                                                                                                                                                   |
| Goals          | Надати менеджеру можливість редагувати список задач проєкту                                                                                                                                                                                             |
| Actors         | Менеджер                                                                                                                                                                                                                                                |
| Trigger        | Менеджер обрав проєкт зі списку і перейшов до вкладки “Задачі”                                                                                                                                                                                          |
| Pre-conditions | - Менеджера авторизовано                                                                                                                                                                                                                                |
| Flow of events | <ol style="list-style-type: none"> <li>1. Менеджер обирає задачу для редагування.</li> <li>2. У відповідній формі менеджер змінює інформацію про задачу.</li> <li>3. Менеджер підтверджує зміни.</li> <li>4. Інформацію про задачу оновлено.</li> </ol> |
| Extension      | -                                                                                                                                                                                                                                                       |
| Post-condition | Інформація про задачу оновлена у системі.                                                                                                                                                                                                               |

Таблиця 2.9 - Варіант використання UC-09

|                |                                                                                                                                                                                                                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Розподіл задач                                                                                                                                                                                                                                                                                                         |
| Use case ID    | UC-09                                                                                                                                                                                                                                                                                                                  |
| Goals          | Надати менеджеру можливість розподіляти задачі проєкту                                                                                                                                                                                                                                                                 |
| Actors         | Менеджер                                                                                                                                                                                                                                                                                                               |
| Trigger        | Менеджер обирає непризначені задачі для розподілу                                                                                                                                                                                                                                                                      |
| Pre-conditions | 1. Обрано хоча б одну задачу серед непризначених                                                                                                                                                                                                                                                                       |
| Flow of events | <ol style="list-style-type: none"> <li>1. Менеджер переходить на вкладку Задачі в деталях проєкту.</li> <li>2. Обирає задачі для розподілу.</li> <li>4. Отримує результати розподілу у вигляді таблиці.</li> <li>5. Підтверджує розподіл.</li> <li>6. Задачі призначено працівникам, які підібрав алгоритм.</li> </ol> |
| Extension      | Якщо жодної задачі не вибрано, кнопка розподілу неактивна                                                                                                                                                                                                                                                              |



Таблиця 2.10 - Варіант використання UC-10

|                |                                                                                                                                                                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Додавання проєкту                                                                                                                                                                                                                                                                                                    |
| Use case ID    | UC-10                                                                                                                                                                                                                                                                                                                |
| Goals          | Надати адміністратору можливість додавати новий проєкт                                                                                                                                                                                                                                                               |
| Actors         | Адміністратор                                                                                                                                                                                                                                                                                                        |
| Trigger        | Потрібно додати новий проєкт у систему                                                                                                                                                                                                                                                                               |
| Pre-conditions | Адміністратора авторизовано                                                                                                                                                                                                                                                                                          |
| Flow of events | <ol style="list-style-type: none"> <li>1. Адміністратор переходить на вкладку Проєкти.</li> <li>2. Натискає кнопку додавання нового проєкту.</li> <li>3. Заповнює необхідні поля.</li> <li>4. Підтверджує збереження проєкту.</li> <li>5. Проєкт успішно збережено, він відображається у списку проєктів.</li> </ol> |
| Extension      | -                                                                                                                                                                                                                                                                                                                    |

Таблиця 2.11 - Варіант використання UC-11

|                |                                                                                                                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Призначення проєкту                                                                                                                                                                                                                                                |
| Use case ID    | UC-11                                                                                                                                                                                                                                                              |
| Goals          | Надати адміністратору можливість редагувати менеджера проєкту                                                                                                                                                                                                      |
| Actors         | Адміністратор                                                                                                                                                                                                                                                      |
| Trigger        | Потрібно змінити менеджера проєкту                                                                                                                                                                                                                                 |
| Pre-conditions | Адміністратора авторизовано                                                                                                                                                                                                                                        |
| Flow of events | <ol style="list-style-type: none"> <li>1. Адміністратор переходить на вкладку Проєкти.</li> <li>2. Обирає нового менеджера проєкту з усіх наявних.</li> <li>3. Натискає кнопку підтвердження оновлення.</li> <li>4. Менеджера проєкту оновлено успішно.</li> </ol> |
| Extension      | -                                                                                                                                                                                                                                                                  |

Таблиця 2.12 - Варіант використання UC-12

|                |                                                                                                                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use case name  | Редагування ролі користувача                                                                                                                                                                                                                                       |
| Use case ID    | UC-12                                                                                                                                                                                                                                                              |
| Goals          | Надати адміністратору можливість редагувати роль користувача                                                                                                                                                                                                       |
| Actors         | Адміністратор                                                                                                                                                                                                                                                      |
| Trigger        | Потрібно змінити роль користувача                                                                                                                                                                                                                                  |
| Pre-conditions | Адміністратора авторизовано                                                                                                                                                                                                                                        |
| Flow of events | <ol style="list-style-type: none"> <li>1. Адміністратор переходить на вкладку Користувачі.</li> <li>2. Обирає нову роль користувача з усіх наявних.</li> <li>3. Натискає кнопку підтвердження оновлення.</li> <li>4. Роль користувача оновлено успішно.</li> </ol> |
| Extension      | -                                                                                                                                                                                                                                                                  |

## 2.2 Розроблення функціональних вимог

У таблиці 2.13 наведено загальну модель вимог, а у таблицях 2.14 - 2.33 наведений опис функціональних вимог до програмного забезпечення. Матрицю трасування можна побачити у таблиці 2.34.

Таблиця 2.13 - Загальна модель вимог

| № | Назва                            | ID вимоги | Пріоритет | Ризик    |
|---|----------------------------------|-----------|-----------|----------|
| 1 | Реєстрація користувача           | FR-01     | Високий   | Високий  |
| 2 | Авторизація користувача          | FR-02     | Високий   | Високий  |
| 3 | Вихід з акаунту                  | FR-03     | Середній  | Низький  |
| 4 | Перегляд компетенцій користувача | FR-04     | Високий   | Низький  |
| 5 | Перегляд задач користувача       | FR-05     | Високий   | Низький  |
| 6 | Редагування компетенцій          | FR-06     | Високий   | Низький  |
| 7 | Додавання компетенцій            | FR-07     | Середній  | Середній |

Продовження таблиці 2.13

| №  | Назва                              | ID вимоги | Пріоритет | Ризик    |
|----|------------------------------------|-----------|-----------|----------|
| 8  | Перегляд проєктів                  | FR-08     | Високий   | Середній |
| 9  | Перегляд деталей проєкту           | FR-09     | Високий   | Середній |
| 10 | Перегляд задач                     | FR-10     | Високий   | Середній |
| 11 | Редагування задач                  | FR-11     | Середній  | Високий  |
| 12 | Перегляд команди проєкту           | FR-12     | Низький   | Низький  |
| 13 | Додавання працівника до проєкту    | FR-13     | Середній  | Середній |
| 14 | Розподіл завдань                   | FR-14     | Високий   | Високий  |
| 15 | Відображення результатів розподілу | FR-15     | Середній  | Високий  |
| 16 | Коригування результатів розподілу  | FR-16     | Середній  | Високий  |
| 17 | Підтвердження розподілу            | FR-17     | Середній  | Високий  |
| 18 | Додавання проєкту                  | FR-18     | Середній  | Низький  |
| 19 | Призначення менеджера              | FR-19     | Середній  | Середній |
| 20 | Редагування ролі користувача       | FR-20     | Середній  | Високий  |

Таблиця 2.14 - Функціональна вимога FR-01

| ID вимоги | Реєстрація користувача                                                            |
|-----------|-----------------------------------------------------------------------------------|
| FR-01     | Неавторизований користувач може зареєструватись в системі, надавши коректні дані. |

Таблиця 2.15 - Функціональна вимога FR-02

|           |                                                                                           |
|-----------|-------------------------------------------------------------------------------------------|
| ID вимоги | Авторизація користувача                                                                   |
| FR-02     | Неавторизований користувач може отримати доступ до даних, використовуючи логін та пароль. |

Таблиця 2.16 - Функціональна вимога FR-03

|           |                                                                   |
|-----------|-------------------------------------------------------------------|
| ID вимоги | Вихід з акаунту                                                   |
| FR-03     | Авторизований користувач може вийти із системи, завершивши сесію. |

Таблиця 2.17 - Функціональна вимога FR-04

|           |                                                                  |
|-----------|------------------------------------------------------------------|
| ID вимоги | Перегляд компетенцій користувача                                 |
| FR-04     | Користувач може переглядати свої компетенції у власному профілі. |

Таблиця 2.18 - Функціональна вимога FR-05

|           |                                                                        |
|-----------|------------------------------------------------------------------------|
| ID вимоги | Перегляд задач користувача                                             |
| FR-05     | Користувач може переглядати призначені йому задачі у власному профілі. |

Таблиця 2.19 - Функціональна вимога FR-06

|           |                                                                         |
|-----------|-------------------------------------------------------------------------|
| ID вимоги | Редагування компетенцій                                                 |
| FR-06     | Користувач може редагувати свої компетенції, змінюючи рівень володіння. |

Таблиця 2.20 - Функціональна вимога FR-07

|           |                                                               |
|-----------|---------------------------------------------------------------|
| ID вимоги | Додавання компетенцій                                         |
| FR-07     | Користувач може додавати компетенції за допомогою інтерфейсу. |

Таблиця 2.21 - Функціональна вимога FR-08

|           |                                                                       |
|-----------|-----------------------------------------------------------------------|
| ID вимоги | Перегляд проєктів                                                     |
| FR-08     | Менеджер може переглядати список проєктів, за які він відповідальний. |

Таблиця 2.22 - Функціональна вимога FR-09

|           |                                                                       |
|-----------|-----------------------------------------------------------------------|
| ID вимоги | Перегляд деталей проєкту                                              |
| FR-09     | Менеджер може переглядати деталі проєктів, за які він відповідальний. |

Таблиця 2.23 - Функціональна вимога FR-10

|           |                                            |
|-----------|--------------------------------------------|
| ID вимоги | Перегляд задач                             |
| FR-10     | Менеджер може переглядати задачі проєктів. |

Таблиця 2.24 - Функціональна вимога FR-11

|           |                                                                      |
|-----------|----------------------------------------------------------------------|
| ID вимоги | Редагування задач                                                    |
| FR-11     | Менеджер може редагувати задачі проєктів, за які він відповідальний. |

Таблиця 2.25 - Функціональна вимога FR-12

|           |                                                                       |
|-----------|-----------------------------------------------------------------------|
| ID вимоги | Перегляд команди проєкту                                              |
| FR-12     | Менеджер може переглядати працівників, що входять до команди проєкту. |

Таблиця 2.26 - Функціональна вимога FR-13

|           |                                                                        |
|-----------|------------------------------------------------------------------------|
| ID вимоги | Додавання працівника до проєкту                                        |
| FR-13     | Менеджер може додати працівника до команди проєкту, увівши його пошту. |

Таблиця 2.27 - Функціональна вимога FR-14

|           |                                                                                          |
|-----------|------------------------------------------------------------------------------------------|
| ID вимоги | Розподіл завдань                                                                         |
| FR-14     | Менеджер може ініціювати процес автоматичного розподілу завдань за допомогою інтерфейсу. |

Таблиця 2.28 - Функціональна вимога FR-15

|           |                                                                             |
|-----------|-----------------------------------------------------------------------------|
| ID вимоги | Відображення результатів розподілу                                          |
| FR-15     | Менеджер повинен побачити результати розподілу після спрацювання алгоритму. |

Таблиця 2.29 - Функціональна вимога FR-16

|           |                                                                                       |
|-----------|---------------------------------------------------------------------------------------|
| ID вимоги | Коригування результатів розподілу                                                     |
| FR-16     | Після автоматичного розподілу задач менеджер може переглядати результати його роботи. |

Таблиця 2.30 - Функціональна вимога FR-17

|           |                                                                                        |
|-----------|----------------------------------------------------------------------------------------|
| ID вимоги | Підтвердження розподілу                                                                |
| FR-17     | Менеджер може обрати задачі, із призначенням яких він згоден і підтвердити збереження. |

Таблиця 2.31 - Функціональна вимога FR-17

|           |                                                                          |
|-----------|--------------------------------------------------------------------------|
| ID вимоги | Додавання проєкту                                                        |
| FR-17     | Адміністратор може додати проєкт, увівши назву та призначивши менеджера. |

Таблиця 2.32 - Функціональна вимога FR-17

|           |                                                 |
|-----------|-------------------------------------------------|
| ID вимоги | Призначення менеджера                           |
| FR-17     | Адміністратор може змінювати менеджера проєкту. |

Таблиця 2.33 - Функціональна вимога FR-17

|           |                                                |
|-----------|------------------------------------------------|
| ID вимоги | Редагування ролі користувача                   |
| FR-17     | Адміністратор може змінювати роль користувача. |

Таблиця 2.34 - Матриця трасування вимог

|       | FR-01 | FR-02 | FR-03 | FR-04 | FR-05 | FR-06 | FR-07 | FR-08 | FR-09 | FR-10 | FR-11 | FR-12 | FR-13 | FR-14 | FR-15 | FR-16 | FR-17 | FR-18 | FR-19 | FR-20 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| UC-1  | +     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| UC-2  |       | +     | +     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| UC-3  |       |       |       | +     | +     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| UC-4  |       |       |       |       |       | +     | +     |       |       |       |       |       |       |       |       |       |       |       |       |       |
| UC-5  |       |       |       |       |       |       |       | +     | +     | +     |       | +     |       |       |       |       |       |       |       |       |
| UC-6  |       |       |       |       |       |       |       |       | +     | +     |       | +     |       |       |       |       |       |       |       |       |
| UC-7  |       |       |       |       |       |       |       |       |       |       |       |       | +     |       |       |       |       |       |       |       |
| UC-8  |       |       |       |       |       |       |       |       |       |       | +     |       |       |       |       |       |       |       |       |       |
| UC-9  |       |       |       |       |       |       |       |       |       |       |       |       |       | +     | +     | +     | +     |       |       |       |
| UC-10 |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | +     |       |       |
| UC-11 |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | +     |       |
| UC-12 |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | +     |

### 2.3 Розроблення нефункціональних вимог

Нефункціональні вимоги описують необхідні характеристики системи, що визначають якість її роботи. Нижче наведено перелік нефункціональних вимог до застосунку:

#### 1. Продуктивність

- час відповідей на запити користувача має бути не більшим за 3 секунди.

#### 2. Безпека

- розмежування доступу за ролями: менеджер, працівник, адміністратор;
- авторизація для доступу до ресурсів;
- хешування паролів;
- використання httpOnly cookies для зберігання JWT токенів.

#### 3. Надійність

- система має забезпечувати стабільну роботу і коректність результатів обчислень.

#### 4. Зручність використання

- максимальна глибина в ієрархії навігації - 2 рівні;



- інтуїтивно зрозуміле управління результатами розподілу завдань.

## 2.4 Аналіз економічних показників програмного забезпечення

Для аналізу економічних показників використано метод Use Case Points.

Метод Use Case Points (UCP) - це спосіб оцінки трудомісткості розробки програмного забезпечення, що застосовується для планування ресурсів у IT-менеджменті.

Кроки методу Use Case Points [12]:

1. Класифікувати чинних персонажів (actors) на простих, середніх і складних, потім вирахувати Unadjusted Actor Weight (UAW), перемноживши кількість actors в кожній категорії на ваговий коефіцієнт і додати результати.
2. Класифікувати варіанти використання (use cases) на прості, середні і складні в залежності від кількості транзакцій. Вирахувати Unadjusted Use-Case Weight (UUCW) аналогічно до UAW.
3. Додати UUCW і UAW для отримання Unadjusted Use-Case Points (UUCP).
4. Оцінити систему за технічними факторами за допомогою ваг, наведених у таблиці 2.39.
5. Оцінити систему за факторами середовища за допомогою ваг, наведених у таблиці 2.41.
6. Розрахувати UCP за формулою та визначити трудовитрати проєкту у людино-годинах.

Кількість UCP в проєкті базується на наступному:

- кількість та складність випадків використання в системі;
- кількість та складність дійових осіб у системі;
- різні не функціональні вимоги (такі як портативність, продуктивність, ремонтпридатність), які не записуються як випадки використання;

- середовище, в якому буде розвиватися проєкт (наприклад, мова, мотивація команди тощо).

Далі описано кроки обчислення [13] за методикою Use Case Points для проєкту.

1. Обчислення невідкоригованих UCP

Для цього опишемо Use Cases та транзакції кожного випадку використання:

1. Реєстрація користувача:

- введення даних користувача;
- валідація;
- створення користувача в БД.

2. Авторизація:

- введення логіну та паролю;
- перевірка облікових даних;
- створення JWT токена.

3. Перегляд інформації користувача:

- запит на отримання задач користувача;
- відображення деталей задачі.

4. Редагування компетенцій:

- запит на отримання компетенцій користувача;
- внесення змін;
- валідація змін;
- збереження оновлених компетенцій.

5. Перегляд проєктів:

- запит на отримання проєктів;
- пошук проєкту за назвою;
- відображення деталей проєкту.

6. Перегляд деталей проєкту:

- запит на отримання задач проєктів;
- пошук серед непризначених задач;

- відображення деталей задачі.
- 7. Додавання працівника в команду:
  - введення логіну працівника;
  - підтвердження вибору працівника;
  - збереження призначення працівника проєкту.
- 8. Редагування задач проєкту:
  - запит на отримання задач проєкту;
  - внесення змін;
  - підтвердження змін;
  - збереження оновленої задачі.
- 9. Автоматизований розподіл завдань:
  - запит на отримання непризначених задач;
  - запит на отримання працівників команди;
  - виконання алгоритму розподілу;
  - відображення результатів розподілу;
  - підтвердження розподілу;
  - збереження змін.

За допомогою таблиці 2.35 визначення складності випадку використання формалізуємо результати у таблиці 2.36.

Таблиця 2.35 - Таблиця складності випадків використання

| Складність випадку використання | Кількість транзакцій | Вага |
|---------------------------------|----------------------|------|
| Проста                          | < 3                  | 5    |
| Середня                         | від 4 до 7           | 10   |
| Складна                         | > 7                  | 15   |

Таблиця 2.36 - Таблиця оцінки складності випадків використання

| ID    | Use Case                         | Кількість транзакцій | Складність | Вага |
|-------|----------------------------------|----------------------|------------|------|
| UC-01 | Реєстрація користувача           | 3                    | Проста     | 5    |
| UC-02 | Авторизація                      | 3                    | Проста     | 5    |
| UC-03 | Перегляд інформації користувача  | 2                    | Проста     | 5    |
| UC-04 | Редагування компетенцій          | 4                    | Середня    | 10   |
| UC-05 | Перегляд проєктів                | 3                    | Проста     | 5    |
| UC-06 | Перегляд деталей проєкту         | 3                    | Проста     | 5    |
| UC-07 | Додавання працівника в команду   | 3                    | Проста     | 5    |
| UC-08 | Редагування задач проєкту        | 4                    | Середня    | 10   |
| UC-09 | Автоматизований розподіл завдань | 6                    | Середня    | 10   |

Обчислимо загальну вагу випадків використання(UUCW) за формулою (2.1):

$$UUCW = 6 \times 5 + 3 \times 10 = 60$$

$$UUCW = \sum(\text{кількість випадків використання} \times \text{ваговий коефіцієнт}) \quad (2.1)$$

2. Проаналізуємо акторів системи:

Таблиця 2.37 - Таблиця складності акторів системи

| Тип Актора | Приклад                         | Вага |
|------------|---------------------------------|------|
| Простий    | Стороння система через API      | 1    |
| Середній   | Стороння система через протокол | 2    |
| Складний   | Користувач через GUI            | 3    |

Таблиця 2.38 - Таблиця оцінки складності акторів системи

| Тип актора | Опис                             | Вага | Кількість             | Результат |
|------------|----------------------------------|------|-----------------------|-----------|
| Простий    | Зовнішня система через API       | 1    | 0                     | 0         |
| Середній   | Зовнішня система через протокол  | 2    | 0                     | 0         |
| Складний   | Людина через графічний інтерфейс | 3    | 2(робітник, менеджер) | 6         |

Обчислимо загальну вагу акторів за формулою (2.2):

$$UAW = 3 \times 2 = 6$$

$$UAW = \sum \text{кількість акторів в категорії} \times \text{ваговий коефіцієнт} \quad (2.2)$$

Розрахуємо невідредаговані Use Case Points(UUCP) за формулою (2.3):

$$UUCP = 6 + 60 = 66$$

$$UUCP = UAW + UUCW \quad (2.3)$$

3. За допомогою таблиці 2.39 оцінимо систему за технічними факторами у таблиці 2.40.

Таблиця 2.39 - Таблиця вагів технічних факторів

| Фактор | Опис                                   | Вага |
|--------|----------------------------------------|------|
| T1     | Розподілена система                    | 2.0  |
| T2     | Продуктивність                         | 1.0  |
| T3     | Ефективність для кінцевого користувача | 1.0  |
| T4     | Складність внутрішньої обробки         | 1.0  |

Продовження таблиці 2.39

| Фактор | Опис                       | Вага |
|--------|----------------------------|------|
| T5     | Повторне використання коду | 1.0  |
| T6     | Простота установки         | 0.5  |
| T7     | Простота використання      | 0.5  |
| T8     | Портативність              | 2.0  |
| T9     | Простота зміни             | 1.0  |
| T10    | Паралельність              | 1.0  |
| T11    | Функції безпеки            | 1.0  |
| T12    | Доступ для сторонніх       | 1.0  |
| T13    | Спеціальне навчання        | 1.0  |

Таблиця 2.40 - Таблиця оцінки системи за технічними факторами

| ID | Фактор                                 | Вага | Вплив(0-5) | Результат |
|----|----------------------------------------|------|------------|-----------|
| T1 | Розподілена система                    | 2.0  | 1          | 2.0       |
| T2 | Продуктивність                         | 1.0  | 3          | 3.0       |
| T3 | Ефективність для кінцевого користувача | 1.0  | 4          | 4.0       |
| T4 | Складність внутрішньої обробки         | 1.0  | 4          | 4.0       |
| T5 | Повторне використання коду             | 1.0  | 3          | 3.0       |
| T6 | Простота установки                     | 0.5  | 2          | 1.0       |

Продовження таблиці 2.40

| ID  | Фактор                | Вага | Вплив(0-5) | Результат |
|-----|-----------------------|------|------------|-----------|
| T7  | Простота використання | 0.5  | 4          | 2.0       |
| T8  | Портативність         | 2.0  | 1          | 2.0       |
| T9  | Простота зміни        | 1.0  | 3          | 3.0       |
| T10 | Паралельність         | 1.0  | 1          | 1.0       |
| T11 | Функції безпеки       | 1.0  | 3          | 3.0       |
| T12 | Доступ для сторонніх  | 1.0  | 1          | 1.0       |
| T13 | Спеціальне навчання   | 1.0  | 1          | 1.0       |

Обчислимо суму технічних факторів за формулою (2.4) та фактор технічної складності за формулою (2.5):

$$TFactor = 30$$

$$TCF = 0.6 + (0.01 \times 30) = 0.9$$

$$TFactor = \sum(\text{вага фактора} \times \text{значення}) \quad (2.4)$$

$$TCF = 0,6 + (0,01 \times TFactor) \quad (2.5)$$

#### 4. Проведемо оцінку за факторами середовища

Таблиця 2.41 - Таблиця вагів факторів середовища

| Фактор | Опис                                    | Вага |
|--------|-----------------------------------------|------|
| E1     | Знайомство з процесом розробки          | 1.5  |
| E2     | Досвід в додатку                        | 0.5  |
| E3     | Досвід в об'єктно-орієнтованому підході | 1.0  |

Продовження таблиці 2.41

| Фактор | Опис                             | Вага |
|--------|----------------------------------|------|
| E4     | Здібності аналітика              | 0.5  |
| E5     | Мотивація                        | 1.0  |
| E6     | Стабільність вимог               | 2.0  |
| E7     | Робітники з частковою зайнятістю | -1.0 |
| E8     | Складність мови програмування    | -1.0 |

Таблиця 2.42 - Таблиця оцінки системи за факторами середовища

| ID | Фактор                                  | Вага | Вплив(0-5) | Результат |
|----|-----------------------------------------|------|------------|-----------|
| E1 | Знайомство з процесом розробки          | 1.5  | 3          | 4.5       |
| E2 | Досвід в додатку                        | 0.5  | 3          | 1.5       |
| E3 | Досвід в об'єктно-орієнтованому підході | 1.0  | 4          | 4.0       |
| E4 | Здібності аналітика                     | 0.5  | 4          | 2.0       |
| E5 | Мотивація                               | 1.0  | 5          | 5.0       |
| E6 | Стабільність вимог                      | 2.0  | 3          | 6.0       |
| E7 | Робітники з частковою зайнятістю        | -1.0 | 0          | 0.0       |
| E8 | Складність мови програмування           | -1.0 | 3          | -3.0      |

Обчислимо суму факторів середовища за формулою (2.6) та фактор складності середовища за формулою (2.7):

$$E_{\text{Factor}} = 20.0$$



$$EFactor = \sum(\text{вага фактора} \times \text{значення}) \quad (2.6)$$

$$EF = 1,4 + (-0,03 \times EFactor) \quad (2.7)$$

### 5. Розрахунок UCP

Наприкінці розрахуємо відкориговані бали варіантів використання Use Case Points(UCP) за формулою (2.8):

$$UCP = 66 \times 0,9 \times 0,8 = 47,52$$

$$UCP = UUCP \times TCF \times EF \quad (2.8)$$

6. Маючи результат у Use Case Points можна оцінити трудовитрати в людино-годинах, використовуючи стандартне значення 20 годин на 1 UCP:

$$PF = 20 \text{ годин/UCP}$$

$$\text{Трудовитрати} = 47,52 \times 20 = 950,4 \text{ людино-години}$$

$$\text{Трудовитрати} = UCP \times PF \quad (2.9)$$

### Висновок:

Для аналізу економічних показників програмного забезпечення було застосовано метод Use Case Points, який дозволяє оцінити трудомісткість розробки на основі вимог до системи. У результаті розрахунків загальна кількість UCP склала 47.52.

Отриманий показник свідчить про середню складність розробки системи. З урахуванням стандартного коефіцієнта продуктивності, трудомісткість становить 950.4 людино-години, що при команді з 3 розробників дозволяє реалізувати проєкт за 8 тижнів. Трудомісткість проєкту допомагає ще на етапі планування визначити приблизну вартість розробки продукту.

Отримані результати дають змогу зробити попередній прогноз щодо обсягу необхідних ресурсів, спланувати бюджет та визначити економічну доцільність реалізації.

Вебзастосунок спрямований на оптимізацію роботи в командах, тому враховуючи високий попит на системи планування й отримані показники, можна зробити висновок, що реалізація системи є доцільною і потенційно прибутковою, адже економічний ефект від впровадження системи полягає в скороченні часу на розподіл завдань та підвищенні ефективності використання ресурсів. В порівнянні з готовими рішеннями, розробка власного застосунку для компанії хоч і має значно вищу початкову вартість, але вигідніша в довгостроковій перспективі через відсутність регулярних платежів.

Метод UCP забезпечує більш об'єктивну оцінку в порівнянні з суб'єктивними методами, що підвищує точність прогнозування. Таким чином, застосування даного методу дає цінну для планування ресурсів інформацію, доводячи доцільність реалізації проєкту.

## 2.5 Постановка завдання на розробку програмного забезпечення

Метою даної дипломної роботи є спрощення процесу управління людськими ресурсами за рахунок розробки веб-застосунку для автоматизованого розподілу завдань на основі матриці компетенцій співробітників. Програмне забезпечення призначене спростити процес керування проєктами, враховуючи технічні навички співробітників та їх рівень, що забезпечить ефективне використання людських ресурсів та підвищить якість виконання проєктів.

Основними функціональними задачами розробки є розробка функціоналу управління основними сутностями системи: задачами та робітниками, формування матриці компетенцій для кожного співробітника із зазначенням рівня володіння кожною технологією, реалізація та впровадження алгоритму автоматизованого розподілу завдань із можливістю редагування результатів алгоритму та ручного призначення завдань.

В результаті виконання дипломної роботи очікується отримати вебзастосунок з інтуїтивно зрозумілим інтерфейсом, який дозволить оптимізувати робоче

навантаження в командах та підвищити загальну ефективність управління проєктами в ІТ-компаніях.

### Висновки до розділу

У даному розділі було проведено формалізацію вимог до вебзастосунка, визначено варіанти використання системи, які описують сценарії взаємодії користувачів із застосунком. Конкретизація очікуваної функціональності системи з точки зору кінцевих користувачів дозволила також сформулювати функціональні вимоги, які описують необхідні можливості системи, такі як авторизація, реєстрація, автоматизований розподіл завдань та інші вимоги щодо управління даними користувачами із різними ролями. Окрім функціональних також було описано нефункціональні вимоги, які дозволяють контролювати надійність, продуктивність та безпеку застосунку. Матриця трасування вимог описує зв'язки між функціональними вимогами та варіантами використання, що дозволяє виявити непокриті користувацькі сценарії та дозволяє уникнути надлишкових вимог.

Опис вимог застосунку є необхідним етапом перед проєктуванням та розробкою програмного забезпечення для структуризації і чіткого розуміння задач, які сформульовано в останньому підрозділі.

### 3 КОНСТРУЮВАННЯ ТА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Архітектура програмного забезпечення

У першому розділі було обрано клієнт-серверну архітектуру з монолітним бекендом. Такий підхід забезпечує чітке розділення відповідальності між клієнтською частиною, що відповідає за інтерфейс користувача, та серверною, яка реалізує бізнес-логіку та доступ до даних. Для комунікації між компонентами використовується Rest API.

Загальна структура системи представлена на рисунку 3.1, де відображено компоненти системи та їх зв'язки.

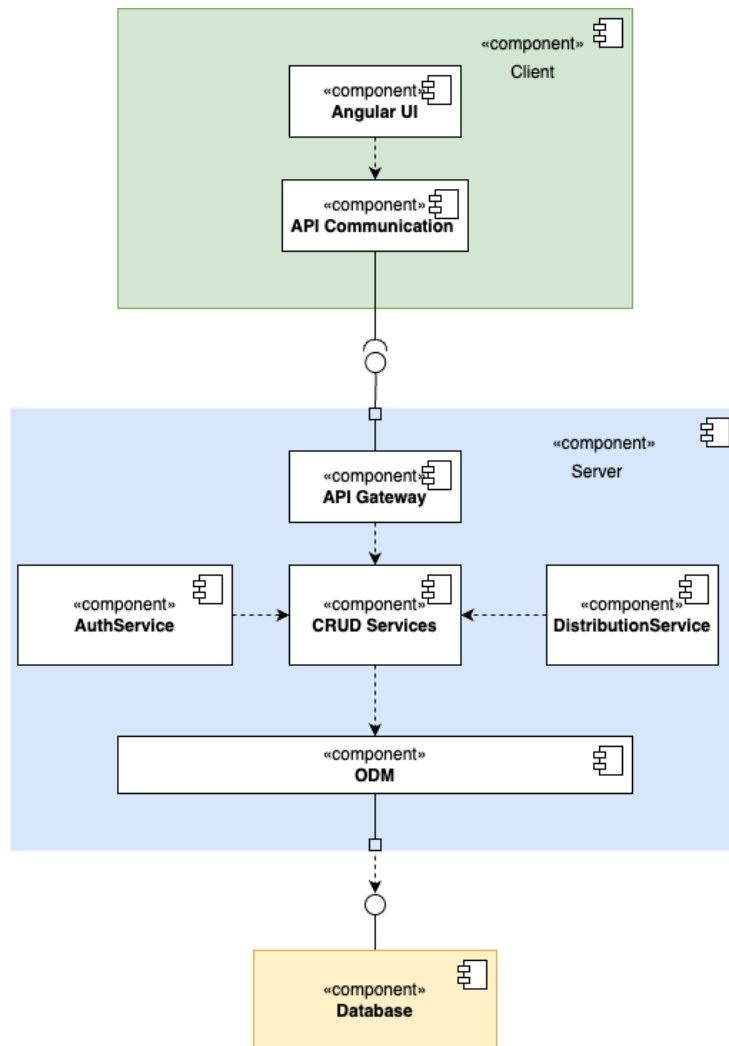


Рисунок 3.1 - Компонентна модель архітектури застосунку

На діаграмі компонентів можна побачити блок Angular Frontend, де компоненти інтерфейсу користувача комунікують із сервісами, призначеними

для взаємодії із сервером. Вони роблять запити на бекенд, де відправлені дані валідуються за допомогою Data Transfer Object (DTO), патерну, за якого обмеження даних попередньо визначаються для передачі між компонентами системи і структуровано валідуються при комунікації між клієнтом і сервером. Валідовані дані надходять до сервісів на сервері, які відповідають за бізнес-логіку та за комунікацію з базою даних за допомогою Object Document Mapping (ODM) для документоорієнтованих баз даних. Shared Data Models є бібліотекою зберігання моделей даних, що використовуються на клієнті і на сервері, що забезпечує узгодженість даних та типізацію об'єктів на обох сторонах. Між усіма компонентами системи дані обробляються у форматі JSON.

### 3.2 Архітектурні рішення та обґрунтування вибору засобів розробки

Для реалізації застосунку було обрано такий технологічний стек:

1. Мова TypeScript - розширення JavaScript, перевагою якої є виявлення помилок на етапі компіляції завдяки статичній типізації.
2. Фреймворк [Nest.js](#) для бекенду є прогресивним [Node.js](#) фреймворком, який має модульну архітектуру, що дозволяє структурувати код у логічні блоки, які імпортуються у основний блок AppModule.
3. Фреймворк Angular для фронтенду має компонентну структуру UI, гнучку маршрутизацію між представленнями та використовує реактивний підхід до обробки даних.
4. Бібліотека mongoose для комунікації серверу із базою даних - це реалізація Object-Document Mapping (ODM). Вона дозволяє інкапсулювати логіку доступу до бази даних за допомогою створення моделей сутності і схем, що відображають структуру сутностей та допомагають валідувати дані при збереженні та оновленні.
5. Систему управління базами даних MongoDB було обрано через гнучку схему даних, що підходить для зберігання компетенцій співробітників,

де структура може бути модифікована у майбутньому. MongoDB є документоорієнтованою нереляційною (NoSQL) базою даних.

6. JSON-подібний формат зберігання BSON (Binary JSON) узгоджується з TypeScript об'єктами, що спрощує роботу з даними. Також MongoDB забезпечує швидкий доступ до даних, що дозволяє виконати нефункціональні вимоги застосунку і легко масштабується горизонтально.

Середовищем розробки обрано WebStorm через, яке призначене спеціально для веб-розробки із технологіями JavaScript/TypeScript, HTML та CSS, що підходить до стеку основних технологій фреймворків Angular та NestJS.

7. Система автодоповнення розрахована на специфіку згаданих технологій, що підвищує якість та швидкість розробки. Для Angular WebStorm пропонує розширений інструментарій для роботи з компонентами, шаблонами та маршрутизацією.

Виконання нефункціональних вимог

Для забезпечення часу відповіді не запити не більше 3 секунд реалізовано заходи на різних рівнях, специфічні для обраних технологій. На рівні доступу до даних використовується проекція даних для зменшення обсягу переданої інформації. У Mongoose застосовано вбудовані інструменти фільтрації і вибору полів для оптимізації запитів за допомогою функції `select`, де вказуються необхідні поля. Також використовуються оператори агрегації для обробки даних на рівні бази даних.

На серверному рівні для пришвидшення відповідей використовується асинхронна обробка запитів та кешування у контролерах за допомогою анотацій `@CacheTTL(cache time to live)`, де в параметрі встановлюється час зберігання у секундах та `@UseInterceptors(CacheInterceptor)`, що додає перехоплювач, який керує кешуванням.

Оптимізація на рівні фронтенду реалізована за допомогою лінивого

завантаження модулів у модулі маршрутизації та використання OnPush стратегії, що зменшує кількість перевірок змін для рендерингу сторінки.

Для безпечного розмежування прав доступ розділений за ролями (менеджер, працівник, адміністратор), усі ресурси системи доступні лише після авторизації. Паролі зберігаються в хешованому вигляді, для авторизації використовується JWT (JSON Web Token), що не вимагає зберігання сесій на сервері, використовує алгоритми шифрування та зберігається у безпечному сховищі HttpOnly cookies.

Надійність роботи застосунку реалізовано через обробку можливих помилок на рівні сервера та клієнта, валідацію даних на фронтенді за допомогою форм та на бекенді за допомогою DTO, кожне поле якого позначено анотаціями з бібліотеки class-validator. Декоратори class-validator роблять код само документованим і захищають застосунок від некоректних або шкідливих даних. Для надійності зберігання даних передбачено резервне копіювання кожного дня опівночі за допомогою бібліотеки NestJS Schedule.

Для зручності використання розроблено структуру навігації з максимальною глибиною 2 рівні та інтуїтивно зрозумілий інтерфейс управління результатами розподілу завдань.

Таблиця 3.1 - Опис утиліт

| № | Назва утиліти | Опис                                                             |
|---|---------------|------------------------------------------------------------------|
| 1 | Postman       | Платформа для розробки та тестування API                         |
| 2 | mongoose      | Object Data Modeling (ODM) бібліотека для MongoDB                |
| 3 | PrimeNG       | Бібліотека UI-компонентів для Angular                            |
| 4 | rxjs          | Бібліотека реактивного програмування з використанням Observables |
| 5 | Schedule      | Модуль для планування задач у NestJS                             |
| 6 | bcrypt        | Бібліотека хешування паролів                                     |

Продовження таблиці 3.1

| № | Назва утиліти   | Опис                                                                                          |
|---|-----------------|-----------------------------------------------------------------------------------------------|
| 7 | class-validator | Бібліотека для валідації об'єктів TypeScript, що використовує декоратори                      |
| 8 | uuid            | Бібліотека генерації унікальних ідентифікаторів, які забезпечують нульову ймовірність колізій |

### 3.3 Конструювання програмного забезпечення

#### 3.3.1 Розробка алгоритму

У першому розділі було розглянуто варіанти вирішення задачі і обрано реалізацію жадібного алгоритму з модифікаціями. Класичний жадібний алгоритм обирає найбільш вигідний варіант за одною простою метрикою, наприклад, призначає задачу працівнику з найвищим рівнем потрібної компетенції. Модифікація у такому випадку проходить задачі у заданому порядку, за пріоритетністю, для кожної задачі створює список кандидатів, які підходять, враховуючи технології, потрібні для виконання задачі, рівень складності задачі та рівень володіння навичкою працівника. Вибір “найкращого” у цьому випадку відбувається за сумарною відповідністю.

Вхідні дані алгоритму:

- працівники проєкту;
- навантаження працівників проєкту (кількість задач в роботі);
- обрані менеджером задачі для розподілу.

Для кожної задачі обов'язковим є визначення пріоритету та складності. Пріоритетність враховується для сортування задач для розподілу, а складність визначає мінімальний рівень володіння технологіями, потрібними для задачі.

Складність може бути одним з трьох значень: базова, середня, висока. Відповідність мінімального рівня, потрібного для задач: базова - 1, середня - 3, висока - 5. Значення співставляються з відповідним рівнем знань технології робітника (від 1 до 5) і результат дозволяє відфільтрувати кандидатів на



виконання задачі. Із схемою алгоритму можна ознайомитися на рисунку 3.2.

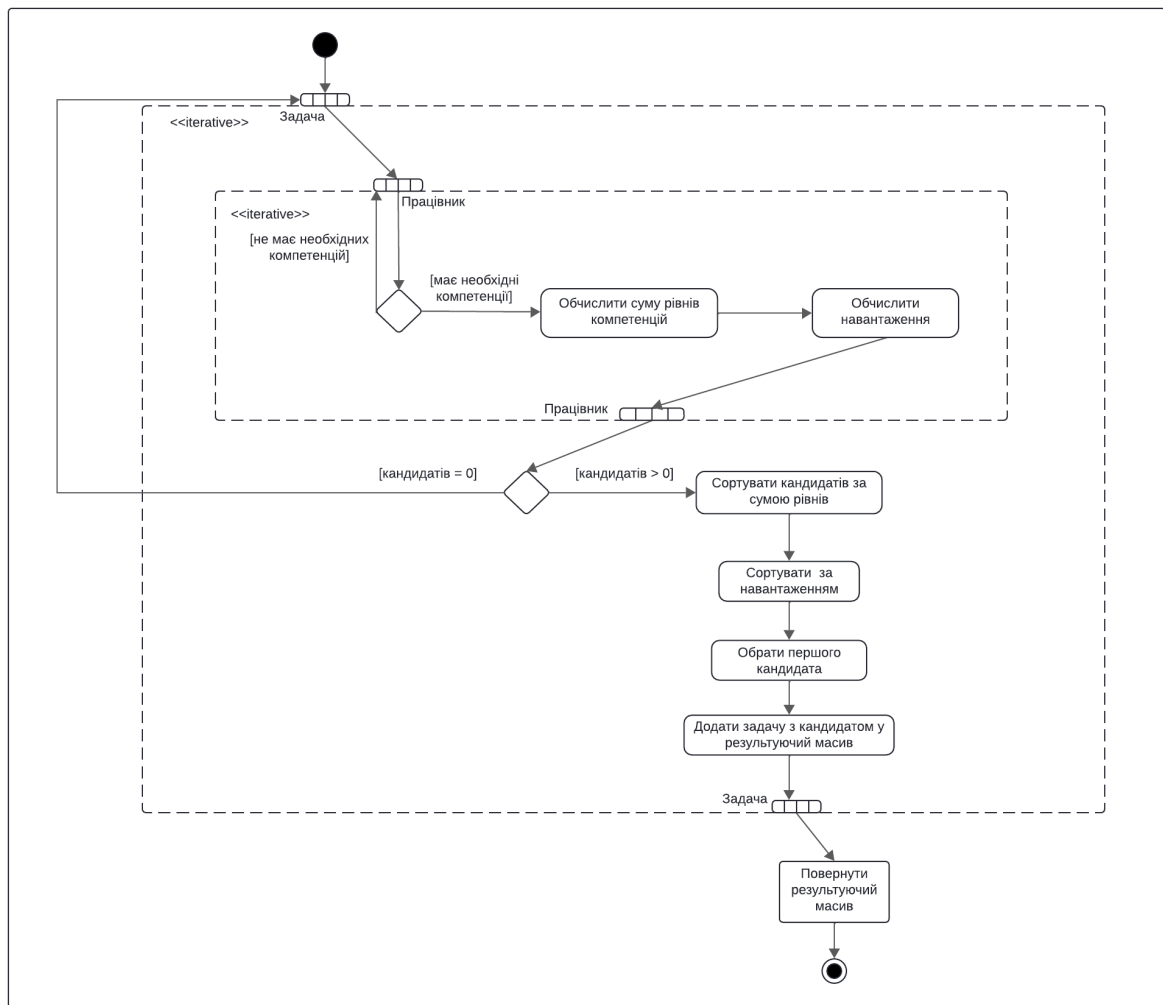


Рисунок 3.2 - Схема алгоритму розподілу

### 3.3.2 Опис структури бази даних

У якості системи управління базами даних використовується MongoDB. База даних серверу призначена для зберігання користувачів, робітників, проєктів та задач. Матриця компетенцій команди формується з компетенцій кожного працівника команди.

Модель бази даних застосунку наведена на рисунку 3.3.

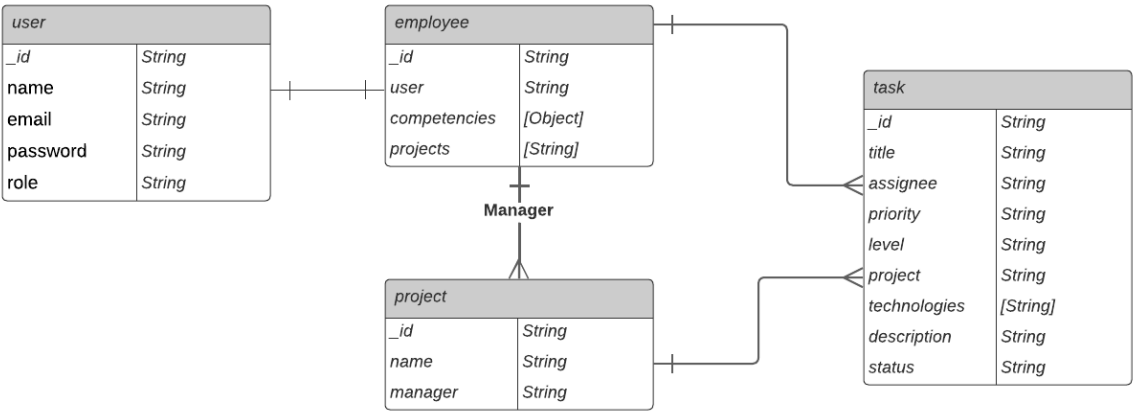


Рисунок 3.3 - Схема бази даних

Колекції для кожної сутності описано у таблицях 3.2 - 3.5.

Таблиця 3.2 - Опис колекції users

| Назва поля | Тип даних | Опис                                                     |
|------------|-----------|----------------------------------------------------------|
| _id        | String    | Унікальний ідентифікатор користувача                     |
| name       | String    | Ім'я користувача                                         |
| email      | String    | Адреса електронної пошти користувача                     |
| password   | String    | Хеш паролю користувача                                   |
| role       | String    | Роль користувача (Працівник, Менеджер або Адміністратор) |

Таблиця 3.3 - Опис колекції employees

| Назва поля | Тип даних | Опис                                                            |
|------------|-----------|-----------------------------------------------------------------|
| _id        | String    | Унікальний ідентифікатор робітника                              |
| user       | String    | Унікальний ідентифікатор користувача для авторизації працівника |

Продовження таблиці 3.3

| Назва поля   | Тип даних | Опис                                               |
|--------------|-----------|----------------------------------------------------|
| competencies | [Object]  | Компетенції користувача                            |
| projects     | [String]  | Посилання на проекти, у яких бере участь працівник |

Таблиця 3.4 - Опис колекції tasks

| Назва поля   | Тип даних | Опис                                       |
|--------------|-----------|--------------------------------------------|
| _id          | String    | Унікальний ідентифікатор користувача       |
| title        | String    | Назва задачі                               |
| status       | String    | Статус задачі, що описує етап виконання    |
| priority     | String    | Пріоритет задачі                           |
| level        | String    | Рівень складності задачі                   |
| project      | String    | Посилання на проект, якому належить задача |
| technologies | [String]  | Технології необхідні для виконання задачі  |
| description  | String    | Текст опису                                |
| assignee     | String    | Посилання на виконавця                     |

Таблиця 3.5 - Опис колекції projects

| Назва поля | Тип даних | Опис                                               |
|------------|-----------|----------------------------------------------------|
| _id        | String    | Унікальний ідентифікатор проєкту                   |
| name       | String    | Назва проєкту                                      |
| manager    | String    | Посилання на працівника, відповідального за проєкт |

### 3.4 Аналіз безпеки даних

У даному розділі буде розглянуто потенційні ризики, пов'язані з безпекою даних та заходи попередження проблем, застосовані у проєкті.

#### 1. Несанкціонований доступ до даних

Облікові дані користувачів можуть бути викраденими, а сесійні токени перехоплені.

Заходи безпеки, застосовані у даному випадку це застосування JWT токенів для авторизації, хешування паролів за допомогою bcrypt, система ролей і дозволів для розділення доступу до даних, зберігання JWT-токенів у HttpOnly cookies, недоступних для JavaScript.

#### 2. NoSQL-ін'єкції [14]

Зловмисники можуть маніпулювати запитами до бази даних через неперевірені вхідні дані, тому для попередження цього у проєкті використовується валідація даних за допомогою бібліотеки class-validator, яка не дозволяє некоректним даним бути обробленими на сервері.

#### 3. Cross-Site Request Forgery (CSRF) [15] та несанкціоновані запити

Існує ризик, що операції над даними можуть виконуватися від імені авторизованого користувача без його відома, а зловмисні сайти можуть робити запити до API. Щоб прибрати вразливості цього типу, необхідно налаштувати Cross-Origin Resource Sharing(CORS) на сервері, чітко визначивши дозволені домени, які можуть комунікувати з API.

## Висновки до розділу

У даному розділі проведено аналіз архітектури проєкту, де розглянуто основні компоненти та зв'язки між ними, обґрунтовано доречність застосування обраних технологій. Було обрано клієнт-серверну архітектуру із монолітним бекендом, що є класичним рішенням для розробки невеликих проєктів. Для імплементації компонентів обрано прогресивний фреймворк NestJS для серверу і Angular для клієнта, які використовують мову TypeScript. Також описано заходи для виконання зазначених у другому розділі нефункціональних вимог для контролю надійності, продуктивності та зручності використання застосунку. Було описано алгоритм розподілу завдань, що є основною бізнес-функцією застосунку. Жадібний алгоритм, модифікований додаванням логіки пов'язаної з компетенціями та технологіями, виявився релевантним для виконання поставленого завдання через відносну швидкість та передбачувані результати.

Останнім пунктом став аналіз безпеки додатку, де було виявлено потенційні ризики, враховуючи описану раніше архітектуру і специфіку застосунку, і запропоновано практичні інструменти підвищення рівня безпеки та уникання зловмисницьких атак.

## 4 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Аналіз якості ПЗ

Для статичного аналізу якості програмного забезпечення було використано утиліту ESLint, яка виявляє порушення стилю коду, потенційні помилки і допомагає оцінювати складність програмної логіки. Метрикою оцінки якості було обрано цикломатичну складність. Цикломатична складність [16] - це кількість незалежних шляхів у вихідному коді. Вона визначає, наскільки складна функція для розуміння та тестування.

Для оцінки цикломатичної складності було активовано правило `complexity` у конфігураційному файлі `eslint.config.mjs`. Порогове значення цикломатичної складності, при перевищенні якого аналізатор виводить попередження, становить 5.

| /Users/anastasiia/task-distributor/backend/src/app/auth/auth.service.ts                 |        |         |                                                                             |                                    |
|-----------------------------------------------------------------------------------------|--------|---------|-----------------------------------------------------------------------------|------------------------------------|
| Line                                                                                    | Column | Type    | Message                                                                     | Rule ID                            |
| 28                                                                                      | 59     | warning | Unexpected any. Specify a different type.                                   | @typescript-eslint/no-explicit-any |
| /Users/anastasiia/task-distributor/backend/src/app/auth/roles-management/roles.guard.ts |        |         |                                                                             |                                    |
| Line                                                                                    | Column | Type    | Message                                                                     | Rule ID                            |
| 15                                                                                      | 20     | warning | Async method 'canActivate' has a complexity of 6. Maximum allowed is 5.     | complexity                         |
| /Users/anastasiia/task-distributor/backend/src/app/distribution/distribution.service.ts |        |         |                                                                             |                                    |
| Line                                                                                    | Column | Type    | Message                                                                     | Rule ID                            |
| 35                                                                                      | 31     | warning | Async method 'distributeTasks' has a complexity of 6. Maximum allowed is 5. | complexity                         |
| 95                                                                                      | 19     | warning | Method 'levelSum' has a complexity of 6. Maximum allowed is 5.              | complexity                         |
| 124                                                                                     | 28     | warning | Arrow function has a complexity of 8. Maximum allowed is 5.                 | complexity                         |
| 0 Errors                                                                                |        |         |                                                                             |                                    |
| 5 Warnings                                                                              |        |         |                                                                             |                                    |

Рисунок 4.1 - Результати статичного аналізу

Результати аналізу демонструють лише невеликі відхилення від встановлених стандартів та відсутність критичних проблем у коді.

Попередження стосуються використання `any` та перевищення значення допустимого рівня цикломатичної складності. Використання типу `any` у методі сервісу аутентифікації потрібне для забезпечення гнучкості використання і є прийнятним, адже це не впливає на якість або безпеку системи. Цикломатична складність методу `canActivate` обумовлена складністю логіки системи авторизації, для якої значення 6 є цілком прийнятним.

Інші перевищення порогу значення цикломатичної складності зустрічаються лише у головному сервісі для імплементації бізнес-логіки, рівні яких допустимими. Для головного методу `distributeTasks()`, який реалізує основний алгоритм розподілу завдань, значення 6 є оптимальним показником, що свідчить про простоту тестування. Інші допоміжні методи слугують для сортування задач та підрахунку суми рівнів володіння технологіями працівника. Рефакторинг цих методів може призвести до надмірного ускладнення читабельності коду та процесу тестування.

Результати свідчать про високий рівень якості коду, що забезпечує підтримуваність та надійність програми.

#### 4.2 Опис процесів тестування

Тестування виконується згідно документу «Програма та методика тестування».

Було виконано ручне тестування програмного забезпечення, опис відповідних тестів наведено у таблицях 4.1 – 4.12.

Таблиця 4.1 – Тест 1.1 Реєстрація користувача

|                         |                                               |
|-------------------------|-----------------------------------------------|
| Початковий стан системи | Користувач знаходиться на сторінці реєстрації |
| Вхідні дані             | Електронна пошта, пароль, ім'я                |

Продовження таблиці 4.1

|                       |                                                                                                                                                                                        |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Опис проведення тесту | У відповідні поля вводяться: коректна електронна пошта, яка до цього не була зареєстрована в системі, пароль від 8 символів. Після цього натискається кнопка підтвердження реєстрації. |
| Очікуваний результат  | Реєстрація проходить успішно, користувач додається у систему і перенаправляється на сторінку авторизації.                                                                              |
| Фактичний результат   | Збігається з очікуваним.                                                                                                                                                               |

Таблиця 4.2 – Тест 1.2 Авторизація користувача

|                         |                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Початковий стан системи | Користувач знаходиться на сторінці авторизації                                                                           |
| Вхідні дані             | Електронна пошта, пароль                                                                                                 |
| Опис проведення тесту   | У відповідні поля вводяться електронна пошта та пароль користувача. Після цього натискається кнопка підтвердження входу. |
| Очікуваний результат    | Авторизація проходить успішно, користувач перенаправляється на початкову сторінку залежно від ролі.                      |
| Фактичний результат     | Збігається з очікуваним.                                                                                                 |

Таблиця 4.3 – Тест 1.3 Додавання проєкту

|                         |                                                 |
|-------------------------|-------------------------------------------------|
| Початковий стан системи | Адміністратор знаходиться на сторінці “Проекти” |
| Вхідні дані             | Назва проєкту, менеджер.                        |



Продовження таблиці 4.3

|                       |                                                                                                                                      |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Опис проведення тесту | У відповідні поля вводяться назва проєкту та менеджер, відповідальний за проєкт. Після цього натискається кнопка збереження проєкту. |
| Очікуваний результат  | Проєкт створено успішно, він відображається у списку.                                                                                |
| Фактичний результат   | Збігається з очікуваним.                                                                                                             |

Таблиця 4.4 – Тест 1.4 Редагування ролі користувача

|                         |                                                                                                                                |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| Початковий стан системи | Адміністратор знаходиться на сторінці “Користувачі”                                                                            |
| Вхідні дані             | Нова роль користувача                                                                                                          |
| Опис проведення тесту   | У рядку користувача, для якого змінюється роль, обирається нова роль. Після цього натискається кнопка підтвердження оновлення. |
| Очікуваний результат    | Роль користувача оновлено успішно.                                                                                             |
| Фактичний результат     | Збігається з очікуваним.                                                                                                       |

Таблиця 4.5 – Тест 1.5 Редагування менеджера проєкту

|                         |                                                                                                                                                 |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Початковий стан системи | Адміністратор знаходиться на сторінці “Проєкти”                                                                                                 |
| Вхідні дані             | Новий менеджер проєкту                                                                                                                          |
| Опис проведення тесту   | У рядку проєкту, для якого змінюється менеджер, обирається новий менеджер з доступних. Після цього натискається кнопка підтвердження оновлення. |

Продовження таблиці 4.5

|                      |                                                 |
|----------------------|-------------------------------------------------|
| Очікуваний результат | Менеджера проєкту користувача оновлено успішно. |
| Фактичний результат  | Збігається з очікуваним.                        |

Таблиця 4.6 – Тест 1.6 Додавання задачі

|                         |                                                                                                                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Початковий стан системи | Менеджер знаходиться на сторінці обраного проєкту                                                                                                                                           |
| Вхідні дані             | Назва задачі, опис, пріоритет, рівень, виконавець, технології, необхідні для виконання, дедлайн виконання.                                                                                  |
| Опис проведення тесту   | Менеджер натискає кнопку “Додати задачу”, відкривається форма редагування задачі. У відповідні поля вводяться вхідні дані. Після цього натискається кнопка підтвердження збереження задачі. |
| Очікуваний результат    | Задачу успішно додано до проєкту.                                                                                                                                                           |
| Фактичний результат     | Збігається з очікуваним.                                                                                                                                                                    |

Таблиця 4.7 – Тест 1.7 Редагування задачі

|                         |                                                                                                            |
|-------------------------|------------------------------------------------------------------------------------------------------------|
| Початковий стан системи | Менеджер знаходиться на сторінці обраного проєкту                                                          |
| Вхідні дані             | Назва задачі, опис, пріоритет, рівень, виконавець, технології, необхідні для виконання, дедлайн виконання. |

Продовження таблиці 4.7

|                       |                                                                                                                                                                                                                          |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Опис проведення тесту | У рядку задачі, яку менеджер хоче оновити, натискається кнопка оновлення, відкривається форма редагування задачі. У відповідні поля вводяться нові дані. Після цього натискається кнопка підтвердження оновлення задачі. |
| Очікуваний результат  | Задачу успішно оновлено.                                                                                                                                                                                                 |
| Фактичний результат   | Збігається з очікуваним.                                                                                                                                                                                                 |

Таблиця 4.8 – Тест 1.8 Розподіл задач

|                         |                                                                             |
|-------------------------|-----------------------------------------------------------------------------|
| Початковий стан системи | Менеджер знаходиться на сторінці непризначених задач обраного проєкту       |
| Вхідні дані             | Обрані менеджером задачі для розподілу                                      |
| Опис проведення тесту   | Менеджер обирає задачі для розподілу, натискає кнопку “Розподілити задачі”. |
| Очікуваний результат    | З’являється вікно із результатами розподілу.                                |
| Фактичний результат     | Збігається з очікуваним.                                                    |

Таблиця 4.9 – Тест 1.9 Підтвердження результатів розподілу

|                         |                                                                                |
|-------------------------|--------------------------------------------------------------------------------|
| Початковий стан системи | Менеджер отримує успішні результати розподілу задач                            |
| Вхідні дані             | Обрані менеджером задачі для оновлення                                         |
| Опис проведення тесту   | Менеджер обирає розподілені задачі для збереження, натискає кнопку “Зберегти”. |
| Очікуваний результат    | Задачі оновлено успішно, вони відображаються у вкладці “Призначені задачі”.    |
| Фактичний результат     | Збігається з очікуваним.                                                       |

Таблиця 4.10 – Тест 1.10 Додавання компетенції

|                         |                                                                                                                                                                               |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Початковий стан системи | Працівника авторизовано, він знаходиться на сторінці “Компетенції”                                                                                                            |
| Вхідні дані             | Технологія, рівень володіння                                                                                                                                                  |
| Опис проведення тесту   | Працівник натискає кнопку “Додати компетенцію”, відкривається форма. У відповідні поля вводиться технологія та рівень її володіння працівника. Натискається кнопка “Зберегти” |
| Очікуваний результат    | Компетенцію додано успішно, вона відображається у вкладці “Компетенції”.                                                                                                      |
| Фактичний результат     | Збігається з очікуваним.                                                                                                                                                      |

Таблиця 4.11 – Тест 1.11 Редагування компетенції

|                         |                                                                                                                                  |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Початковий стан системи | Працівника авторизовано, він знаходиться на сторінці “Компетенції”                                                               |
| Вхідні дані             | Технологія, рівень володіння                                                                                                     |
| Опис проведення тесту   | У рядку компетенції, яку працівник хоче оновити, обирається новий рівень володіння, натискається кнопка підтвердження оновлення. |
| Очікуваний результат    | Компетенцію оновлено успішно.                                                                                                    |
| Фактичний результат     | Збігається з очікуваним.                                                                                                         |

Таблиця 4.12 – Тест 1.12 Перегляд призначених задач

|                         |                                                               |
|-------------------------|---------------------------------------------------------------|
| Початковий стан системи | Працівника авторизовано, він знаходиться на сторінці “Задачі” |
| Вхідні дані             | -                                                             |

Продовження таблиці 4.12

|                       |                                                                        |
|-----------------------|------------------------------------------------------------------------|
| Опис проведення тесту | -                                                                      |
| Очікуваний результат  | Задачі, призначені авторизованому користувачу, відображаються списком. |
| Фактичний результат   | Збігається з очікуванням.                                              |

Для аналізу сервісу з розподілу завдань було використано Jest - стандартний інструмент unit-тестування NestJS. Тестування `DistributionService` включало перевірку основного бізнес-методу `distributeTasks()`, а саме коректність розподілу завдань та відповідність функціональним вимогам. Було протестовано стандартний сценарій розподілу, логіку вибору працівників за навантаженням і пріоритизацію задач. За допомогою можливостей мокування Jest вдалося ізолювати тестування від зовнішніх залежностей і алгоритм успішно спрацював на тестових даних, що підтвердило надійність і коректність його роботи.

|                                                       |      |                                               |
|-------------------------------------------------------|------|-----------------------------------------------|
| ✓ Test Results                                        | 9 ms | ✓ Tests passed: 3 of 3 tests - 9 ms           |
| ✓ distribution.spec.ts                                | 9 ms | jest DistributionService distribution.spec.ts |
| ✓ DistributionService                                 | 9 ms |                                               |
| ✓ distributeTasks                                     | 9 ms |                                               |
| ✓ should distribute tasks to qualified employees      | 6 ms |                                               |
| ✓ should assign task to employee which is less loaded | 2 ms |                                               |
| ✓ should assign higher priority tasks first           | 1 ms |                                               |

Рисунок 4.2 - Результати виконання модульних тестів

### 4.3 Опис контрольного прикладу

Головним функціоналом програмного забезпечення є автоматизований розподіл задач, яким користувач може скористатися, маючи роль менеджера.

На рисунку 4.3 авторизований менеджер має призначені проєкти, один з яких обрано для демонстрації функціоналу.

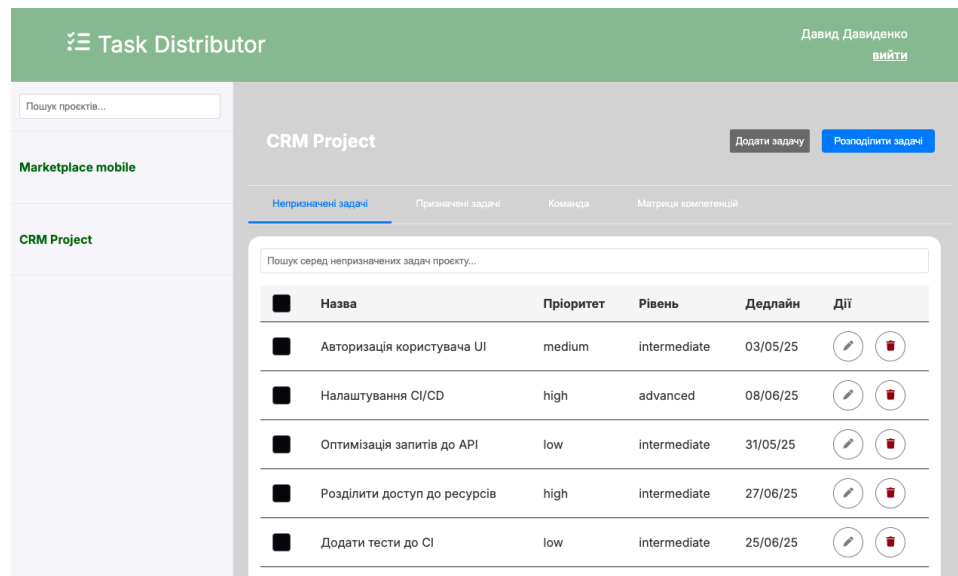


Рисунок 4.3 - Деталі проєкту у профілі менеджера

Щоб запустити процес розподілу завдань менеджер має обрати задачі з непризначених і натиснути кнопку “Розподілити задачі”. Результат розподілу відображається у діалоговому вікні, що має дві вкладки: задачі, для яких вдалося знайти виконавця(рисунок 4.4) та ті, для яких не було знайдено кандидатів (рисунок 4.5).

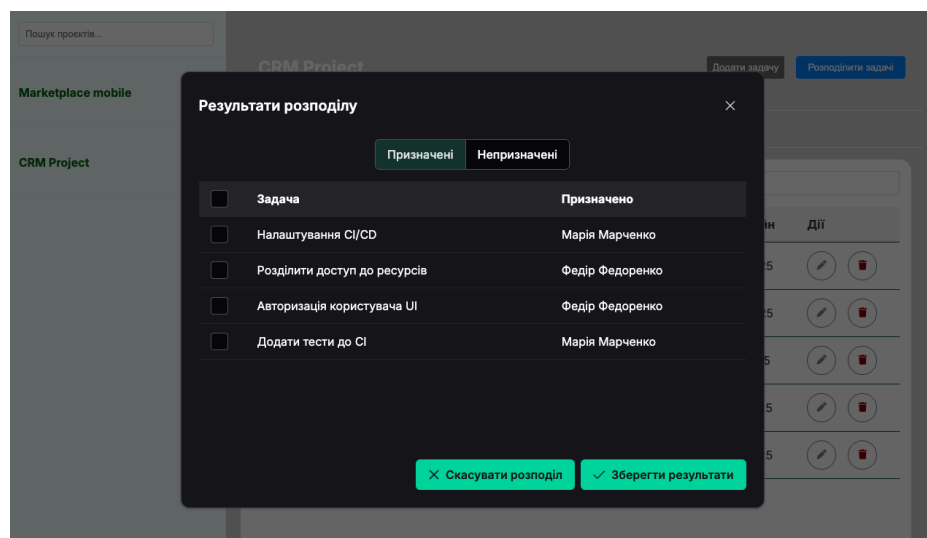


Рисунок 4.4 - Результати успішного розподілу

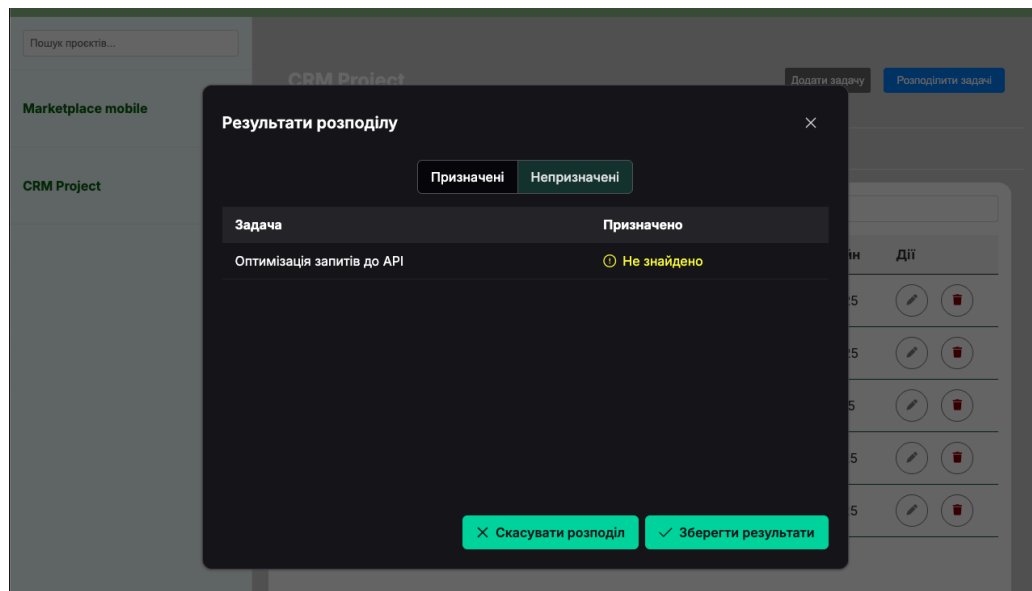


Рисунок 4.5 - Непризначені задачі

Наступним кроком менеджера є вибір задач, які він хоче оновити згідно з результатами розподілу та натиснути кнопку “Зберегти результати” для підтвердження оновлення завдань у системі.

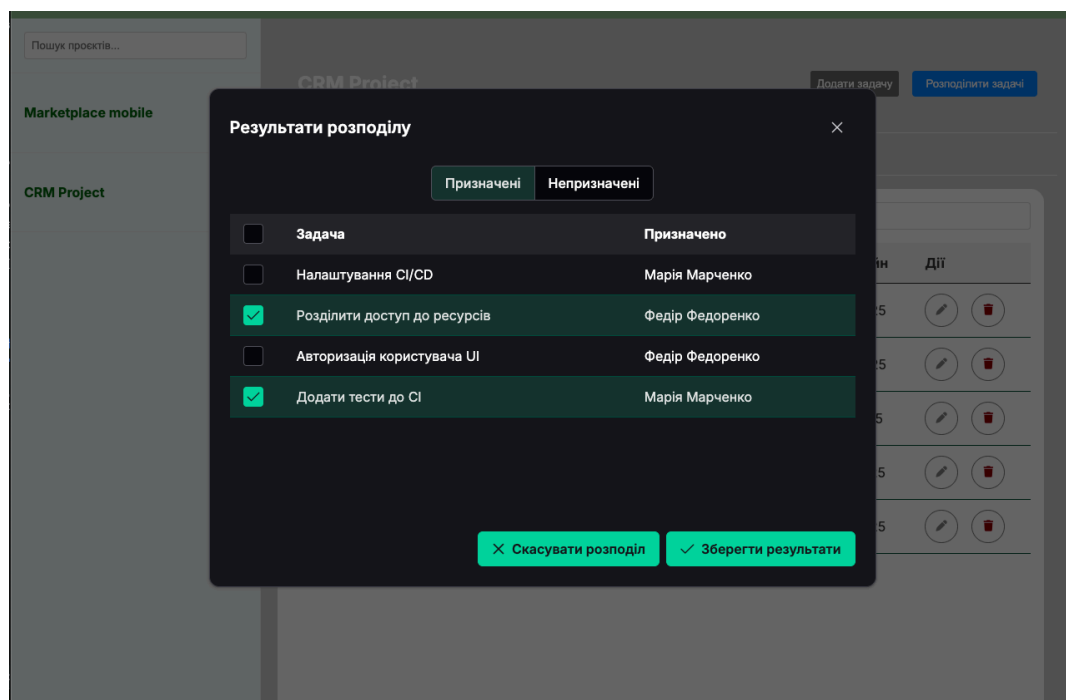


Рисунок 4.6 - Обрані менеджером задачі для призначення  
Результатом є відображення призначених задач у відповідній вкладці.

The screenshot displays a web application for managing a CRM project. On the left is a sidebar with a search bar and two menu items: 'Marketplace mobile' and 'CRM Project'. The main area is titled 'CRM Project' and contains buttons for 'Додати задачу' (Add task) and 'Розподілити задачу' (Distribute task). Below these are tabs for 'Непризначені задачі' (Unassigned tasks), 'Призначені задачі' (Assigned tasks), 'Команда' (Team), and 'Матриця компетенцій' (Competency matrix). The 'Assigned tasks' tab is active, showing a search bar and a table of tasks.

| Назва                        | Пріоритет | Рівень       | Призначено      | Дедлайн  | Дії             |
|------------------------------|-----------|--------------|-----------------|----------|-----------------|
| Розділити доступ до ресурсів | high      | intermediate | Федір Федоренко | 27/06/25 | [Edit] [Delete] |
| Додати тести до CI           | low       | intermediate | Марія Марченко  | 25/06/25 | [Edit] [Delete] |

Рисунок 4.7 - Результати збереження результатів призначення

### Висновки до розділу

У даному розділі було проаналізовано якість розробленого програмного забезпечення шляхом проведення оцінки статичним аналізатором ESLint. Цикломатична складність методів не перевищує допустимої норми, що свідчить про прийнятний рівень складності та демонструє простоту тестування коду.

Програмне забезпечення також було протестовано вручну відповідно до сценаріїв, описаних у тест-кейсах. Результати були позитивними, що підтвердило виконання функціональних вимог. Крім того, було створено unit-тести для перевірки коректності роботи алгоритму розподілу задач. Виконання тестів дозволяє контролювати поведінку алгоритму, що є важливим для подальшого супроводження програмного забезпечення.

Наприкінці було описано контрольний приклад процесу розподілу завдань менеджером, який продемонстрував практичне застосування автоматизованого розподілу.

Результати проведеного аналізу підтверджують надійність і коректність роботи розробленого програмного забезпечення, що є необхідною умовою для його розгортання та подальшого технічного супроводу.



## 5 РОЗГОРТАННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 5.1 Розгортання програмного забезпечення

Для розгортання програмного забезпечення було обрано хостингові платформи Vercel та Render, базу даних розгорнуто за допомогою хмарного сервісу MongoDB Atlas. Vercel спеціалізується на фронтенд застосунках та автоматично інтегрується з GitHub. Платформа забезпечує доставку контенту через мережу CDN та автоматичну оптимізацію статичних ресурсів. Render надає стабільне середовище виконання для [Node.js](#) додатків, що автоматично розподіляє навантаження та забезпечує безпечне зберігання API ключів та конфігураційних параметрів. Обидва рішення пропонують безкоштовні послуги. MongoDB Atlas також має безкоштовний тарифний план, що надає 512 МБ сховища та до 100 одночасних користувачів. Сервіс надає детальну аналітику використання, метрики продуктивності та сповіщення, що дозволяє вчасно виявляти проблеми. Стек забезпечує надійну та масштабовану інфраструктуру застосунку з можливістю потенційного розширення у міру зростання потреб розробки. На рисунках 5.1 - 5.10 описано процес розгортання програмного забезпечення за допомогою вищезазначених рішень.

На рисунку 5.1 створено кластер у MongoDB Atlas для розгортання бази даних.

**Deploy your cluster**

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

| Option                                | Price                                 | Description                                                                                       | Storage | RAM    | vCPU    |
|---------------------------------------|---------------------------------------|---------------------------------------------------------------------------------------------------|---------|--------|---------|
| <input type="radio"/> M10             | \$0.09/hour                           | Dedicated cluster for development environments and low-traffic applications.                      | 10 GB   | 2 GB   | 2 vCPUs |
| <input type="radio"/> Flex            | From \$0.011/hour<br>Up to \$30/month | For application development and testing, with on-demand burst capacity for unpredictable traffic. | 5 GB    | Shared | Shared  |
| <input checked="" type="radio"/> Free | Free                                  | For learning and exploring MongoDB in a cloud environment.                                        | 512 MB  | Shared | Shared  |

**Free forever!** Your free cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

**Configurations**

Name: You cannot change the name once the cluster is created.

Provider: ☒ AWS ☐ Google Cloud ☐ Azure

**Quick setup**

☒ Automate security setup

☒ Preload sample dataset

Рисунок 5.1 - Налаштування кластера у MongoDB Atlas

Також було створено користувача бази даних для доступу до даних та згенеровано рядок для з'єднання з сервером при подальшому розгортанні. На рисунку 5.2 зображено конфігурацію з'єднання.

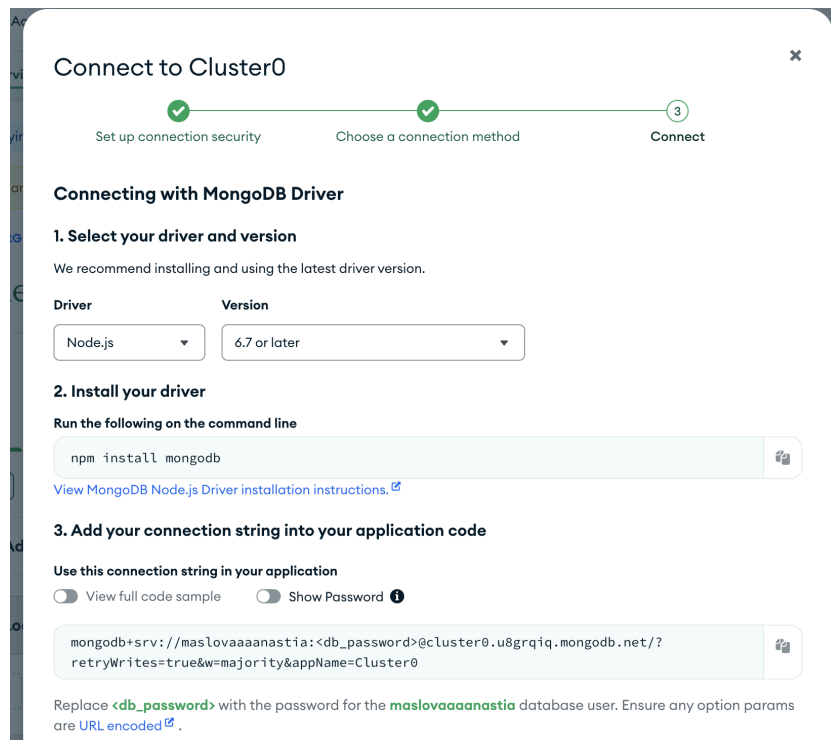


Рисунок 5.2 - Налаштування з'єднання із базою даних

На рисунку 5.3 представлено команди збірки проєкту за допомогою утиліти nx у файлі проєкту package.json для розгортання серверної частини.

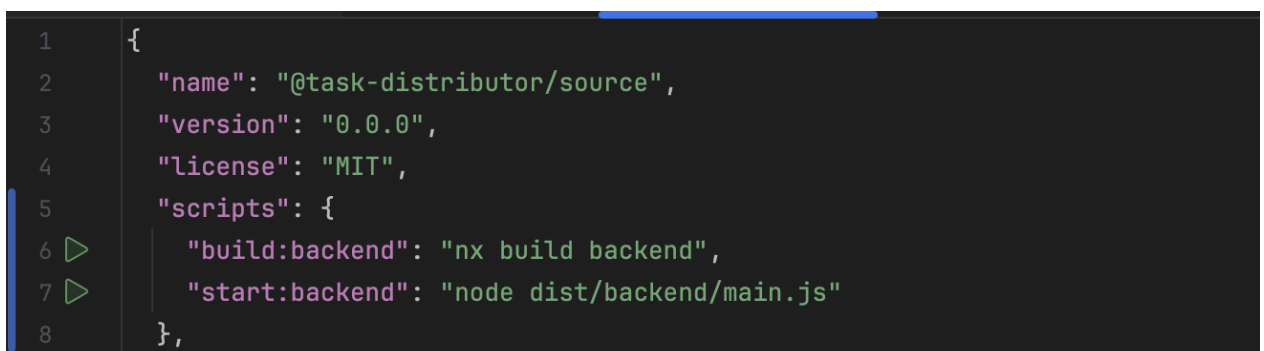


Рисунок 5.3 - Конфігурація для розгортання

Налаштуємо CORS у файлі [main.ts](#), для того, щоб розгорнутий клієнтський додаток міг комунікувати з сервером. На рисунку 5.4 в перелік адрес, які можуть отримати доступ, додано адресу клієнтської частини застосунку.

```

8  async function bootstrap() : Promise<void> { Show usages  anastasiia *
9      const app : INestApplication<any> = await NestFactory.create(AppModule);
10     const globalPrefix = 'api';
11     app.setGlobalPrefix(globalPrefix);
12     app.useGlobalPipes(
13         new ValidationPipe({
14             whitelist: true,
15             transform: true,
16         }),
17     );
18
19     app.use(cookieParser());
20     app.enableCors({
21         origin: ['https://task-distributor.vercel.app', 'http://localhost:4200'],
22         credentials: true,
23     });
24     const port : string | 3000 = process.env.PORT || 3000;
25     await app.listen(port);
26     Logger.log(
27         `Application is running on: http://localhost:${port}/${globalPrefix}`
28     );
29 }
30
31 bootstrap();

```

Рисунок 5.4 - Файл [main.ts](#) із CORS налаштуваннями

Для розгортання бекенду було встановлено Render на акаунт GitHub для взаємодії платформ. На рисунку 5.5 створено новий проєкт, а на рисунку 5.6 визначено команди збірки та запуску застосунку.

The screenshot shows the 'New Web Service' page in the Render dashboard. At the top, there's a header with 'My Workspace' and 'New Web Service' tabs. Below this, a message says 'You are deploying a Web Service'. The 'Source Code' section shows a repository 'nastia485 / task-distributor' updated 33m ago. The 'Name' field is filled with 'task-distributor-backend'. The 'Project' field is optional. A message prompts the user to 'Create a new project to add this to?' with a '+ Create a project' button. The 'Language' dropdown is set to 'Node'.

Рисунок 5.5 - Створення нового проєкту у Render

The screenshot shows the 'New Project' configuration page on the Render platform. It includes several sections with labels and input fields:

- Language:** A dropdown menu set to 'Node'.
- Branch:** A dropdown menu set to 'master'.
- Region:** A dropdown menu set to 'Frankfurt (EU Central)'.
- Root Directory:** A text input field containing 'backend'.
- Build Command:** A text input field containing 'backend/ \$ npm install && npm run build'.
- Start Command:** A text input field containing 'backend/ \$ npm run start:prod'.

Рисунок 5.6 - Налаштування проєкту

На рисунку 5.7 продемонстровано налаштування змінних середовища у файлі [environment.ts](#) у клієнтському додатку, що потрібно для коректної взаємодії із сервером.

```
export const environment = { Show usages  anastasiia *
  production: true,
  apiUrl: 'https://task-distributor-backend-kpa8.onrender.com/api'
};
```

Рисунок 5.7 - Змінні середовища на фронтенді

Vercel було встановлено на акаунт GitHub аналогічно до Render.

На рисунку 5.8 зображено додавання нового проєкту з посиланням на репозиторій, на рисунку 5.9 зазначено команди збірки, встановлення залежностей та директорію dist, яка містить результати компіляції.

The screenshot shows the 'New Project' configuration page on the Vercel platform. It includes several sections with labels and input fields:

- Importing from GitHub:** A section showing the repository 'nastia485/task-distributor' and the branch 'master'.
- Choose where you want to create the project and give it a name:** A section with two sub-sections:
  - Vercel Team:** A dropdown menu set to 'nastia485's projects'.
  - Project Name:** A text input field containing 'task-distributor'.
- Framework Preset:** A dropdown menu set to 'Angular'.
- Root Directory:** A text input field containing './'.

Рисунок 5.8 - Створення проєкту у Vercel

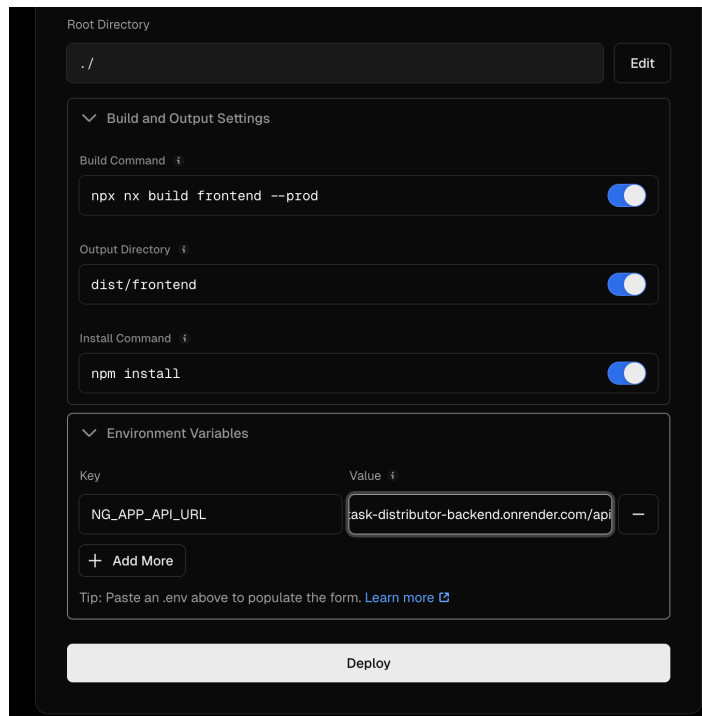


Рисунок 5.9 - Налаштування проєкту у Vercel

Результатом розгортання є адреса ресурсу на хостингу та інформація про стан розгортання проєкту.

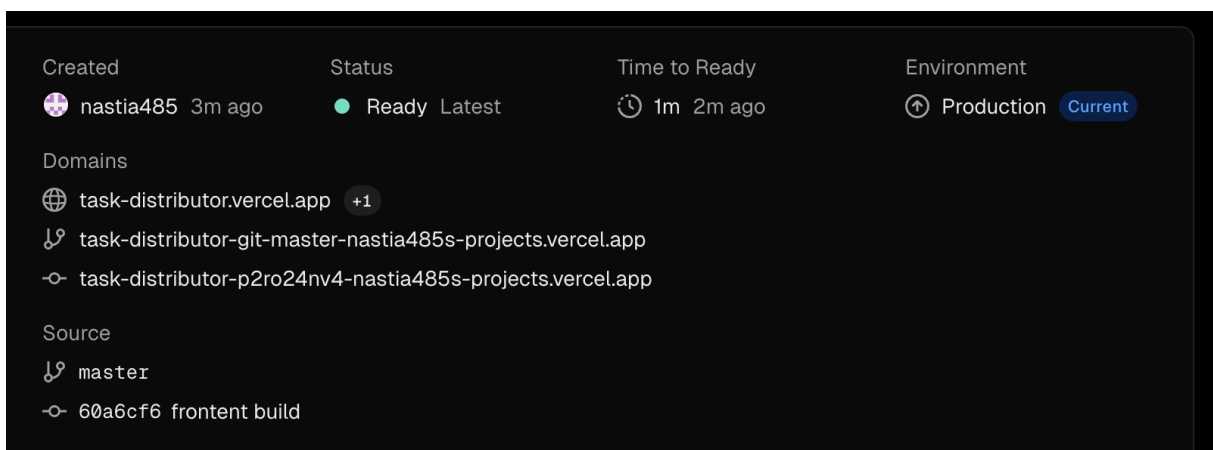


Рисунок 5.10 - Результат розгортання клієнтського додатку у Vercel

## 5.2 Супровід програмного забезпечення

Інструкція користувача наведена в окремому документі “Керівництво користувача”.

Система складається з двох основних компонентів: клієнтської та серверної частини, які розгортаються на різних платформах. Для

автоматизації процесу розгортання використовується інтеграція платформ із GitHub. Оновлення системи починається із внесення змін до головної гілки репозиторію, що запускає два паралельні процеси розгортання.

Vercel автоматично виявляє зміни в Angular додатку, виконує збірку проєкту та розповсюджує оновлену версію через глобальну мережу CDN. Render встановлює залежності через пакетний менеджер npm, збирає NestJS проєкт за допомогою NX та перезапускає сервер.

У разі виявлення критичних помилок на обох платформах передбачено механізм відкату, що автоматично ініціює розгортання останньої стабільної версії.

База даних, розміщена на MongoDB Atlas, забезпечує щоденне створення резервних копій із можливістю відновлення стану системи за будь-який з останніх семи днів.

#### Висновки до розділу

У розділі було описано інструменти та процеси розгортання програмного забезпечення. Для клієнтської частини було обрано платформу Vercel, яка дозволяє досягти високої швидкості доставки оновлень і мінімізувати час простоїв. Для серверної частини було використано хмарну платформу Render, яка є багатофункціональним рішенням для розгортання серверних компонентів. Платформи пропонують безкоштовні тарифи, можливості безперервної доставки та інтеграції, що підвищує надійність застосунку і дозволяє оперативно впроваджувати зміни.

Базу даних було розміщено на сервісі MongoDB Atlas, який гарантує щоденне резервне копіювання з можливістю відновлення стану системи, що є критично важливим для забезпечення цілісності даних.

## ВИСНОВКИ

У результаті виконання дипломного проєкту було виконано проєктування вебзастосунку для автоматизованого розподілу задач, який дозволяє оптимізувати робоче навантаження в командах та підвищити загальну ефективність управління проєктами в ІТ-компаніях.

У ході реалізації було проаналізовано предметну область та наявні аналогічні рішення, що дозволило сформулювати вимоги до системи та завдання на розробку. Після визначення архітектури було імплементовано модифікований жадібний алгоритм розподілу завдань з урахуванням технічних навичок та поточного навантаження співробітників.

Для перевірки відповідності програмного забезпечення висунутим вимогам було проведено тестування за користувацькими сценаріями. Додатково було створено автоматизовані модульні тести для перевірки коректності роботи алгоритму, а також проведено статичний аналіз коду. Результати тестування підтвердили працездатність системи, а обчислення цикломатичної складності показало прийнятний рівень, що свідчить про якість і підтримуваність коду.

Застосунок реалізує усі заплановані функції: управління проєктами та задачами, розподіл завдань з урахуванням технічних навичок співробітників. Застосунок дозволяє ефективно використовувати кадровий потенціал, що підтверджує досягнення поставленої мети щодо спрощення процесу керування проєктами та підвищення його ефективності.

Розроблене рішення може бути впроваджене у робочі процеси ІТ-компаній для автоматизації розподілу навантаження та оптимізації використання людських ресурсів. Подальший розвиток системи може включати інтеграцію з популярними системами управління проєктами, а також розширення функціоналу шляхом додавання аналітичних звітів для надання рекомендацій менеджерам щодо необхідних навичок для успішного виконання проєкту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Schwalbe K. Introduction to project management. 2016. URL: <https://intropm2.com/wp-content/uploads/2017/09/6e-ch-1-sep-6-2017.pdf> (дата звернення: 28.03.2025).
- 2) Платформа управління проєктами TempoPlanner. URL: <https://www.tempo.io/products/capacity-planner> (дата звернення: 04.04.2025).
- 3) Платформа управління проєктами Monday. URL: <https://monday.com/> (дата звернення: 04.04.2025).
- 4) Платформа управління проєктами ClickUp. URL: <https://app.clickup.com/login> (дата звернення: 04.04.2025).
- 5) Платформа управління проєктами Asana. URL: <https://asana.com> (дата звернення: 04.04.2025).
- 6) Multi-criteria performance analysis for decision making in project management. 2011. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0263786310001419> (дата звернення: 15.04.2025).
- 7) Gabrovšek B., Novak T., Povh J., Poklukar D. R., Žerovnik J. Multiple Hungarian Method for k-Assignment Problem. Mathematics. 2020. Vol. 8, No. 11. URL: <https://www.mdpi.com/2227-7390/8/11/2050> (дата звернення: 16.04.2025).
- 8) Karloff H. Linear Programming. Boston : Birkhäuser, 1991. 142 p.
- 9) Forrest S. Genetic algorithms. ACM Computing Surveys. 1996. Vol. 28, No. 1. P. 77-80.
- 10) Jungnickel D. Graphs, Networks and Algorithms. 4th ed. Berlin : Springer, 2013. 685 p.
- 11) Lile E. A. Client/server architecture: A brief overview. 1994. URL: <https://www.proquest.com/openview/6f660e1578ee4f9741d31bf0bb2aef3f/1> (дата звернення: 16.04.2025).



- 12) Сидоров М. О., Родіонов П. О., Марченко О. І. Методичний посібник для проведення практичних робіт з дисципліни «Економіка ІТ-індустрії та підприємництво». 40 с.
- 13) Estimating With Use Case Points. 2005. URL: <https://www.mountangoatsoftware.com/articles/estimating-with-use-case-points> (дата звернення: 01.05.2025).
- 14) Бондаренко О., Ушкаленко І. Безпека WEB-додатків: актуальні проблеми та їх аналіз. 2017. URL: <http://repository.vsau.org/getfile.php/17100.PDF> (дата звернення: 03.05.2025).
- 15) Kombade R. D., Meshram B. B. CSRF Vulnerabilities and Defensive Techniques. International Journal of Computer Network and Information Security. 2012. Vol. 4, No. 1. P. 31-37.
- 16) Watson A. H., Wallace D. R., McCabe T. J. Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric. NIST Special Publication 500-235. 2013. 114 p.

ДОДАТОК А. Звіт на СПІВПАДІННЯ ТЕКСТУ

Звіт подібності

метадані

Назва організації  
**National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute**  
Заголовок  
**ІП-311\_Маслова\_ПЗ**  
Автор  
Науковий керівник / Експерт  
**ІП-311\_МасловаОлійник Ю.О.**  
підрозділ  
**ФІОТ, К-а інформатики та програмної інженерії**

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

|                        |                                                                                     |     |
|------------------------|-------------------------------------------------------------------------------------|-----|
| Заміна букв            |  | 2   |
| Інтервали              |  | 377 |
| Мікропробіли           |  | 0   |
| Білі знаки             |  | 0   |
| Парафрази (SmartMarks) |  | 146 |

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

| 10 найдовших фраз   |                                                                                                                                                                   | Копір тексту                              |        |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|--------|
| ПОРЯДКОВИЙ<br>НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)                                                                                                                          | КІЛЬКІСТЬ ІДЕНТИЧНИХ<br>СЛІВ (ФРАГМЕНТІВ) |        |
| 1                   | ІП-311_Химич_ПЗ<br>6/2/2025<br>National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (ФІОТ, К-а інформатики та програмної інженерії)    | 72                                        | 0.80 % |
| 2                   | ІП-13_Макарчук_ПЗ<br>5/27/2025<br>National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (ФІОТ, К-а інформатики та програмної інженерії) | 60                                        | 0.66 % |

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ АВТОМАТИЗОВАНОГО РОЗПОДІЛУ  
ЗАВДАНЬ НА ОСНОВІ МАТРИЦІ КОМПЕТЕНЦІЙ СПІВРОБІТНИКІВ**

**Текст програми**

КП.ІП-з1108.045440.03.12

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ Юрій ОЛІЙНИК

Нормоконтроль:

\_\_\_\_\_ Олександр СТЕЛЬМАХ

Виконавець:

\_\_\_\_\_ Анастасія МАСЛОВА

Київ – 2025

**Посилання на репозиторій з повним текстом програмного коду**  
<https://github.com/nastia485/task-distributor>

### **Файл distribution.service.ts**

```
interface EmployeeState {
  employee: Employee;
  levelSum: number;
  dueDateLoad: number;
}

@Injectable()
export class DistributionService {
  private distributionResults: DistributionResult[] = [];
  private readonly logger = new Logger(TaskService.name);
  constructor(
    private taskService: TaskService,
    private employeeService: EmployeeService) {
  }

  public async distributeTasks(distData: DistributionDto):
  Promise<DistributionResult[]> {
    this.distributionResults = [];

    this.logger.verbose('Starting task distribution');
    const tasksToDistribute: Task[] = await
    this.taskService.getByIds(distData.taskIds);
    const tasks = this.sortTasksByDueDate(tasksToDistribute);
    const employees: Employee[] = await
    this.employeeService.findByProject(distData.projectId);
    let employeeStates: EmployeeState[];
    for (const task of tasks) {
      this.logger.verbose('Searching candidates for: ' +
      task.title + ', tech: ' + task.technologies + ', level: ' +
      task.level);
      employeeStates = [];
      for (const employee of employees) {
        const levelSum = this.levelSum(task, employee);

        if (levelSum > 0) {
          employeeStates.push({
            employee: employee,
            levelSum: levelSum,
            dueDateLoad: await this.getEmployeeLoad(task,
            employee)
          });
        }
      }
      if (employeeStates.length > 0) {
        this.logger.verbose('Found candidates: ');
        for (const state of employeeStates) {
```

```

        this.logger.verbose(state.employee.name + ' with
competencies ' + JSON.stringify(state.employee.competencies) +
'\n');
    }

    const sortedStates = employeeStates.sort((a, b) => {
        if (a.dueDateLoad !== b.dueDateLoad) {
            return a.dueDateLoad - b.dueDateLoad;
        }
        return a.levelSum - b.levelSum;
    });

    const bestState = sortedStates[0];
    this.logger.verbose('The best candidate is ' +
bestState.employee.name + ' with load ' + bestState.dueDateLoad
+ ' and summary level of skills ' + bestState.levelSum);

    this.distributionResults.push({ task, assignee:
bestState.employee });
    } else {
        this.logger.verbose('Candidates were not found');
    }
}
return this.distributionResults;
}

    public async assignSelectedTasks(selectedDistribution:
DistributionResultDto[]): Promise<TaskDto[]> {
    const updatedTasks: TaskDto[] = [];
    for (const assignment of selectedDistribution) {
        updatedTasks.push(await
this.taskService.updateTask(assignment.task._id, { assignee:
assignment.assignee._id }));
    }
    return updatedTasks;
}

    private levelSum(task: Task, employee: Employee): number {
    const requiredMinLevel = this.defineMinLevel(task.level);
    const coincidences = [];
    let levelSum = 0;
    for (const technology of task.technologies) {
        for (const competency of employee.competencies) {
            if (technology === competency.technology &&
competency.level >= requiredMinLevel) {
                levelSum += competency.level;
                coincidences.push(technology);
                break;
            }
        }
    }
}

```

```

    return (coincidences.length === task.technologies.length) ?
levelSum : 0;
}

```

```

private defineMinLevel(level: string): number {
  switch (level) {
    case TaskLevel.Basic:
      return 1;
    case TaskLevel.Intermediate:
      return 3;
    case TaskLevel.Advanced:
      return 5;
  }
}

```

```

private sortTasksByDueDate(tasks: Task[]): Task[] {
  return [...tasks].sort((a, b) => {
    const priorityOrder = { 'high': 3, 'medium': 2, 'low': 1 };
    const priorityA = priorityOrder[a.priority] || 0;
    const priorityB = priorityOrder[b.priority] || 0;

    if (priorityA !== priorityB) {
      return priorityB - priorityA;
    }

    if (!a.dueDate && !b.dueDate) {
      return 0;
    }

    if (!a.dueDate) {
      return 1;
    }

    if (!b.dueDate) {
      return -1;
    }

    const dateA = new Date(a.dueDate).getTime();
    const dateB = new Date(b.dueDate).getTime();
    return dateA - dateB;
  });
}

```

```

private async getEmployeeLoad(task: Task, employee: Employee) {

```

```

const savedAssigned = await
this.taskService.findAssignedByEmployee(employee._id);
const currentAssignments =
this.distributionResults.filter(res => res.assignee ===
employee)
    .map(assignment => assignment.task);
const allAssigned = [...savedAssigned,
...currentAssignments];

const assignedBefore = allAssigned.filter(t => {
    return new Date(t.dueDate) < new Date(task.dueDate);
});

return assignedBefore.length ?? 0;
}
}

```

### Файл `distribution-result.component.ts`

```

@Component({
    selector: 'app-distribution-result-dialog',
    imports: [CommonModule, Button, Dialog, ReactiveFormsModule,
TableModule, SelectButton, FormsModule],
    providers: [MessageService],
    templateUrl: './distribution-result-dialog.component.html',
    styleUrls: ['./distribution-result-dialog.component.scss']
})
export class DistributionResultDialogComponent implements OnInit
{
    public unassignedResults: DistributionResult[] = [];

    distributionResult = input<DistributionResult[]>([]);

    initialTasksToDistributeIds = input<string[]>([]);

    selectedPairs: DistributionResult[] = [];

    isVisible = model<boolean>(false);

    distributionCompleted = output<boolean>();

    stateOptions: any[] = [{ label: 'Призначені', value:
'assigned' }, { label: 'Непризначені', value: 'unassigned' }];

    table = 'assigned';

    private taskService = inject(TaskService);

    private distributionService = inject(DistributionService);

```

```

public constructor() {
  effect(async () => {
    const unassignedIds =
      this.initialTasksToDistributeIds().filter(taskId =>
        !this.distributionResult().map(result =>
result.task._id).includes(taskId));
    this.unassignedResults = [];
    for (const id of unassignedIds) {
      this.unassignedResults.push({ task: await
this.taskService.getById(id), assignee: undefined });
    }
  });
}

public cancel() {
  this.isVisible.set(false);
}

public async saveResults(): Promise<void> {
  await
this.distributionService.updateSelectedTasks(this.selectedPairs)
;
  this.isVisible.set(false);
  this.distributionCompleted.emit(true);
}
}

```

### Файл **competency-matrix.component.ts**

```

@Component({
  selector: 'app-competency-matrix',
  imports: [CommonModule, TableModule],
  templateUrl: './competency-matrix.component.html',
  styleUrls: ['./competency-matrix.component.scss']
})
export class CompetencyMatrixComponent implements OnInit {

  private employeeService = inject(EmployeeService);

  public employees: Employee[] = [];
  public teamCompetencies: Technology[] = [];

  projectId = input<string | undefined>();

  async ngOnInit(): Promise<void> {
    await this.getMatrixData();
  }

  public async getMatrixData() {
    this.employees = await
this.employeeService.getByProject(this.projectId()!);
  }
}

```



```

    const skillsSet = new Set<Technology>();
    this.employees.forEach(employee => {
      employee.competencies.forEach(skill => {
        skillsSet.add(skill.technology);
      });
    });

    this.teamCompetencies = Array.from(skillsSet);
  }

  public getSkillLevel(member: Employee, skill: Technology):
  number | undefined {
    return member.competencies.find(competency =>
    competency.technology === skill)?.level;
  }

  public getSkillLevelClass(member: Employee, skill:
  Technology): string {
    const level = this.getSkillLevel(member, skill);

    if (level === undefined || level === null) {
      return 'level-empty';
    }

    if (level >= 1 && level <= 5) {
      return `level-${level}`;
    }

    return 'level-empty';
  }
}

```

### Файл employee.service.ts

```

@Injectable()
export class EmployeeService {
  constructor(
    @InjectModel('Employee') private employeeModel:
    Model<EmployeeDocument>,
    private authService: AuthService
  ) {
  }

  async addCompetency(employeeId: string, competency:
  Competency): Promise<EmployeeDocument> {
    const employee = await this.findEmployee(employeeId);
    employee.competencies.push(competency);
    return employee.save();
  }

  async updateCompetency(employeeId: string, competency:
  Competency): Promise<EmployeeDocument> {
    const employee = await this.findEmployee(employeeId);
  }
}

```

```

    const competencyIndex = employee.competencies.findIndex(
      comp => comp.technology === competency.technology
    );

    if (competencyIndex === -1) {
      throw new NotFoundException(`Competency with technology
"${competency.technology}" not found`);
    }

    employee.competencies[competencyIndex].level =
competency.level;

    const result = await this.updateEmployee(employeeId,
employee);
    return result;
  }

  async deleteCompetency(employeeId: string, competency:
Competency): Promise<EmployeeDocument> {
    const employee = await this.findEmployee(employeeId);

    const competencyIndex = employee.competencies.findIndex(
      comp => comp.technology === competency.technology
    );

    if (competencyIndex === -1) {
      throw new NotFoundException(`Competency with technology
"${competency.technology}" not found`);
    }

    employee.competencies.splice(competencyIndex, 1);

    const result = await this.updateEmployee(employeeId,
employee);
    return result;
  }

  async addMember(data: any): Promise<Employee> {
    let employee = await this.findByEmail(data.data.email);
    if (!employee) {
      throw new NotFoundException(`Employee with email
${data.data.email} is not found`);
    }
    employee = await this.employeeModel.findByIdAndUpdate(
      employee._id,
      { $push: { projects: data.data.projectId } },
      { new: true }
    );
    return employee;
  }
}

```

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ АВТОМАТИЗОВАНОГО РОЗПОДІЛУ  
ЗАВДАНЬ НА ОСНОВІ МАТРИЦІ КОМПЕТЕНЦІЙ СПІВРОБІТНИКІВ**

**Програма та методика тестування**

КП.ІП-з1108.045440.04.51

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ Юрій ОЛІЙНИК

Нормоконтроль:

\_\_\_\_\_ Олександр СТЕЛЬМАХ

Виконавець:

\_\_\_\_\_ Анастасія МАСЛОВА

Київ – 2025

## **ЗМІСТ**

|                                       |          |
|---------------------------------------|----------|
| <b>1 ОБ'ЄКТ ВИПРОБУВАНЬ</b>           | <b>3</b> |
| <b>2 МЕТА ТЕСТУВАННЯ</b>              | <b>4</b> |
| <b>3 МЕТОДИ ТЕСТУВАННЯ</b>            | <b>5</b> |
| <b>4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ</b> | <b>6</b> |

## **1 ОБ'ЄКТ ВИПРОБУВАНЬ**

Об'єктом випробування є вебзастосунок для автоматизованого розподілу завдань на основі матриці компетенцій співробітників. Необхідно перевірити відповідність функціональним вимогам, коректність роботи алгоритму розподілу та якість програмного коду.

## 2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
- перевірка збереження даних;
- перевірка сумісності вебзастосунку з останніми версіями сучасних браузерів (Chrome, Opera, Firefox);
- знаходження проблем, помилок і недоліків з метою їх усунення;
- перевірка зручності графічного інтерфейсу;
- перевірка коректності роботи алгоритму розподілу.

### 3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовуються такі методи:

- статичне тестування – перевіряється програма разом з усією документацією, яка аналізується на предмет дотримання стандартів програмування;
- динамічне тестування – застосовується в процесі виконання програми. Коректність програмного засобу перевіряється на певній кількості тестів;
- функціональне тестування – полягає у перевірці відповідності реальної поведінки програмного забезпечення очікуваній;
- системне тестування – перевіряється усе програмне забезпечення в цілому;
- мануальне тестування – тестування без використання автоматизації, тест-кейси пише особа, що тестує програмне забезпечення;
- тестування «сірої скриньки» – об'єктом тестування тут є деякі особливості внутрішньої поведінки програми. Перевіряється коректність вихідних даних при заданих вхідних, застосовується для тестування окремих алгоритмів (функцій).

## 4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Під час проведення тестування будуть використовуватись наступні допоміжні засоби:

- утиліта ESLint для статичного тестування;
- бібліотека Jest для unit-тестування.

Порядок проведення тестування буде наступним:

- динамічне тестування на відповідність функціональним вимогам;
- тестування зручності використання;
- тестування інтерфейсу користувача;
- тестування у різних браузерях.



Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ АВТОМАТИЗОВАНОГО РОЗПОДІЛУ  
ЗАВДАНЬ НА ОСНОВІ МАТРИЦІ КОМПЕТЕНЦІЙ СПІВРОБІТНИКІВ**

**Керівництво користувача**

КП.ІП-311.045440.05.34

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ Юрій ОЛІЙНИК

Нормоконтроль:

\_\_\_\_\_ Олександр СТЕЛЬМАХ

Виконавець:

\_\_\_\_\_ Анастасія МАСЛОВА

Київ – 2025

**ЗМІСТ**

|     |                                                 |   |
|-----|-------------------------------------------------|---|
| 1   | ПРИЗНАЧЕННЯ ПРОГРАМИ                            | 3 |
| 2   | ПІДГОТОВКА ДО РОБОТИ з ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ | 4 |
| 2.1 | Системні вимоги для коректної роботи            | 4 |
| 2.2 | Завантаження застосунку                         | 4 |
| 2.3 | Перевірка коректної роботи                      | 4 |
| 3   | Виконання програми                              | 6 |

## **1 Призначення програми**

“Task Distributor” - це вебзастосунок призначений для спрощення розподілу завдань на основі їх компетенцій співробітників. Менеджери мають змогу керувати проєктами завдяки можливості додавати на редагувати задачі, додавати працівників у команду проєкту та призначати задачі. Автоматизований розподіл здійснюється з урахуванням знань працівників та їх рівня, які вони можуть вносити у систему самостійно. Для глобального централізованого управління проєктами адміністратори можуть створювати проєкти та призначати відповідальних менеджерів.

Кінцева версія програмного забезпечення представлена у вигляді застосунку, що доступний користувачам через веббраузер.

Фінальна збірка містить необхідні налаштування середовища, конфігурації файли та тестові дані, необхідні для розгортання та перевірки функціональності системи.

## 2 ПІДГОТОВКА ДО РОБОТИ з ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

### 2.1 Системні вимоги для коректної роботи

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 4 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт.

Рекомендована конфігурація технічних засобів:

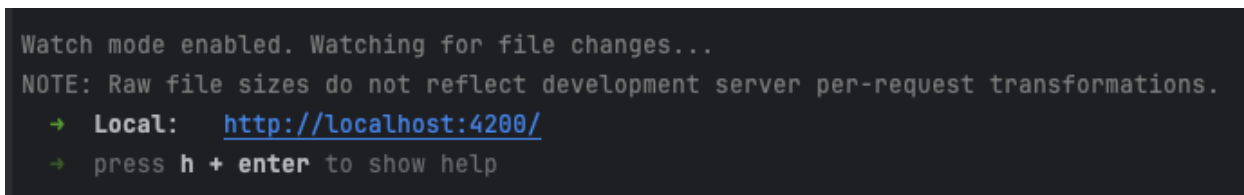
- тип процесору: Intel Core i7;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт.

### 2.2 Завантаження застосунку

На даний момент застосунок можна встановити власноруч, клонувавши код з репозиторію на GitHub та встановивши усі необхідні залежності за допомогою проєктного менеджера npm. Запуск застосунку відбувається за допомогою команди `npm serve`.

### 2.3 Перевірка коректної роботи

По завершенню налаштувань за при запуску застосунку у консолі має з'явитися адреса, за якою застосунок доступний локально, як на рисунку 2.1.



```
Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
→ Local: http://localhost:4200/
→ press h + enter to show help
```

Рисунок 2.1 - Повідомлення про успішний запуск застосунку

Перейшовши за вказаною адресою, користувач має побачити початкову сторінку авторизації(рисунки 2.2).

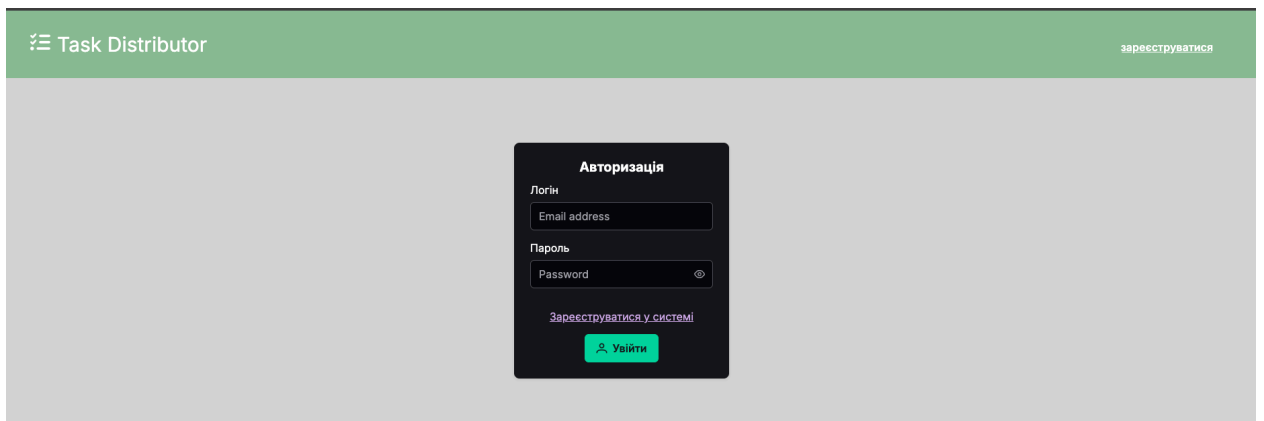


Рисунок 2.2 - Початкова сторінка застосунку

### 3 ВИКОНАННЯ ПРОГРАМИ

При запуску програмного застосунку користувачу буде показано сторінку авторизації (рисунок 3.1).

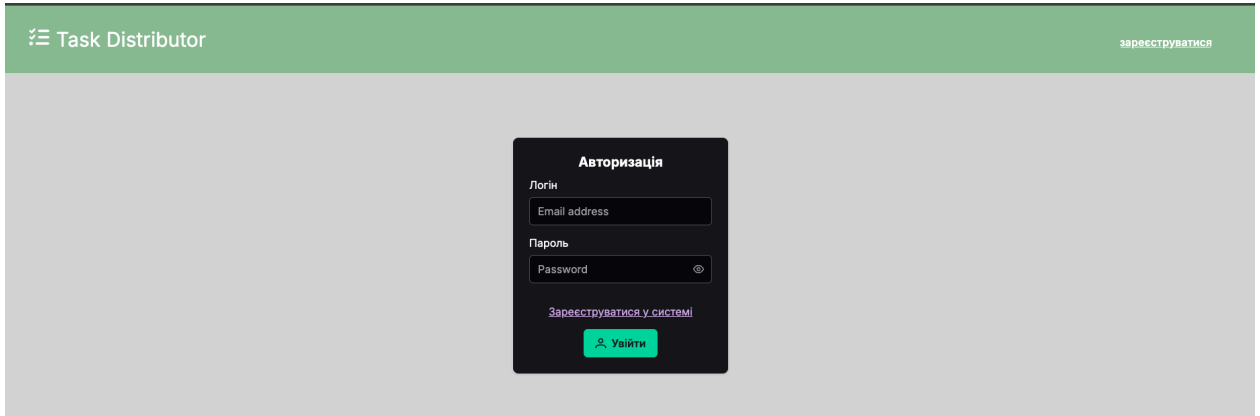


Рисунок 3.1 – Початкова сторінка застосунку

При авторизації користувача як менеджера він потрапить на сторінку проєктів, за які він відповідальний. Обравши проєкт, його деталі будуть відображені в основній частині сторінки. Вкладки містять наступну інформацію про проєкт: непризначені задачі (рисунок 3.2), призначені (рисунок 3.3), працівники проєкту (3.4), матриця компетенцій працівників проєкту(3.5).

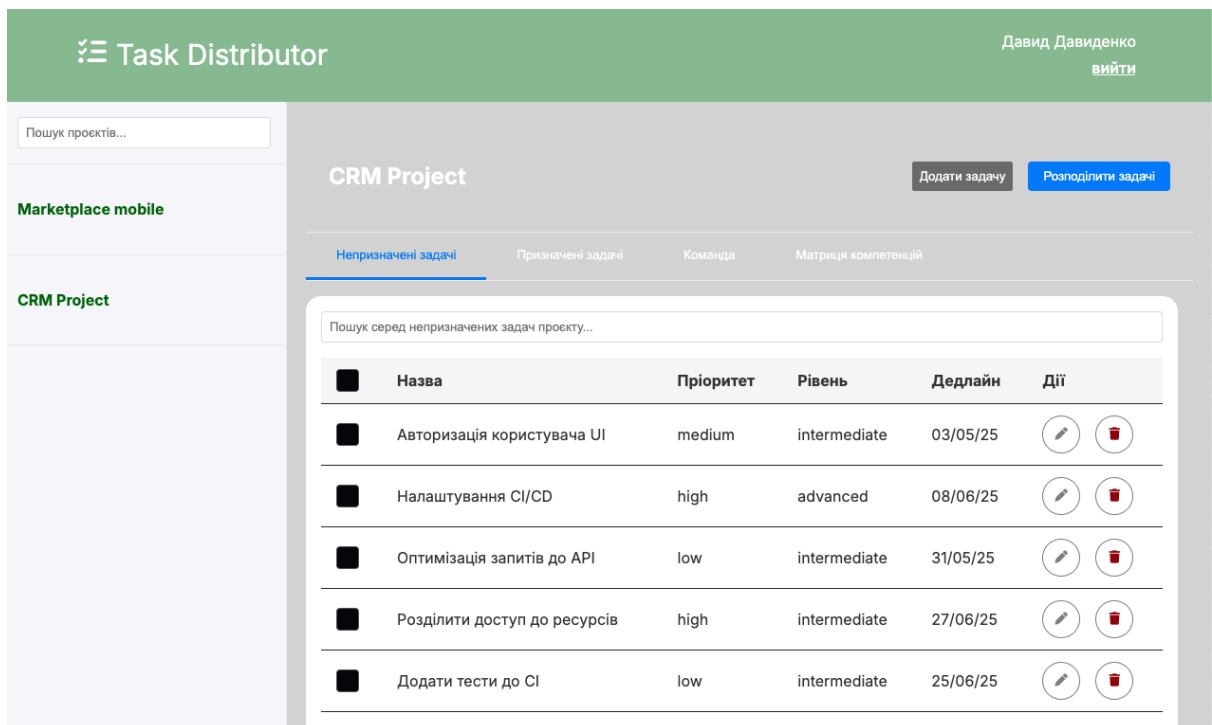


Рисунок 3.2 – Непризначені задачі проєкту

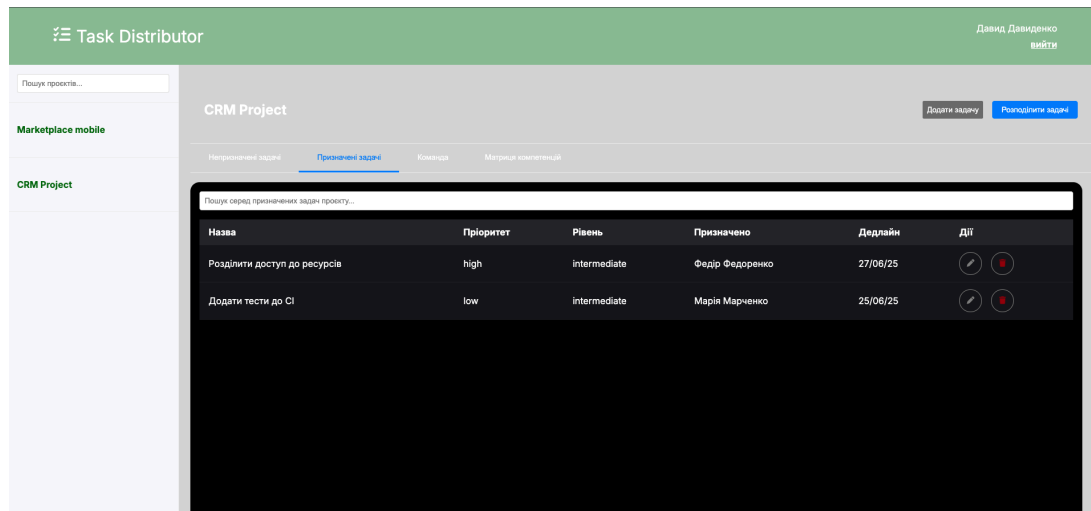


Рисунок 3.3 – Призначені задачі проекту

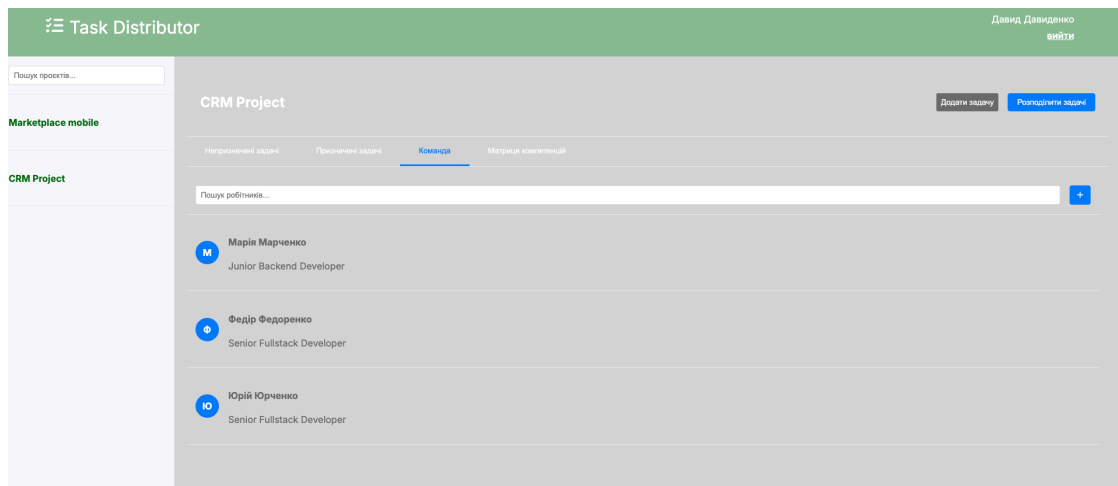


Рисунок 3.4 – Працівники проекту

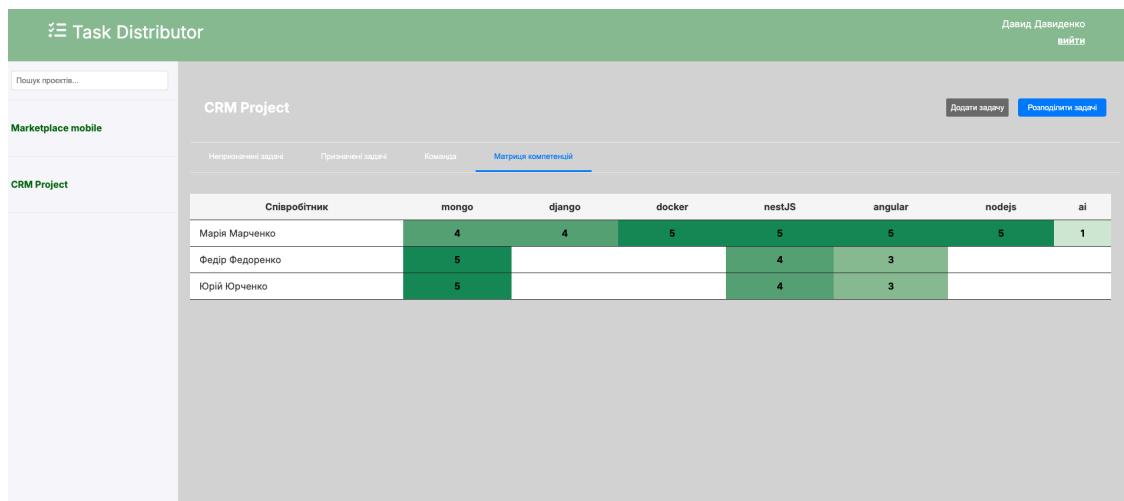


Рисунок 3.5 – Матриця компетенцій працівників

Щоб розподілити задачі, менеджер має обрати їх у вкладці “Непризначені” і натиснути кнопку “Розподілити задачі”. Результатом розподілу є список задач, для яких вдалося знайти виконавця(рисунок 3.6) і для яких не знайшлося кандидатів (рисунок 3.7).

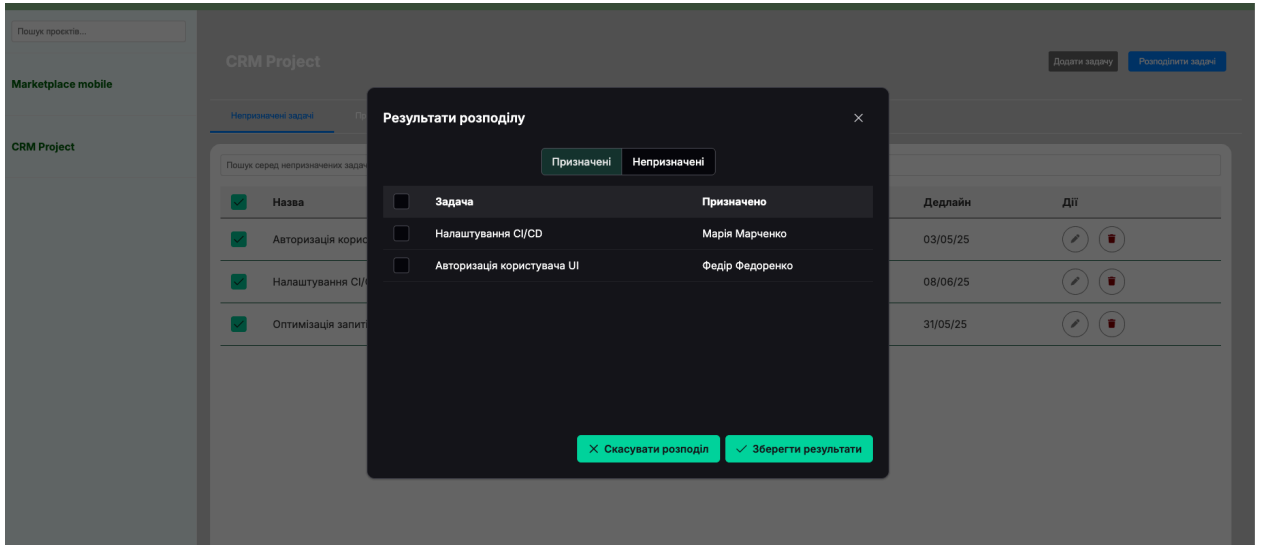


Рисунок 3.6 - Успішні результати розподілу

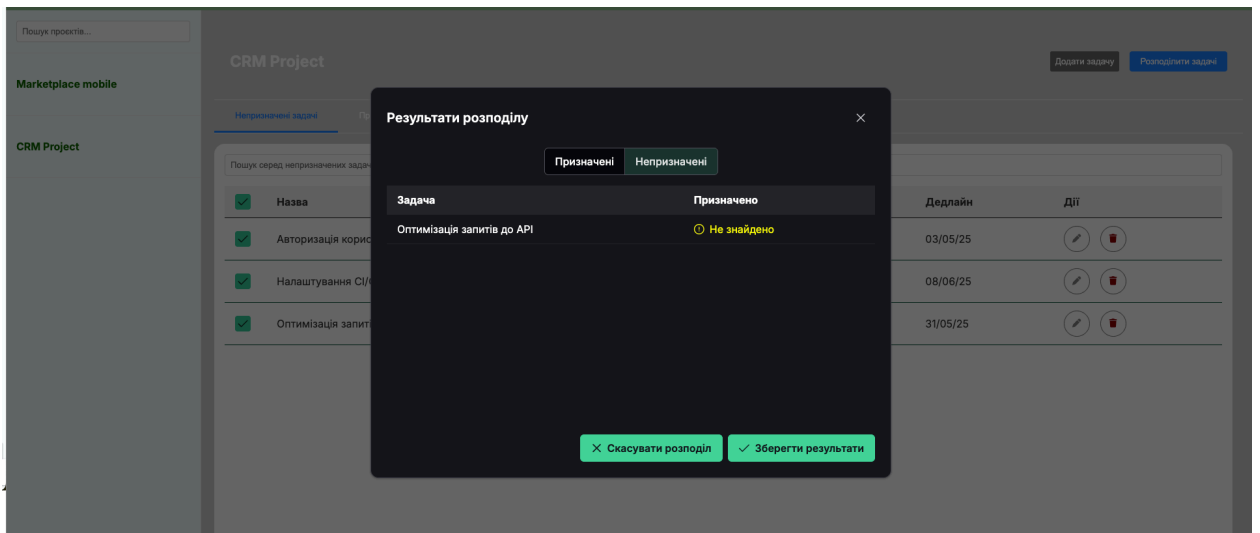


Рисунок 3.7 - Непризначені задачі

Щоб додати нову задачу, необхідно натиснути кнопку “Додати задачу”. Відкриється форма для введення інформації (рисунок 3.8).



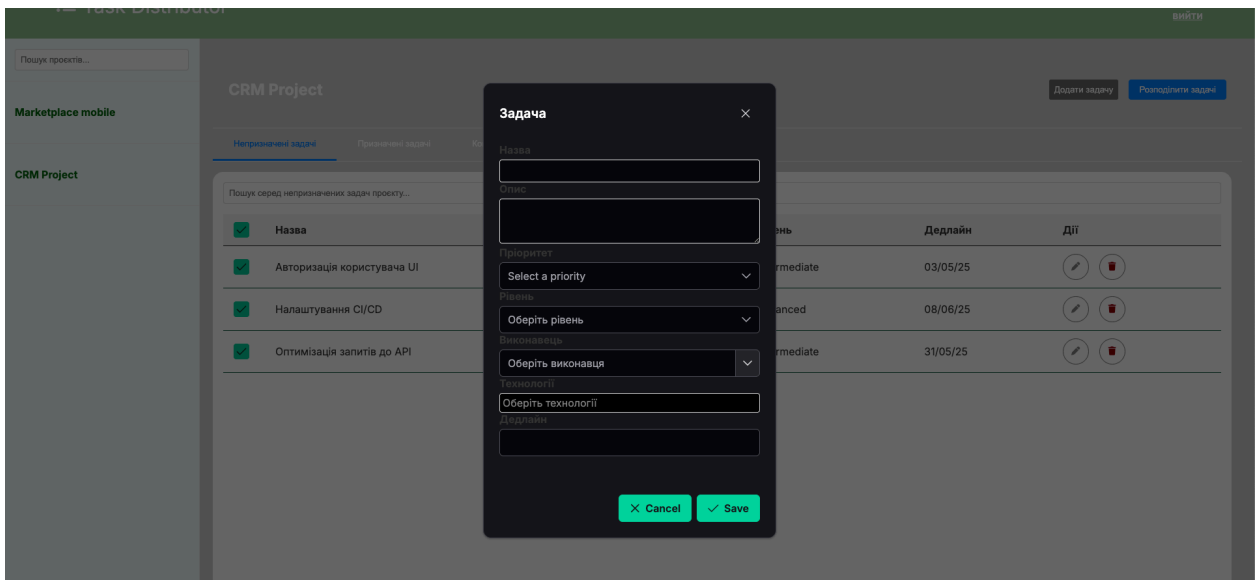


Рисунок 3.8 - Створення нової задачі

Щоб додати працівника у команду, менеджер має ввести його електронну адресу (рисунок 3.9).

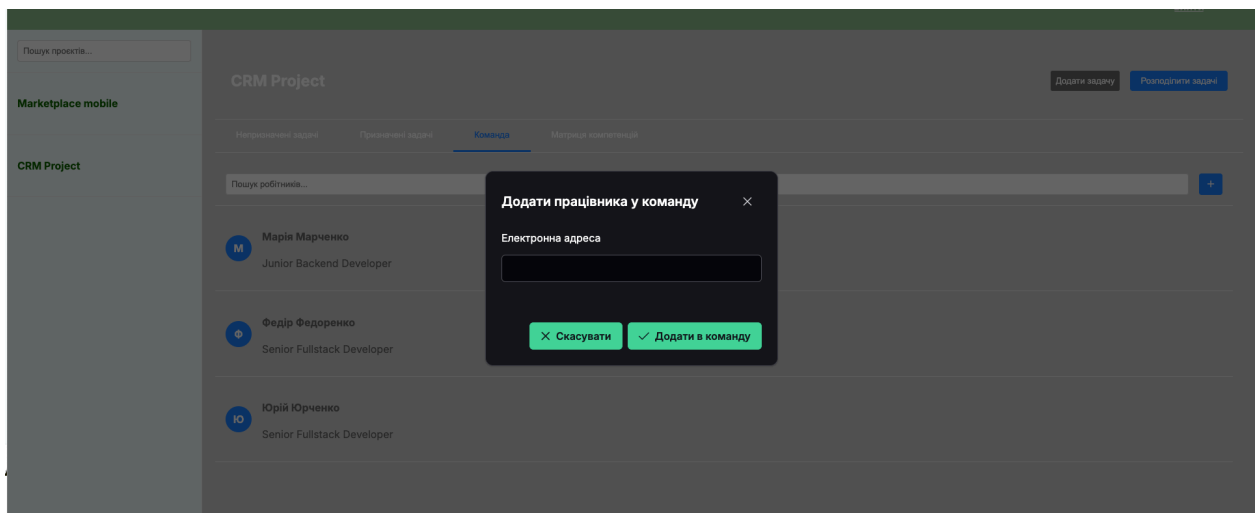


Рисунок 3.9 - Додавання працівника до проєкту

Авторизувавшись як працівник, користувач отримує доступ до своїх компетенцій (рисунок 3.10), маючи змогу додати нову (рисунок 3.11).

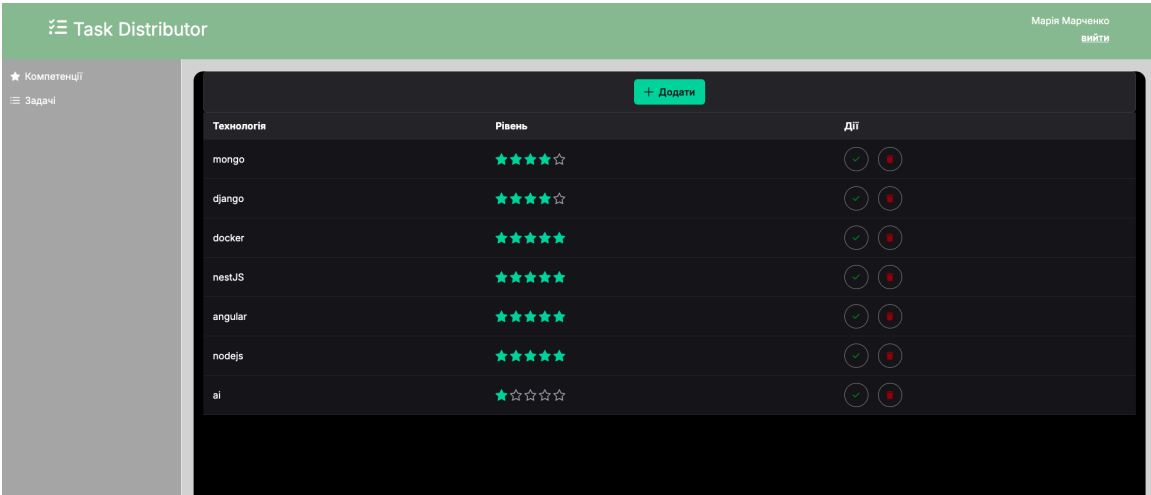


Рисунок 3.10 - Компетенції працівника

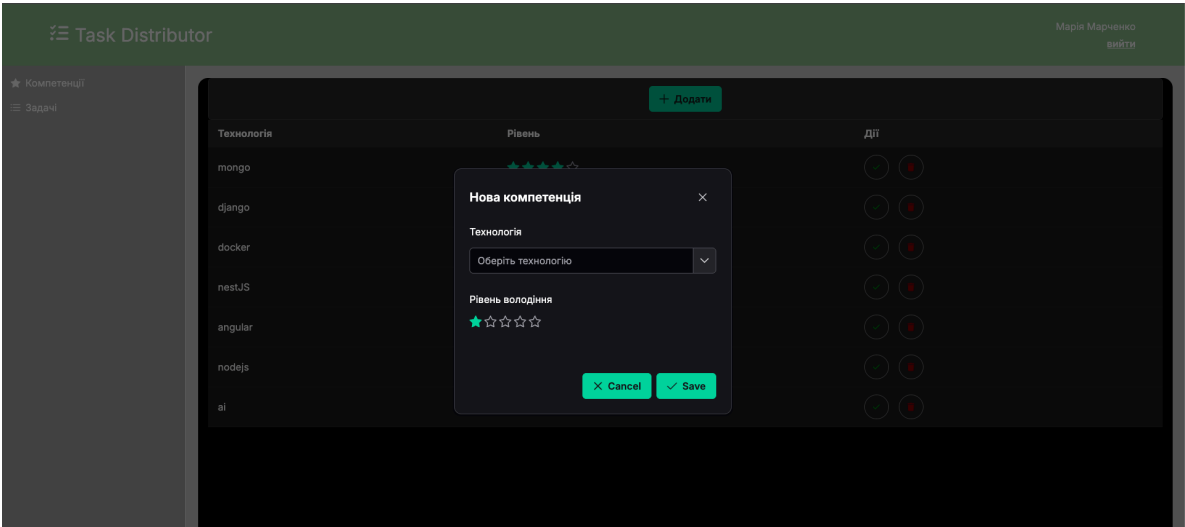


Рисунок 3.11 - Форма додавання нової компетенції

Адміністратор може керувати проєктами(рисунок 3.12), додавати нові (рисунок 3.13) та керувати ролями користувачів (рисунок 3.14).

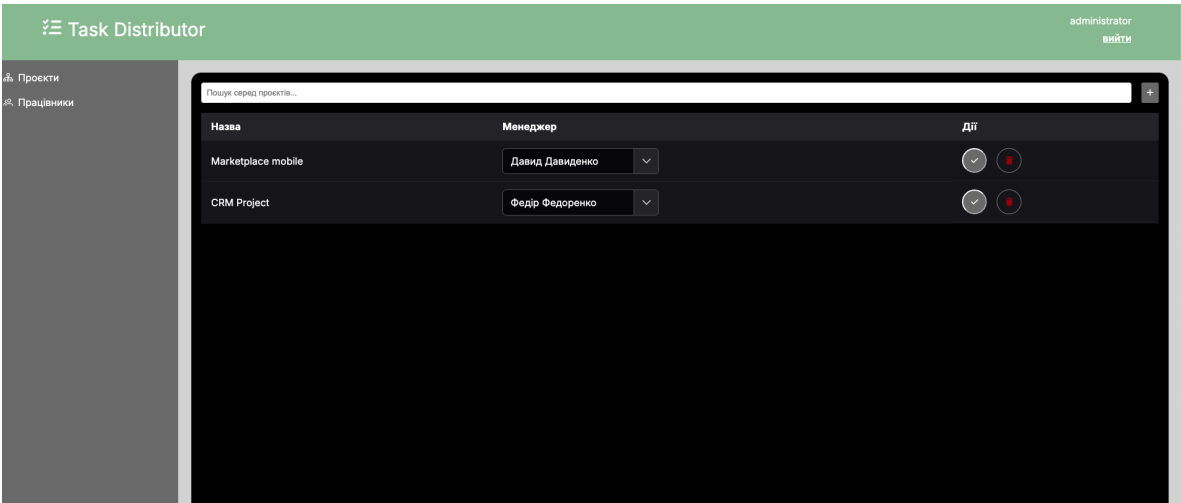


Рисунок 3.12 - Перегляд проєктів системи

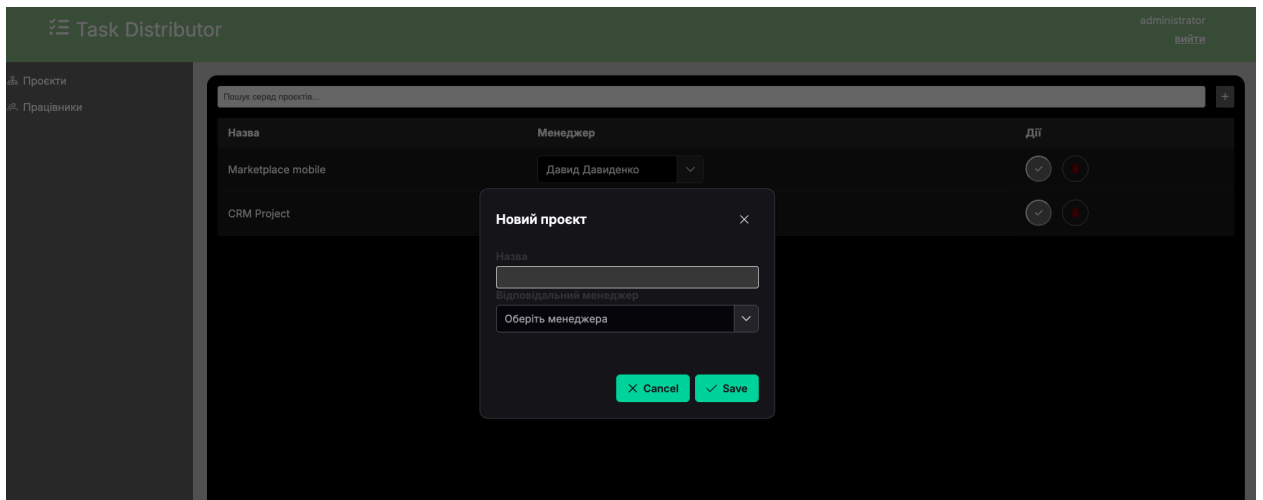


Рисунок 3.13 - Додавання проекту

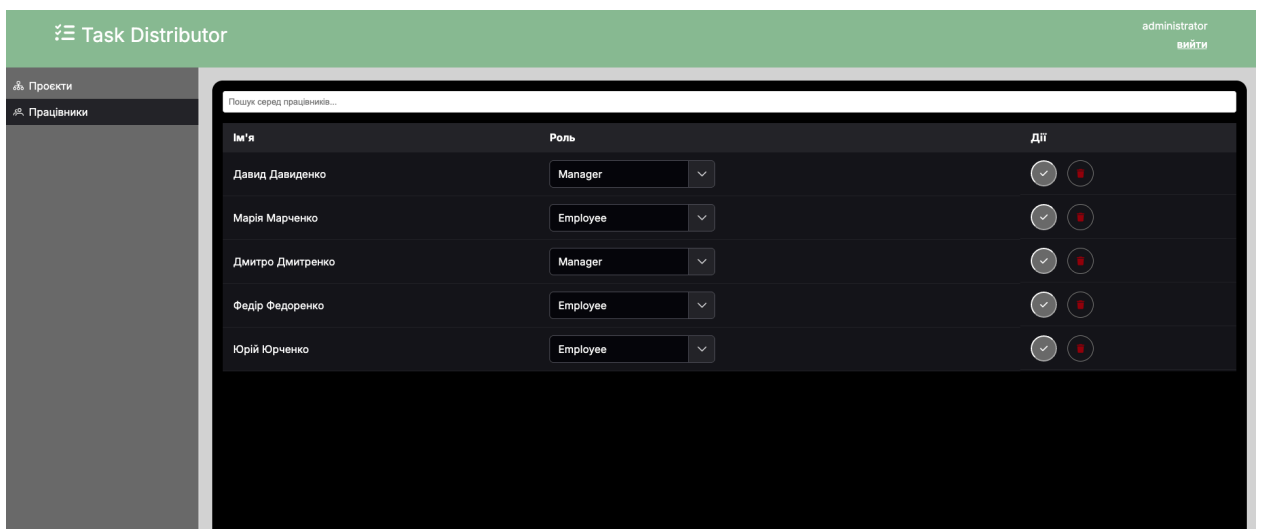


Рисунок 3.14 - Перегляд користувачів

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ АВТОМАТИЗОВАНОГО РОЗПОДІЛУ  
ЗАВДАНЬ НА ОСНОВІ МАТРИЦІ КОМПЕТЕНЦІЙ СПІВРОБІТНИКІВ**

**Графічний матеріал**

КП.ІП-з1108.045440.06.99

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ Юрій ОЛІЙНИК

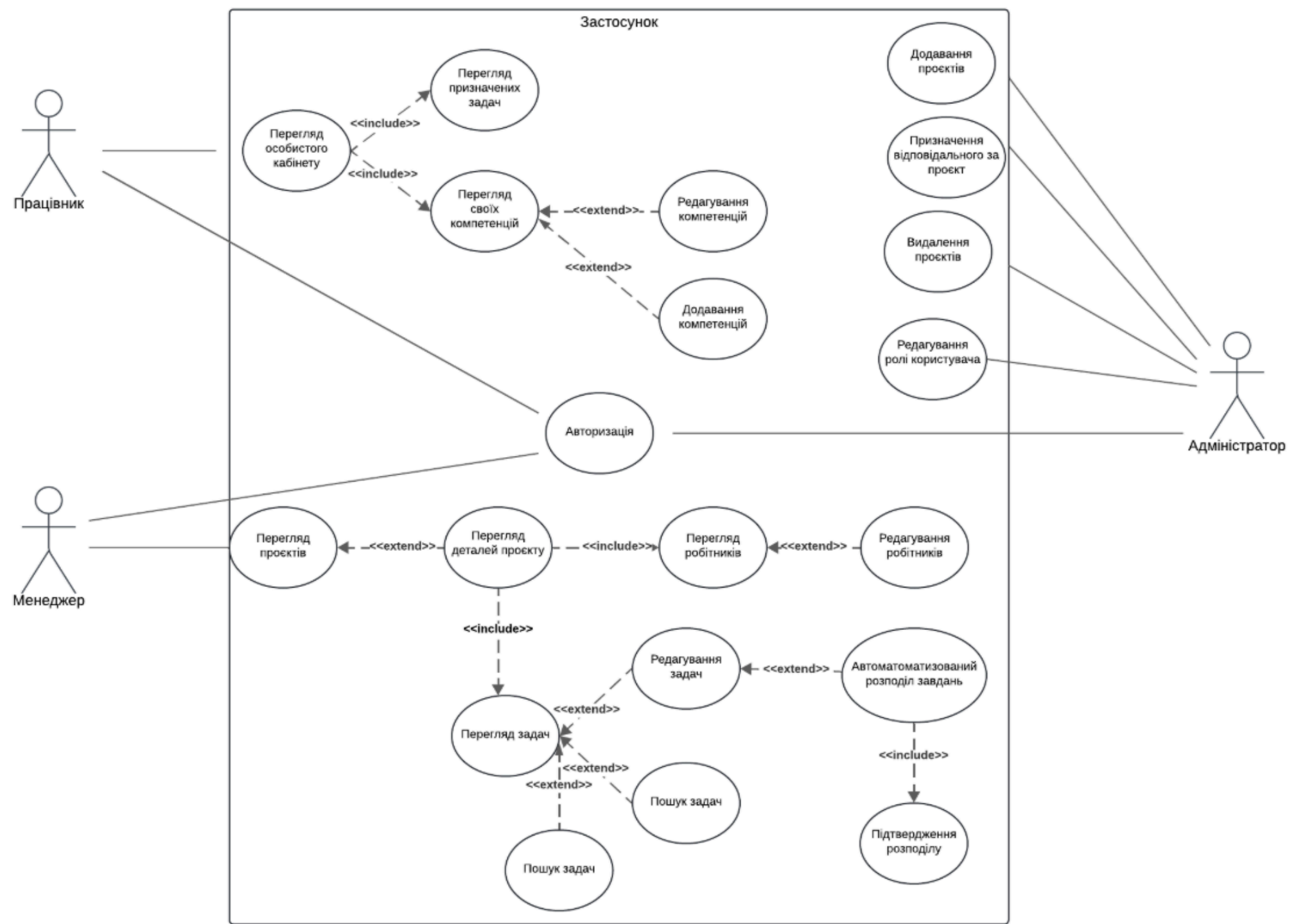
Нормоконтроль:

\_\_\_\_\_ Олександр СТЕЛЬМАХ

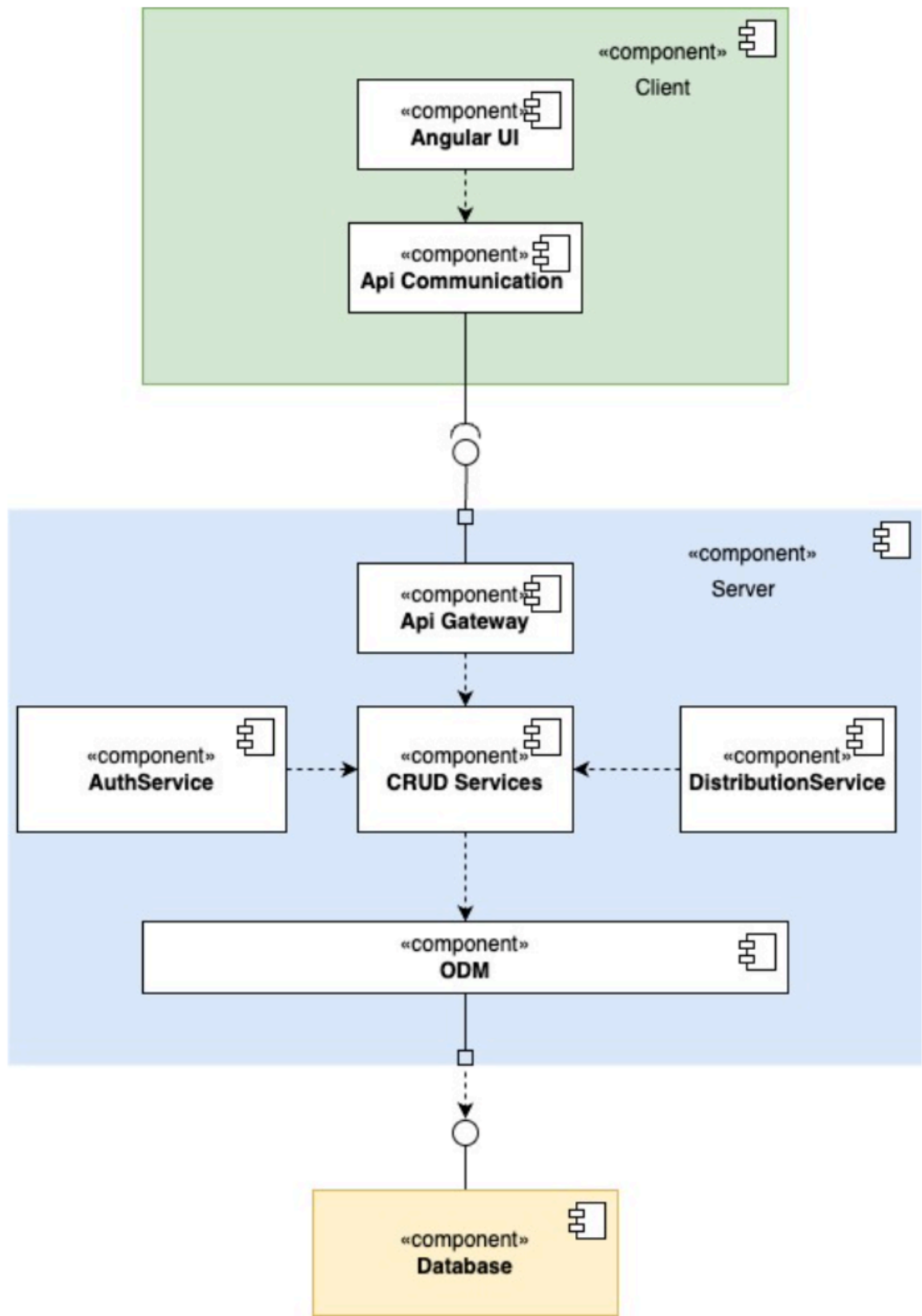
Виконавець:

\_\_\_\_\_ Анастасія МАСЛОВА

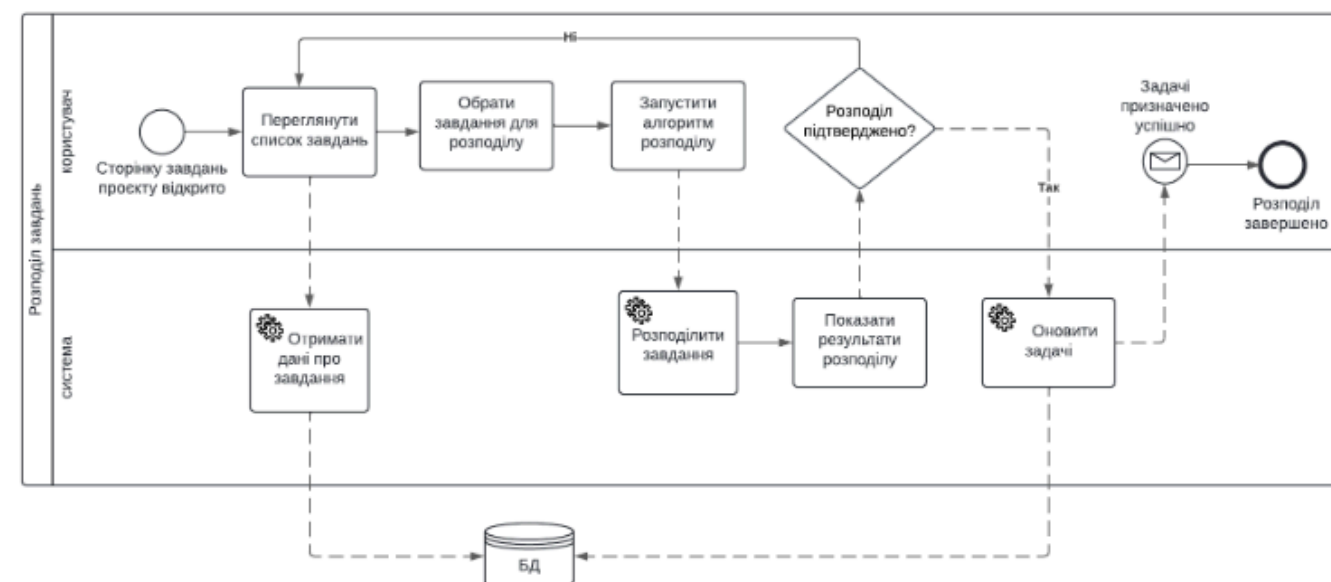
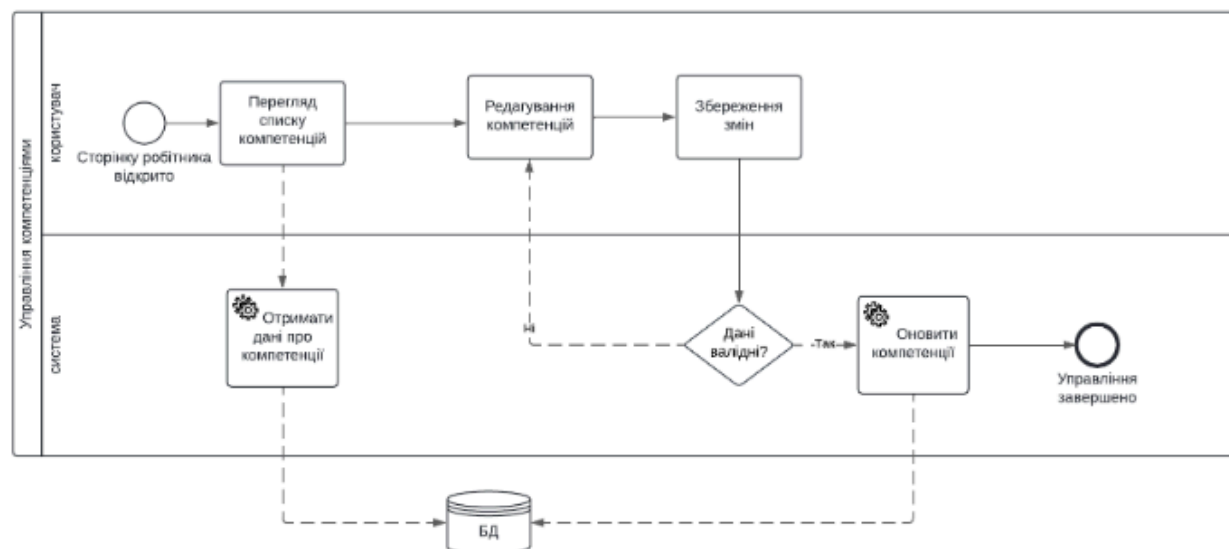
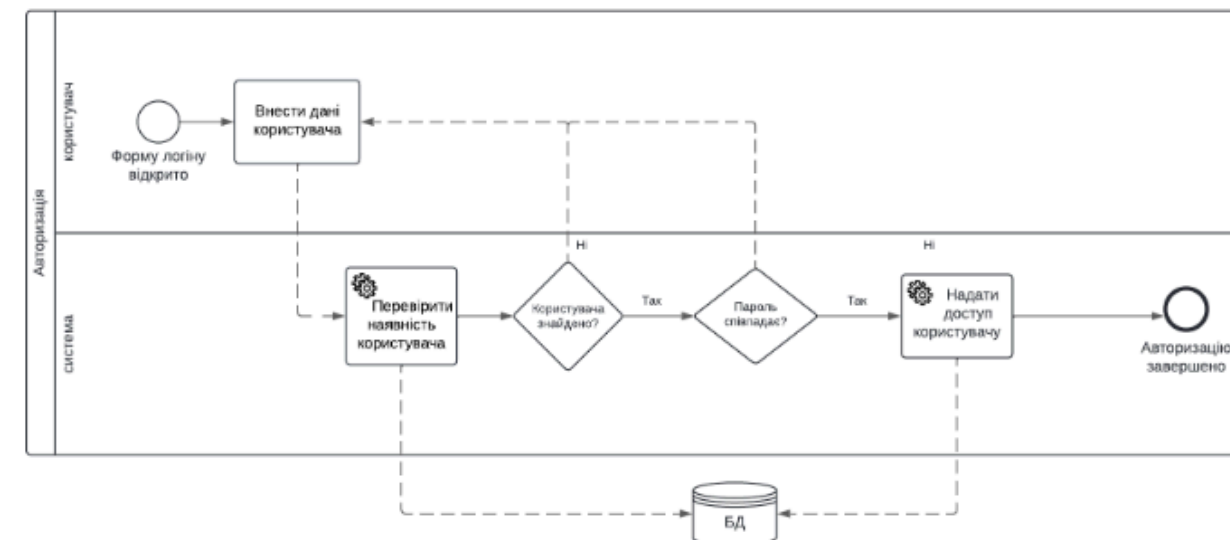
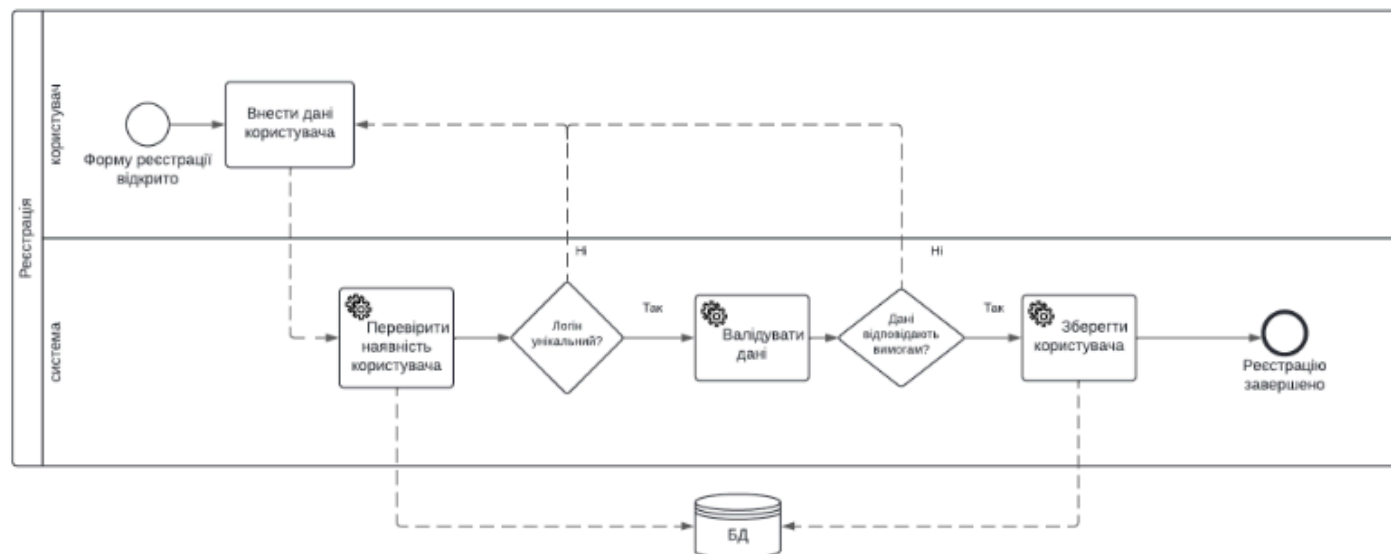
Київ – 2025



|           |      |                |        |      |                                                                                                        |                                                       |  |  |      |           |         |  |
|-----------|------|----------------|--------|------|--------------------------------------------------------------------------------------------------------|-------------------------------------------------------|--|--|------|-----------|---------|--|
|           |      |                |        |      | КПІ.ІП-31108.045440.06.99.CCB                                                                          |                                                       |  |  |      |           |         |  |
|           |      |                |        |      | Схема структурна варіантів<br>використань                                                              | Літера                                                |  |  | Маса |           | Масштаб |  |
| Зм.       | Арк. | № документа    | Підпис | Дата |                                                                                                        |                                                       |  |  |      |           |         |  |
| Розробив  |      | Маслова А. О.  |        |      |                                                                                                        |                                                       |  |  |      |           |         |  |
| Перевірів |      | Олійник Ю. О.  |        |      |                                                                                                        |                                                       |  |  |      |           |         |  |
| Т. контр. |      |                |        |      |                                                                                                        |                                                       |  |  |      |           |         |  |
|           |      |                |        |      |                                                                                                        | Аркуш 1                                               |  |  |      | Аркушів 1 |         |  |
| Н. контр. |      | Стельмах О. П. |        |      | Вебзастосунок для автоматизованого розподіл<br>завдань на основі матриці компетенцій<br>співробітників | КПІ ім.Ігоря Сікорського<br>Кафедра ІПІ<br>гр. ІП-311 |  |  |      |           |         |  |
| Затвердив |      | Жаріков Е.В.   |        |      |                                                                                                        |                                                       |  |  |      |           |         |  |



|           |      |                |       |      |                                                                                                   |                                                       |      |           |
|-----------|------|----------------|-------|------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------|------|-----------|
|           |      |                |       |      | КПІ.ІП-з1108.045440.06.99.CCM                                                                     |                                                       |      |           |
|           |      |                |       |      | Схема структурна компонентів програмного забезпечення                                             | Літера                                                | Маса | Масштаб   |
| Зм.       | Арк. | № докум.       | Підп. | Дата |                                                                                                   |                                                       |      |           |
| Розробив  |      | Маслова А. О.  |       |      |                                                                                                   |                                                       |      |           |
| Перевірів |      | Олійник Ю. О.  |       |      |                                                                                                   |                                                       |      |           |
| Т. контр. |      |                |       |      | Вебзастосунок для автоматизованого розподілу завдань на основі матриці компетенцій співробітників | Аркуш 1                                               |      | Аркушів 1 |
|           |      |                |       |      |                                                                                                   |                                                       |      |           |
| Н. контр. |      | Стельмах О. П. |       |      |                                                                                                   | КПІ ім.Ігоря Сікорського<br>Кафедра ІПІ<br>гр. ІП-з11 |      |           |
| Затвердив |      | Жаріков Е.В.   |       |      |                                                                                                   |                                                       |      |           |



|           |      |                |        |      |                                                                                                   |  |  |  |
|-----------|------|----------------|--------|------|---------------------------------------------------------------------------------------------------|--|--|--|
|           |      |                |        |      | КП.ІП-31108.045440.06.99.ССД                                                                      |  |  |  |
|           |      |                |        |      | Схема структурна діяльності                                                                       |  |  |  |
| Зм.       | Арк. | № документа    | Підпис | Дата |                                                                                                   |  |  |  |
| Розробив  |      | Маслова А. О.  |        |      |                                                                                                   |  |  |  |
| Перевірив |      | Олійник Ю. О.  |        |      |                                                                                                   |  |  |  |
| Т. контр. |      |                |        |      | Вебзастосунок для автоматизованого розподілу завдань на основі матриці компетенцій співробітників |  |  |  |
| Н. контр. |      | Стельмах О. П. |        |      |                                                                                                   |  |  |  |
| Затвердив |      | Жаріков Е.В.   |        |      |                                                                                                   |  |  |  |
|           |      |                |        |      |                                                                                                   |  |  |  |
|           |      |                |        |      | Літера                                                                                            |  |  |  |
|           |      |                |        |      | Маса                                                                                              |  |  |  |
|           |      |                |        |      | Масштаб                                                                                           |  |  |  |
|           |      |                |        |      | Аркуш 1                                                                                           |  |  |  |
|           |      |                |        |      | Аркушів 1                                                                                         |  |  |  |
|           |      |                |        |      | КПІ ім.Ігоря Сікорського                                                                          |  |  |  |
|           |      |                |        |      | Кафедра ІПІ                                                                                       |  |  |  |
|           |      |                |        |      | гр. ІП-311                                                                                        |  |  |  |