

БАКАЛАВРСЬКА РОБОТА

БР. ІІ - 27.00.00.000 ІІЗ

Група ІІ-21-2

Кулаков Максим

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Кулаков Максим Юрійович

(прізвище, ім'я, по батькові)

УДК 004
(індекс)

БАКАЛАВРСЬКА РОБОТА

Методологія масштабування системи керування задачами розробки по

рівню команди

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Кулаков М.Ю.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Дмитрик Тарас Богданович, асистент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Теоретичні основи керування задачами та масштабування в розробці ПЗ	02.05.2025	виконано
2	Аналіз вимог для розробки система управління завданнями	11.05.2025	виконано
3	Представлення існуючих підходів для масштабування керування задачами на рівні команди	21.05.2025	виконано
4	Проектування ER діаграми	28.05.2025	виконано
5	Програмна реалізація методів масштабування системи керування задачами	03.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 79 сторінок, 48 рисунків, список використаних джерел із 35 найменуваннями.

Метою даної роботи є розробка масштабованої системи управління задачами для командної розробки програмного забезпечення з урахуванням сучасних методологій Agile та відповідних фреймворків масштабування.

Об'єкт дослідження - процеси управління завданнями та проектами в розподілених командах розробки програмного забезпечення.

Предмет дослідження - методи та засоби реалізації масштабованої системи управління задачами на основі Agile-фреймворків.

В першому розділі проаналізовано методології управління завданнями та основи масштабування Agile з описом відповідних фреймворків, що стали основою концепції розробки.

В другому розділі досліджено сучасні підходи до управління проектами й завданнями, визначено вимоги до системи та побудовано логічну модель майбутньої реалізації.

В третьому розділі розроблено архітектуру, інтерфейси та функціональні модулі системи управління задачами, проведено тестування, що підтвердило її функціональність і масштабованість.

Висновок: розроблено алгоритм автентифікації, ER-діаграму, інтерфейсну частину за ролями та реалізовано функціональні модулі, що підтримують масштабування системи відповідно до зростання кількості команд.

КЛЮЧОВІ СЛОВА: AGILE, МАСШТАБУВАННЯ, СИСТЕМА УПРАВЛІННЯ ЗАДАЧАМИ, КОМАНДА РОЗРОБКИ, SAFE, SCRUM@SCALE, ПРОЄКТНЕ УПРАВЛІННЯ, АРХІТЕКТУРА СИСТЕМИ, ER-ДІАГРАМА, МОДУЛЬНА СТРУКТУРА

ANNOTATION

The bachelor's thesis contains 79 pages, 48 figures, a list of used sources with 35 names.

The purpose of this work is to develop a scalable task management system for team software development taking into account modern Agile methodologies and relevant scaling frameworks.

The object of the study is the processes of task and project management in distributed software development teams.

The subject of the study is methods and means of implementing a scalable task management system based on Agile frameworks.

The first section analyzes task management methodologies and the basics of Agile scaling with a description of the relevant frameworks that became the basis of the development concept.

The second section examines modern approaches to project and task management, identifies system requirements, and builds a logical model for future implementation.

In the third section, the architecture, interfaces, and functional modules of the task management system are developed, and testing is conducted, which confirms its functionality and scalability.

Conclusion: an authentication algorithm, ER diagram, role-based interface part have been developed, and functional modules have been implemented that support system scaling in accordance with the growth in the number of teams.

KEYWORDS: AGILE, SCALING, TASK MANAGEMENT SYSTEM, DEVELOPMENT TEAM, SAFE, SCRUM@SCALE, PROJECT MANAGEMENT, SYSTEM ARCHITECTURE, ER DIAGRAM, MODULAR STRUCTURE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	9
ВСТУП	10
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ КЕРУВАННЯ ЗАДАЧАМИ ТА МАСШТАБУВАННЯ В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
1.1. Особливості розробка системи управління завданнями для оптимізації процесів організації	13
1.2. Аналіз вимог для розробки система управління завданнями	14
1.3. Методологія розробки системи	16
1.4. Огляд існуючих підходів та фреймворків для масштабування Agile ...	17
1.4.1. Scaled Agile Framework (SAFe)	18
1.4.2. Large-Scale Scrum (LeSS).....	19
1.4.3. Фреймворк Scrum@Scale.....	21
РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ ІСНУЮЧИХ ПІДХОДІВ ТА ІНСТРУМЕНТІВ ДЛЯ МАСШТАБУВАННЯ КЕРУВАННЯ ЗАДАЧАМИ НА РІВНІ КОМАНДИ	24
2.1. Основи управління проектами та завданнями в інженерії програмного забезпечення	24
2.1.1. Передумови	24
2.1.2. Управління проектами	25
2.1.3 Управління завданнями	25
2.2. Значення управління завданнями для ефективності функціонування організацій	27

					БР.ІІ – 27.00.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розроб.		Кулаков М.Ю.			Методологія масштабування системи керування задачами розробки по рівню команди Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
Перевір.		Дмитрик Т.Б.					6	
Реценз.						ІФНТУНГ ІІІ-21-2		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В.В.						

2.3. Представлення функцій системи управління задачами по рівню команди.....	30
2.4. Специфікація вимог.....	33
2.4.1. Нефункціональні вимоги (вимоги до продуктивності).....	34
2.4.2. Функціональні вимоги.....	34
2.5. Логічна модель системи.....	35
2.6. Проектування логічної діаграми верхнього рівня.....	37
2.7. Розробка алгоритму процесу автентифікації користувача в системі....	40

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ МАСШТАБУВАННЯ СИСТЕМИ КЕРУВАННЯ ЗАДАЧАМИ РОЗРОБКИ ПО РІВНЮ

КОМАНДИ.....	43
3.1. Архітектура та технічні аспекти запропонованої системи управління задачами.....	43
3.2. Проектування ER діаграми.....	45
3.3. Діаграми Ганта та PERT як основні інструменти планування та візуалізації проектів.....	48
3.4. Архітектура інтерфейсу користувача та функціональність системи управління завданнями за ролями.....	50
3.5. Функціональні аспекти інтерфейсу управління проектами та завданнями.....	56
3.5.1. Інтерфейс управління проектами.....	56
3.5.2. Інтерфейс управління задачами.....	58
3.6. Інтерфейс спеціалізованих модулів системи управління завданнями: облік часу та фінансове управління.....	61
3.6.1. Модуль обліку робочого часу.....	61
3.6.2. Модуль візуалізації розкладу.....	62
3.6.3. Модуль управління задачами.....	63
3.6.4. Модуль управління фінансами (Рахунки-фактури).....	63

3.7. Інтерфейс функціональності для адміністратора та клієнта	66
3.7.1. Модуль управління клієнтами	66
3.7.2. Модуль звітності	67
3.7.3. Модуль управління користувачами.....	69
3.7.4. Модуль візуалізації задач (Календар)	69
3.7.5. Модуль конфігурації робочого простору	70
3.7.6. Інтерфейс для ролі клієнта	71
3.8. Тестування системи	72
ВИСНОВКИ.....	75
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	77
БІБЛІОГРАФІЧНА ДОВІДКА	

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ART – Agile Release Train

CPM – Critical Path Method (Метод критичного шляху)

ДПД – Діаграма потоку даних

EAT – Executive Action Team

EMS – Executive MetaScrum

IEEE – Institute of Electrical and Electronics Engineers

LeSS – Large-Scale Scrum

MVB – Minimum Viable Bureaucracy

PDMA – Plan Do Check Access

PERT – Program Evaluation and Review Technique

PMI – Project Management Institute

SAFe – Scaled Agile Framework

Scrum@Scale – Scrum@Scale

TMS – Task Management System

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасна розробка програмного забезпечення характеризується високою динамікою змін, багаторівневою командною структурою та необхідністю швидкого реагування на потреби бізнесу. Зростання кількості проєктів, які реалізуються за участю кількох команд, висуває нові вимоги до систем управління завданнями, особливо в контексті їх масштабування. Традиційні підходи не завжди здатні ефективно підтримувати координацію дій, прозорість процесів та управління залежностями між командами. Саме тому актуальним є розроблення методології масштабування системи управління задачами, що дозволяє адаптувати її до потреб різнорівневих команд і забезпечити цілісність управлінських процесів у межах великомасштабної розробки програмного забезпечення.

Управління задачами є критичним компонентом процесу розробки програмного забезпечення, оскільки воно визначає ефективність планування, розподілу ресурсів, контролю виконання та комунікації між учасниками проєкту. Із впровадженням Agile-методологій, які акцентують увагу на гнучкості, самоорганізації команд та ітеративності процесів, виникла потреба у створенні систем управління задачами, які можуть масштабуватись відповідно до структури та розміру організації.

Актуальність роботи

Актуальність теми зумовлена зростаючим попитом на ефективні засоби управління завданнями у великих розробницьких командах, що працюють за Agile-методологіями. З появою нових викликів, таких як децентралізація команд, гібридні моделі співпраці, висока конкуренція та швидкий темп змін у вимогах до ПЗ, особливого значення набуває масштабованість систем управління. Попри велику кількість існуючих інструментів, більшість з них мають обмежену гнучкість, інтеграцію або не забезпечують повноцінну підтримку ролей, звітності, тайм-менеджменту й фінансового обліку. Саме

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

тому розробка кастомізованої, функціонально насиченої системи управління задачами із підтримкою масштабування командної роботи є своєчасною та затребуваною.

Однак в умовах багатоконандної розробки постають нові виклики: координація між командами, управління міжпроектними залежностями, уніфікація інструментів контролю та адаптація до різних ролей користувачів (розробників, менеджерів, клієнтів). Сучасні фреймворки для масштабування Agile, такі як SAFe, LeSS, Scrum@Scale, пропонують методологічну базу для вирішення цих проблем, однак їх ефективне застосування потребує відповідного технічного інструментарію — системи, здатної підтримувати гнучку структуру, розмежування доступів та автоматизацію ключових процесів.

Метою даної роботи є розробка масштабованої системи управління задачами для командної розробки програмного забезпечення з урахуванням сучасних методологій Agile та відповідних фреймворків масштабування.

В ході дослідження було проаналізовано теоретичні засади, вимоги до таких систем, запропоновано логічну та функціональну архітектуру, а також реалізовано інтерфейси для різних ролей користувачів. Особливу увагу приділено створенню модульної структури, яка дозволяє забезпечити адаптивність системи, її розширюваність та відповідність принципам Agile. Запропонований підхід сприяє ефективному управлінню як окремими завданнями, так і загальною динамікою розробки у великомасштабних проєктах.

Завдання дослідження

1. Провести аналіз теоретичних основ управління завданнями та масштабування процесів у розробці ПЗ.
2. Дослідити існуючі методології та фреймворки масштабування Agile (SAFe, LeSS, Scrum@Scale).
3. Сформувати вимоги до системи управління задачами.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

4. Розробити логічну модель та архітектуру системи.

5. Реалізувати програмну систему з підтримкою ключових функцій масштабованого управління.

6. Провести тестування системи на відповідність вимогам.

Об'єкт дослідження - процеси управління завданнями та проектами в розподілених командах розробки програмного забезпечення.

Предмет дослідження - методи та засоби реалізації масштабованої системи управління задачами на основі Agile-фреймворків.

Методи дослідження:

- аналіз і синтез літературних джерел;
- порівняльний аналіз фреймворків масштабування Agile;
- методи структурного та об'єктно-орієнтованого проектування;
- UML-моделювання;
- методи програмної реалізації (фреймворки, API);
- модульне та інтеграційне тестування.

Наукова новизна

Наукова новизна полягає в розробці та реалізації комплексної системи управління задачами, що інтегрує сучасні фреймворки масштабування Agile, підтримує багаторівневу взаємодію команд та забезпечує гнучку архітектуру з орієнтацією на розширюваність і адаптивність.

Практичне застосування

Результати роботи можуть бути використані в ІТ-компаніях для підвищення ефективності управління завданнями, автоматизації робочих процесів, інтеграції з інструментами планування та оптимізації взаємодії між ролями всередині проектних команд.

Бакалаврська робота містить 79 сторінок, 48 рисунків, 3 розділи список використаних джерел із 35 найменуваннями.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ КЕРУВАННЯ ЗАДАЧАМИ ТА МАСШТАБУВАННЯ В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Особливості розробка системи управління завданнями для оптимізації процесів організації

Система управління завданнями (Task Management System, TMS) є спеціалізованим веб-додатком, призначеним для оптимізації процесів організації, призначення та моніторингу виконання завдань у середовищі малих та середніх підприємств.

Архітектура системи включає низку функціональних підмодулів, що забезпечують повний життєвий цикл управління задачами та проектами.



Рисунок 1.1 - Основна функціональність системи управління задачами

Ключова функціональність охоплює (рис. 1.1):

- Можливості коментування завдань для забезпечення ефективної комунікації між членами команди.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

- Механізми завантаження та скачування файлів, інтегровані із завданнями, для централізованого зберігання пов'язаних документів.
- Функціонал призначення завдань конкретним співробітникам.
- Модулі для управління проектами та завданнями, включаючи їх створення, редагування та архівування.
- Інструменти для керування базами даних співробітників та користувачів системи.

Однією з ключових переваг розробленої системи є її ергономічний інтерфейс користувача, який сприяє інтуїтивно зрозумілій взаємодії адміністраторів та кінцевих користувачів, забезпечуючи ефективне виконання різноманітних операцій в онлайн-режимі. Система оснащена інформативною панеллю моніторингу (dashboard), яка візуалізує актуальний статус проектів та активності користувачів за допомогою графічних елементів, надаючи комплексну картину стану робіт на одному екрані. Вбудована функція пошуку забезпечує швидкий доступ до необхідної інформації про завдання та проекти. Для візуалізації робочого процесу та відстеження прогресу проектів в системі імплементовано принципи методології Kanban. Підтримується ієрархічна структура управління, що дозволяє визначати ключові етапи (віхи) та декомпозицію складних задач на підзавдання, забезпечуючи структурований підхід до моніторингу виконання.

Додаткова функціональність системи включає механізми управління правами доступу для підтримки множинних клієнтських облікових записів, можливості створення унікальних нотаток для внутрішнього використання та генерації індивідуалізованих рахунків-фактур.

1.2. Аналіз вимог для розробки система управління завданнями

Пропонована система управління завданнями являє собою програмний інструмент, функціонал якого спрямований на сприяння організаціям в

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

ефективному завершенні проектів шляхом оптимізації процесів організації та пріоритезації пов'язаних завдань, забезпечуючи при цьому дотримання встановлених термінів. Даний інструмент може застосовуватися як індивідуальними користувачами, так і в межах команд або організацій в цілому. Використання системи управління завданнями дозволяє ефективно керувати завданнями, здійснювати моніторинг витраченого часу та налагоджувати взаємодію між різними командами. Ефективність діяльності організації значною мірою корелює з адекватністю та продуктивністю її системи управління завданнями. Важливу роль в управлінні проектами та координації роботи персоналу відіграє менеджер завдань.

Запропонована система розробляється з метою оптимізації комунікаційних потоків між співробітниками та керівним складом. Система забезпечує оперативності розподілу завдань, що сприяє мінімізації кількості та тривалості формальних зустрічей. Керівники отримують інструменти для періодичного надання співробітникам детальної інформації щодо поточних завдань. Процес управління завданнями реалізовано з акцентом на зручність та ефективність для обох сторін взаємодії. Контроль та моніторинг виконання завдань здійснюється на рівні Координатора системи (або інша роль, визначена в структурі), який відповідає за підтримку цілісності бази даних та загальну продуктивність системи. Координатор системи володіє відповідними повноваженнями для нагляду за функціонуванням всієї системи.

Мета проекту розробки:

1. Провести порівняльний аналіз функціональних можливостей та обмежень систем управління завданнями, що використовуються у різних бізнес-організаціях, та розробити прототип цільової системи.
2. Здійснити розробку програмного забезпечення системи управління завданнями згідно із затвердженою специфікацією.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

3. Виконати верифікацію та валідацію розробленого програмного забезпечення з метою підтвердження відповідності функціональним вимогам та досягнення повної функціональності.

Очікувані результати проекту:

1. Формування аналітичного звіту, що містить результати порівняльного дослідження функцій та обмежень систем управління завданнями, що застосовуються у бізнес-середовищі.

2. Розробка повнофункціонального програмного забезпечення системи управління завданнями, адаптованого для потреб малих та середніх підприємств, що забезпечує ефективне призначення та управління робочими завданнями.

1.3. Методологія розробки системи

Розробка системи управління завданнями здійснювалася відповідно до визначеної методології, яка базувалася на ключових етапах життєвого циклу розробки програмного забезпечення (ЖЦРПЗ). Структура методології включала наступні послідовні кроки:

1. Дослідження та аналіз.

Проведення поглибленого літературного дослідження та порівняльного аналізу існуючих систем управління завданнями з метою ідентифікації їхніх функціональних можливостей, потенційних сфер застосування та виявлення наявних прогалин чи обмежень.

2. Збір та формалізація вимог.

Детальне визначення функціональних та нефункціональних вимог до цільової системи на основі результатів етапу дослідження та аналізу потреб малих і середніх підприємств.

3. Проектування та концептуалізація.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Розробка архітектури системи, проектування структури бази даних та визначення основних модулів. Розгляд та оцінка різних концепцій дизайну користувацького інтерфейсу з подальшим вибором оптимального рішення. Створення детального проекту інтерфейсу користувача (UI design).

4. Реалізація (Розробка програмного забезпечення).

Безпосереднє написання програмного коду системи відповідно до затвердженої архітектури та специфікацій вимог, включаючи розробку клієнтської та серверної частин. На цьому етапі також може передбачатися створення прототипу.

5. Тестування. Проведення комплексного тестування розробленого програмного забезпечення з метою виявлення та усунення дефектів, а також підтвердження відповідності системи визначеним вимогам та досягнення повної функціональності.

Такий підхід дотримується класичних принципів ЖЦРПЗ, забезпечуючи структурований процес від початкового дослідження до фінального тестування розробленого програмного забезпечення системи управління задачами.

1.4. Огляд існуючих підходів та фреймворків для масштабування Agile

Масштабування гнучких (Agile) методологій стає критично важливим для організацій, що зростають, оскільки прості застосування Scrum чи Kanban на рівні однієї команди виявляються недостатніми для координації роботи багатьох команд над складними продуктами чи проектами. Для вирішення цієї задачі розроблено низку фреймворків та підходів, серед яких найбільш відомими є Scaled Agile Framework (SAFe), Large-Scale Scrum (LeSS) та Scrum@Scale. Кожен з них пропонує своє бачення та інструменти для ефективного масштабування Agile-практик на рівні підприємства.

					БР.ІІІ – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Основна одиниця в SAFe на програмному рівні, що об'єднує 5-12 Agile-команд (50-125 осіб), які спільно планують, розробляють та випускають ціннісне інкремент рішення.

- Program Increment (PI).

Фіксований проміжок часу (зазвичай 8-12 тижнів), протягом якого ART працює над досягненням поставлених цілей програми. Завершується PI-плануванням та демонстрацією результатів.

- Ціннісні потоки (Value Streams).

SAFe фокусується на організації роботи навколо потоків створення цінності, що охоплюють усі кроки від ідеї до поставки цінності кінцевому споживачеві.

- Ролі.

Вводить специфічні ролі для масштабування, такі як Release Train Engineer (RTE), Product Manager, System Architect, Solution Train Engineer (STE) тощо, на додаток до стандартних Scrum-ролей.

Переваги SAFe:

- Високий рівень структурованості та деталізації, що полегшує впровадження у великих та складних організаціях.

- Забезпечує вирівнювання цілей на всіх рівнях організації (від стратегії портфеля до завдань команди).

- Пропонує комплексний набір практик для різних аспектів розробки (DevOps, Lean Portfolio Management тощо).

- Широко визнаний та має велику кількість навчальних ресурсів.

1.4.2. Large-Scale Scrum (LeSS)

LeSS є більш мінімалістичним фреймворком для масштабування Scrum. Він розширює стандартний Scrum-фреймворк для застосування до кількох команд, що працюють над одним продуктом. LeSS прагне застосувати принципи Scrum з мінімально можливою кількістю додаткових правил, ролей

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

та артефактів. Існує дві конфігурації: Basic LeSS (для 2-8 команд) та LeSS Huge (для понад 8 команд).

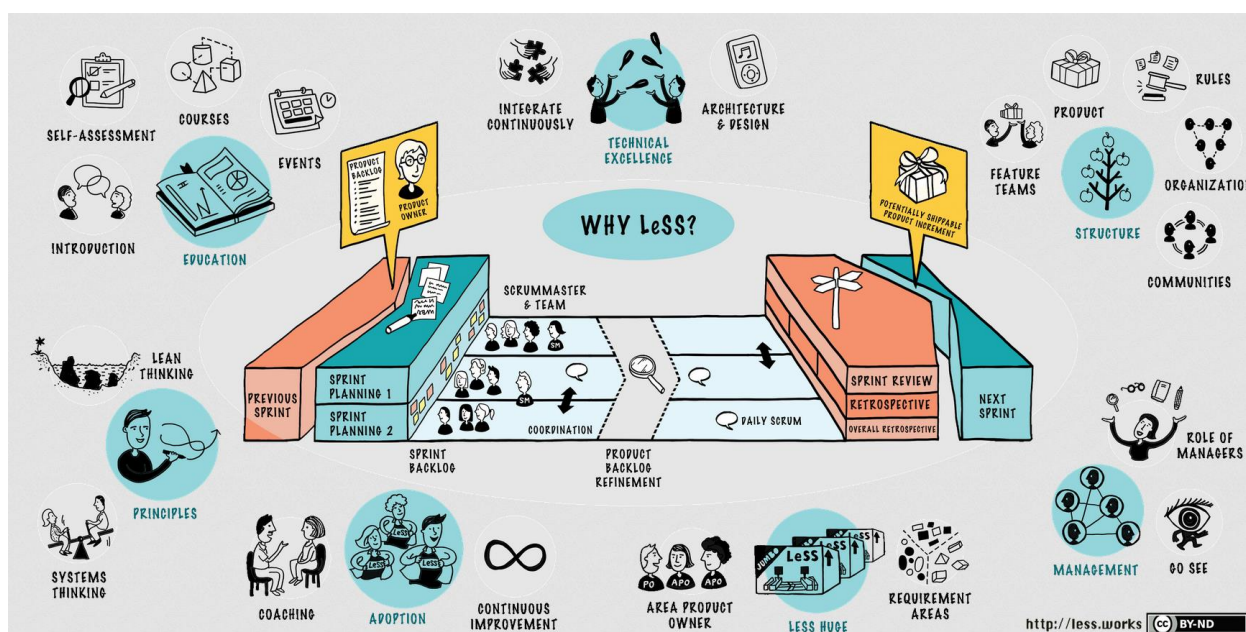


Рисунок 1.3 – Огляд Large Scale Scrum

Ключові принципи та компоненти LeSS:

- Scrum як основа.

LeSS максимально зберігає ролі (один Власник Продукту для всіх команд, Scrum-майстри, команди розробки), артефакти (один Product Backlog для всіх команд, Team Backlog) та події (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective) зі стандартного Scrum.

- Один Product Backlog, один Власник Продукту.

Ключовий принцип, що забезпечує єдине бачення продукту та пріоритетів для всіх команд.

- Команди, орієнтовані на клієнта/продукт.

Команди є крос-функціональними та фокусуються на створенні цінності для клієнта.

- Масштабування шляхом "демасштабування" (De-scaling).

LeSS заохочує спрощення організаційної структури, усунення зайвих ролей та процесів.

- Дослідження (Exploration).

Наголошує на важливості дослідження та навчання через експерименти.

Переваги LeSS:

- Зберігає гнучкість та простоту базового Scrum.
- Сприяє фокусу на продукті та клієнті завдяки єдиному Product Backlog та Власнику Продукту.
- Менш обтяжливий з точки зору ролей та процесів порівняно з SAFe.
- Заохочує організаційні зміни для підтримки гнучкості.

1.4.3. Фреймворк Scrum@Scale

Scrum@Scale є фреймворком, розробленим Джеффом Сазерлендом, одним із засновників Scrum. Він базується на масштабуванні фундаментальних елементів Scrum – циклів Scrum Master та Product Owner – та їхній взаємодії. Scrum@Scale описує архітектуру масштабованого Scrum, яка дозволяє організації динамічно адаптуватися до мінливих умов.

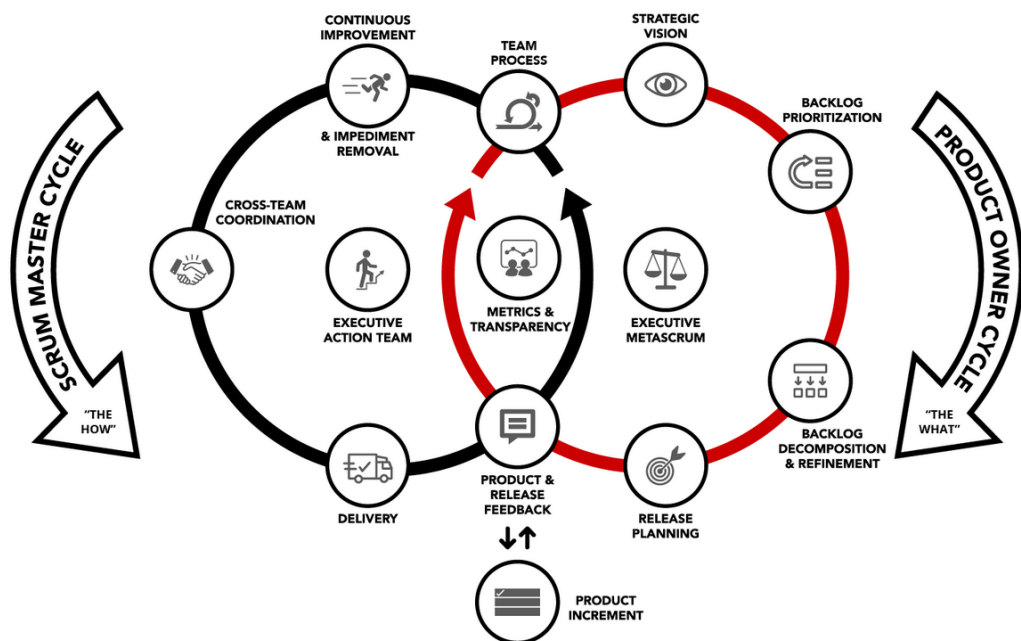


Рисунок 1.4 - Фреймворк Scrum@Scale

Ключові принципи та компоненти Scrum@Scale:

- Scrum Master Cycle та Product Owner Cycle.

Фреймворк визначає два основні цикли: цикл Скрам-майстра (фокус на оптимізації процесів та усуненні перешкод) та цикл Власника Продукту (фокус на визначенні та поставці найбільшої цінності).

- Scrum of Scrums (SoS).

Механізм координації для кількох команд. В Scrum@Scale цей концепт розширюється до Scrum of Scrum of Scrums тощо, створюючи ієрархію (або мережу) SoS.

- Executive Action Team (EAT).

Команда керівників, відповідальна за усунення системних перешкод та трансформацію організації.

- Executive MetaScrum (EMS).

Команда керівників, що відповідає за пріоритезацію на рівні організації та синхронізацію Product Backlog.

- Мінімально життєздатна бюрократія (Minimum Viable Bureaucracy - MVB).

Принцип, що полягає у створенні найменшої необхідної структури та процесів для ефективного функціонування.

- Масштабування через мережу.

На відміну від жорстких ієрархій, Scrum@Scale передбачає масштабування через гнучкі мережі команд.

Переваги Scrum@Scale:

- Глибоко інтегрований з базовим Scrum, що полегшує перехід для команд, які вже використовують Scrum.

- Гнучкий та адаптивний, може застосовуватися в різних організаційних структурах.

- Наголошує на ролі керівництва (через EAT та EMS) у підтримці та масштабуванні Agile.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

- Спрямований на мінімізацію зайвої бюрократії.

Отже, SAFe, LeSS та Scrum@Scale пропонують різні шляхи для масштабування Agile. SAFe є найбільш всеохоплюючим і підходить для великих, складних організацій, яким потрібна детальна структура та керівництво. LeSS є більш легким фреймворком, що ідеально підходить для організацій, які хочуть масштабувати чистий Scrum з мінімальною кількістю змін, особливо коли кілька команд працюють над одним продуктом. Scrum@Scale є гнучким, мережево-орієнтованим підходом, який масштабує самі цикли Scrum і потребує активної участі керівництва для усунення системних перешкод. Вибір фреймворку залежить від розміру організації, її культури, рівня зрілості Agile-практик та специфіки продукту чи проекту, над яким працюють команди.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ ІСНУЮЧИХ ПІДХОДІВ ТА ІНСТРУМЕНТІВ ДЛЯ МАСШТАБУВАННЯ КЕРУВАННЯ ЗАДАЧАМИ НА РІВНІ КОМАНДИ

2.1. Основи управління проектами та завданнями в інженерії програмного забезпечення

2.1.1. Передумови

Після програмної кризи, що спостерігалася на початку 1970-х років, у професійній спільноті сформувалося усвідомлення критичного значення інженерії програмного забезпечення як дисципліни, що надає набір методів, інструментів та процедур для створення якісного програмного забезпечення в умовах обмежених бюджетів та часових рамок. Незважаючи на це, згідно з виданням звіту CHAOS за 2024 рік, лише 37% усіх проектів у сфері інформаційних технологій (ІТ) були визнані успішними, тобто такими, що були завершені вчасно, в межах бюджету та з реалізацією усіх передбачених функцій та можливостей. За результатами дослідження компанії ІВМ щодо причин невдач ІТ-проектів, неякісне управління проектами безпосередньо спричинило 54% таких випадків. У зв'язку з цим, провідні професійні організації, такі як Асоціація обчислювальної техніки (АСМ) та IEEE Computer Society, послідовно підкреслюють важливість інтеграції навчання з управління ІТ-проектами до освітніх програм для студентів відповідних спеціальностей у рамках своїх спільних робочих груп.

Порівняно з випускниками інших дисциплін, зокрема медицини чи авіації, фахівці у сфері ІТ часто розпочинають професійну діяльність із недостатнім рівнем практичних навичок, релевантних реальним проектам. Це зумовлює необхідність набуття досвіду в умовах реальних робочих завдань, де наслідки неефективного планування або помилкових рішень можуть призвести до зриву проекту або значних фінансових втрат [4].

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

2.1.2. Управління проектами

Галузь управління проектами характеризується постійним розвитком. Зростаюча стандартизація процесів, удосконалення методик та активний розвиток спеціалізованого програмного забезпечення та додатків сприяють перетворенню управління проектами з "мистецтва" на більш науково обґрунтовану дисципліну. Значним досягненням у цій галузі є концепція Системи загальної вартості управління (Total Cost Management, TCM). Відповідно до визначення AACE International, TCM означає "системний підхід до контролю вартості протягом життєвого циклу будь-якої компанії, програми, об'єкта, проекту, продукту або послуги". В контексті реалізації проекту, TCM представляє собою сукупність практик та процесів, які організація застосовує для контролю витрат протягом життєвого циклу проекту. Підхід TCM базується на циклі Plan Do Check Access (PDMA) [5].

Традиційне визначення проекту, згідно з Інститутом управління проектами (2008), визначає його як «тимчасове починання, спрямоване на створення унікального продукту, послуги або результату». Проект також може бути розглянутий як чітко визначений набір взаємопов'язаних завдань, які мають бути завершені для досягнення встановлених цілей проекту. У типовому проекті виконання багатьох завдань відбувається паралельно. Важливою характеристикою проектів є наявність залежностей між завданнями, які часто визначають умови, за яких виконання одного завдання має бути завершено до початку іншого [6].

2.1.3 Управління завданнями

Управління завданнями (Task Management), яке не слід плутати з більш широкою дисципліною управління проектами, є практикою, що охоплює процеси виконання окремого завдання від ініціації до завершення. Управління завданнями на рівні команди є складовою частиною управління проектами, що сприяє відстеженню всієї роботи персоналу та організацій в

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

цілому, забезпечуючи виконання всіх елементів роботи. Це є особливо актуальним в умовах високого робочого навантаження або ускладненої комунікації [7]. Ефективне програмне забезпечення для управління завданнями зазвичай включає функціональні можливості для пріоритизації завдань, підтримки співпраці, відображення статусу виконання, інтеграції з календарем та оцінки трудомісткості [8]. Зокрема, часто використовують метод, який використовує концепції матриці завдань та матриці Ейзенхауера для структуризації завдань проекту та їх класифікації за пріоритетністю. Серед поширених інструментів управління завданнями можна виділити Trello, Todo list тощо.

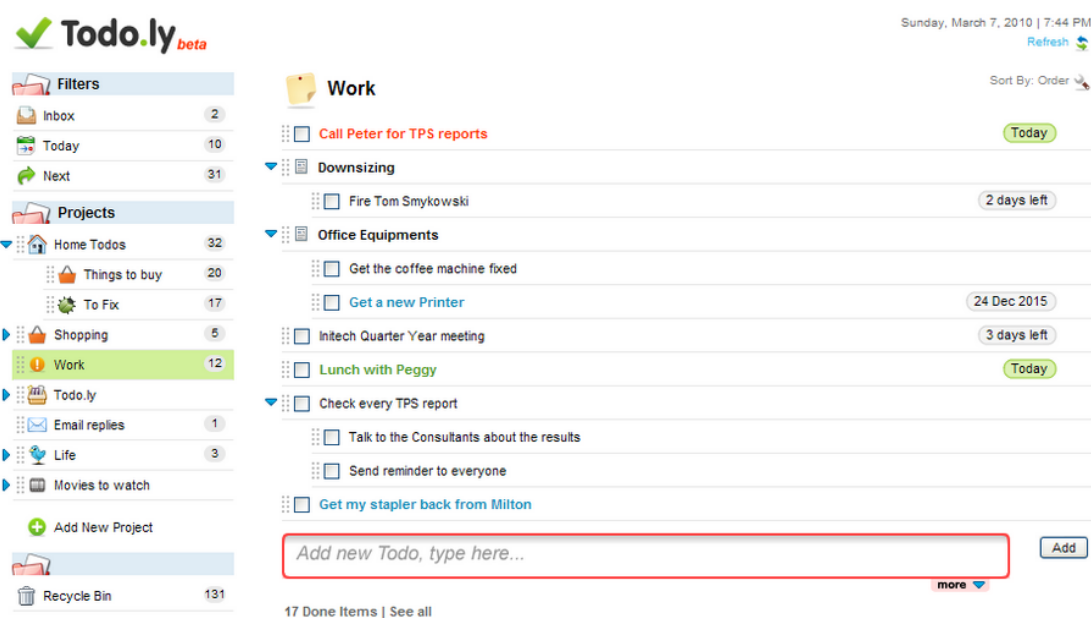


Рисунок 2.1 – Приклад Todo list системи як інструменту управління задачами

Todo list (Список справ) — це простий, але ефективний інструмент управління задачами, який дозволяє організувати, планувати й контролювати виконання завдань. Він широко використовується як в особистому, так і в професійному контексті, зокрема в рамках проектного менеджменту, Agile-методологій, тайм-менеджменту тощо. Todo list є універсальним інструментом управління задачами, який поєднує простоту у

використанні з високою ефективністю. Він сприяє підвищенню організованості, відповідальності й продуктивності як окремих осіб, так і команд.

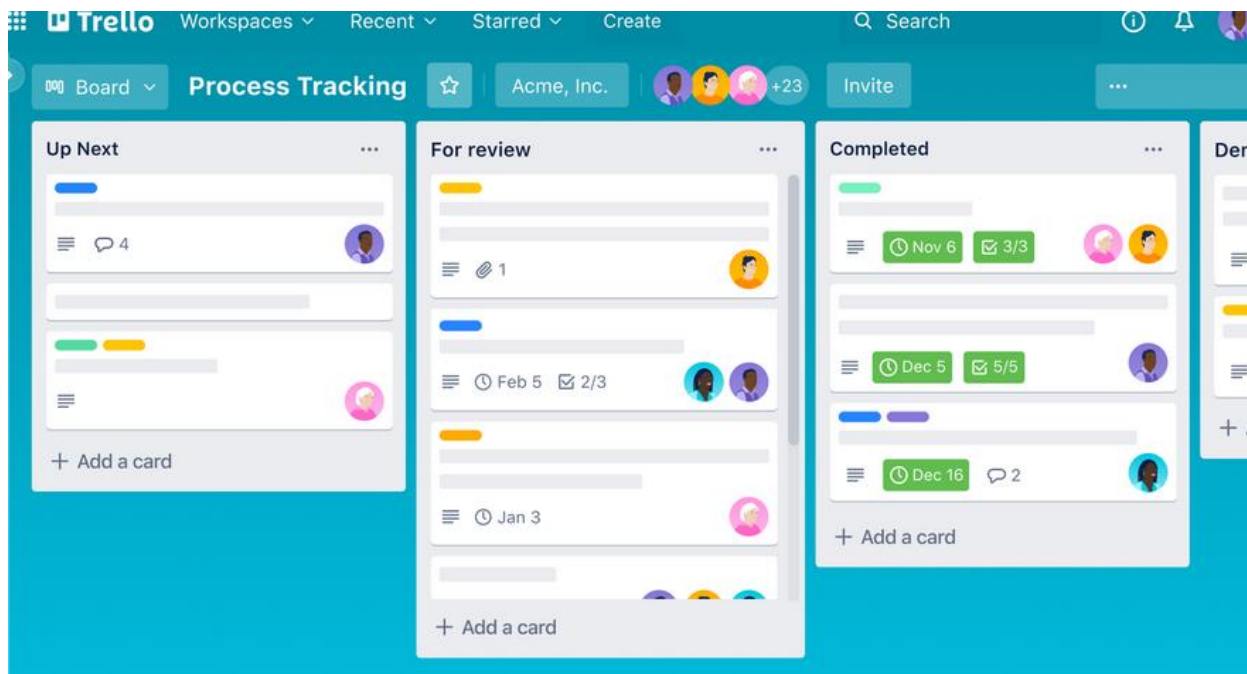


Рисунок 2.2 – Приклад системи Trello як інструменту управління задачами

2.2. Значення управління завданнями для ефективності функціонування організацій

Управління завданнями є критично важливим компонентом загальної системи управління проектами, оскільки воно безпосередньо впливає на результативність виконання проектів, дотримання термінів, оптимальне використання ресурсів та досягнення стратегічних цілей організації. Ефективне управління завданнями дозволяє керівникам проектів підтримувати контроль над процесами, забезпечуючи чітке планування, моніторинг, звітність, аналіз і коригування діяльності на всіх етапах життєвого циклу проекту.

Ключовими складовими управління завданнями є визначення обсягів робіт, встановлення пріоритетів, делегування відповідальності, відстеження

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

прогресу та аналіз ефективності. Для реалізації цих функцій необхідна сучасна система управління завданнями, яка забезпечує доступ до актуальної інформації в режимі реального часу, дозволяє автоматизувати рутинні процеси та сприяє підвищенню прозорості управлінських рішень. Такі системи створюють умови для інтеграції завдань між членами команди, оптимізації комунікації та уникнення дублювання зусиль.

У контексті складних, багатокомпонентних проєктів особливо важливою є здатність керівника проєкту координувати взаємодію спеціалістів із різними професійними компетенціями. Це вимагає не лише високого рівня адаптивності, а й готовності до впровадження новітніх цифрових рішень, які дозволяють гнучко налаштовувати інструменти управління відповідно до специфіки кожного конкретного проєкту [12].

Важливу роль в оптимізації управління завданнями відіграє впровадження Agile-методологій, які ґрунтуються на принципах гнучкості, ітеративності та тісної співпраці в команді. В Agile-середовищі управління завданнями відбувається за допомогою коротких ітерацій (спринтів), щоденних мітингів (Daily Stand-up), візуалізації поточного стану задач (наприклад, на дошках Kanban або Scrum) та постійного зворотного зв'язку. Це дозволяє командам швидко адаптуватися до змін, оперативно вирішувати проблеми й забезпечувати безперервне постачання цінності замовнику.

Управління завданнями в Agile також акцентує увагу на колективній відповідальності, прозорості процесів і максимальному залученні стейкхолдерів до формування вимог. Завдяки цьому підходу суттєво підвищується рівень залученості команди, а процес ухвалення рішень стає більш гнучким і ефективним. Цифрові інструменти, такі як Jira, Trello, Asana, ClickUp, Azure DevOps та інші, надають можливість візуалізувати завдання, контролювати навантаження учасників команди та оцінювати продуктивність за допомогою метрик, таких як velocity, burn-down chart, lead time тощо.

					БР.ІІІ – 27.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

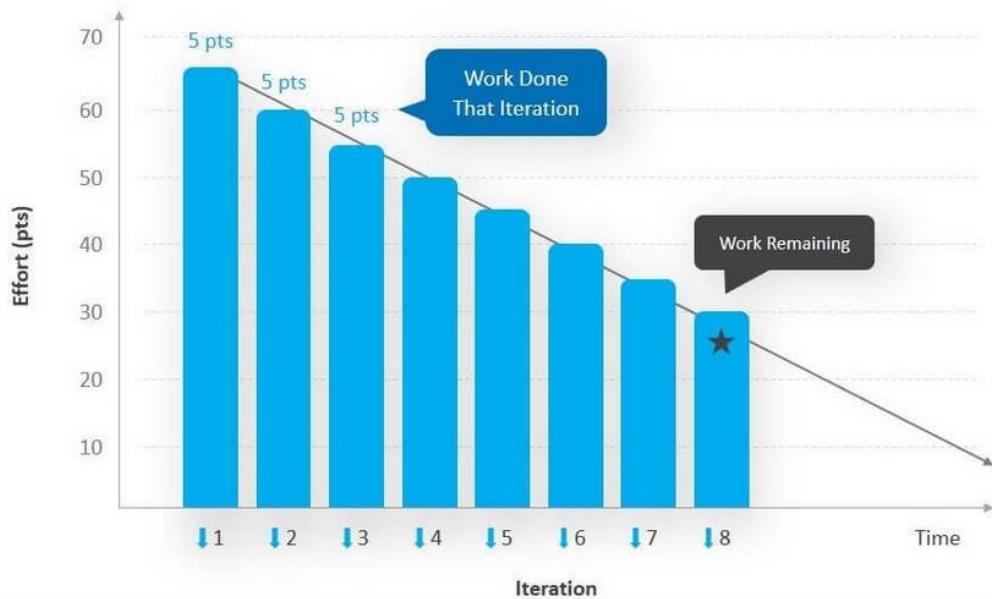


Рисунок 2.3 - Burn-down chart як метрика оцінки продуктивності задач

Приклади використання Agile-інструментів у практиці:

- Spotify застосовує унікальну модель Agile-організації, що включає автономні команди (скводи) з використанням Kanban-дощок у Trello та Jira для відстеження задач, спринтів і залежностей.

- Microsoft інтегрував Scrum і Kanban у Microsoft Azure DevOps для управління життєвим циклом програмного забезпечення, що дозволяє масштабувати Agile-підхід навіть у великих проєктах.

- ING Bank здійснив трансформацію організаційної структури за допомогою Agile, зокрема впровадивши інструменти, як-от Jira та Confluence, для щоденного управління задачами міжкросфункціональних команд.

- IBM використовує Agile@Scale з інтеграцією інструментів управління завданнями (Jira, Miro, Asana) для підтримки глобальних DevOps-команд, підвищуючи прозорість і узгодженість у виконанні технічних задач.

Крім того, в умовах сучасної економіки, що характеризується високою динамікою змін, зростанням рівня конкуренції та цифровізацією бізнес-процесів, здатність організації ефективно управляти завданнями виступає джерелом конкурентних переваг. Це дає змогу забезпечити швидку реакцію

на зміни зовнішнього середовища, скорочення витрат, підвищення якості результатів та рівня задоволеності зацікавлених сторін.

Таким чином, управління завданнями є не лише технічною функцією, а й стратегічним інструментом забезпечення гнучкості, ефективності та інноваційного розвитку організації, особливо в умовах впровадження Agile-підходів до проєктної діяльності.

2.3. Представлення функцій системи управління задачами по рівню команди

Система управління завданнями є невід’ємним інструментом у структурі сучасного організаційного менеджменту, що дозволяє підвищити ефективність виконання проєктів, оптимізувати використання ресурсів та забезпечити контроль над прогресом.

Таблиця 2.1 - Порівняльна таблиця функціональних можливостей популярних систем управління завданнями

Функція / Система	Jira	Trello	Asana	ClickUp	Monday.com
Інформаційна панель (Dashboard)	✓	✓	✓	✓	✓
Kanban-дошка	✓	✓	✓	✓	✓
Управління проєктами	✓	✗	✓	✓	✓
Тайм-трекінг (Timesheets)	✓	✗	✓	✓	✓
Управління рахунками	✗	✗	✗	✓	✓
Колаборація (коментарі, файли)	✓	✓	✓	✓	✓
Налаштування робочого простору	✓	✗	✓	✓	✓

Однією з основних цілей таких систем є забезпечення ефективного управління часом через пріоритезацію завдань і проєктів, як в індивідуальній,

так і в колективній роботі. Чітке розуміння цілей, визначення пріоритетів та наявних ресурсів є ключовими передумовами успішного управління завданнями. Для малих та середніх підприємств, зокрема у сфері розробки програмного забезпечення, впровадження ефективної системи управління завданнями сприяє підвищенню продуктивності персоналу та якості виконання проєктних робіт.



Рисунок 2.4 - Функціональні компоненти системи управління задачами

Незважаючи на те, що функціональність систем управління завданнями може варіюватися залежно від специфіки організації, основні функціональні компоненти таких систем зазвичай охоплюють наступні аспекти (рис. 2.4):

1. Інформаційна панель управління (Dashboard)

Забезпечує візуальне представлення загального стану проєктів, кількості завдань, помилок та активності учасників. Дашборд надає користувачам змогу швидко оцінити прогрес по кожному завданню, визначити критичні точки та актуальні завдання, які потребують першочергової уваги.

2. Управління проєктами

Система дозволяє створювати нові проєкти, призначати відповідальні команди, додавати клієнтів з можливістю налаштування рівня доступу, а також встановлювати бюджетні обмеження та контрольні віхи. Також реалізовано функціонал для фіксації змін статусу завдань, ведення хронології активностей та візуалізації динаміки прогресу.

3. Дошка завдань (Task Board)

Інструмент для створення, структурування та пріоритезації завдань у межах конкретного проєкту. Завдання можуть бути призначені конкретним членам команди з встановленням дедлайнів. Передбачено можливість створення підзавдань, додавання коментарів та прикріплення файлів, що забезпечує контекстуалізований підхід до виконання задач.

4. Управління завданнями на основі Kanban

Забезпечує візуальне керування статусами завдань у вигляді колонок ("Очікує", "У процесі", "Завершено" тощо), що дає змогу ефективно моніторити завантаження команди та своєчасно ідентифікувати вузькі місця в робочому процесі. Kanban-дошка також підтримує гнучке перетягування завдань між статусами та візуалізацію продуктивності.

5. Багатокористувацький доступ

Система підтримує запрошення необмеженої кількості користувачів з можливістю надання різних рівнів доступу до проєктів і робочих просторів. Окремий модуль дозволяє переглядати обсяг участі кожного користувача у проєктах та завданнях.

6. Табелі обліку робочого часу (Timesheets)

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

Інструмент для фіксації фактичного часу, витраченого на виконання конкретних завдань і проєктів. Користувачі можуть зазначати дату, тривалість роботи, а також додавати опис виконаної діяльності, що сприяє підвищенню прозорості та точності аналізу продуктивності.

7. Управління файлами

Надається можливість завантаження, зберігання та прикріплення необхідних документів до завдань або проєктів. Це забезпечує централізований доступ до важливих матеріалів у межах команди.

8. Управління рахунками-фактурами

Функціонал включає створення, редагування та друк рахунків-фактур для проєктів, з можливістю призначення умов оплати, податкових ставок, знижок і вибору замовників. Це дозволяє організаціям поєднувати проєктне управління з базовими елементами фінансового контролю.

9. Налаштування робочого простору

Можливість адаптації середовища системи до специфіки бізнесу — зміна логотипу, вибір валют, редагування реквізитів оплати та податкових налаштувань. Користувачі також можуть персоналізувати вигляд та структуру завдань відповідно до внутрішніх процесів.

Таким чином, система управління завданнями виконує не лише операційні, а й стратегічні функції в рамках організаційного управління, сприяючи підвищенню гнучкості, адаптивності та цілеспрямованості команди. Її використання особливо актуальне в умовах динамічного середовища, притаманного сучасним ІТ-компаніям, які впроваджують гнучкі методології управління, такі як Agile.

2.4. Специфікація вимог

Розробка системи здійснюється відповідно до специфікації вимог, яка деталізує необхідні характеристики цільового програмного продукту.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

Вимоги до системи можуть бути класифіковані за основними категоріями:

- Нефункціональні вимоги, що включають вимоги до продуктивності (Performance Requirements).

- Функціональні вимоги (Functional Requirements).

Точне та повне розуміння специфікації вимог є критично важливим фактором успіху проекту, оскільки воно забезпечує відповідність розробленої системи очікуванням та потребам кінцевих користувачів.

2.4.1. Нефункціональні вимоги (вимоги до продуктивності)

До нефункціональних вимог, що визначають якісні характеристики системи та обмеження, належать:

- Незалежність від СУБД: Система має бути спроектована таким чином, щоб забезпечувати незалежність від конкретної реалізації системи управління базами даних.

- Час відповіді: Час відповіді системи на запити користувача має бути мінімальним для забезпечення комфортної взаємодії.

- Продуктивність: Система повинна демонструвати високий рівень загальної продуктивності під типовим та піковим навантаженням.

- Безпека: Рівень безпеки системи має бути високим для запобігання несанкціонованому доступу, модифікації або видаленню даних з боку злоумисників.

2.4.2. Функціональні вимоги

Функціональні вимоги визначають конкретні функції та операції, які система повинна виконувати. Для ролі Адміністратора системи передбачено реалізацію наступних ключових функцій:

- Управління проектами: Перегляд існуючих, створення нових та видалення проектів, а також можливість оновлення їх деталей.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

- Призначення завдань: Призначення завдань конкретним користувачам у межах будь-якого проекту, включаючи функціональність взаємодії з клієнтами щодо завдань.

- Управління рахунками-фактурами: Створення, видалення та оновлення рахунків-фактур, а також редагування або видалення конкретних елементів інформації в них.

- Конфігурація системи: Налаштування загальних параметрів функціонування системи, таких як параметри сайту, конфігурація завдань, налаштування оплати, податкові параметри та шаблони рахунків-фактур.

2.5. Логічна модель системи

Логічна діаграма потоку даних (ДПД) системи, представлена далі (рис. 2.5), ілюструє процеси трансформації даних та їх рух в системі за результатами проведеного аналізу.

Система взаємодіє з користувачем (клієнтом) через інтерфейсні форми, що реалізовані на фронтенді. Для ініціації сеансу роботи користувачеві надається форма автентифікації, яка вимагає введення адреси електронної пошти та пароля. Після введення даних здійснюється їх валідація шляхом перевірки облікових даних користувача у системі реєстрації. На основі визначених прав доступу (дозволів) здійснюється завантаження відповідного інтерфейсу системи та надання доступу до дозволеної функціональності.

Розглянемо основні елементи та потоки цієї діаграми.

1. Request for Development (Запит на розробку)

→ Початкова точка процесу. Ініціатор надсилає запит, який потрапляє в систему для обробки.

2. Processing Request & Calculation of Estimated Effort (Обробка запиту та розрахунок зусиль)

→ Запит аналізується, оцінюється обсяг робіт.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

- Вихід: -Request Detail- передається в Data Store, а також у наступні етапи.

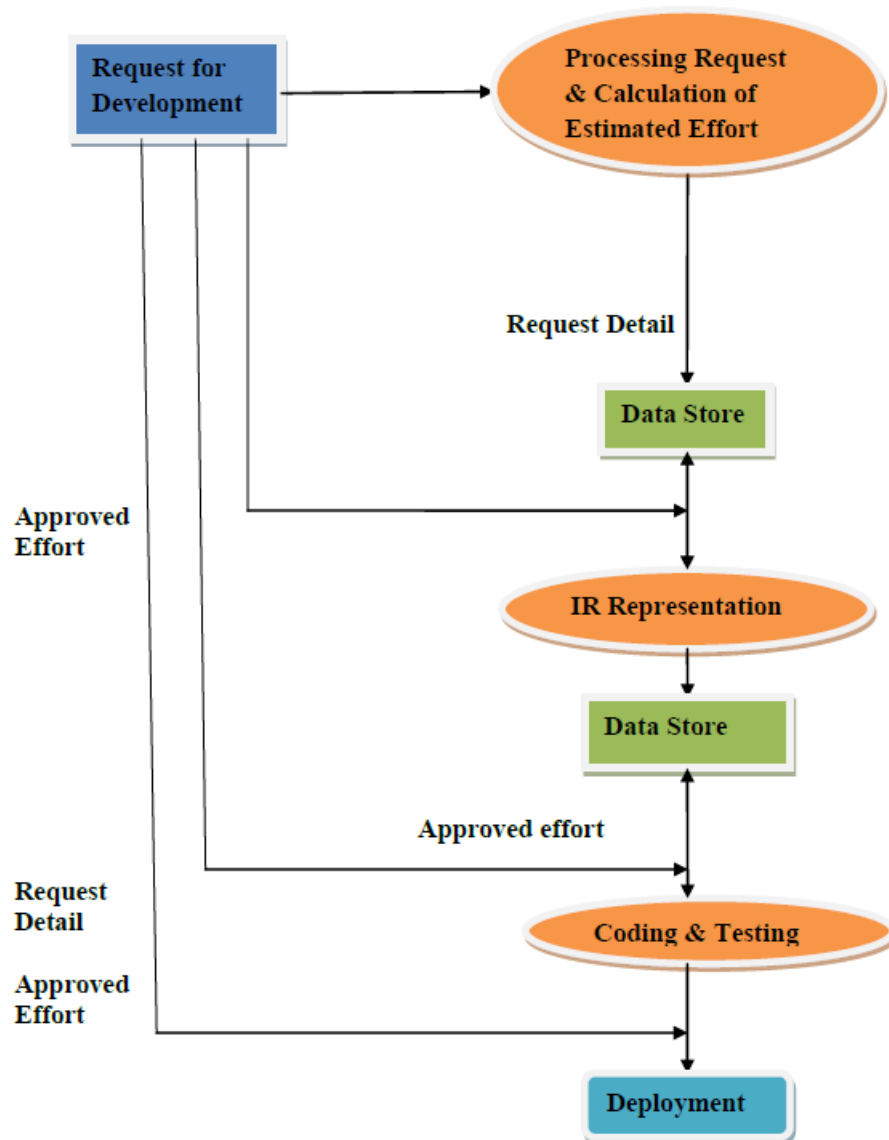


Рисунок 2.5 - Логічна діаграма потоку даних

3. Data Store (1)

→ Зберігає деталі запиту. Є проміжним сховищем даних для подальшого використання.

4. IR Representation

→ Представляє собою проміжну форму (інтерфейсне представлення або інтерпретаційне представлення) для подальшої обробки.

5. Data Store (2)

→ Зберігає деталі IR-представлення, які використовуються під час реалізації.

6. Coding & Testing (Кодування та тестування)

→ Основний етап розробки програмного забезпечення. Після затвердження зусиль відбувається створення та перевірка програмного продукту.

7. Deployment (Впровадження)

→ Фінальний етап, де готовий продукт впроваджується у виробниче середовище.

Опишемо потоки даних:

- Request Detail та Approved Effort циркулюють між етапами, забезпечуючи необхідну інформацію для прийняття рішень та виконання робіт.

- Кожен етап логічно пов'язаний зі сховищами даних, що дозволяє зберігати та повторно використовувати дані.

2.6. Проектування логічної діаграми верхнього рівня

На рисунку 2.6 подана логічна діаграма верхнього рівня, що ілюструє основний потік процесів та функціональних модулів системи керування завданнями. Діаграма має ієрархічний характер та відображає взаємодію користувача із системою на ключових етапах.

Структуру можна описати наступним чином:

1. Початковий вхід до системи.

Процес розпочинається із загальної сутності "TASK MANAGEMENT SYSTEM". Першим кроком для зареєстрованих користувачів є взаємодія з "Login form for registered" (Форма входу для зареєстрованих).

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

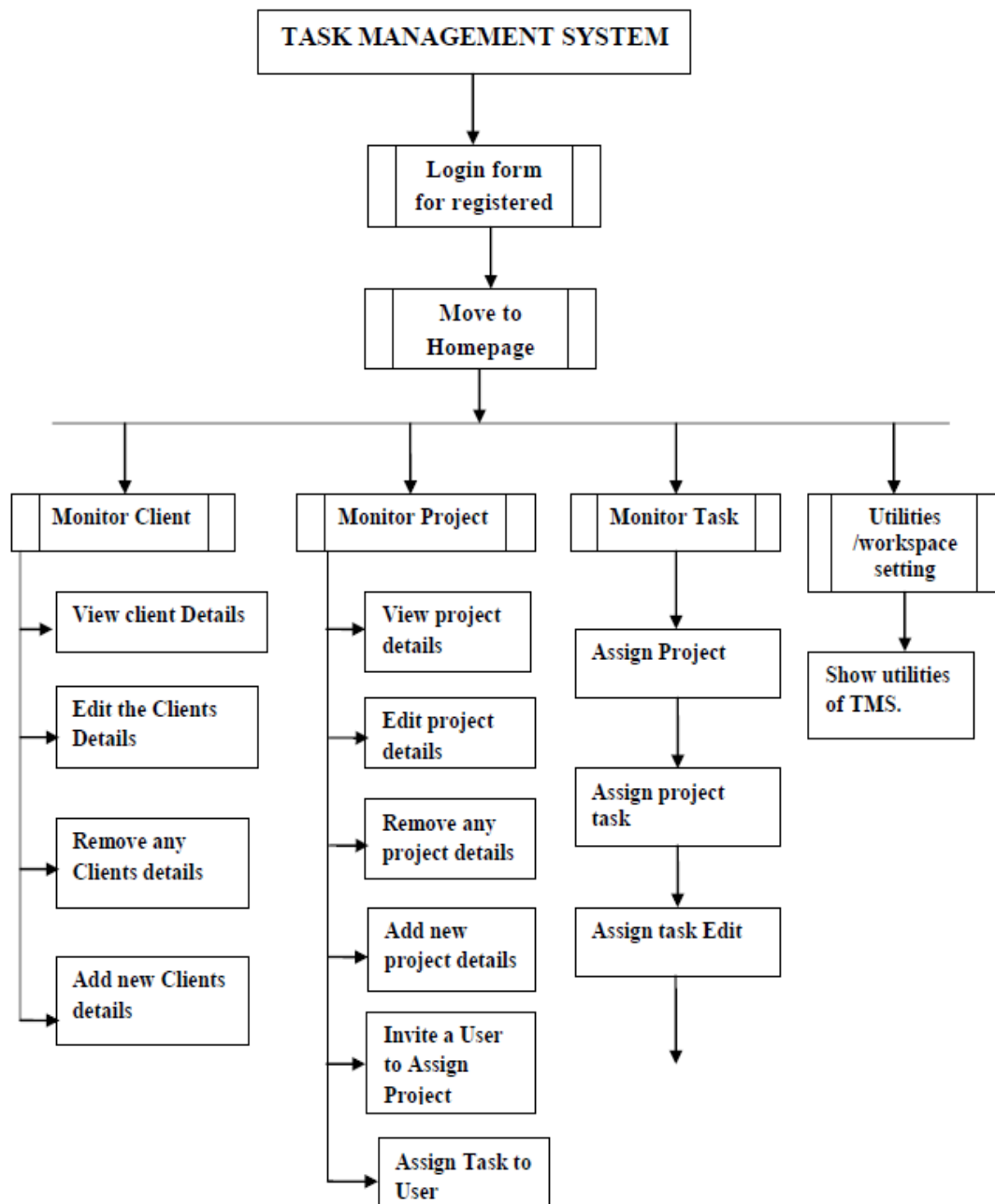


Рисунок 2.6 - Логічна діаграма верхнього рівня

2. Доступ до головної сторінки.

Після успішної автентифікації користувач переміщується до "Move to Homepage" (Перехід на головну сторінку).

3. Основні функціональні гілки.

З головної сторінки система розгалужується на чотири основні функціональні області або модулі:

- Monitor Client (Моніторинг клієнтів).

Ця гілка охоплює функції, пов'язані з управлінням клієнтською інформацією, включаючи "View Client Details" (Перегляд деталей клієнта), "Edit the Clients Details" (Редагування деталей клієнтів), "Remove any Clients details" (Видалення деталей клієнтів) та "Add new Clients details" (Додавання нових деталей клієнтів).

- Monitor Project (Моніторинг проектів).

Цей розділ присвячений функціям управління проектами. Він включає "View project details" (Перегляд деталей проекту), "Edit project details" (Редагування деталей проекту), "Remove any project details" (Видалення деталей проекту), "Add new project details" (Додавання нових деталей проекту), а також функції взаємодії з користувачами в контексті проекту: "Invite a User to Assign Project" (Запрошення користувача до призначення проекту) та "Assign Task to User" (Призначення завдання користувачу).

- Monitor Task (Моніторинг завдань).

Ця гілка зосереджена на функціях, пов'язаних із самими завданнями. Вона включає "Assign Project" (Призначення проекту - розташування цього блоку під "Monitor Task" виглядає нелогічним з точки зору типового процесу, але так показано на діаграмі-), "Assign project task" (Призначення завдання проекту) та "Assign task Edit" (Редагування призначення завдання).

- Utilities /workspace setting (Утиліти / налаштування робочого простору).

Ця гілка відображає доступ до допоміжних функцій та налаштувань системи, що представлено блоком "Show utilities of TMS." (Показати утиліти TMS).

Загалом, діаграма представляє високорівневе декомпозицію системи управління завданнями за ключовими функціональними областями, починаючи від точки входу користувача.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

2.7. Розробка алгоритму процесу автентифікації користувача в системі

На рисунку 2.7 подано схему процесу автентифікації користувача та подальшої взаємодії з функціональністю системи управління завданнями або проектами. Вона демонструє логіку роботи з системою після спроби входу.

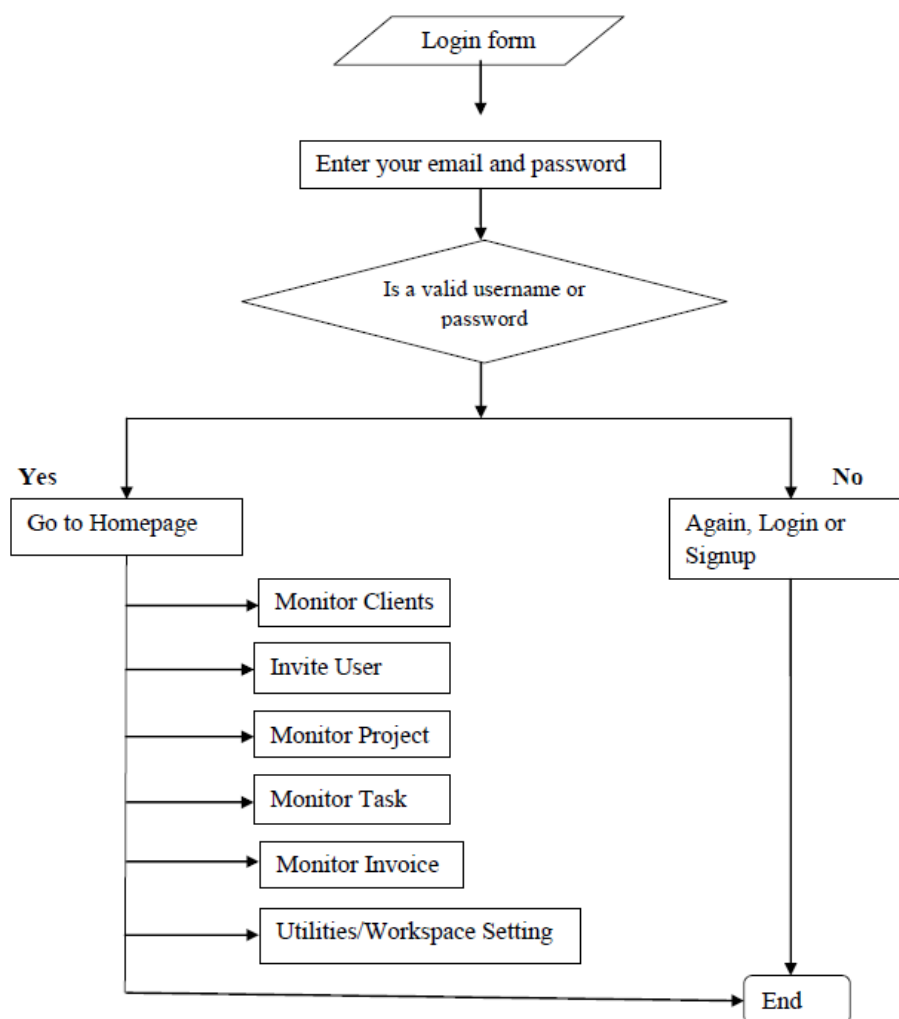


Рисунок 2.7 – Алгоритм процесу автентифікації користувача в системі

Опис етапів алгоритму (рис. 2.7):

1. Login Form (Форма входу)

→ Користувач ініціює процес, відкриваючи форму авторизації.

2. Enter your email and password (Введення електронної пошти та пароллю)

→ Користувач надає свої облікові дані для перевірки.

3. Is a valid username or password? (Чи правильні дані?)

→ Система перевіряє правильність облікових даних:

- Якщо так (Yes) – користувач переходить до домашньої сторінки.

- Якщо ні (No) – повертається до авторизації або реєстрації.

Успішний вхід (гілка "Yes"):

Після переходу на Homepage (Домашню сторінку) користувач отримує доступ до таких функцій:

- Monitor Clients — перегляд та керування клієнтами.

- Invite User — запрошення нових користувачів до системи.

- Monitor Project — моніторинг поточних проєктів.

- Monitor Task — перегляд і контроль завдань.

- Monitor Invoice — перегляд рахунків-фактур.

- Utilities / Workspace Setting — налаштування середовища роботи.

Неуспішний вхід (гілка "No"):

Користувач перенаправляється до повторної авторизації або реєстрації, після чого процес завершено.

Успішна авторизація відкриває доступ до ключових адміністративних та управлінських функцій. Такий підхід забезпечує захищене та контрольоване середовище для командної роботи в цифровому просторі.

2.8. Діаграма потоку даних

Діаграми потоку даних є центральним інструментом та основою, з якої розробляються інші компоненти. Логічний переклад даних від входу до виходу через процеси може бути визначений окремо від фізичних

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

компонентів системи. Діаграму потоку даних також називають графіком потоку даних або діаграмою бульбашок.

Найважливішим аспектом діаграм потоку даних є збалансованість, що означає, що діаграма потоку даних на одному рівні відповідно відтворюється в діаграмі потоку даних на наступному рівні при розробці проекту. У діаграмах потоку даних можуть виникати два типи помилок: синтаксичні та семантичні. Семантичні помилки складніші за синтаксичні через набір правил, яких необхідно дотримуватися для їх виявлення.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ МАСШТАБУВАННЯ СИСТЕМИ КЕРУВАННЯ ЗАДАЧАМИ РОЗРОБКИ ПО РІВНЮ КОМАНДИ

3.1. Архітектура та технічні аспекти запропонованої системи управління задачами

Запропонована система являє собою веб-орієнтований додаток управління завданнями (Task Management System), яка розробляється як комплексна платформа для забезпечення ефективної взаємодії між адміністраторами, співробітниками та клієнтами. Система передбачає три основні ролі користувачів: Адміністратор, Співробітник та Клієнт. Роль Адміністратора включає відповідальність за моніторинг усіх завдань у рамках проектів. Даний додаток може застосовуватися організаціями для виконання повсякденних операційних завдань.

Структура системи складається з п'яти основних функціональних модулів:

- Модуль автентифікації користувачів (входу до системи).
- Модуль реєстрації нових користувачів.
- Модуль управління проектами.
- Модуль управління клієнтами.
- Модуль конфігурації робочого простору та утиліт.

Основною метою системи управління завданнями є забезпечення ефективного виконання завдань компанії. Дана частина роботи представляє архітектуру та реалізацію системи управління завданнями, включаючи проектування структури даних та підхід до розподілу робочих завдань.

Реалізація веб-додатку системи управління завданнями базується на сучасному стеку технологій:

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

а) MySQL є реляційною системою управління базами даних (РСУБД), яка характеризується високою швидкістю та зручністю використання, що робить її придатною для застосування як у малих, так і у великих бізнес-середовищах. Система розповсюджується під ліцензією з відкритим кодом та підтримує стандартизовану мову запитів SQL. MySQL функціонує на різноманітних операційних системах та взаємодіє з багатьма мовами програмування (наприклад, PHP, PERL, C, C++, JAVA тощо).

б) PHP є серверною мовою сценаріїв, яка може бути вбудована в HTML. Вона широко застосовується для розробки повнофункціональних веб-сайтів, включаючи платформи електронної комерції, а також для управління динамічним вмістом, взаємодії з базами даних та моніторингу сесій. PHP підтримує підключення до численних популярних систем баз даних, включаючи MySQL, PostgreSQL, Oracle, Sybase, Informix та Microsoft SQL Server. Доступність PHP на більшості хостинг-провайдерів та його широке розповсюдження як середовища виконання на веб-серверах робить його ефективним рішенням для малих підприємств та приватних клієнтів.

в) Laravel є одним з провідних PHP-фреймворків, який здобув значну популярність протягом останніх п'яти років як серед веб-розробників, так і у бізнес-структурах по всьому світу. За даними опитування SitePoint, фреймворк Laravel, що реалізує архітектурний патерн MVC, є найбільш затребуваним як для особистих, так і для комерційних проєктів. Laravel класифікується як фреймворк повного стеку, оскільки він забезпечує функціональність для обробки запитів від веб-серверу до управління базами даних та генерації HTML-вмісту.

Апаратні вимоги для клієнтської частини:

- Тип комп'ютера: PC-AT сумісний.
- Процесор: Intel Dual Core або вище.
- Обсяг оперативної пам'яті (ОЗУ): від 1 ГБ.
- Пристрої введення: Маніпулятор типу "Миша", Клавіатура.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Програмні вимоги для серверної частини:

- Операційна система: Сімейство Windows (версії 8, 10) або дистрибутиви Linux.

- Веб-сервер: Apache.

Використані інструменти та технології

- Програмний фреймворк: Laravel

- Мова програмування серверної частини: PHP

- Технології клієнтської частини (Frontend): JavaScript (JS), HTML, CSS

- Система управління базами даних (СУБД): MySQL

3.2. Проектування ER діаграми

ER-діаграма (діаграма сутність-зв'язок) відображає структуру бази даних системи управління проектами та завданнями. Вона містить сутності, їх атрибути та зв'язки між ними (рис. 3.1).

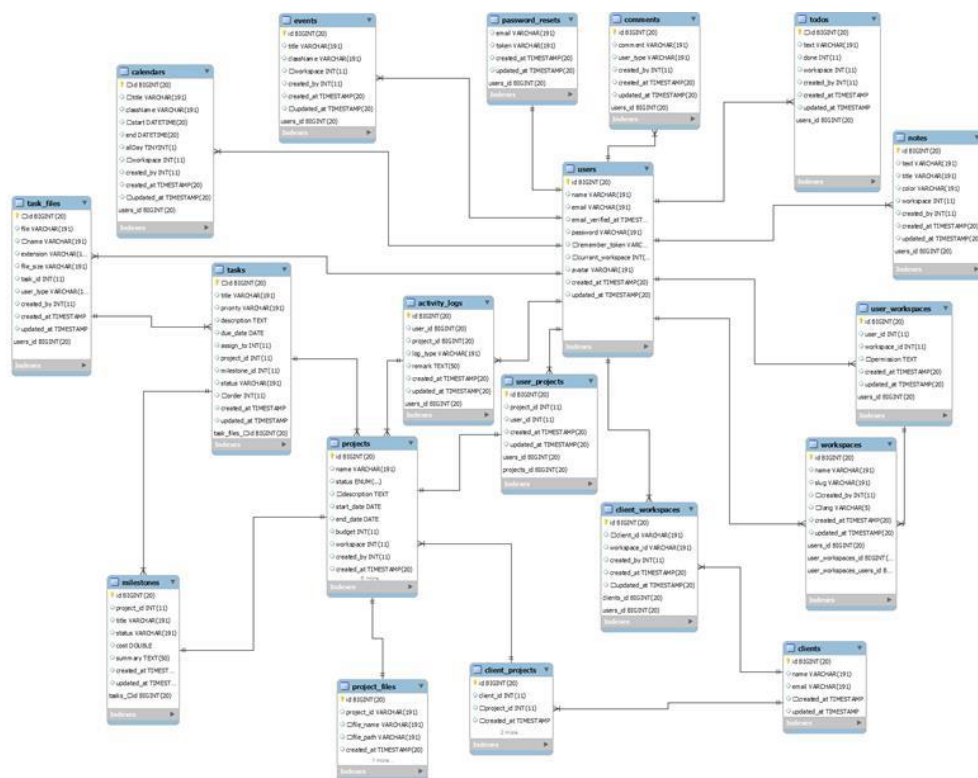


Рисунок 3.1 – ER-діаграма системи

Змн.	Арк.	№ докум.	Підпис	Дата

Нижче подано узагальнений опис основних компонентів та взаємозв'язків між ними.

Ключові сутності та їх призначення:

1. users

- Зберігає інформацію про користувачів (ім'я, email, пароль тощо).

- Пов'язана з багатьма іншими таблицями: `projects`, `tasks`, `comments`, `activity_logs`, `user_workspaces`, `user_projects`.

2. projects

- Містить дані про проекти (назва, опис, статус, дедлайн).

- Пов'язана з: `users`, `tasks`, `milestones`, `project_files`, `client_projects`, `user_projects`.

3. tasks

- Описує завдання в межах проектів (назва, опис, статус, термін, виконавець).

- Має зв'язки з: `projects`, `task_files`, `milestones`, `comments`.

4. workspaces

- Робочі простори, які об'єднують користувачів і проекти.

- Пов'язана з `user_workspaces`, `projects`, `clients`.

5. user\workspaces

- Багато-до-багатьох зв'язок між `users` і `workspaces`.

6. user\projects

- Багато-до-багатьох зв'язок між `users` і `projects`.

7. milestones

- Віхи проекту, що слугують етапами виконання завдань.

- Прив'язані до проектів і можуть містити терміни виконання.

8. task_files / project_files

- Файли, прикріплені до завдань або проектів.

9. comments

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

- Коментарі, пов'язані із завданнями. Кожен коментар має автора (`user_id`) і належить до певного завдання (`task_id`).

10. activity_logs

- Журнал дій користувачів у системі (логування подій).

11. clients

- Дані про клієнтів, з якими пов'язано проекти.

12. client_projects

- Таблиця зв'язку між клієнтами та проектами.

13. events / calendars

- Планувальні таблиці для зберігання подій і розкладів, пов'язаних з користувачами або завданнями.

14. notes

- Нотатки, створені користувачами (можливо, як особисті чи проектні замітки).

15. password_resets

- Таблиця для керування запитом на відновлення пароля.

Основні зв'язки між сутностями:

- users ↔ workspaces (через `user_workspaces`)

- users ↔ projects (через `user_projects`)

- users ↔ tasks (через поля `assigned_to`, `created_by`)

- projects ↔ tasks

- projects ↔ milestones

- projects ↔ clients (через `client_projects`)

- tasks ↔ comments, tasks ↔ task_files

- projects ↔ project_files

Ця ER-діаграма ілюструє модульну та масштабовану архітектуру бази даних, орієнтовану на управління проектами, завданнями та співпрацею між користувачами. Схема підтримує багатокористувацьке середовище,

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

клієнтську взаємодію, логування дій, прикріплення файлів, нотатки та управління доступом.

3.3. Діаграми Ганта та PERT як основні інструменти планування та візуалізації проектів

Діаграма Ганта є різновидом стовпчастої діаграми, що використовується для візуального представлення розкладу виконання завдань проекту у часі. Цей інструмент широко застосовується для планування проектів незалежно від їх масштабу, надаючи зручний спосіб відображення запланованих робіт на конкретні дати. Вона також дозволяє наочно представити загальні терміни проекту (дати початку та завершення) в одному графічному форматі.

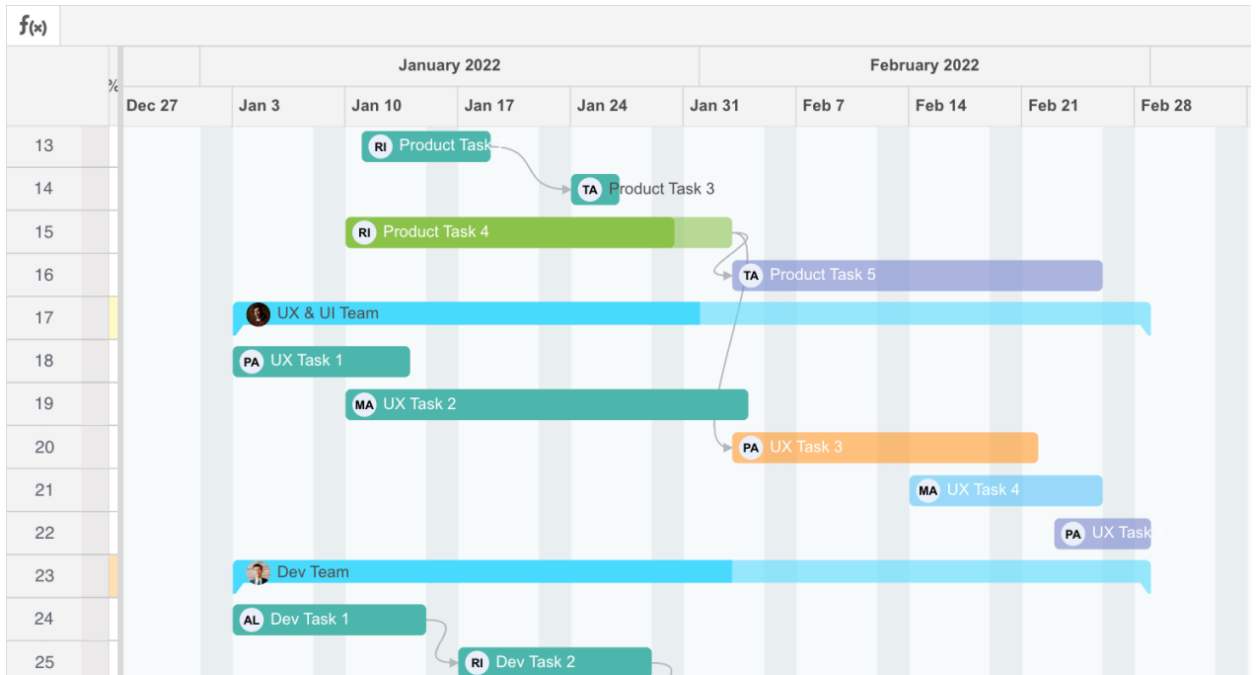


Рисунок 3.2 – Діаграма Ганта

Діаграма Ганта може використовуватися для ілюстрації системного підходу до процесу передачі технологій, що охоплює документовані знання та досвід, отримані на етапах розробки та/або комерціалізації, компетентній

та уповноваженій стороні. Процес передачі технологій включає як передачу відповідної документації, так і демонстрацію здатності сторони, що отримує, коректно виконувати критично важливі складові технології, що має бути підтверджено обома сторонами та, за необхідності, відповідними регуляторними органами. Актуальний статус виконання робіт за розкладом може бути відображений на діаграмі Ганта шляхом затінення стовпців відповідно до відсотка завершеності завдання та додаванням вертикальної лінії, що позначає поточну дату (лінія "СЬОГОДНІ").

Діаграма PERT є інструментом управління проектами, призначеним для планування, організації та координації комплексу завдань у межах проекту. PERT (Program Evaluation and Review Technique - Техніка оцінки та перегляду програми) — це метод управління, розроблений Військово-морськими силами США у 1950-х роках для контролю за реалізацією програми ракет підводних човнів "Поларіс". Схожий за призначенням метод критичного шляху (Critical Path Method - CPM) був розроблений приблизно в той же період для управління проектами в приватному секторі.

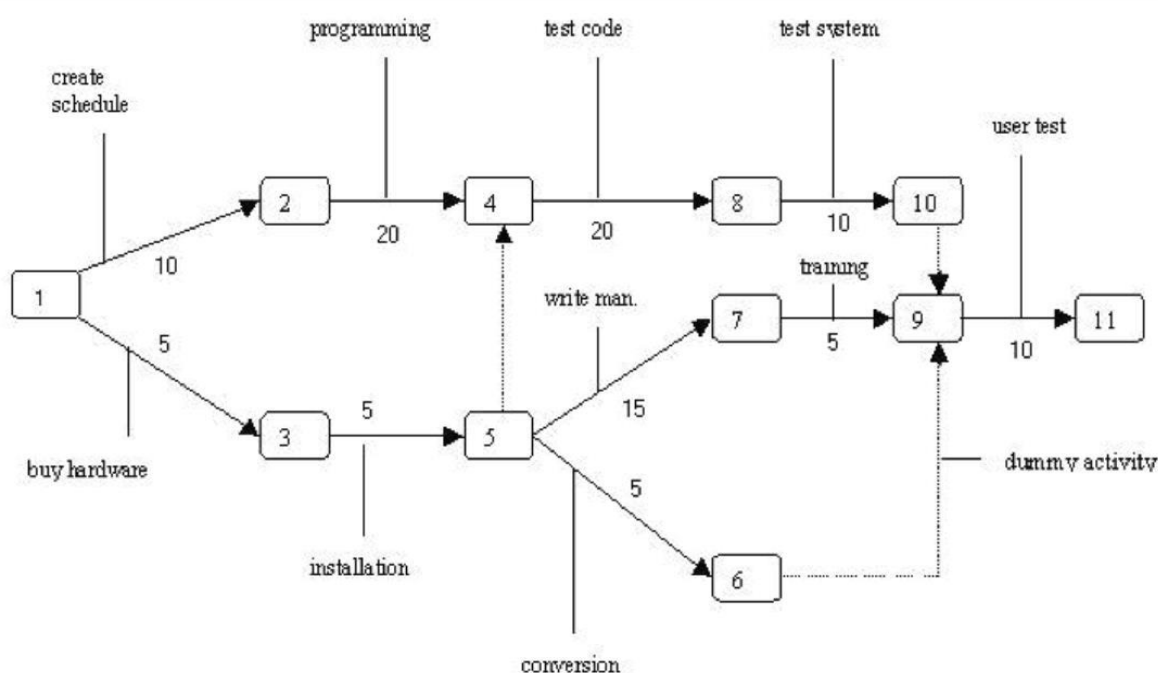


Рисунок 3.3 - Діаграма PERT

Змн.	Арк.	№ докум.	Підпис	Дата

Діаграма PERT дозволяє менеджеру проекту здійснювати аналіз проектних завдань та оцінювати час, необхідний для виконання кожної окремої діяльності. На основі цієї інформації менеджер проекту може оцінити мінімальний термін, необхідний для завершення проекту в цілому. Ці дані також є корисними для розробки бюджету проекту та визначення необхідних ресурсів для його успішної реалізації.

Діаграма PERT являє собою мережеву діаграму, яка представляє проект як сукупність пронумерованих вузлів (що можуть зображуватися у вигляді кіл або прямокутників), які символізують події або ключові етапи (віхи) проекту. Ці вузли з'єднані позначеними спрямованими лініями (векторами), що відображають завдання проекту. Послідовність виконання завдань визначається напрямком стрілок на лініях. Діаграма PERT, як поширена методика візуалізації в управлінні проектами, іноді надається перевага над діаграмою Ганта через її здатність більш чітко ілюструвати залежності між завданнями. З іншого боку, інтерпретація діаграми PERT може бути значно складнішою, особливо у випадках комплексних проектів. Через це менеджери проектів часто застосовують обидві техніки для повноцінного планування та контролю.

3.4. Архітектура інтерфейсу користувача та функціональність системи управління завданнями за ролями

Модуль системи управління завданнями призначений для комплексного управління всіма аспектами проектів в межах організації, надаючи диференційований доступ до функціональності залежно від ролі користувача. Взаємодія користувача із системою реалізується через послідовність інтерфейсних екранів, що забезпечують доступ до відповідних модулів та операцій.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

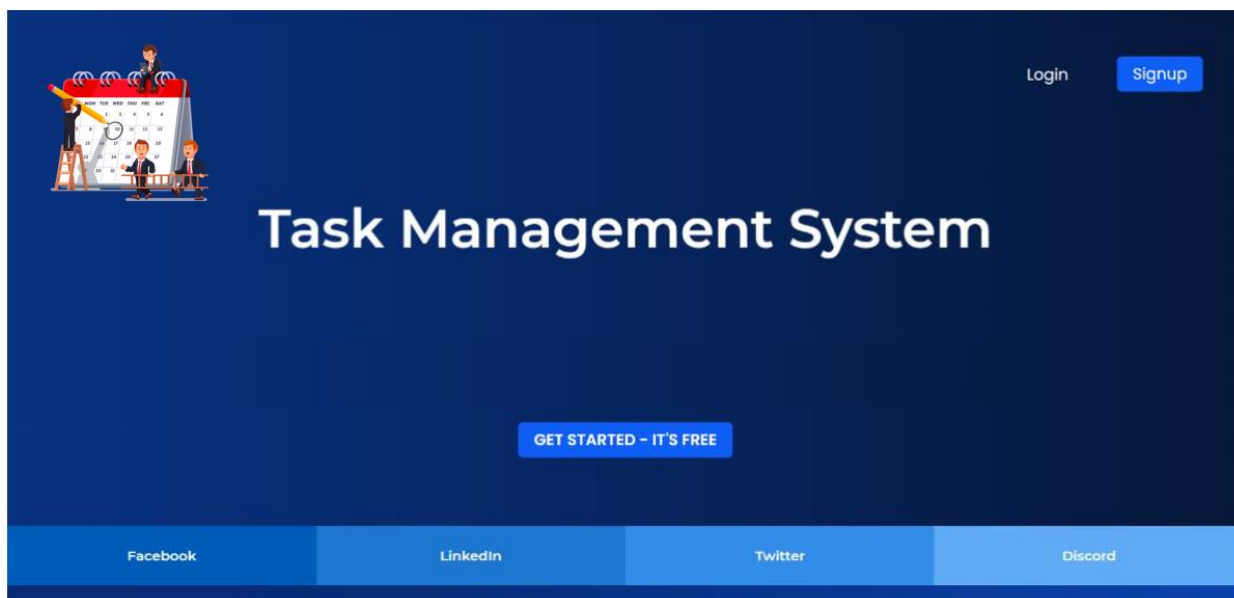


Рисунок 3.3 – Головна сторінка системи

Сторінка входу (Login Page) є початковою точкою доступу до TMS. Цей інтерфейс надає можливість користувачам здійснити вхід до системи шляхом введення своїх облікових даних (адреси електронної пошти та пароля). Реалізовано механізм валідації введених даних шляхом їх перевірки у базі зареєстрованих користувачів. Екран також містить функціональність для переходу до процедури реєстрації нових користувачів. Інтерфейс автентифікації передбачає можливість входу до системи з різними ролями, зокрема як стандартний користувач або клієнт. Після успішної автентифікації в ролі Адміністратора користувач отримує доступ до відповідної панелі моніторингу.

Сторінка реєстрації (Registration Page) призначена для створення нових облікових записів у системі. Для успішної реєстрації користувачеві необхідно надати своє ім'я, адресу електронної пошти, пароль, а також визначити назву робочого простору, до якого буде належати обліковий запис. Після завершення процедури реєстрації та активації функції створення облікового запису (наприклад, натисканням відповідної кнопки) система здійснює перенаправлення користувача на основну панель моніторингу.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Sign Up

Full Name

Workspace Name

Email Address

Password

*A min 8 characters, upper and lower character, number and special character are required

Confirm Password

Accept terms and privacy

Register

OR

Sign In

Рисунок 3.4 - Сторінка реєстрації

Login

User Login

Client Login

Email

Password

Remember Me

Login

[Forgot your password?](#)

OR

Sign Up

Рисунок 3.5 - Сторінка входу в систему

Екран Адміністратора (Панель управління) - цей інтерфейс є центральним пунктом управління системою для користувачів з роллю Адміністратора.

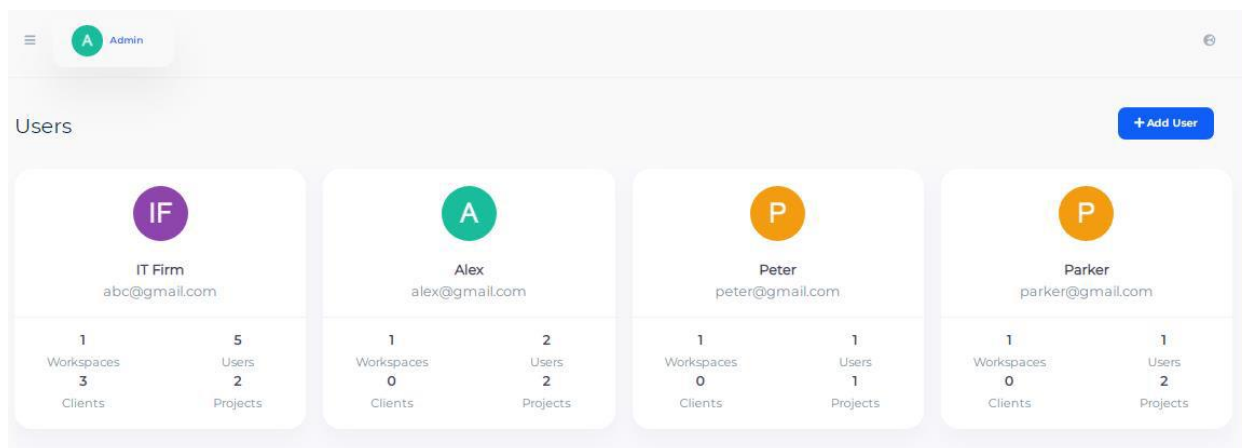


Рисунок 3.6 – Панель адміністратора

Панель моніторингу надає агреговану інформацію та ключові показники, що стосуються функціонування організації в системі. Зокрема, вона відображає:

- Перелік усіх зареєстрованих користувачів, асоційованих з даною організацією.
- Статистичні дані щодо кількості створених робочих просторів, загальної кількості користувачів, клієнтів та проектів у системі (або пов'язаних з діяльністю адміністратора). Інтерфейс надає прямий доступ до функцій управління користувачами (наприклад, через кнопку "Додати користувача" для ініціації процедури створення нового користувача) та доступу до власного профілю адміністратора (наприклад, через елемент інтерфейсу "Адміністратор").

Екран додавання користувача (Add User Page) - цей екран є частиною модуля управління користувачами і призначений для внесення даних нового користувача системи. Інтерфейс містить поля для введення відповідної інформації. Збереження даних нового користувача та створення його

облікового запису здійснюється шляхом активації відповідної функції (наприклад, кнопки "Створити").

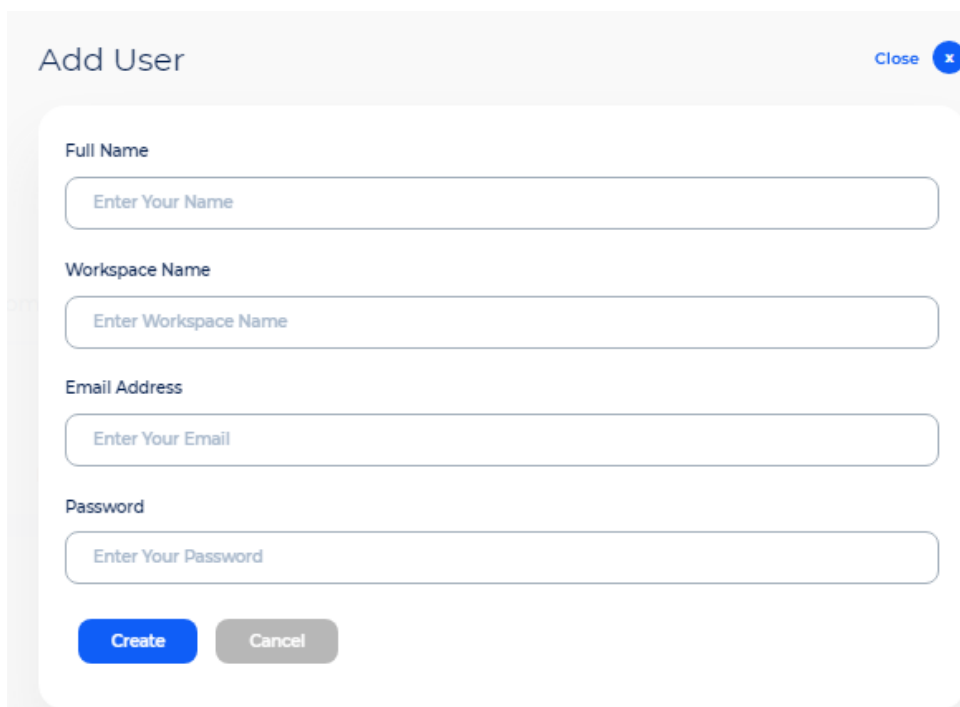


Рисунок 3.7 – Сторінка додавання користувача

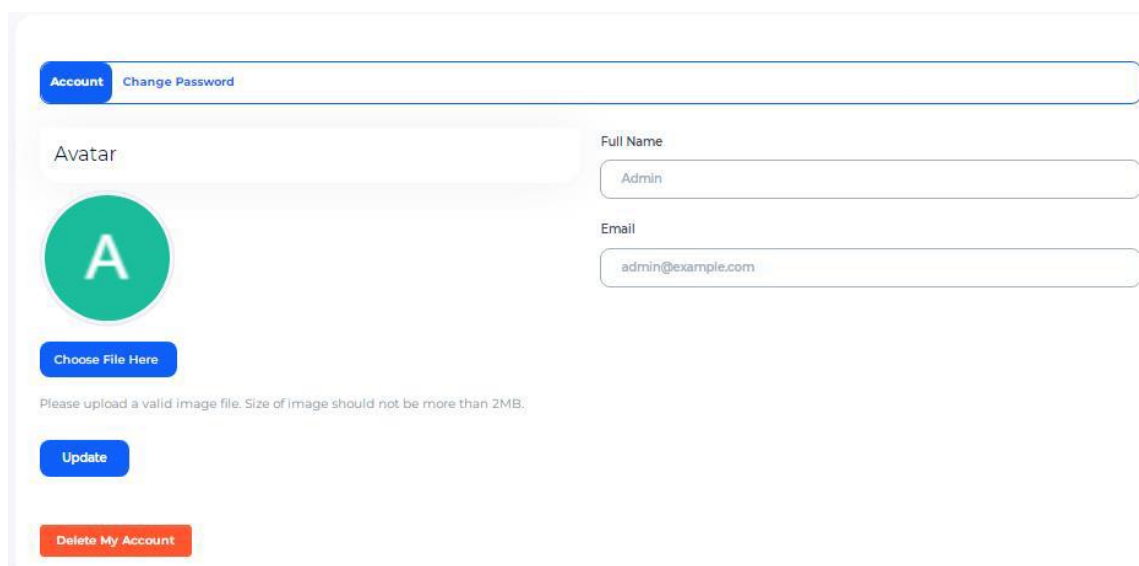


Рисунок 3.8 - Екран профілю адміністратора

Екран профілю Адміністратора (Administrator Profile Page) подано на рисунку 3.8. Доступ до цього екрану зазвичай здійснюється через

навігаційний шлях, наприклад, "Адміністратор > Мій профіль". Інтерфейс дозволяє Адміністратору переглядати та модифікувати інформацію свого профілю, включаючи зміну імені, адреси електронної пошти та зображення профілю. Крім того, на даній сторінці реалізована функція оновлення пароля облікового запису Адміністратора.

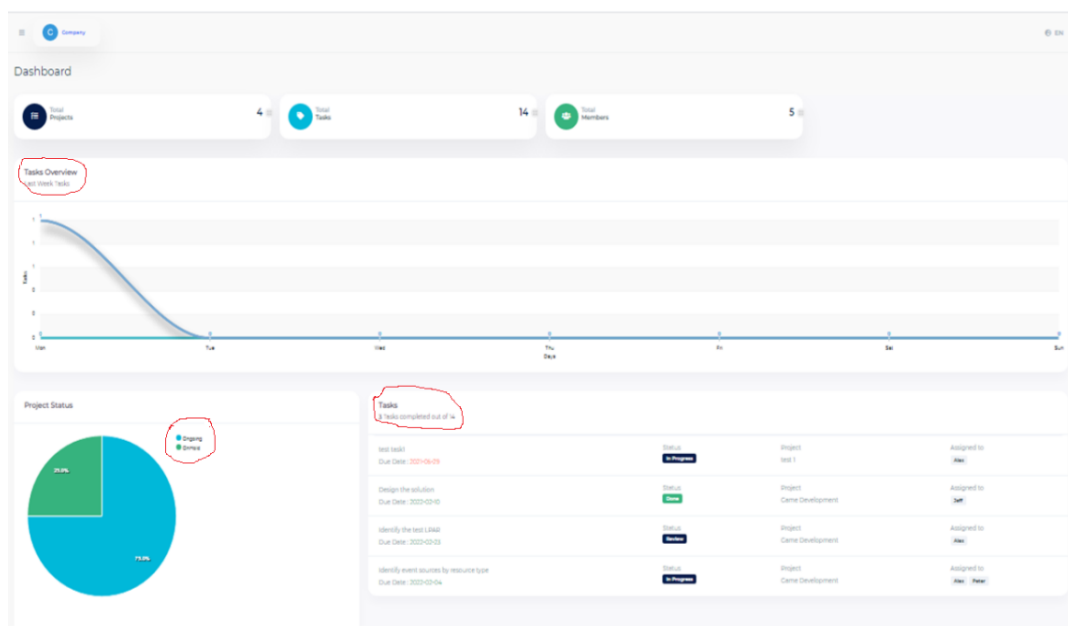


Рисунок 3.9 - Панель управління зареєстрованого користувача

Панель управління зареєстрованого користувача є основним інтерфейсом для користувачів зі стандартними правами доступу (не Адміністратор). Ця панель надає персоналізований огляд активності та статистичних даних, релевантних конкретному користувачеві. Ключові інформаційні блоки включають:

- Статистичні показники щодо кількості проектів, завдань та інших учасників, діяльність яких відстежується користувачем або до яких він причетний.
- Графічна візуалізація огляду виконання завдань за попередній тиждень.

- Відображення поточного статусу проектів, у яких бере участь користувач.
- Кількість завдань, успішно завершених даним користувачем.

3.5. Функціональні аспекти інтерфейсу управління проектами та завданнями

Інтерфейсна структура системи управління завданнями (TMS) включає набір екранів, призначених для комплексного управління проектами та їхніми складовими – завданнями. Дана частина роботи деталізує функціональність, доступну користувачам при роботі з модулями управління проектами та завданнями.

3.5.1. Інтерфейс управління проектами

Екран огляду проектів - цей інтерфейс надає користувачеві агрегований перелік проектів, до яких він має доступ або за які відповідає. Проекти на даному екрані класифікуються за трьома основними статусами: "Поточні", "Завершені" та "На утриманні". Інтерфейс також містить функціональність для ініціації процедури створення нового проекту (наприклад, через активацію відповідної кнопки "Створити").

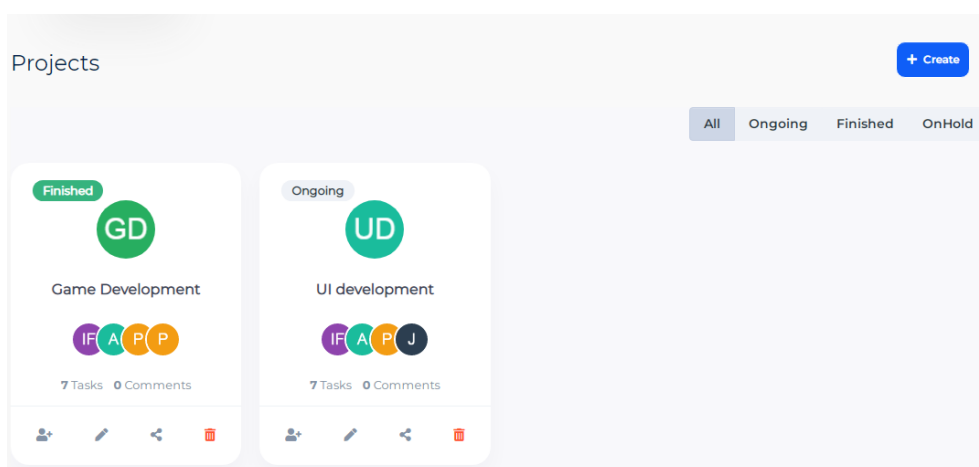


Рисунок 3.10 - Екран огляду проектів

Екран створення проекту (рис. 3.11) призначений для внесення вихідних даних нового проекту. Користувачеві надається набір полів для введення інформації, релевантної проекту (назва, опис, терміни тощо). Збереження конфігурації нового проекту та його реєстрація в системі здійснюється шляхом активації функції збереження (наприклад, кнопки "Створити").

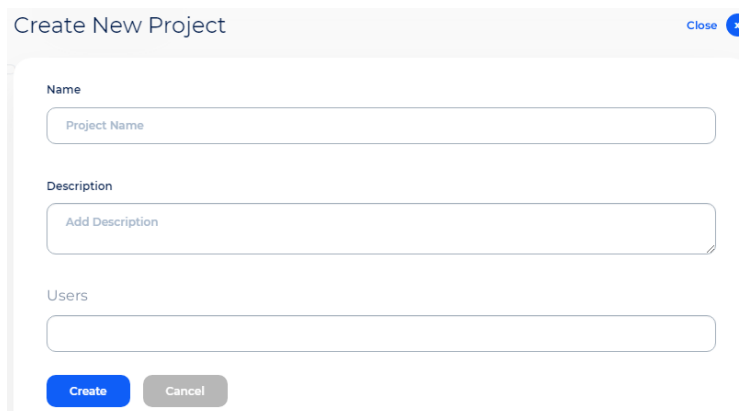


Рисунок 3.11 - Екран створення проекту

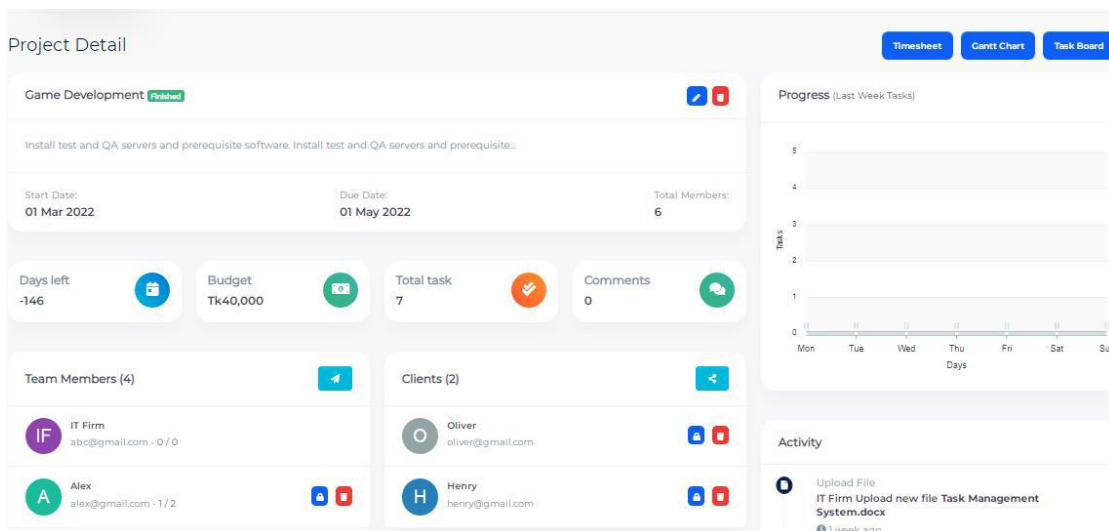


Рисунок 3.12 - Екран деталей проекту

Екран деталей проекту (рис. 3.12) доступ до детальної інформації про конкретний проект надається через інтерфейс деталей проекту (типовий навігаційний шлях: Проекти > [Іконка назви проекту/Назва проекту]). На

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

даному екрані відображаються ключові атрибути проекту, включаючи його опис, дату початку, термін виконання, загальну кількість учасників, бюджет, кількість днів, що залишилися до завершення, та інші релевантні показники.

Функціональність екрану деталей проекту охоплює:

- Управління учасниками проекту: Запрошення нових та видалення існуючих учасників.
- Взаємодія з клієнтами: Надання доступу до інформації про проект новим клієнтам.
- Управління віхами: Створення, редагування та видалення ключових етапів проекту (віх).
- Управління файлами: Можливості для завантаження та скачування файлів будь-якого формату, пов'язаних із проектом.
- Візуалізація: Графічне представлення огляду завдань проекту та їх поточного статусу.
- Моніторинг активності: Відображення хронологічного списку подій та дій, пов'язаних із проектом.

3.5.2. Інтерфейс управління задачами

Екран дошки задач (Task Board Page) доступний за навігаційним шляхом, наприклад, Проекти > Кнопка "Дошка завдань". Цей інтерфейс реалізує візуалізацію робочого процесу за методологією Kanban. Завдання проекту представлені на дошці та розподілені за чотирма основними станами виконання, що відображаються у вигляді списків (колонок): "До виконання" (To Do), "В процесі" (In Progress), "Перевірка" (Review) та "Виконано" (Done). Для кожного завдання, відображеного на дошці, передбачені інтерактивні елементи (наприклад, посилання або кнопки), що надають можливість здійснити його перегляд, редагування або видалення. З цього екрану також забезпечується доступ до функції створення нового завдання та можливість повернення до екрану деталей відповідного проекту.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

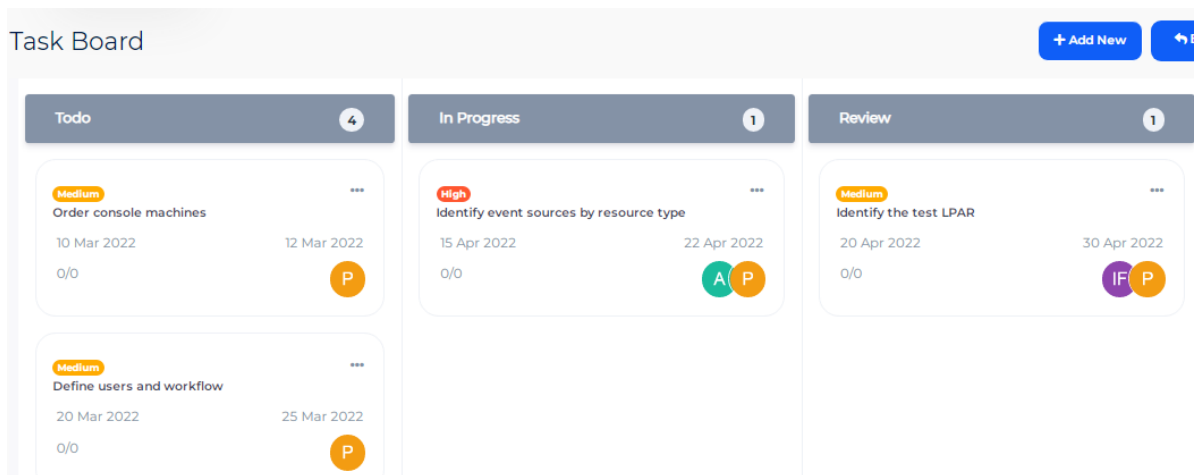


Рисунок 3.13 - Екран дошки задач

Екран створення задачі (Create Task Page) призначений для специфікації параметрів нового завдання в межах проекту. Користувачеві надаються поля для введення необхідної інформації про завдання. Збереження конфігурації нового завдання здійснюється шляхом активації функції збереження.

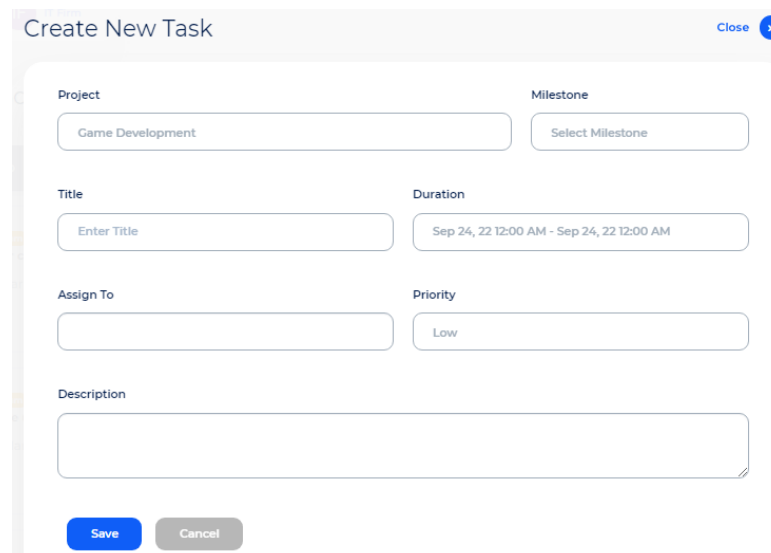


Рисунок 3.14 - Екран створення задачі

Екран перегляду задачі (Task View Page): Доступний за навігаційним шляхом, наприклад, Проекти > Дошка завдань > Перегляд. Цей інтерфейс надає деталізовану інформацію про конкретне завдання, включаючи його

опис, призначеного виконавця та інші атрибути. На даному екрані також реалізовано функціональність для взаємодії, що включає додавання коментарів, управління пов'язаними файлами та декомпозицію завдання на підзавдання.

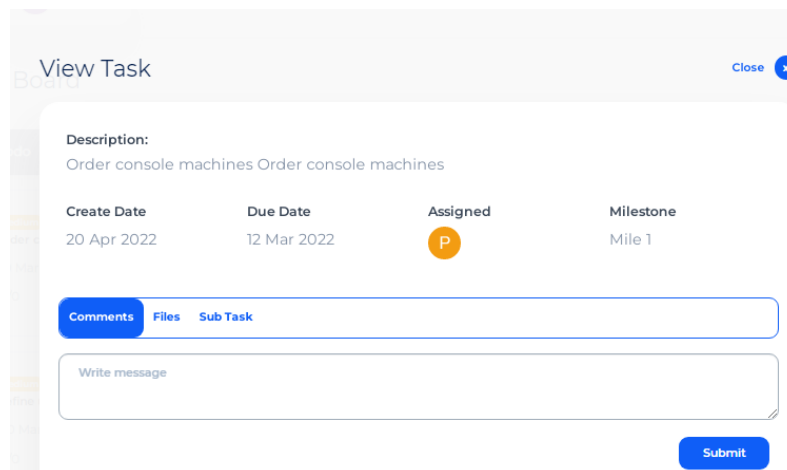


Рисунок 3.15 - Екран перегляду задачі

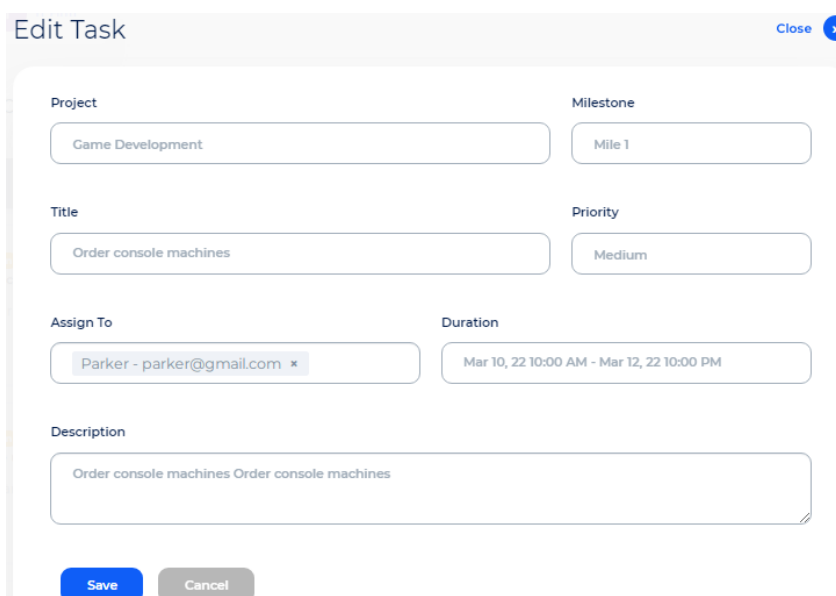


Рисунок 3.16 - Екран редагування задачі

Екран редагування задачі (Task Edit Page): Доступний за навігаційним шляхом, наприклад, Проекти > Дошка задач > Редагувати. Цей інтерфейс призначений для модифікації існуючих атрибутів завдання. Користувач може

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

внести необхідні зміни до інформації про завдання, а збереження оновлених даних здійснюється шляхом активації функції збереження.

3.6. Інтерфейс спеціалізованих модулів системи управління завданнями: облік часу та фінансове управління

Система управління завданнями (TMS) також включає спеціалізовані модулі, що розширюють її функціональність для підтримки обліку робочого часу, візуалізації проектного розкладу та фінансового управління, зокрема роботи з рахунками-фактурами.

3.6.1. Модуль обліку робочого часу

Екран таблиця обліку робочого часу (Timesheet Page): Доступ до цього інтерфейсу здійснюється через навігаційний шлях, наприклад, Проекти > Кнопка "Табель обліку робочого часу". Даний модуль дозволяє користувачам реєструвати кількість годин, витрачених на виконання конкретних завдань протягом певного дня. Система агрегує ці дані, надаючи тижневий огляд загального часу, інвестованого у проект. Ініціація створення нового запису в таблиці обліку робочого часу відбувається шляхом взаємодії з відповідним елементом інтерфейсу (наприклад, позначеним колом).

Task	Monday 11 April	Tuesday 12 April	Wednesday 13 April	Thursday 14 April	Friday 15 April	Saturday 16 April	Sunday 17 April	Total
Project Name: Game Development								
Task Name: Design the solution	-	01:00	-	02:30	-	03:30	-	07:00
Project Name: Game Development								
Task Name: Identify event sources by resource type	-	-	-	-	-	03:00	-	03:00
Total	-	01:00	-	06:30	-	06:30	-	10:00

Рисунок 3.17 - Екран таблиця обліку робочого часу

Екран створення табеля обліку робочого часу - цей інтерфейс призначений для внесення необхідної інформації щодо відпрацьованого часу. Збереження даних здійснюється шляхом активації функції збереження (наприклад, кнопки "Зберегти").

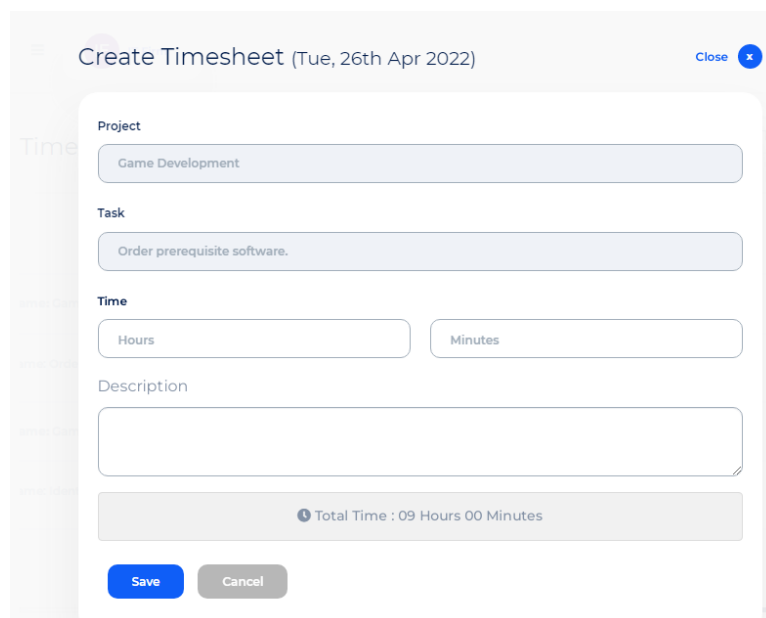


Рисунок 3.18 - Екран створення табеля обліку робочого часу

3.6.2. Модуль візуалізації розкладу

Екран діаграми Ганта (Gantt Chart Page) доступний за навігаційним шляхом, наприклад, Проекти > Кнопка "Діаграма Ганта".

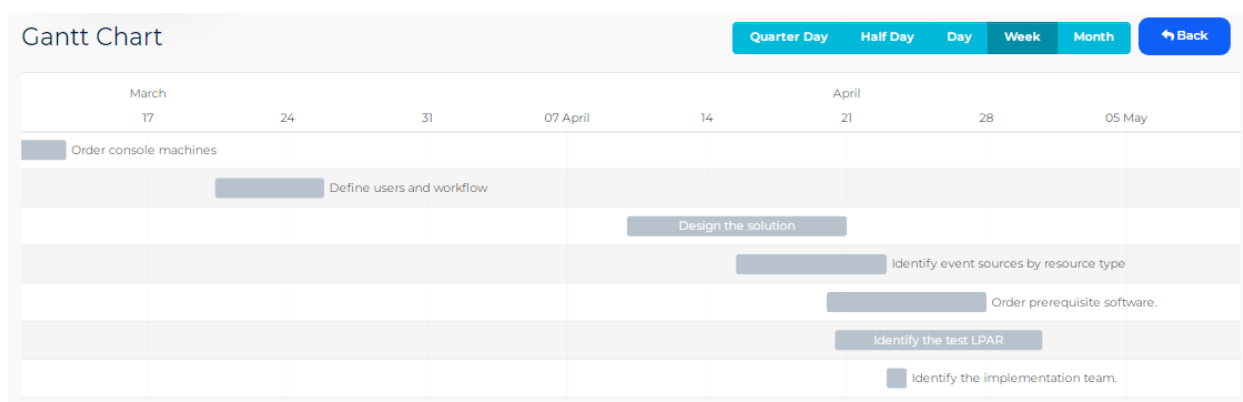


Рисунок 3.19 - Екран діаграми Ганта

Цей інтерфейс візуалізує тривалість виконання завдань проекту у часі. Діаграма підтримує відображення розкладу з різним ступенем деталізації: чверть дня, половина дня, день, тиждень та місяць, що дозволяє здійснювати моніторинг прогресу на різних рівнях гранулярності.

3.6.3. Модуль управління задачами

Екран індексу задач (Task Index Page): Доступ до цього екрану надається через відповідний пункт у бічному меню системи ("Завдання"). Інтерфейс представляє перелік усіх доступних користувачеві завдань. З цього екрану реалізована можливість безпосереднього редагування та видалення завдань.

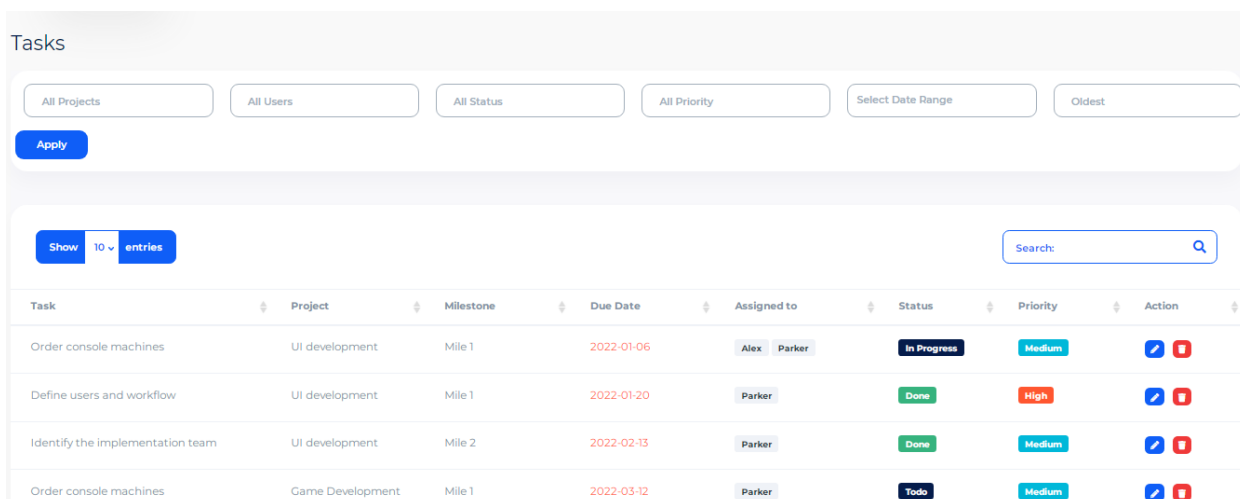


Рисунок 3.20 - Екран індексу задач

3.6.4. Модуль управління фінансами (Рахунки-фактури)

Екран індексу рахунків-фактур (Invoice Index Page). Доступ до цього інтерфейсу здійснюється через відповідний пункт у бічному меню ("Рахунки-фактури"). На даному екрані відображаються деталі рахунків-фактур, які можуть бути пов'язані з конкретними проектами. З цієї сторінки користувач має можливість ініціювати створення нового рахунку-фактури, а також редагувати чи видаляти існуючі записи.

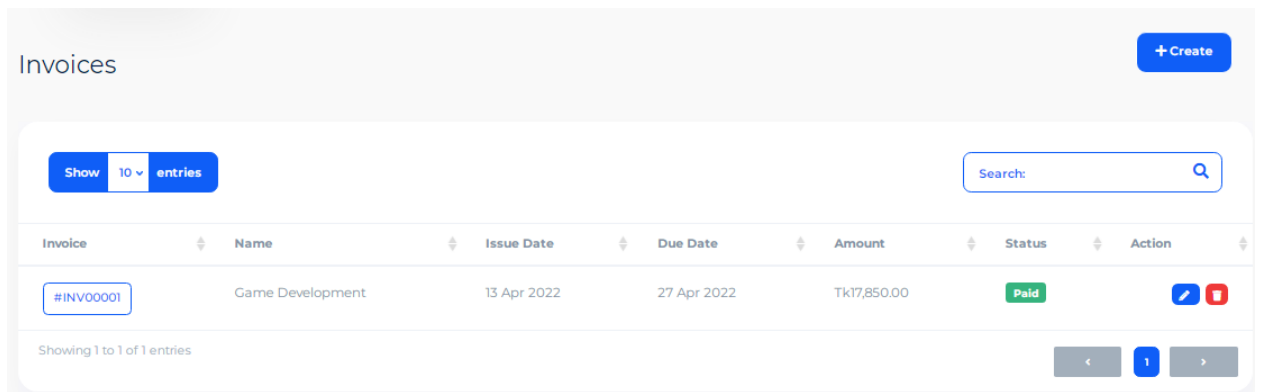


Рисунок 3.21 - Екран індексу рахунків-фактур

Екран створення рахунку-фактури (Create Invoice Page) доступний за навігаційним шляхом, наприклад, Рахунки-фактури > Кнопка "Створити". Цей інтерфейс призначений для внесення всієї необхідної інформації для формування нового рахунку-фактури. Збереження даних та створення документа виконується шляхом активації функції збереження (наприклад, кнопки "Зберегти").

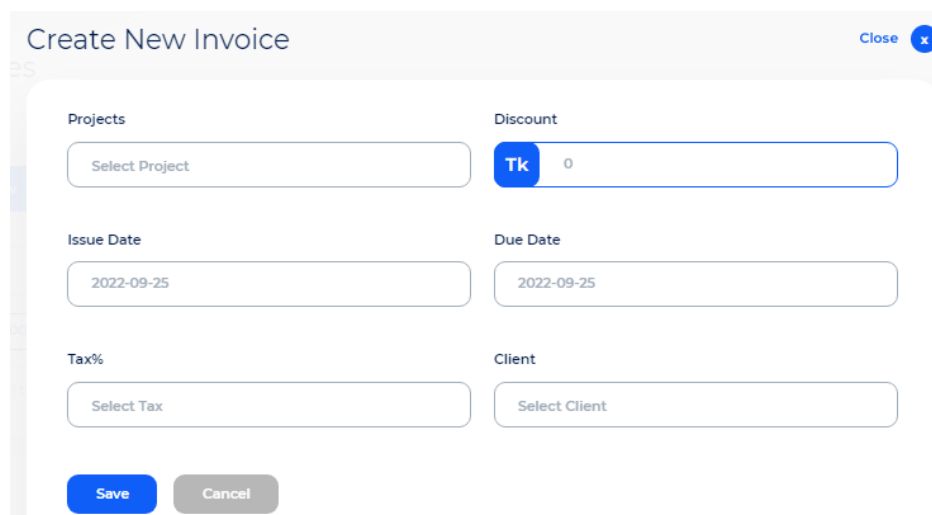


Рисунок 3.22 - Екран створення рахунку-фактури

Екран деталей рахунку-фактури (Invoice Details Page) відкривається після вибору конкретного рахунку-фактури (наприклад, натисканням на номер рахунку-фактури). На цьому екрані відображається повна інформація про рахунок-фактуру, включаючи платіжну адресу, адресу клієнта, резюме

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

замовлення/послуг тощо. Система надає функціональність для генерації та друку рахунка-фактури у форматі PDF. Також на цій сторінці відображається перелік усіх платежів, асоційованих з даним рахунком-фактурою.

Invoices

#INV00001

From:
CGIT
Muradpur, Chattogram
Chattogram, Muradpur - 2020
Bangladesh
02-12677

To:
Oliver
Dhanmondi, Dhaka
Dhaka, Dhanmondi - 4001
Bangladesh
oliver@gmail.com
02-4564

Project:
Game Development

Status:
Paid

Issue Date:
13 Apr 2022

Due Date:
27 Apr 2022

Рисунок 3.23 - Екран деталей рахунку-фактури

Екран додавання елементу до рахунку-фактури (Add Item Page) доступний за навігаційним шляхом, Рахунки-фактури > Кнопка "Номер рахунку-фактури" > Кнопка "Додати елемент". Цей інтерфейс дозволяє додавати позиції до рахунку-фактури, зокрема завдання з відповідними цінами, пов'язані з конкретним проектом.

Add Item

Close

Tasks

Select Task

Price

Tk 0

Add Cancel

Рисунок 3.24 - Екран додавання елементу до рахунку-фактури

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

Екран редагування рахунку-фактури (Invoice Edit Page): Доступний за навігаційним шляхом, наприклад, Рахунки-фактури > Кнопка "Номер рахунку-фактури" > Кнопка "Редагувати рахунок-фактуру". Цей інтерфейс призначений для внесення змін до існуючого рахунку-фактури. Після корегування інформації її збереження здійснюється шляхом активації функції збереження (наприклад, кнопки "Зберегти").

Рисунок 3.25 - Екран редагування рахунку-фактури

3.7. Інтерфейс функціональності для адміністратора та клієнта

Система управління завданнями (TMS) надає інтерфейси для управління користувачами та клієнтами, генерації аналітичних звітів, конфігурації робочого простору, а також специфічні можливості для ролі клієнта, що розширює адміністративні та моніторингові функції системи.

3.7.1. Модуль управління клієнтами

Екран індексу клієнтів доступ до цього інтерфейсу надається через відповідний елемент у бічному навігаційному меню ("Клієнти"). Екран відображає перелік зареєстрованих клієнтів організації. Функціональність

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

даного інтерфейсу дозволяє здійснювати додавання нових клієнтів, а також редагування або видалення існуючих записів про клієнтів.

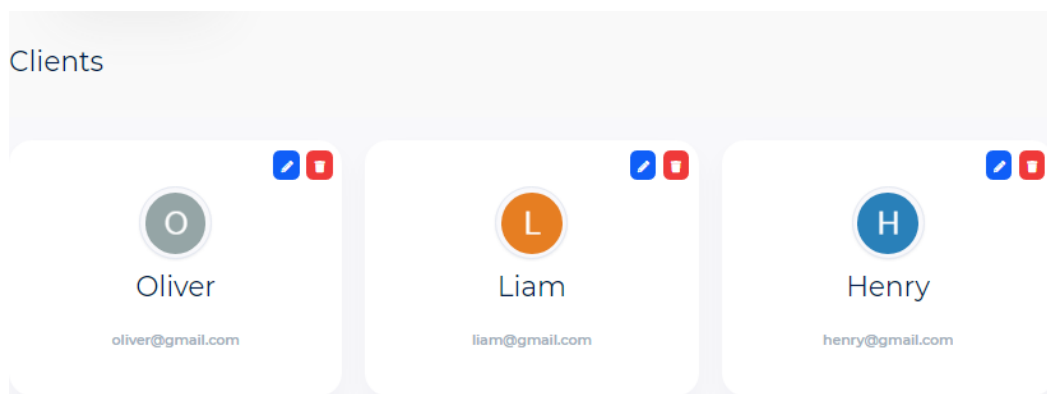


Рисунок 3.26 - Екран індексу клієнтів

Екран додавання клієнта доступний за навігаційним шляхом, наприклад, Клієнти > Додати клієнта. Цей інтерфейс призначений для внесення всієї необхідної інформації для реєстрації нового клієнта в системі. Створення запису клієнта здійснюється шляхом активації функції збереження (наприклад, кнопки "Створити").

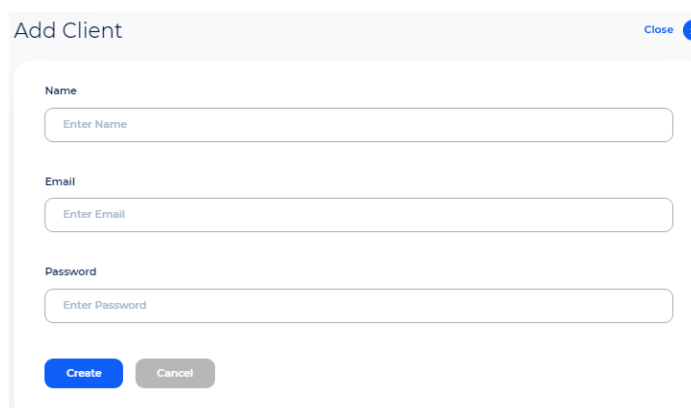


Рисунок 3.27 - Екран додавання клієнта

3.7.2. Модуль звітності

Система передбачає функціональність генерації звітів, доступ до якої надається через випадаючий список у відповідному розділі.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

Звіт про задачу доступний за навігаційним шляхом Звіти > Звіт про завдання. Цей звіт надає можливість фільтрувати інформацію про завдання за назвою проекту та датою створення або виконання. Звіт може бути експортований у форматі PDF для подальшого аналізу або архівування.

Task Report of Workspace 1

Task	Project	Milestone	Due Date	Assigned to	Status	Priority
Order console machines	Game Development	Mile 1	2022-03-12	Parker	Todo	Medium
Define users and workflow	Game Development		2022-03-25	Parker	Todo	Medium
Identify event sources by resource type	Game Development	Mile 3	2022-04-22	Alex Parker	In Progress	High
Order prerequisite software.	Game Development		2022-04-27	Peter	Todo	Low
Identify the test LPAR	Game Development		2022-04-30	IT Firm Peter	Review	Medium
Design the solution	Game Development		2022-04-20	Alex	Done	Low
Identify the implementation team.	Game Development	Mile 2	2022-04-23	Parker	Todo	Medium

Рисунок 3.28 - Звіт про задачу

Звіт про таблиць обліку робочого часу доступний за навігаційним шляхом Звіти > Звіт про таблиць обліку робочого часу. Цей звіт дозволяє фільтрувати дані обліку робочого часу за назвою проекту та іменем користувача. Звіт також підтримує експорт у форматі PDF.

Weekly Timesheet of Workspace 1

Task	Monday 18 April	Tuesday 19 April	Wednesday 20 April	Thursday 21 April	Friday 22 April	Saturday 23 April	Sunday 24 April	Total
Project Name: Game Development								
Task Name: Design the solution								
IT Firm	-	-	00:30	-	-	-	-	00:30
Task Name: Identify event sources by resource type								
IT Firm	01:00	-	00:30	-	-	-	-	01:30
Task Name: Order prerequisite software.								
IT Firm	-	-	-	03:10	-	01:20	-	04:30

Рисунок 3.29 - Звіт про таблиць обліку робочого часу

3.7.3. Модуль управління користувачами

Екран індексу користувачів. Доступ до цього інтерфейсу надається через відповідний елемент у бічному навігаційному меню ("Користувачі"). Екран відображає перелік користувачів, зареєстрованих у системі. Користувач з відповідними правами (наприклад, власник робочого простору) може звідси ініціювати процес запрошення інших користувачів (учасників) до системи. На цій сторінці також відображаються статистичні дані щодо загальної кількості призначених проєктів та завдань для кожного користувача.

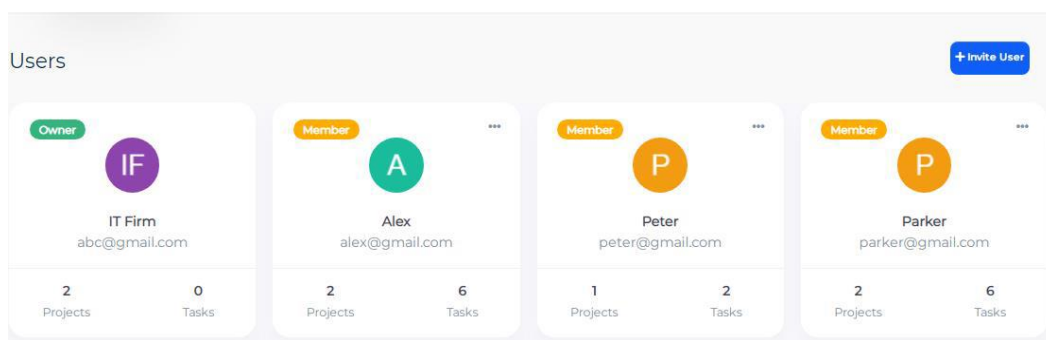


Рисунок 3.30 - Екран індексу користувачів

3.7.4. Модуль візуалізації задач (Календар)

Екран календаря. Доступ до цієї функції надається через меню "Календар". Інтерфейс календаря візуалізує усі завдання системи у форматі календаря, що дозволяє відстежувати терміни та розподіл робіт у часі.

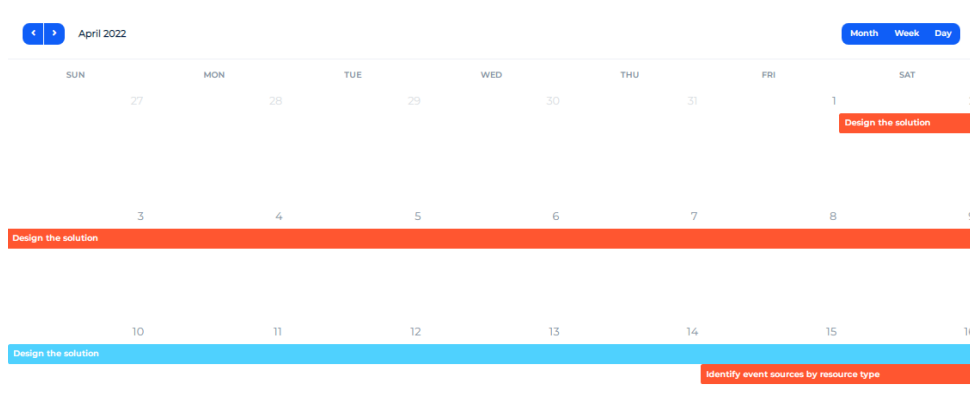


Рисунок 3.31 - Екран календаря

Змн.	Арк.	№ докум.	Підпис	Дата

3.7.5. Модуль конфігурації робочого простору

Екран налаштувань робочого простору доступний за навігаційним шляхом Налаштування робочого простору -> Налаштування сайту. Цей інтерфейс дозволяє здійснювати базові налаштування візуального оформлення та ідентифікації робочого простору, зокрема змінювати логотип сайту та його назву.

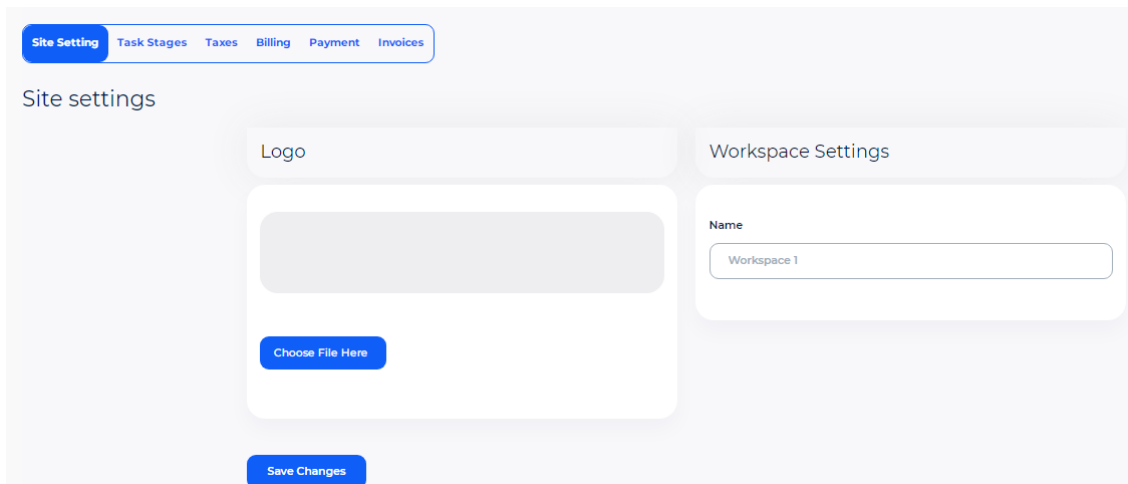


Рисунок 3.32 - Екран налаштувань робочого простору

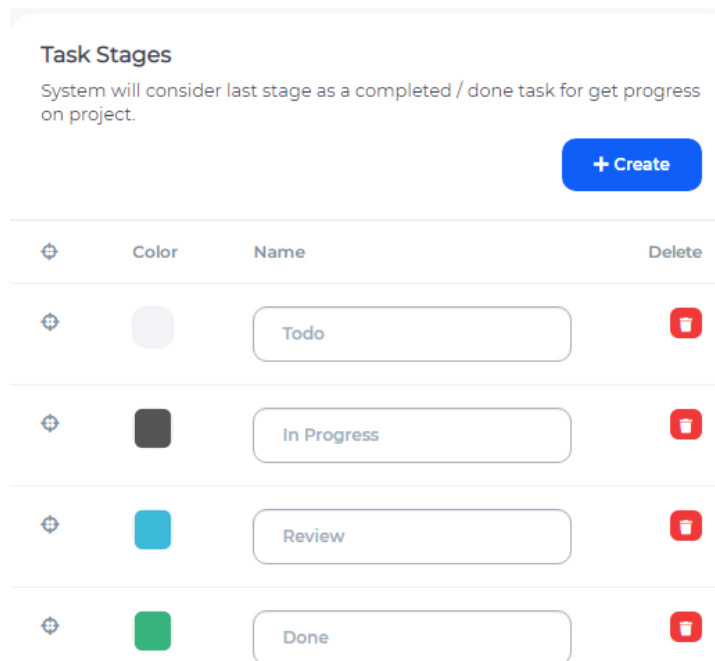
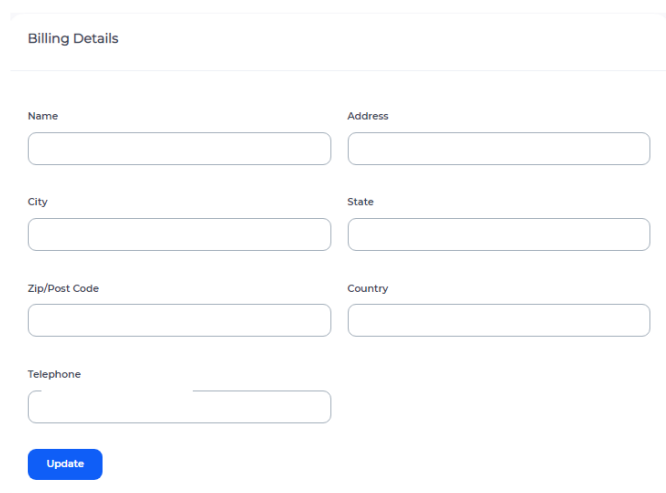


Рисунок 3.33 - Екран етапів задач

Екран етапів задач (рис. 3.33) доступний за навігаційним шляхом Налаштування робочого простору -> Етапи завдань. Цей інтерфейс призначений для конфігурації життєвого циклу завдань шляхом визначення та управління етапами виконання. В системі передбачено логіку, за якою досягнення останнього визначеного етапу вважається завершенням завдання, що використовується для розрахунку прогресу проекту. Інтерфейс дозволяє створювати рівні етапів завдань, а також створювати та редагувати конкретні етапи.

Екран деталей оплати доступний за навігаційним шляхом Налаштування робочого простору -> Оплата. Цей інтерфейс призначений для оновлення та збереження платіжних реквізитів, пов'язаних з робочим простором або обліковим записом.



The image shows a web form titled "Billing Details". It contains the following fields:

- Name:
- Address:
- City:
- State:
- Zip/Post Code:
- Country:
- Telephone:

At the bottom of the form is a blue button labeled "Update".

Рисунок 3.34 - Екран деталей оплати

3.7.6. Інтерфейс для ролі клієнта

Панель моніторингу клієнта. Цей інтерфейс є основним екраном для користувачів з роллю Клієнта, надаючи доступ до інформації, релевантної їхнім проектам та взаємодії з організацією. Панель включає доступ до Профілю клієнта, де може відображатись та редагуватись контактна інформація та інші реквізити клієнта.

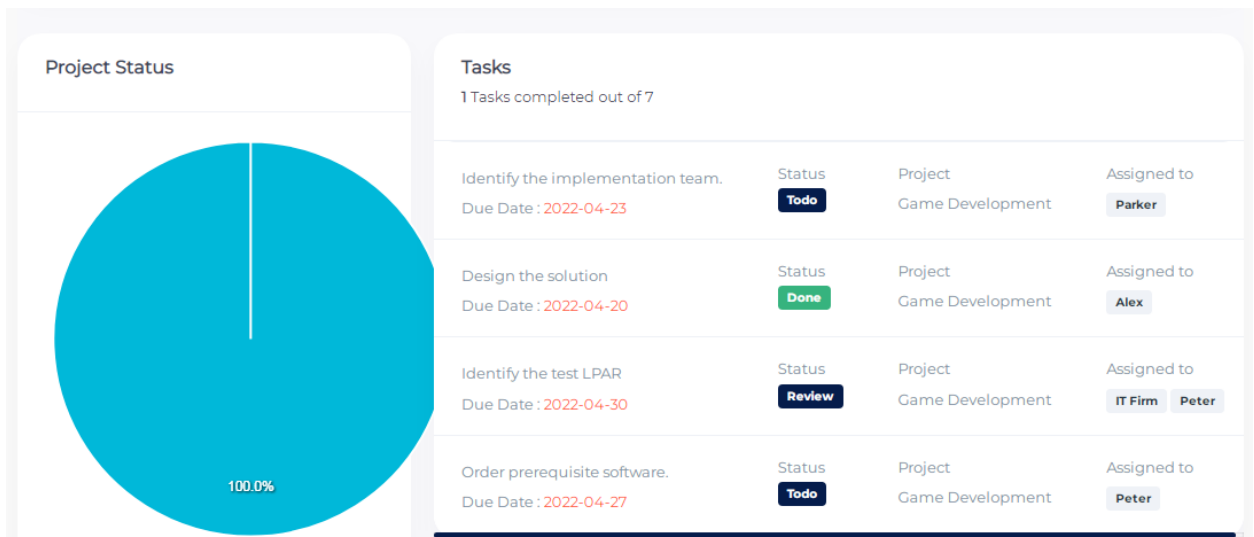


Рисунок 3.35 - Панель моніторингу клієнта

Account Change Password **Billing Details**

Address
Dhanmondi, Dhaka

City State

Zip/Post Code Country

Telephone

Update

Рисунок 3.36 – Екран профілю клієнта

3.8. Тестування системи

Тестування означає процес аналізу програмного елемента для виявлення різниць між існуючими або необхідними умовами та оцінки характеристик програмних елементів. Перед випуском програмного забезпечення його необхідно ретельно протестувати за допомогою численних тестових випадків та режимів, щоб переконатися, що воно не містить помилок і споживає мінімально можливу кількість місця та часу. Тестові

випадки були вибрані заздалегідь, з визначеними очікуваними результатами та записаними фактичними результатами для порівняння. Вибір тестових випадків здійснюється за допомогою техніки "Тестування білої скриньки" для перевірки внутрішньої логіки програмування та ефективності, а також техніки "Тестування чорної скриньки" для перевірки відповідності програмного забезпечення вимогам, щоб знайти максимальну кількість помилок з мінімальними зусиллями та часом. Хоча тестові випадки розробляються з урахуванням цикломатичної складності (СУС), умовного тестування, програмний код все ще не є оптимальним, оскільки не враховуються всі інші можливі альтернативні частини в програмному забезпеченні. На рівні інтеграції програмне забезпечення проходить тестування третіми сторонами, що ще більше підвищить оптимальність та ефективність програмного забезпечення.

Навантажувальне тестування (Load Testing) — це нефункціональна процедура тестування програмного забезпечення, призначена для оцінки його продуктивності під заздалегідь визначеним очікуваним навантаженням. Воно дозволяє визначити поведінку програмного додатку в умовах одночасної роботи численних користувачів. Метою проведення навантажувального тестування перед розгортанням програмного забезпечення є ідентифікація та усунення вузьких місць продуктивності, а також забезпечення його стабільності та безперебійної роботи.

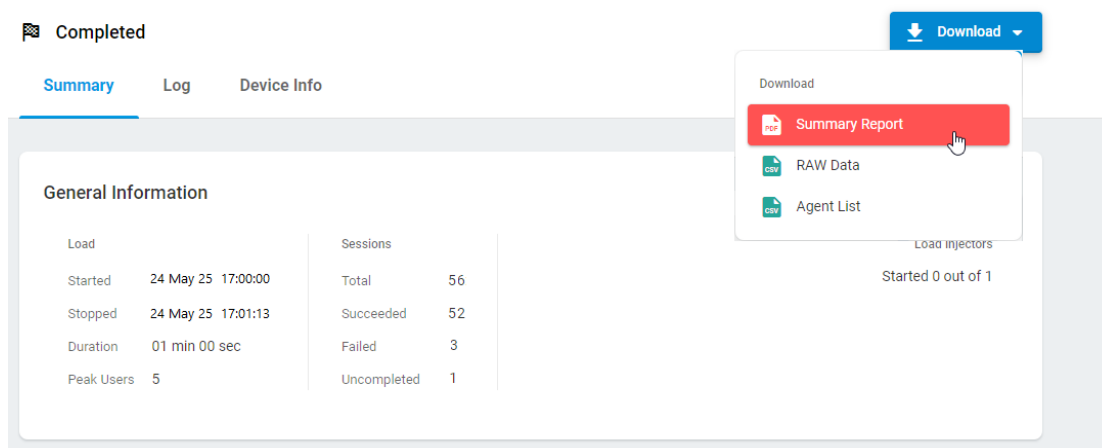


Рисунок 3.37 - Звіт навантажувального тестування

Навантажувальне тестування мого вебсайту було проведено за допомогою сервісу dotcom-monitor.com. Отримані результати представлено на рисунку 3.37.

Звіти навантажувального тестування зазвичай включають наступні розділи: Загальні відомості (Summary), Журнал (Log) та Інформація про пристрій (Device Info). Звіти, що містять необроблені дані (Raw Data), представляють статистичні показники, такі як час початку (Start Time), тривалість (Duration), час з'єднання (Connection Time), час отримання першого пакету (First Packet time), та інші параметри тестування у форматі файлу CSV.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

ВИСНОВКИ

В дипломній роботі були визначені теоретичні та практичні аспекти побудови системи керування задачами з урахуванням вимог масштабованості на рівні команди. На основі аналізу сучасних підходів до управління програмними проектами та застосування Agile-методологій встановлено, що ефективне масштабування систем управління завданнями є ключовим чинником у підтримці продуктивності та узгодженості команд розробки в умовах зростаючої складності та динаміки проектного середовища.

У першому розділі досліджено теоретичні засади управління завданнями, проаналізовано вимоги до проектування таких систем, розглянуто методологію розробки, а також надано детальний огляд існуючих фреймворків для масштабування Agile-підходів — зокрема, SAFe, LeSS і Scrum@Scale. Ці фреймворки дозволяють розширювати практики Scrum на рівні організації, забезпечуючи координацію між кількома командами та уніфікацію процесів розробки.

У другому розділі було систематизовано підходи до управління проектами та завданнями в інженерії ПЗ, а також розглянуто особливості побудови функціональності системи на рівні команд. Було сформульовано функціональні та нефункціональні вимоги до системи, зокрема щодо продуктивності, масштабованості та гнучкості в адаптації до різних командних структур. Представлено логічну модель системи, схеми потоків даних, а також алгоритм автентифікації користувачів, що забезпечує безпечний доступ до функціональності відповідно до ролей.

У третьому розділі представлено програмну реалізацію системи з урахуванням принципів масштабованої архітектури. Розроблена ER-діаграма демонструє чітку структуру сутностей (користувачі, проекти, завдання, клієнти, робочі простори тощо) та їх взаємозв'язки, що забезпечують розширюваність і гнучкість системи при збільшенні кількості команд. Також

					БР.ІП – 27.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

реалізовано модульну побудову інтерфейсів для різних ролей: адміністратора, менеджера, розробника, клієнта. Особливу увагу приділено спеціалізованим модулям: обліку робочого часу, управлінню фінансами, моніторингу активностей, візуалізації розкладу та звітності.

Проведене тестування підтвердило коректність функціонування системи, її здатність обробляти паралельні запити з боку користувачів у рамках окремих команд, а також дотримання вимог до масштабованості та безпеки. Інтерфейсна частина орієнтована на інтуїтивну взаємодію з користувачем, а реалізовані механізми дозволяють ефективно інтегрувати систему в організаційну структуру ІТ-компанії будь-якого розміру.

Таким чином, запропонована методологія масштабування системи керування задачами забезпечує комплексне вирішення завдань координації, контролю, комунікації та продуктивності в командах розробки. Результати роботи можуть бути використані для впровадження у реальні ІТ-проекти, а також як основа для подальших досліджень у галузі автоматизації та оптимізації процесів управління життєвим циклом програмного забезпечення.

					БР.ІП – 27.00.00.000 ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kerzner, H. (2017). Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley.
2. PMI (2021). A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Seventh Edition. Project Management Institute.
3. Beck, K., Beedle, M., van Bennekum, A. et al. (2001). Manifesto for Agile Software Development. <https://agilemanifesto.org>
4. Highsmith, J. (2010). Agile Project Management: Creating Innovative Products. Addison-Wesley.
5. Kniberg, H., & Ivarsson, A. (2012). Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds. Spotify White Paper.
6. Microsoft DevOps Resource Center. (2023). Agile Tools and Practices in Azure DevOps. Microsoft Docs.
7. Why Is Task Management Important For Your Organization - <https://www.taskopad.com/blog/why-task-tracking-is-important/#:~:text=Task%20management%20is%20necessary%20for>
8. McKinsey & Company. (2018). How ING transformed its agile operating model. Retrieved from: <https://www.mckinsey.com>
9. IBM Agile Academy. (2022). Scaling Agile at IBM: From Teams to Enterprises. IBM White Paper.
10. What is a Sprint Burndown Chart & What is its Significance? - | Unichrone - <https://unichrone.com/blog/agile/sprint-burndown-chart/>
11. ressman, R. S. (2010). Software Engineering: A Practitioner's Approach (7th ed.). McGraw-Hill Education.
12. Kerzner, H. (2017). Project Management: A Systems Approach to Planning, Scheduling, and Controlling (12th ed.). Wiley.
13. The Standish Group. (2011). CHAOS Report 2011. (Note: Specific report details vary).

					БР.ІІІ – 27.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

- 14.Boehm, B. W. (1981). Software Engineering Economics. Prentice Hall.
- 15.Sauer, C. (1993). Why Information Systems Fail: A Case Study Approach. Blackwell Scientific Publications.
- 16.Anderson, D. J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.
- 17.Leffingwell, D. (2019). SAFe 4.5 Distilled: Achieving Business Agility with the Scaled Agile Framework. Pearson. (Reflecting a SAFe version around the time of potential study).
- 18.Larman, C., & Vodde, B. (2015). Large-Scale Scrum: More with LeSS. Addison-Wesley Professional.
- 19.Sutherland, J., & Viktorov, V. (2018). Scrum@Scale: The Definitive Guide. Scrum Inc.
- 20.Wiegers, K. E., & Beatty, J. (2013). Software Requirements (3rd ed.). Microsoft Press.
- 21.Robertson, S., & Robertson, J. (2012). Mastering the Requirements Process: Practical Steps for Succeeding with Real-World Requirements (3rd ed.). Addison-Wesley Professional.
- 22.DeMarco, T. (1978). Structured Analysis and System Specification. Yourdon Press.
- 23.Meredith, J. R., & Mantel Jr., S. J. (2011). Project Management: A Managerial Approach (8th ed.). Wiley.
- 24.Shtub, A., Bard, J. F., & Globerson, S. (2005). Project Management: Engineering, Management, and Construction (3rd ed.). Prentice Hall.
- 25.Lester, A. (2017). Project Planning, Scheduling & Control: A Hands-On Guide to Bringing Projects in On Time and On Budget (6th ed.). Butterworth-Heinemann.
- 26.Bentley, L. D., & Whitten, J. L. (2007). System Analysis & Design Methods (7th ed.). McGraw-Hill Irwin.

					БР.ІІІ – 27.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

27. Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley Professional.
28. Highsmith, J. (2009). Agile Project Management: Creating Innovative Products (2nd ed.). Addison-Wesley Professional.
29. Cohn, M. (2005). Agile Estimating and Planning. Prentice Hall.
30. Adzic, G. (2011). Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing. Self-Published.
31. Lerdorf, R., Tatroe, K., & MacIntyre, P. (2006). Programming PHP (2nd ed.). O'Reilly Media.
32. Self-correction: Stick to more foundational or formal academic sources.
33. Fenton, N. E., & Pfleeger, S. L. (1997). Software Metrics: A Rigorous and Practical Approach. PWS Publishing.
34. AACE International. (Year). Total Cost Management Framework. AACE International.
35. Dingsøyr, T., Dyba, T., & Moe, N. B. (Eds.). (2014). Agile Software Development: Current Research and Future Directions. Springer.

					БР.ІІІ – 27.00.00.000 ІІЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: “ **Методологія масштабування системи керування задачами розробки по рівню команди** ”

Обсяг пояснювальної записки: 79 аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента _____