



Я, Руденко Максим Олександрович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«19» грудня 2025р

Handwritten signature of Maxim Rudenko in black ink, featuring a stylized 'M' and 'R'.

Максим РУДЕНКО

## Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологійКафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехнікиРівень вищої освіти другий (магістерський)Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(код і повна назва)

Тип програми Освітньо-професійнаОсвітня програма Комп'ютерно-інтегровані технологічні процеси і виробництва

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу Руденку Максиму Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення CRM системи для автоматизації бізнес-процесів на виробничому підприємствізатверджена наказом університету від «10» листопада 2024 р. №. 1029Ст2. Термін подання здобувачем роботи до екзаменаційної комісії 19 грудня 2025 р.

3. Вихідні дані до роботи \_\_\_\_\_

3.1 CRM система для автоматизації бізнес-процесів на виробничому підприємстві;

3.2 Середовище розробки Visual Studio Code;

3.3 Засоби моделювання MySQL Workbench, Draw.io;

3.3 Мова програмування PHP;

3.4 Технічне забезпечення ПК з процесором Core i7.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

4.1 Вступ;

4.2 Аналіз предметної області та постановка завдання;

4.3 Огляд існуючих CRM та їх придатність для виробничих підприємств;

4.4 Проектування та розроблення crm-системи;

4.5 Тестування та оцінка надійності розробленої CRM-системи;

4.6 Питання пов'язані з охороною праці;

4.7 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Графічний матеріал у вигляді презентації – 15 арк. ф. А 4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.09.25-15.09.25	Виконано
2	Аналіз літератури та супутніх джерел	15.09.25-23.09.25	Виконано
3	Аналіз предметної області	24.09.25-29.10.25	Виконано
4	Розробка вимог до CRM системи	30.09.25-15.10.25	Виконано
5	Розробка CRM системи	16.10.25-01.11.25	Виконано
6	Аналіз результатів проведеної роботи	01.11.25-15.11.25	Виконано
7	Оформлення пояснювальної записки	16.11.25-30.11.25	Виконано
8	Оформлення графічної частини	01.12.25-03.12.25	Виконано
9	Підготовка презентації	04.12.25-07.12.25	Виконано
10	Подання роботи на нормоконтроль	07.12.25-16.12.25	Виконано
11	Подання кваліфікаційної роботи на перевірку академічної доброчесності	15.12.25-16.12.25	Виконано
12	Подання роботи на рецензування	17.12.25-18.12.25	Виконано
13	Попередній захист	18.12.25	Виконано
14	Подання роботи до екзаменаційної комісії	19.12.25	Виконано

Дата видачі завдання «01» вересня 2025 р.

Здобувач



(підпис)

Максим РУДЕНКО

Керівник роботи

доц. Світлана СОТНИК

(підпис)

(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 110 с., 2 табл., 27 рис., 2 додатки, 20 джерел.

АВТОМАТИЗАЦІЯ, УПРАВЛІННЯ, CRM-СИСТЕМА, БАЗА ДАНИХ, ІНТЕРФЕЙС, ВИРОБНИЧЕ ПІДПРИЄМСТВО.

Метою роботи є підвищення ефективності управління клієнтами, завданнями та персоналом на виробничому підприємстві шляхом розроблення веб-орієнтованої CRM-системи для автоматизації бізнес-процесів і контролю виконання робіт у реальному часі.

Об'єкт дослідження – процеси управління бізнес-діяльністю виробничого підприємства.

Предмет дослідження – методи та алгоритми автоматизації процесів управління, планування та контролю виконання виробничих завдань у веб-орієнтованій CRM-системі.

У кваліфікаційній роботі розглянуто актуальні питання розробки веб-орієнтованої CRM-системи, орієнтованої на специфіку невеликого виробничого підприємства, яке поєднує базові інструменти управління клієнтами, проектами та завданнями. Особливу увагу приділено розробці структури та алгоритмів роботи системи у невеликих виробничих підприємствах.

Результати роботи відповідають Цілям сталого розвитку 9 та 12: «Промисловість, інновації та інфраструктура», «Відповідальне споживання та виробництво».

Практична цінність роботи полягає в можливості використання розробленої CRM-системи в реальних умовах діяльності виробничого підприємства.

## ABSTRACT

The master's thesis: 110 p., 2 table, 27 figures, 2 appendice, 20 sources.

AUTOMATION, MANAGEMENT, CRM SYSTEM, DATABASE, INTERFACE,  
MANUFACTURING ENTERPRISE.

The purpose of this work is to increase the efficiency of managing customers, tasks, and personnel at a manufacturing enterprise through the development of a web-based CRM system for business process automation and real-time monitoring of task execution.

The object of the research is the management processes of business activities within a manufacturing enterprise.

The subject of the research is the methods and algorithms for automating the processes of management, planning, and monitoring the execution of production tasks in a web-based CRM system.

This thesis examines relevant issues of developing a web-oriented CRM system tailored to the specifics of a small manufacturing enterprise, combining basic tools for managing customers, projects, and tasks. Particular attention is paid to the development of the structure and operational algorithms of the system in the context of small-scale manufacturing companies.

The results of the work are aligned with Sustainable Development Goals 9 and 12: "Industry, Innovation and Infrastructure" and "Responsible Consumption and Production."

The practical value of the work lies in the possibility of applying the developed CRM system under real operating conditions of a manufacturing enterprise.

## ЗМІСТ

Перелік скорочень .....	9
Вступ.....	10
1 Аналіз предметної області та постановка завдання.....	13
1.1 Функціональні можливості CRM-систем .....	13
1.1.1 Класифікація CRM-систем.....	15
1.1.2 Переваги та недоліки CRM-систем .....	18
1.1.3 Специфіка управління проєктами на виробничому підприємстві .....	20
1.2 Огляд існуючих CRM та їх придатність для виробничих підприємств .....	21
1.3 Обґрунтування необхідності розробки спеціалізованого рішення.....	26
1.4 Висновки до першого розділу.....	28
2 Проєктування та розроблення CRM-системи .....	29
2.1 Вибір технологічного стеку для реалізації CRM-системи.....	29
2.2 Вибір архітектури CRM-системи .....	31
2.3 Розроблення UML діаграм до розроблюваної CRM-системи .....	35
2.3.1 Розроблення UML-діаграми прецедентів використання CRM-системи .....	35
2.3.2 Розроблення UML-діаграми класів прецеденту «Керування завданнями» .....	39
2.3.3 Розроблення UML-діаграми послідовностей для успішного основного сценарію прецеденту «Цикл роботи з задачами». .....	41
2.4 Постановка завдання на розроблення CRM-системи.....	44
2.5 Проєктування схеми бази даних (ER-діаграма).....	45
2.6 Розробка запитів до бази даних за допомогою мови програмування PHP .....	48
2.7 Розроблення логіки взаємодії користувача зі сторінками CRM-системи.....	53
2.8 Розроблення логіки онлайн чату для CRM-системи .....	64
2.9 Реалізація інтерфейсу користувача .....	70
3 Тестування та експериментальні дослідження .....	85
3.1 Функціональне тестування CRM-системи .....	85
3.2 Оцінка надійності CRM-системи на основі статистичної моделі.....	87

3.3 Аналіз стійкості розробленої CRM-системи для виробничого підприємства	89
3.4 Охорона праці .....	92
Висновки .....	95
Перелік джерел посилання .....	97
Додаток А Опубліковані результати.....	100
Додаток Б Демонстраційний матеріал.....	109

## ПЕРЕЛІК СКОРОЧЕНЬ

САУ – система автоматичного управління;

СУБД – система управління базами даних;

CRM – Customer Relationship Management;

ERP – Enterprise resource planning;

UML – Unified Modeling Language.

## ВСТУП

У сучасних умовах розвитку економіки виробничі підприємства стикаються з високим рівнем конкуренції, швидкими змінами ринку та зростаючими вимогами клієнтів до якості продукції та сервісу. Для збереження конкурентоспроможності підприємствам необхідно оперативно адаптувати бізнес-середовище, ефективно управляти взаємовідносинами з клієнтами та оптимізувати внутрішні процеси. Однак у процесі розвитку виробничі компанії зіштовхуються з низкою проблем: збільшенням кількості замовників, необхідністю систематизації даних про клієнтів, контролем виконання замовлень, підвищенням продуктивності персоналу та забезпеченням прозорості бізнес-процесів. Розв'язання цих завдань потребує впровадження сучасних інформаційних систем, серед яких особливе місце займають CRM-системи (Customer Relationship Management).

CRM-системи є важливими інструментами управління відносинами з клієнтами, оскільки вони дозволяють підприємству систематизувати інформацію про замовників, автоматизувати процеси взаємодії, вести історію співпраці, формувати звітність та аналізувати ефективність діяльності. Для виробничих підприємств це має особливе значення, адже їхня робота часто пов'язана з великою кількістю замовлень, виробничих циклів і партнерів, що вимагає чіткого контролю термінів, ресурсів і якості обслуговування.

Розроблення та впровадження інформаційних систем управління є важливим напрямом цифрової трансформації сучасних підприємств. У межах магістерської підготовки особливу увагу приділено поєднанню теоретичних засад автоматизації з практичними аспектами проектування та реалізації програмних рішень. Виконання кваліфікаційної роботи спрямоване на узагальнення та систематизацію знань з проектування інформаційних систем, веб-технологій, баз даних і методів

автоматизації бізнес-процесів, а також на формування здатності застосовувати ці знання для розв'язання прикладних інженерних завдань у виробничому середовищі.

Робота над магістерським проектом передбачає самостійне прийняття технічних і архітектурних рішень, вибір оптимальних інструментів розроблення та обґрунтування функціональних можливостей створюваної системи. У процесі дослідження здійснюється аналіз предметної області, формування вимог до програмного продукту, моделювання його структури та реалізація основних функцій. Такий підхід забезпечує розвиток професійних компетентностей у сфері автоматизації, програмної інженерії та інформаційних технологій, а також сприяє підготовці здобувача до подальшої фахової діяльності.

Актуальність дослідження зумовлена тим, що значна частина сучасних CRM-рішень є універсальними, надмірно складними або фінансово затратними для впровадження на малих і середніх виробничих підприємствах. Тому розроблення власної веб-орієнтованої CRM-системи, адаптованої під специфіку виробничої компанії, є важливим та практично значущим завданням.

Метою роботи є підвищення ефективності управління клієнтами, завданнями та персоналом на виробничому підприємстві шляхом розроблення веб-орієнтованої CRM-системи для автоматизації бізнес-процесів і контролю виконання робіт у реальному часі.

Об'єкт дослідження – процеси управління бізнес-діяльністю виробничого підприємства.

Предмет дослідження – методи та алгоритми автоматизації процесів управління, планування та контролю виконання виробничих завдань у веб-орієнтованій CRM-системі. Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести аналіз існуючих підходів до впровадження CRM-систем у діяльність виробничих підприємств та визначити їхні сильні й слабкі сторони;
- визначити бізнес-функції та вимоги до системи;

- обґрунтувати вибір технологічного стеку, який забезпечить ефективну реалізацію системи при мінімальних витратах ресурсів;
- розробити структуру та архітектуру веб-орієнтованої CRM-системи з урахуванням особливостей виробничої діяльності;
- реалізувати CRM-систему, що дозволить автоматизувати ключові процеси управління клієнтами, замовленнями, проектами та завданнями;
- провести тестування прототипу та оцінити його ефективність у контексті виробничого підприємства;
- сформулювати висновки та надати рекомендації щодо розвитку системи;
- оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [1] і методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка освітньої програми «Комп'ютерно-інтегровані технологічні процеси і виробництва» [2-12].

Тема розроблення CRM-системи для автоматизації бізнес-процесів на виробничому підприємстві безпосередньо узгоджується з Цілями сталого розвитку ООН, зокрема з Ціллю 8 «Гідна праця та економічне зростання» та Ціллю 9 «Індустріалізація, інновації та інфраструктура». Впровадження таких інформаційних систем сприяє підвищенню ефективності виробництва, оптимізації використання ресурсів, цифровій трансформації бізнесу та підвищенню конкурентоспроможності підприємств у глобальному середовищі. Реалізація проекту спрямована на підтримку інноваційної діяльності та сталого економічного розвитку шляхом автоматизації управлінських процесів і підвищення якості прийняття рішень.

Матеріали публікації тез на дотичну тему до теми кваліфікаційної роботи, що пройшли апробацію матеріалів у формі доповіді на конференції наведені у додатку А [13].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

## 1.1 Функціональні можливості CRM-систем

CRM-система – управління взаємовідносинами з клієнтами. По-суті, це програмне забезпечення для автоматизації та оптимізації процесів взаємодії з клієнтами, яке дозволяє збирати, зберігати та аналізувати дані про них, управляти продажами, маркетингом та сервісом, підвищувати лояльність клієнтів, збільшувати прибуток компанії та підтримувати внутрішні бізнес-процеси підприємств.

Основним призначенням CRM-системи є підвищення ефективності взаємодії між підприємством та його клієнтами, що, у свою чергу, веде до зростання продуктивності бізнесу, оптимізації витрат і підвищення прибутковості. В умовах сучасного бізнесу, коли конкуренція зростає, а вимоги клієнтів постійно змінюються, ефективна взаємодія з клієнтами стає ключовим чинником успіху будь-якого підприємства. Саме CRM-системи дозволяють не лише накопичувати та зберігати дані про клієнтів, а й забезпечувати їх структурований аналіз для прийняття обґрунтованих управлінських рішень. Головною метою використання таких систем є формування довготривалих і взаємовигідних відносин між компанією та її клієнтами, підвищення рівня задоволеності споживачів, оптимізація бізнес-процесів і, як результат, зростання прибутковості підприємства.

CRM-системи виконують широкий спектр функцій, які охоплюють різні напрями діяльності компанії, наприклад, управління продажами, сервісне обслуговування клієнтів, взаємодія між співробітниками, аналітика та звітність, управління маркетингом, управління контактами (рис. 1.1).

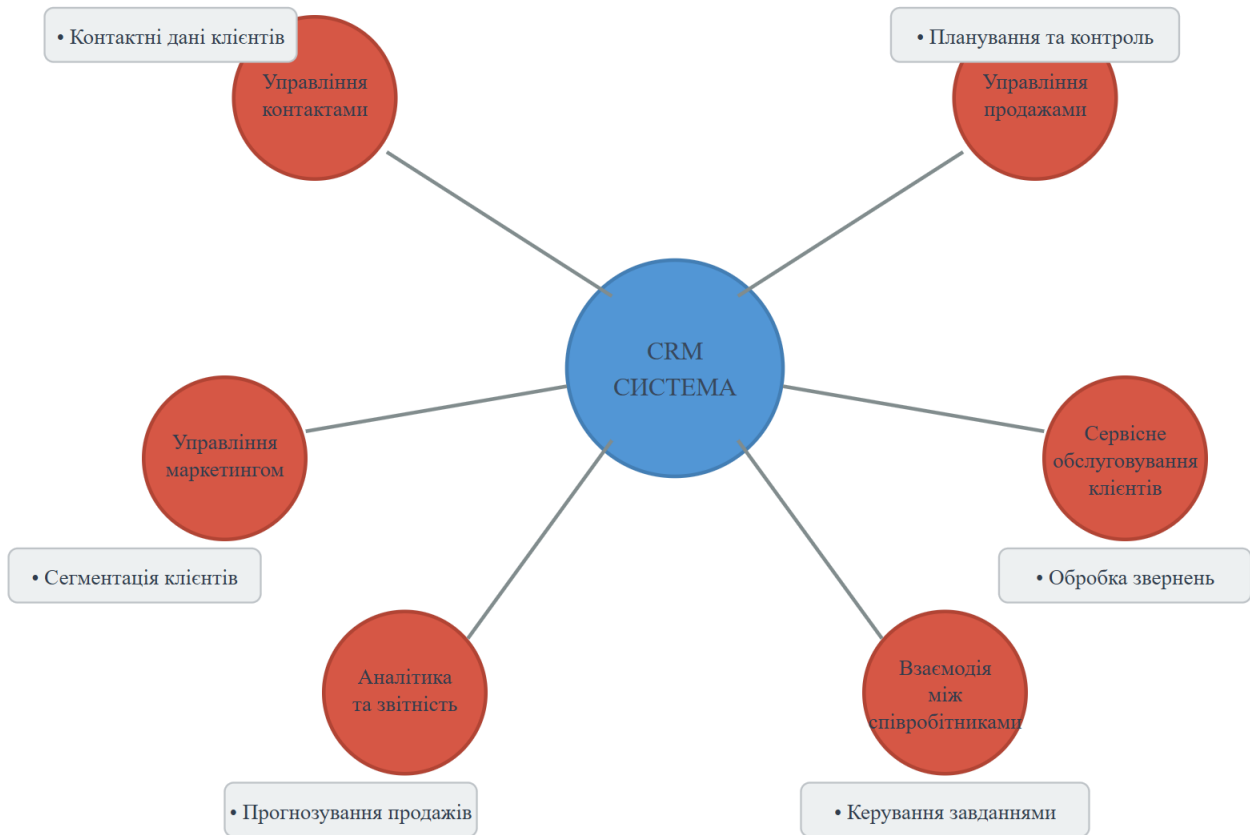


Рисунок 1.1 – Функції CRM-систем

Однією з базових функцій є управління контактами, що передбачає централізоване збереження всієї інформації про клієнтів, такої як контактні дані, історія взаємодії, угоди, звернення до служби підтримки. Це дозволяє швидко отримувати доступ до потрібних відомостей і забезпечує персоналізовану комунікацію з клієнтом. Важливою функцією є також управління продажами, яке включає планування та контроль роботи відділу продажів, відстеження статусів угод, аналіз ефективності менеджерів. Завдяки цьому компанія отримує можливість чітко бачити етапи руху клієнта від ліда до укладеної угоди та оптимізувати бізнес-процеси, що супроводжують продаж [14].

Не менш значущою функцією є управління маркетингом. CRM-системи дозволяють сегментувати клієнтську базу, формувати цільові аудиторії для проведення рекламних кампаній, автоматизувати розсилки та аналізувати їх

результати. Це сприяє підвищенню ефективності маркетингової діяльності та дає змогу компанії зосередитися на найбільш перспективних клієнтах. Крім того, важливим є забезпечення якісного сервісного обслуговування клієнтів. Сучасні CRM-системи включають модулі для підтримки клієнтів, що дозволяють обробляти звернення, фіксувати проблеми, контролювати їх вирішення та забезпечувати своєчасний зворотний зв'язок. Це формує довіру клієнтів до компанії та підвищує їхню лояльність.

Ще одним ключовим напрямом використання CRM є аналітика. Система надає інструменти для побудови звітів, прогнозування продажів, аналізу поведінки клієнтів та ефективності бізнес-процесів. Використання аналітичних можливостей дозволяє виявляти тенденції на ринку, своєчасно реагувати на зміни попиту, оптимізувати стратегії розвитку та забезпечувати конкурентоспроможність підприємства.

Також дуже важливим є реалізація через CRM-систему взаємодії між співробітниками. Система надає можливість керування завданнями, контролювати їх виконання, нагадує про дедлайни. У системі організовано внутрішній обмін інформацією між співробітниками. Така система забезпечує координацію дій між різними відділами компанії.

### 1.1.1 Класифікація CRM-систем

На сьогодні ринок CRM-рішень є досить різноманітним, для обґрунтованого вибору або розробки оптимальної системи саме для виробничого підприємства необхідно провести аналіз існуючих підходів до класифікації CRM-систем. Це дозволить визначити їхній функціонал, архітектурні особливості та сфери найефективнішого застосування [15].

CRM-системи можна класифікувати за кількома критеріями:

- за моделлю розгортання;
- за функціональним призначенням;

- за масштабом підприємства;
- за галузевою специфікою.

Найбільш поширеним є поділ на локальні та хмарні системи.

В результаті, визначено, що локальні CRM встановлюються безпосередньо на серверах підприємства і забезпечують повний контроль над даними та конфігурацією, але потребують значних витрат на обладнання та технічне обслуговування.

Тож, хмарні CRM працюють у вигляді веб-застосунків і надаються як сервіс «SaaS». Вони не потребують складної інфраструктури на боці підприємства, легко масштабуються та дозволяють отримувати доступ до системи з будь-якого місця, де є Інтернет.

Для малих і середніх компаній саме хмарні рішення часто є більш привабливими завдяки нижчій вартості впровадження та зручності у використанні.

З точки зору функціонального призначення CRM-системи можна поділити на три основні типи:

- операційні;
- аналітичні;
- колаборативні.

Розглянемо спочатку операційні CRM зосереджені на автоматизації щоденних бізнес-процесів, таких як: управління продажами, ведення бази клієнтів, контроль угод, організація роботи з контактами та завданнями.

Далі розглянемо аналітичні CRM, які роблять акцент на збиранні, обробці й аналізі даних про клієнтів і ринок. Вони допомагають виявляти закономірності, прогнозувати попит і приймати обґрунтовані управлінські рішення.

Розглянемо колаборативні CRM, які забезпечують ефективну взаємодію між різними підрозділами компанії, що особливо актуально для підприємств, де важлива тісна координація між відділом продажів, маркетингом і технічною підтримкою. На практиці більшість сучасних рішень поєднують елементи цих трьох напрямів.

За масштабом підприємства CRM-системи поділяють на рішення для малого, середнього та великого бізнесу.

Для невеликих компаній характерні прості та зручні інтерфейси з базовим функціоналом, який дозволяє вести клієнтську базу, контролювати угоди та виконання завдань.

Середній бізнес потребує більшої кількості інструментів, зокрема модулів аналітики, управління маркетингом та інтеграції з іншими системами (наприклад, бухгалтерськими або ERP).

Для великих корпорацій призначені масштабні CRM-платформи, які підтримують складну ієрархію користувачів, інтеграцію з ERP і BI-системами та глибоку кастомізацію під потреби різних відділів.

Окрему групу становлять галузеві CRM-системи, створені з урахуванням специфіки конкретної сфери. Наприклад, для виробничих підприємств CRM може містити інструменти управління постачальниками та партнерами, планування виробничих замовлень і відстеження етапів виконання.

Для сфери освіти CRM пристосовані для ведення обліку студентів, комунікації з викладачами та контролю навчальних процесів.

У медичній сфері CRM забезпечують збереження медичних карток пацієнтів, контроль запису на прийом і взаємодію з лабораторіями.

Таким чином, галузеві CRM-рішення враховують особливості ринку та допомагають максимально ефективно налаштувати роботу підприємства. Застосування спеціалізованих CRM-систем сприяє підвищенню прозорості операцій, оптимізації внутрішніх процесів та зменшенню кількості помилок, пов'язаних з обробкою інформації. Така класифікація демонструє, що вибір CRM-системи завжди залежить від розміру підприємства, галузі, фінансових можливостей та стратегічних цілей бізнесу. Тому під час впровадження CRM-системи важливо враховувати функціональні можливості програмного продукту.

### 1.1.2 Переваги та недоліки CRM-систем

Розглянемо позитивні та негативні аспекти CRM-систем. До їх переваг відносять:

- можливість централізованого зберігання всієї інформації;
- оптимізація рутинних завдань;
- полегшення планувань та контроль роботи;
- підвищення якості обслуговування клієнтів;
- зручний обмін інформацією між співробітниками;
- зростання продуктивності співробітників.

Розглянемо більш докладно кожен з вище згаданих аспектів:

– можливість централізованого зберігання всієї інформації забезпечує зручний доступ до даних, прозорість взаємодії та мінімізує ризик втрати важливих відомостей. Завдяки цьому компанія отримує цілісне уявлення про історію співпраці з кожним клієнтом, що сприяє персоналізації сервісу та підвищенню рівня задоволеності споживачів;

– оптимізація рутинних завдань є також дуже важливою перевагою так як вона дозволяє значно знизити витрати часу на виконання рутинних завдань, підвищити продуктивність працівників і уникнути помилок, пов'язаних із людським фактором;

– полегшення планувань та контроль роботи – це забезпечує прозорість процесів у компанії, адже керівництво завжди може відслідковувати ефективність роботи, наприклад, відділу продажів, маркетингу чи служби підтримки. Що у свою чергу, сприяє підвищенню дисципліни, контролю та відповідальності співробітників;

– підвищення якості обслуговування клієнтів є критично важливим, саме тому компанії й вирішують застосовувати CRM-системи. Вони дуже допомагають персоналізувати комунікацію з клієнтами, формувати індивідуальні пропозиції, швидко реагувати на потреби чи проблеми споживачів. Такий підхід підвищує рівень задоволеності клієнтів і сприяє формуванню довгострокових відносин.

Завдяки аналітичним можливостям системи компанія має змогу отримати глибоке розуміння поведінки клієнтів, їхніх уподобань та очікувань, що дозволяє будувати більш ефективні маркетингові стратегії та прогнозувати майбутні потреби ринку;

- зручний обмін інформацією між співробітниками – такі системи забезпечують комунікацію між співробітниками в середині системи. Завдяки цьому співробітника не потрібно витрачати час на перемикання між різними сервісами та витрачати час на це. При цьому історія спілкувань між співробітниками завжди залишається збереженою у системі, що дозволяє швидко вирішувати різні нестандартні ситуації;

- зростання продуктивності співробітників – так як система надає швидкий доступ до всієї інформації, щодо клієнтів, задач та документації в одному місці, менеджери можуть працювати набагато швидше та продуктивніше.

Тепер розглянемо найбільш поширені недоліки CRM-систем:

- висока вартість впровадження та обслуговування;
- складність впровадження;
- залежність від технічної інфраструктури;
- безпека даних;
- надмірна складність використання.

Розглянемо кожен недолік більш докладно:

- висока вартість впровадження та обслуговування є одним з головним недоліків, так як малі та середні компанії, що хочуть покращити свої показники, не завжди спроможні витрачати великі кошти на впровадження таких систем. Крім вартості самої системи, компанії часто стикаються з додатковими витратами на налаштування, інтеграцію з іншими програмними продуктами, навчання персоналу та підтримку системи;

- складність впровадження є також важливим обмеженням, так як перехід на CRM вимагає адаптації бізнес-процесів, а іноді й зміни організаційної культури

компанії. Це може викликати опір з боку працівників, які звикли працювати за старими методами;

– залежність від технічної інфраструктури є доволі великим недоліком. У разі використання хмарних CRM-систем компанія стає залежною від надійності інтернет-з'єднання та постачальника послуг, тоді як локальні системи потребують потужного обладнання й регулярного обслуговування;

– безпека даних є дуже важливим аспектом. Зберігання великого обсягу клієнтської інформації у єдиній системі вимагає ретельного захисту, адже витік чи втрата даних можуть мати серйозні наслідки як для репутації компанії, так і для її фінансової стабільності;

– надмірна складність використання – це та перепона, з якою стикаються працівники компанії, під час впровадження CRM-систем. У прагненні надати широкий функціонал розробники часто створюють інтерфейси, які потребують значного часу для вивчення. Це ускладнює роботу для нових співробітників і може призвести до зниження ефективності використання системи. У результаті компанії іноді застосовують лише невелику частину можливостей CRM, не використовуючи її потенціал повною мірою.

Таким чином, CRM-системи поєднують у собі як значні переваги, що здатні суттєво покращити діяльність підприємства, так і певні обмеження, які потребують уважного врахування під час вибору та впровадження. Успішність їх використання залежить від правильної стратегії впровадження, готовності персоналу до змін та здатності компанії інвестувати у розвиток сучасних інформаційних технологій.

### 1.1.3 Специфіка управління проектами на виробничому підприємстві

Специфіка управління проектами на виробничому підприємстві має свої особливості, вони зумовлені характером його діяльності, високим рівнем залежності від ресурсів та необхідністю чіткого планування.

На відміну від сфери послуг чи ІТ-компаній, де акцент робиться переважно на людських ресурсах та інформаційних технологіях, виробничі підприємства повинні враховувати складну інфраструктуру, виробничі потужності, технологічні процеси, логістику та постачання матеріалів. Управління проєктами у такому середовищі тісно пов'язане з виробничим циклом і потребує особливої уваги до оптимізації використання обладнання, контролю якості, зменшення виробничих витрат та забезпечення безперервності процесів. Крім цього, кожен проєкт на виробництві повинен узгоджуватися з технічними стандартами, нормами безпеки та вимогами до сертифікації продукції, що накладає додаткові обмеження на строки, бюджет і організацію робіт. Саме тому організація й точне планування всіх бізнес-процесів на виробничому підприємстві є критично важливими.

## 1.2 Огляд існуючих CRM та їх придатність для виробничих підприємств

Сучасний ринок програмного забезпечення пропонує широкий спектр CRM-систем, які відрізняються функціональністю, гнучкістю налаштувань, моделями ліцензування та орієнтацією на різні галузі бізнесу. Загалом більшість таких систем розроблені як універсальні рішення, які можна адаптувати під різні види діяльності, проте їхня придатність до використання на виробничих підприємствах залежить від того, наскільки добре вони інтегруються з виробничими процесами та наскільки ефективно враховують особливості роботи промислових компаній.

Для виробничих підприємств важливими є такі властивості CRM: можливість зберігати та обробляти великий обсяг даних про клієнтів та партнерів, інтеграція з системами планування ресурсів підприємства Enterprise resource planning (ERP), управління замовленнями та постачанням, а також гнучкі інструменти аналітики, які дозволяють прогнозувати попит і планувати виробничі потужності.

Однією з найбільш відомих CRM-систем, яку широко використовують у світі, є Salesforce.

Це хмарне рішення, яке пропонує комплексний підхід до управління відносинами з клієнтами. Ця система включає в себе продажі, маркетинг, сервісне обслуговування та аналітику.

Salesforce відзначається високим рівнем масштабованості, що робить її придатною як для малих компаній, так і для великих виробничих корпорацій.

Для виробничих підприємств особливо цінною є можливість інтеграції Salesforce з ERP-системами, такими як SAP чи Oracle, що дозволяє об'єднати управління клієнтами з управлінням виробничими процесами та ланцюгами постачання. Наприклад, менеджери з продажів можуть у режимі реального часу відстежувати статус виконання замовлення на виробництві, планувати строки доставки продукції, координувати дії з відділом логістики та контролювати фінансові показники.

Також, Salesforce має широкий набір аналітичних інструментів, які дозволяють будувати прогнози щодо попиту на продукцію, аналізувати ефективність роботи відділів і виявляти ризики, пов'язані з простроченням контрактів.

Проаналізуємо основні функції, що доступні користувачу цієї системи на її різних сторінках, більш докладно:

- віджет «Ключові угоди» (Key Deals) показує список поточних угод з продажу з фільтрами, наприклад, які скоро закриваються або за відповідальністю користувача;
- останні записи (Recent Records), швидкий доступ до яких, швидко нагадає користувачу про те, що він переглядав нещодавно;
- статистика продуктивності (Performance Metrics) відображає ключові показники, наприклад, для відділу продажів за поточний квартал чи інший період, щоб користувач бачив прогрес у досягненні цілей;
- завдання та події на сьогодні (Today's Tasks / Today's Events) – показує список запланованих зустрічей, подій або завдань, які потрібно виконати сьогодні;

– дашборди і звіти у вигляді графіків (Reports / Dashboard snapshots) – відображення візуальних даних (графіків, діаграм) прямо на домашній сторінці для швидкого огляду тенденцій;

– швидкий доступ до ярликів (Quick Links / Navigation / Favorites) дає можливість швидко переходити до часто використовуваних об'єктів, записів або переглядів.

Зовнішній вигляд головної сторінки CRM-системи може відрізнятись та залежить від налаштувань профіля конкретного користувача та виду його діяльності в компанії.

Домашня сторінка CRM-системи «Salesforce» з базовими налаштуваннями зображена на рисунку 1.2.

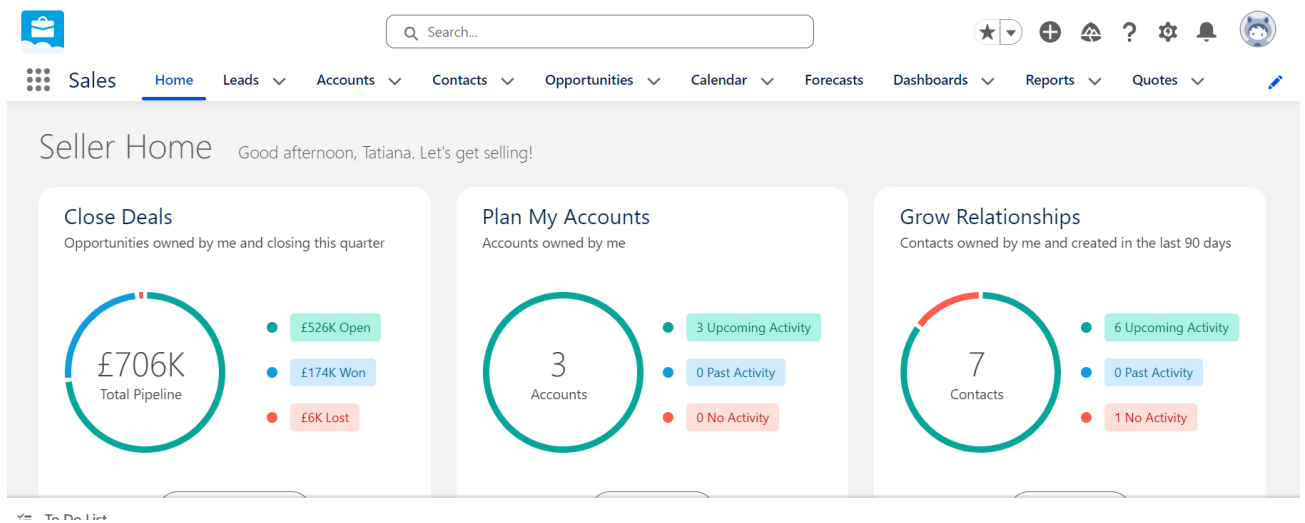


Рисунок 1.2 – Домашня сторінка CRM-системи «Salesforce»

Ще однією прогресивною CRM-системою є «ZOHO CRM» – це гнучка платформа для управління відносинами з клієнтами. Система «ZOHO CRM» є універсальним рішенням для бізнесу, яка допомагає керувати продажами, автоматизувати рутинні завдання та вдосконалювати відносини з клієнтами.

Платформа «ZOHO CRM» має вбудовані додатки та модулі, які легко масштабуються під потреби компанії.

«ZOHO CRM» є також хмарним рішенням, яке пропонує багатий набір функцій для управління клієнтами, продажами, маркетингом і сервісом.

Zoho дозволяє централізовано зберігати контакти, вести історію взаємодій із клієнтами, автоматизувати робочі процеси, наприклад, надсилання слід-листів, планування нагадувань, а також прогнозувати продажі та створювати візуальні звіти і дашборди, які надають огляд ключових показників компанії.

Серед особливостей є можливість налаштовувати робочі воронки продажів під специфічні етапи, формування сторінок і полів під внутрішні процеси компанії, комунікація з клієнтами через email, а також аналітика з використанням інструментів бізнес-інтелекту, що дає змогу бачити тенденції, відслідковувати результативність менеджерів і відділів.

Також ця система має інтеграційні можливості. Система може легко поєднуватися з іншими сервісами Zoho, такими як Zoho Mail, Zoho Desk, Zoho Projects. Це дозволяє мати всі потрібні інструменти в одному місці для побудови єдиного цифрового середовища. Такий підхід допомагає спростити координацію між відділами підприємства та створює передумови для більш ефективного управління виробничими процесами.

Ще потрібно відзначити, що до цієї системи можна підключити інструмент створення звітів ZOHO Reports. За допомогою цього конструктора можна створити практично будь-який звіт з будь-якого напрямку.

Зовнішній вигляд CRM-системи «ZOHO CRM» зображено на рисунку 1.3.

	Deal Name	Amount	Stage	Closing Date	Account Name
<input type="checkbox"/>	Sigma Corporation	\$ 340,000.00	Qualification	05/31/2024	Sigma Corporation
<input type="checkbox"/>	Scriptorium - New York	\$ 540,000.00	Qualification	04/18/2024	Scriptorium
<input type="checkbox"/>	Commercial Press - Miami	\$ 500,000.00	Qualification	04/26/2024	Commercial Press
<input type="checkbox"/>	Maithili Inc	\$ 12,000.00	Negotiation/Review	03/17/2024	Maithili Inc
<input type="checkbox"/>	King - Mariner	\$ 230,000.00	Needs Analysis	03/17/2024	King
<input type="checkbox"/>	King	\$ 600,000.00	Identify Decision Makers	03/17/2024	King
<input checked="" type="checkbox"/>	Truhlar And Truhlar Attys	\$ 450,000.00	Identify Decision Makers	03/17/2024	Truhlar And Truhlar
<input checked="" type="checkbox"/>	Commercial Press	\$ 450,000.00	Value Proposition	03/17/2024	Commercial Press
<input checked="" type="checkbox"/>	Morlong Associates	\$ 350,000.00	Value Proposition	03/17/2024	Morlong Associates
<input checked="" type="checkbox"/>	Chapman	\$ 700,000.00	Proposal/Price Quote	03/17/2024	Chapman
<input checked="" type="checkbox"/>	Printing Dimensions	\$ 250,000.00	Value Proposition	03/17/2024	Printing Dimensions
<input checked="" type="checkbox"/>	Feltz Printing Service	\$ 450,000.00	Identify Decision Makers	03/17/2024	Feltz Printing Service
<input type="checkbox"/>	Chemel	\$ 700,000.00	Needs Analysis	03/17/2024	Chemel
<input type="checkbox"/>	Chanay	\$ 550,000.00	Identify Decision Makers	03/17/2024	Chanay
<input checked="" type="checkbox"/>	Benton	\$ 2,500,000.00	Qualification	03/17/2024	Benton

Рисунок 1.3 – Зовнішній вигляд CRM-системи «ZOHO CRM»

Таким чином, проведено порівняльний аналіз переваг та недоліків існуючих систем для конкретних потреб виробничих підприємств.

Salesforce – це одна з найпотужніших CRM-платформ на ринку, яка пропонує дуже широкий набір можливостей, високий рівень кастомізації та насичений екосистемою доповнень і інтеграцій. Zoho CRM натомість більше орієнтована на малий та середній бізнес, має доступніший ціновий рівень, велику кількість базових функцій «з коробки», простіший інтерфейс і меншу криву навчання.

Спочатку розглянемо основні переваги системи «Salesforce»:

- гнучкий інструментарій для управління продажами, маркетингом і сервісним обслуговуванням, що дозволяє виробничому підприємству налаштувати воронки продажів, етапи замовлень, процеси погодження документів, контролювати якість та виконання замовлень;

- має аналітичні інструменти для аналізу різних показників різних відділів та робити прогнози щодо ситуації на ринку;

- система має власний marketplace, що дозволяє підключати готові модулі чи доповнення;

- відстеження всіх етапів, на яких знаходяться проекти, у реальному часі.

Розглянемо основні недоліки цієї системи:

- занадто високі витрати на ліцензію та на впровадження цієї системи;

- деякі функції відкриваються тільки у найдорожчих тарифах;

- складність системи;

- велика кількість модулів, що можуть обтяжувати систему.

Далі розглянемо основні переваги системи «ZOHO CRM»:

- багато базових функцій доступні вже в нижчих тарифах;

- має безкоштовну версію з мінімумом функціоналу для обмеженої кількості користувачів;

- дозволяє ефективно вести облік клієнтів і контактів, автоматизувати процеси продажів і обробки замовлень, формувати базові звіти, працювати з кількома воронками продажів, налаштовувати робочі процеси і тригери без надмірної складності;

- має можливість інтеграції з іншими модулями і з різними каналами комунікації.

Також розглянемо основні недоліки цієї системи:

- інтерфейс Zoho є перевантаженим;

- деякі функції доступні у найпотужніших пакетах, купити, які може дозволити собі не кожна компанія;

- складність системи.

### 1.3 Обґрунтування необхідності розробки спеціалізованого рішення

Спеціалізоване рішення для розроблення веб-орієнтованої CRM-система може стати вигіднішим рішенням для невеликого виробничого підприємства порівняно з

використанням готових платформ на кшталт Salesforce чи Zoho CRM з кількох важливих причин. Насамперед це пов'язано з тим, що готові рішення зазвичай орієнтовані на широкий спектр бізнесів і надають величезну кількість інструментів, значна частина яких може бути зайвою для невеликого підприємства.

У результаті підприємство змушене оплачувати функціонал, яким воно практично не користується, або ж витратити час на налаштування та відключення непотрібних модулів. Натомість власна CRM-система може бути спроектована виключно під ті бізнес-процеси, які справді потребують автоматизації, що забезпечує простоту використання та відсутність перевантаження зайвими можливостями.

Ще одним аргументом є контроль над даними. У випадку із SaaS-рішеннями, такими як Salesforce чи Zoho, всі дані зберігаються на сторонніх серверах, що зазвичай викликає занепокоєння у підприємств, пов'язаних із конфіденційністю комерційної інформації.

Власна CRM-система дозволяє зберігати дані локально або на захищених корпоративних серверах, що підвищує безпеку та довіру співробітників до системи.

Таким чином, спеціалізоване рішення CRM-системи є більш вигідним варіантом для невеликого виробничого підприємства, оскільки вона дозволить створити інструмент, який повністю відповідає потребам конкретного бізнесу, мінімізувати витрати у довгостроковій перспективі, забезпечить простоту впровадження, гнучкість подальшої модифікації та високий рівень захисту даних. Це робить її більш раціональним і стратегічно виправданим вибором у порівнянні з використанням універсальних готових рішень.

## 1.4 Висновки до першого розділу

У першому розділі було зроблено комплексний аналіз предметної області, що охоплює як теорію, так і практичні аспекти використання CRM-систем у сучасному бізнес-середовищі.

Було розглянуто функціональні можливості CRM-систем, зокрема управління контактами, продажами, маркетингом, сервісним обслуговуванням та аналітикою, а також визначено їхній вплив на оптимізацію бізнес-процесів підприємства.

Ще було проведено класифікацію CRM-систем. Це дозволило зрозуміти, що вибір конкретного рішення завжди залежить від особливостей підприємства, його розміру та фінансових можливостей.

Здійснено критичний огляд переваг і недоліків CRM-систем. Такий аналіз дав змогу оцінити як потенціал використання CRM, так і виклики, які підприємства повинні враховувати під час їх впровадження.

Крім того, у розділі здійснено огляд сучасних CRM-рішень, таких як Salesforce та Zoho CRM. Проаналізовано їхні основні модулі та особливості.

Таким чином, результати проведеного аналізу предметної області надали цілісне уявлення про структуру сучасних CRM-систем. Виявлені переваги й недоліки, а також вивчення практичних рішень дозволить сформулювати обґрунтовану постановку завдання щодо створення CRM-системи, яка буде відповідати потребам виробничого підприємства.

## 2 ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ CRM-СИСТЕМИ

### 2.1 Вибір технологічного стеку для реалізації CRM-системи

Для розробки веб-орієнтованої CRM-системи буде використано сучасний набір інструментів та технологій, які забезпечать зручність у створенні, тестуванні й подальшому супроводі програмного забезпечення.

Серверна частина системи базуватиметься на мові програмування PHP, за допомогою якої буде реалізовано основну бізнес-логіку застосунку оскільки це одна з найпоширеніших мов для веброзробки, має широку спільноту користувачів та велику підтримку у документації. Також ця мова програмування має розвинену екосистему бібліотек і фреймворків. Крім того, PHP чудово поєднується з більшістю реляційних СУБД, підтримується усіма популярними вебсерверами, що забезпечує гнучкість у розгортанні.

Для зберігання та обробки даних, було вирішено застосовувати СУБД MySQL, оскільки вона є швидкою, стабільною та відкритою системою керування базами даних. Забезпечує високу продуктивність при роботі з великими обсягами даних. MySQL відзначається простотою адміністрування та широкою підтримкою в документації. Це робить її доцільним вибором для такого проєкту. У якості альтернативи можна було б використати PostgreSQL, яка забезпечує більш розвинені механізми роботи з складними запитамі, або ж SQLite, що є зручною для невеликих застосунків, але має обмеження у масштабуванні.

Для адміністрування СУБД було обрано інструмент phpMyAdmin. оскільки він надає зручний веб-інтерфейс для роботи з MySQL і дозволяє виконувати всі основні операції. Його перевага полягає у простоті використання та широкій підтримці. Альтернативними рішеннями можуть бути MySQL Workbench. Він має більш розширені функції візуалізації структури БД та розробки складних SQL-запитів. Ще

одним варіантом є DBeaver, він є універсальним кросплатформним інструментом і підтримує роботу не лише з MySQL, а й з великою кількістю інших СУБД.

У якості локального сервера планується використати середовище OpenServer із вебсервером Apache 2.4, що дасть змогу ефективно налагоджувати та тестувати роботу застосунку. Вибір цього рішення зумовлений його простотою встановлення, підтримкою різних версій PHP, MySQL та Apache, а також зручним інтерфейсом для керування конфігураціями. Apache є надійним вебсервером із багаторічною історією розвитку. Він забезпечує гнучкі налаштування, стабільну роботу та широку підтримку. Альтернативою OpenServer може виступити XAMPP, який також включає Apache, MySQL та PHP. Ще одним варіантом є MAMP, він орієнтований на macOS та Windows і має простий графічний інтерфейс із готовим набором інструментів для локальної веброзробки.

Для клієнтської частини CRM-системи буде розроблено вебінтерфейс із використанням HTML та CSS, а для створення адаптивного дизайну сторінок застосовуватиметься фреймворк Bootstrap, що дозволить коректно відобразити інтерфейс на різних пристроях.

Розроблення програмних компонентів виконуватиметься у середовищі Microsoft Visual Studio Code 2024.

Для моделювання структури системи та створення UML-діаграм передбачається використання програмного продукту DrawIO.

Додатково у процесі розробки можуть бути задіяні JavaScript для реалізації динамічних елементів інтерфейсу та CSS-бібліотеки іконок Font Awesome для покращення візуального представлення елементів системи.

Такий стек технологій забезпечить реалізацію надійної та зрозумілої CRM-системи, яка поєднуватиме простоту у використанні та достатню функціональність для потреб невеликого виробничого підприємства.

## 2.2 Вибір архітектури CRM-системи

Дуже важливим етапом створення будь-якого програмного забезпечення є вибір його архітектури, оскільки саме архітектурні рішення визначають структурну організацію системи, взаємодію її компонентів, можливість подальшого масштабування та супроводу. Для веб-орієнтованої CRM-системи архітектура є фундаментом, який забезпечує ефективну роботу всіх модулів, від інтерфейсу користувача до механізмів обробки даних на сервері. Правильний вибір архітектурного підходу дозволяє досягти балансу між продуктивністю, надійністю, безпекою та зручністю експлуатації системи.

Далі розглянемо процес вибору архітектурного підходу.

На етапі проектування будь-якої системи, потрібно правильно обрати архітектурний підхід, який буде у ній застосовуватись. Він визначає логічну структуру застосунку, взаємозв'язок його компонентів та спосіб обміну даними між ними. Для розробки веб-орієнтованої CRM-системи було обрано трирівневу клієнт-серверну архітектуру, яка забезпечує чітке розмежування функцій між клієнтським інтерфейсом, бізнес-логікою та рівнем зберігання даних. Цей підхід підвищує гнучкість, масштабованість та надійність системи. Така модель системи якнайкраще підійде для задач автоматизації бізнес-процесів на виробничому підприємстві.

Клієнт – це комп'ютер на стороні користувача, який відправляє запит до сервера для надання інформації або виконання певних дій.

Сервер – це більш потужний комп'ютер або обладнання, призначене для вирішення певних завдань з виконання програмних кодів, виконання сервісних функцій за запитом клієнтів, надання користувачам доступу до певних ресурсів, зберігання інформації і баз даних.

Архітектура системи складається з трьох основних рівнів: клієнтського рівня (presentation layer), рівня застосунків (application layer) та рівня даних (data layer).

Клієнтський рівень є інтерфейсом взаємодії користувача з CRM системою. Взаємодія здійснюється через веб-браузер (наприклад, Google Chrome, Mozilla Firefox, Opera), який забезпечує відображення сторінок і передавання запитів до сервера за протоколом HTTP/HTTPS.

Основними функціями клієнтського рівня є:

- відображення контенту є функцією на рівні якої реалізується графічна оболонка системи. Таку оболонку називають веб-інтерфейсом. Він створюється за допомогою HTML, CSS та JavaScript. Ця функція забезпечує відображення сторінок, форм, таблиць, панелей керування та інших елементів користувацького інтерфейсу;
- збір і передача даних надає можливість користувачу вводити інформацію, наприклад, створювати завдання, редагувати дані клієнтів, формувати звіти. Після чого браузер формує та надсилає HTTP-запити на сервер;

Рівень застосунків є посередником між клієнтом і базою даних. Він реалізує основну бізнес-логіку CRM-системи. На цьому рівні розгортається сервер застосунків, який у даному проекті буде працювати на базі Apache 2.4 із використанням PHP 8.0.

Основними завданнями рівня застосунків є:

- обробка запитів відбувається завдяки тому, що сервер приймає запити, які надходять від клієнтів, інтерпретує їх, перевіряє коректність введених даних і спрямовує до відповідних компонентів системи;
- виконання бізнес логіки реалізуються шляхом формування правил роботи CRM системи. Це створення, оновлення та видалення записів про клієнтів, завдання, проекти й звіти. Також забезпечується перевірка прав доступу користувачів до різних функцій, логування дій та контроль статусів завдань;
- взаємодія з базою даних здійснюється шляхом формування сервером застосунків SQL-запитів до сервера баз даних MySQL та виконанням операцій вибірки (SELECT), додавання (INSERT), оновлення (UPDATE) та видалення (DELETE);

– формування відповідей відбувається після виконання запиту. Сервер формує відповідь, генеруючи динамічні веб-сторінки (HTML або JSON-відповіді) та передає їх клієнту. Це дозволяє відображати актуальний стан даних у веб-інтерфейсі.

Рівень даних відповідає за зберігання, обробку та захист інформації, що використовується в CRM-системі. Для розроблювальної CRM було обрано систему керування базами даних MySQL 8.0. Вона забезпечує високу продуктивність, масштабованість і підтримку транзакцій.

Завданнями рівня даних є:

- зберігання інформації у базі даних про користувачів, проекти, завдання, звіти, сповіщення та історію взаємодій;
- обробка SQL запитів, які надходять від рівня застосунків, сервер бази даних повертає результати у вигляді html-сторінок, таблиць або повідомлень про успішне виконання операцій;
- цілісність та безпека забезпечується завдяки механізмам транзакцій, зовнішнім ключам, індексам і контролем доступу. Такий підхід гарантує узгодженість даних і обмежує доступ до конфіденційної інформації, тільки для користувачів з відповідними правами.

Перевагою трирівневої клієнт-серверної архітектури є те, що вона дозволяє розподілити навантаження між різними компонентами системи, підвищити масштабованість і спростити технічну підтримку. Кожен рівень можна оновлювати або змінювати незалежно від інших. Це забезпечує гнучкість системи, у разі необхідності проведення подальшої модернізації CRM-системи. Крім того, чітке розмежування логіки, інтерфейсу та даних сприяє стабільній роботі системи навіть за умов збільшення кількості користувачів або розширення функціоналу. Це робить клієнт-серверний підхід оптимальним для розвитку системи в реальних умовах виробничого підприємства (рис. 2.1).

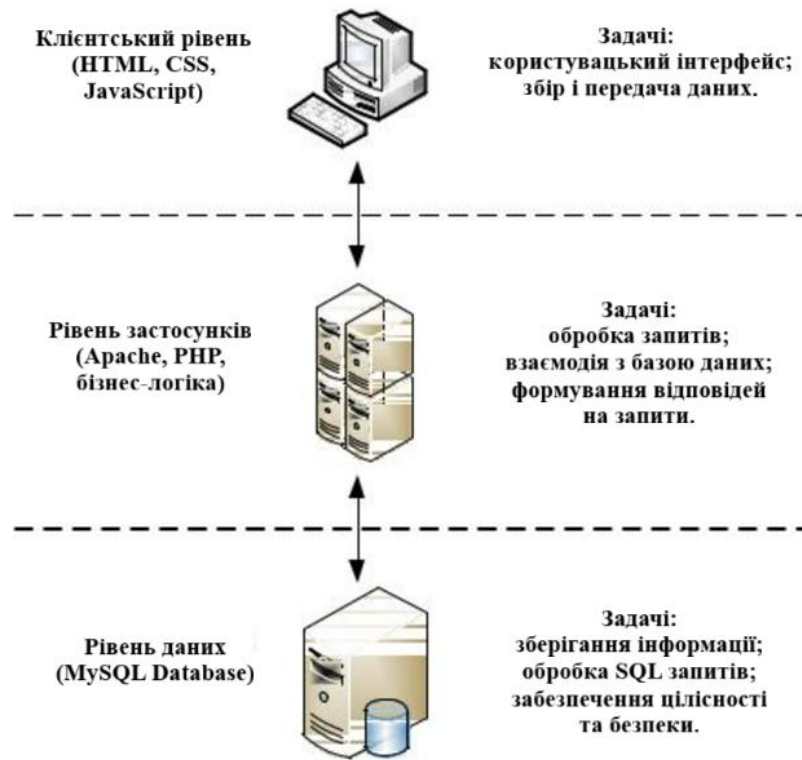


Рисунок 2.1 – Трирівнева архітектура веб-застосунка CRM-системи

Кожен рівень виконує чітко визначені функції, що забезпечує узгоджену роботу системи в цілому. Такий підхід дозволяє оптимізувати обмін даними, підвищити ефективність обробки запитів та забезпечити стабільне функціонування навіть при збільшенні кількості користувачів. Архітектура створює гнучке середовище для подальшого розширення та інтеграції нових модулів CRM-системи без потреби у зміні всієї її структури.

## 2.3 Розроблення UML діаграм до розроблюваної CRM-системи

Уніфікована мова моделювання (англ. Unified Modeling Language, UML) – це універсальний інструмент, який може використовуватися для проєктування широкого спектра програмних систем, незалежно від сфери їх застосування, типу організації, масштабів чи рівня складності проєкту. UML забезпечує єдиний стандартизований синтаксис для опису об'єктів і їхніх взаємозв'язків у системі.

UML-діаграми (Unified Modeling Language diagrams) – це універсальний інструмент моделювання, який використовується для візуального опису, проєктування та документування програмних систем. Використання UML-діаграм дозволяє наочно відобразити архітектуру розроблюваної системи, її основні компоненти, взаємозв'язки між ними, а також процеси обробки даних і взаємодії користувачів із системою.

### 2.3.1 Розроблення UML-діаграми прецедентів використання CRM-системи

Діаграми прецедентів (use case diagrams) – це вид діаграм у мові моделювання UML, які візуально показують взаємодію між акторами (користувачами, системами) та прецедентами (діями або сценаріями) в системі. Вони описують функціональні вимоги з точки зору користувача, а не внутрішню структуру системи.

На діаграмі прецедентів акторами можуть бути реальні користувачі або інші системи. Прецеденти зображаються у вигляді овалів і позначають функції або сценарії, які система надає цим акторам. Зв'язки між акторами та прецедентами показують, які саме дії доступні кожному з них. Така візуалізація допомагає зрозуміти межі системи, її основних користувачів і ролі, які вони мають. Також така діаграма відображає які потреби задовольняє система та допомагає виявити всі основні сценарії роботи з нею.

Такі діаграми широко застосовуються на ранніх етапах розроблення програмного забезпечення, під час аналізу вимог. Вони слугують інструментом для комунікації між аналітиками, замовниками та розробниками, оскільки зрозумілі навіть тим, хто не має технічної освіти. За допомогою цих діаграм можна узгодити бачення системи з усіма сторонами, визначити основні сценарії її використання та пріоритети реалізації функціоналу.

Крім основних зв'язків між акторами та прецедентами, у діаграмі можуть відображатися спеціальні типи відношень include та extend. Відношення include показує, що певний прецедент завжди включає поведінку іншого. Відношення extend описує ситуацію, коли додаткова поведінка виконується лише за певних умов. Актори, що є користувачами системи, а також функції самої CRM зображені на діаграмі прецедентів (рис. 2.2).

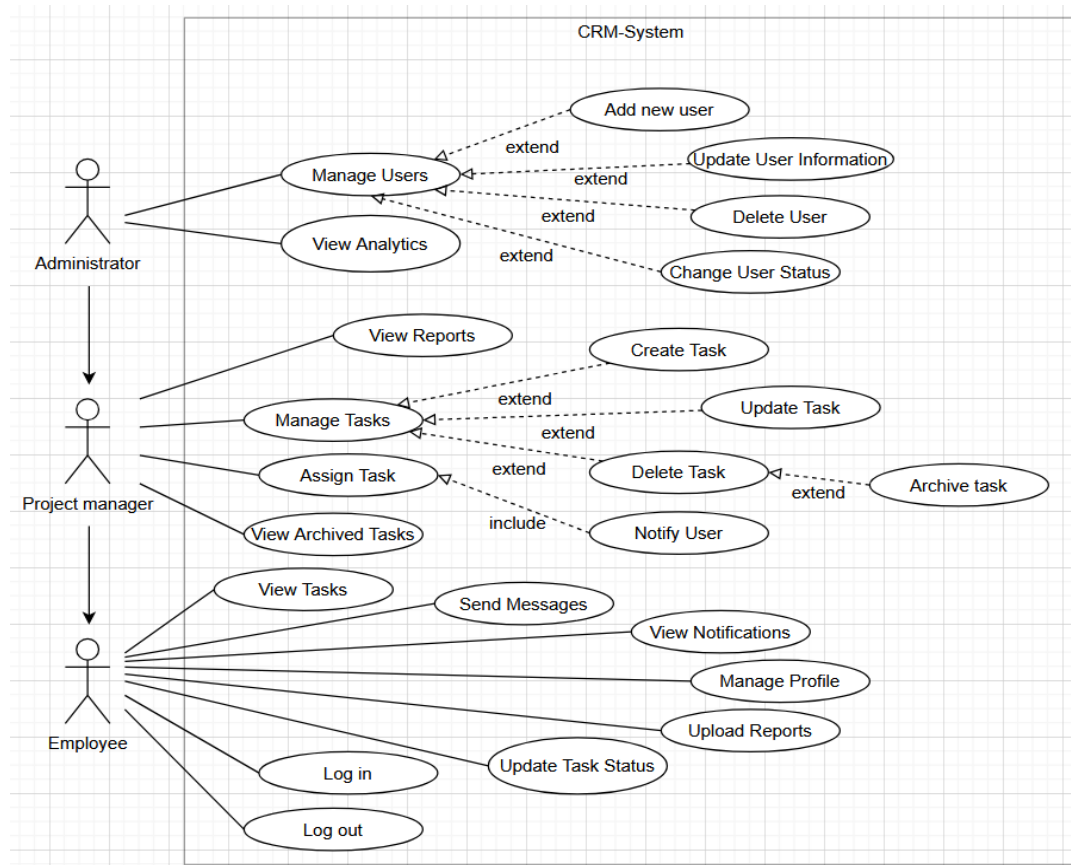


Рисунок 2.2 – Діаграма прецедентів CRM-системи

На діаграмі прецедентів зображено наступні актори:

– «Employee» – це зареєстрований у системі користувач, що виконує призначені завдання, може змінювати їх статус у процесі виконання, відправляти повідомлення та отримувати сповіщення;

– «Project manager» – це користувач системи, що має змогу керувати завданнями, розподіляти їх між співробітниками, переглядати звіти й архівувати завершені проекти. Також користувач з таким статусом наслідує всі можливості, що є користувача з статусом працівник;

– «Administrator» – це людина, що має права суперкористувача в CRM-системі та має доступ до всіх функцій та підсистем цього застосунку. Основні функції такого користувача є керування користувачами та перегляд аналітики.

У таблиці 2.1 представлено опис кожного прецеденту.

Таблиця 2.1 – Описи прецедентів

Назва прецеденту	Опис прецедентів
Manage Users	Дозволяє адміністратору керувати користувачами системи.
View Analytics	Дозволяє адміністратору переглядати статистики виконання завдань або активності користувачів.
Add New User	Дозволяє адміністратору додавати нових користувачів до системи.
Update User Information	Дозволяє адміністратору оновлювати різну інформацію про користувачів системи
Delete User	Дозволяє адміністратору видаляти користувачів з системи.

Продовження таблиці 2.1

Назва прецеденту	Опис прецедентів
Change User Status	Дозволяє адміністратору змінювати статуси користувачів системи
Manage Tasks	Дозволяє менеджеру проєкту керувати завданнями.
Create Task	Дозволяє менеджеру проєкту додавати нові завдання.
Update Task	Дозволяє менеджеру проєкту змінювати інформацію щодо завдання.
Delete Task	Дозволяє менеджеру проєкту видаляти вже виконанні/не актуальні завдання.
Archive task	Дозволяє менеджеру архівувати інформацію про вже виконані завдання, щоб зберігалась історія виконаних завдань.
Assign Task	Дозволяє менеджеру проєкту розподіляти завдання між співробітниками та назначати конкретне завдання конкретному працівнику.
Notify User	Дозволяє оповіщати користувача при призначенні йому завдання.
View Archived Tasks	Дозволяє менеджеру передивлятися завдання, що вже були завершені та перенесені у архів.
View Tasks	Дозволяє працівнику переглядати назначені йому завдання.

Продовження таблиці 2.1

Назва прецеденту	Опис прецедентів
Send Messages	Дозволяє працівнику надсилати повідомлення.
View Notifications	Дозволяє працівнику переглядати сповіщення.
Manage Profile	Дозволяє працівнику керувати своїм профілем.
Upload Reports	Дозволяє працівнику завантажувати звіти з зроблених завдань.
Update Task Status	Дозволяє працівнику змінювати статус завдання під час його виконання.
Log in	Дозволяє працівнику авторизуватись у системі.
Log out	Дозволяє працівнику вийти з системи.

### 2.3.2 Розроблення UML-діаграми класів прецеденту «Керування завданнями»

Діаграма класів – це одна з основних діаграм у мові моделювання UML, яка використовується для відображення статичної структури системи. Вона показує основні класи об'єктів, їх атрибути, операції (методи), а також зв'язки між ними. Діаграма класів дозволяє побачити логічну організацію системи, визначити її основні елементи та взаємозалежності. Такі діаграми застосовують для аналізу, проектування й документування програмного забезпечення.

Основна мета діаграм класів полягає в тому, щоб надати розробникам і аналітикам узагальнене, але структуроване уявлення про систему. Вона допомагає зрозуміти, які сутності існують у предметній області, які між ними існують відносини та як вони взаємодіють між собою.

Діаграми класів застосовуються на різних етапах розроблення програмного забезпечення. На етапі аналізу вони дозволяють моделювати предметну область і визначати основні сутності. На етапі проєктування діаграми класів деталізуються, описуючи структуру програмних модулів, інтерфейси та логіку взаємодії компонентів системи. Завдяки цьому можна зрозуміти концепцію розроблюваної системи та її подальшу технічну реалізацією у вигляді коду.

Діаграма відображає логічну структуру основних класів, що беруть участь у процесі створення, призначення, виконання та контролю задач у CRM-системі виробничого підприємства. Вона показує, як пов'язані між собою користувачі, менеджер проєкту, задачі, статуси задач і звіти про їх виконання [16-18]. Розроблена діаграма класів CRM-системи зображена на рисунку 2.3.

Розглянемо докладніше кожен концептуальний клас у цій діаграмі:

- Project\_Manager – містить інформацію про проєктного менеджера, що створює нові задачі, перевіряє звіти з виконаних завдань та затверджує їх, призначає працівникам певні задачі та змінює статуси на яких залежно від стадії виконання знаходиться завдання;

- Task – містить всю інформацію щодо завдань, які знаходяться у процесі виконання. Таку як id, назву завдання, його опис, пріоритетність його виконання, дату його створення, термін його виконання, інформацію про людину, що його створила та інформацію про призначеного працівника, який буде виконувати це завдання;

- User – містить інформацію про всіх користувачів системи. Зберігає id, ім'я користувача, його пароль для входу у систему, його адресу електронної пошти, та роль у системі;

- Report – містить інформацію про звіти які завантажують працівники. Зберігає id, назву завдання, файл звіту та ким було завантажено звіт;

- Task\_Status – містить інформацію на якій стадії виконання наразі знаходиться завдання.

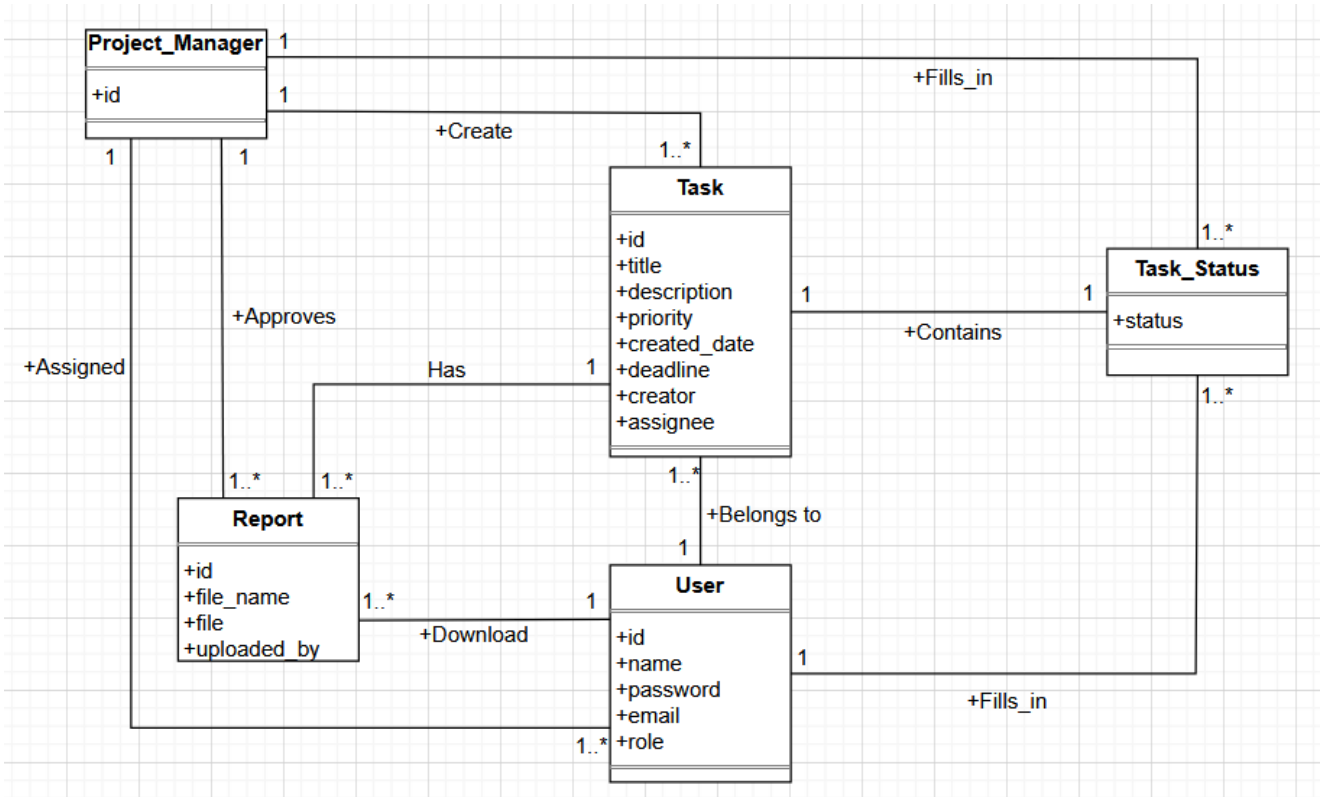


Рисунок 2.3 – Діаграма класів CRM-системи для прецеденту «Керування задачами»

Дана діаграма класів може слугувати відправною точкою для подальшої розробки ER-діаграми. Вона дозволяє відстежити всю логіку бізнес-процесів та зрозуміти, якою повинна бути структура інтерфейсу користувача, щоб він включав у себе доступ до всіх необхідних функцій.

### 2.3.3 Розроблення UML-діаграми послідовностей для успішного основного сценарію прецеденту «Цикл роботи з задачами»

UML-діаграма послідовностей це один із типів діаграм у мові моделювання UML, що використовується для опису динамічної поведінки системи, тобто взаємодії між об'єктами або компонентами у певному часовому порядку. На відміну від статичних діаграм, таких як діаграма класів, діаграма послідовностей зосереджується не на структурі системи, а на процесах, що відбуваються під час виконання конкретного сценарію або прецеденту. Основна мета цієї діаграми

полягає у відображенні обміну повідомленнями між учасниками системи, які можуть бути користувачами, програмними компонентами або зовнішніми сервісами.

Кожен учасник процесу позначається вертикальною лінією життя (lifeline). Вона символізує його існування протягом певного часу. На цій лінії можуть розташовуватись прямокутники, які є активними областями (activation bars), вони відображають моменти, коли об'єкт виконує певну дію або обробляє повідомлення. Взаємодія між об'єктами показується стрілками, що позначають виклики методів або передавання повідомлень. Суцільними стрілками позначають синхронні виклики, наприклад, коли відправник очікує відповідь, а пунктирними позначають асинхронні або зворотні повідомлення, наприклад, повернення результату чи підтвердження виконання дії.

UML-діаграми дозволяють детально описати логіку взаємодії між компонентами під час реалізації певного функціонального сценарію. Завдяки ним можна зрозуміти порядок викликів методів, залежності між об'єктами та часову послідовність подій. Такі діаграми допомагають розробникам, аналітикам і тестувальникам однаково інтерпретувати процеси в системі, забезпечуючи узгоджене бачення бізнес-логіки.

Така діаграма допомагає виявити зайві або відсутні кроки у процесі, оптимізувати потоки даних і запобігти помилкам на етапі реалізації. Часто такі діаграми створюють на основі описів прецедентів використання, щоб показати, як саме реалізується сценарій на рівні системних компонентів. Також їх широко застосовуються у розробленні програмного забезпечення, особливо при моделюванні складних бізнес-процесів, інтеграції сервісів або проектуванні взаємодії клієнт-серверних систем. Розроблена діаграма послідовностей прецеденту процесу виконання завдання зображена на рисунку 2.4.

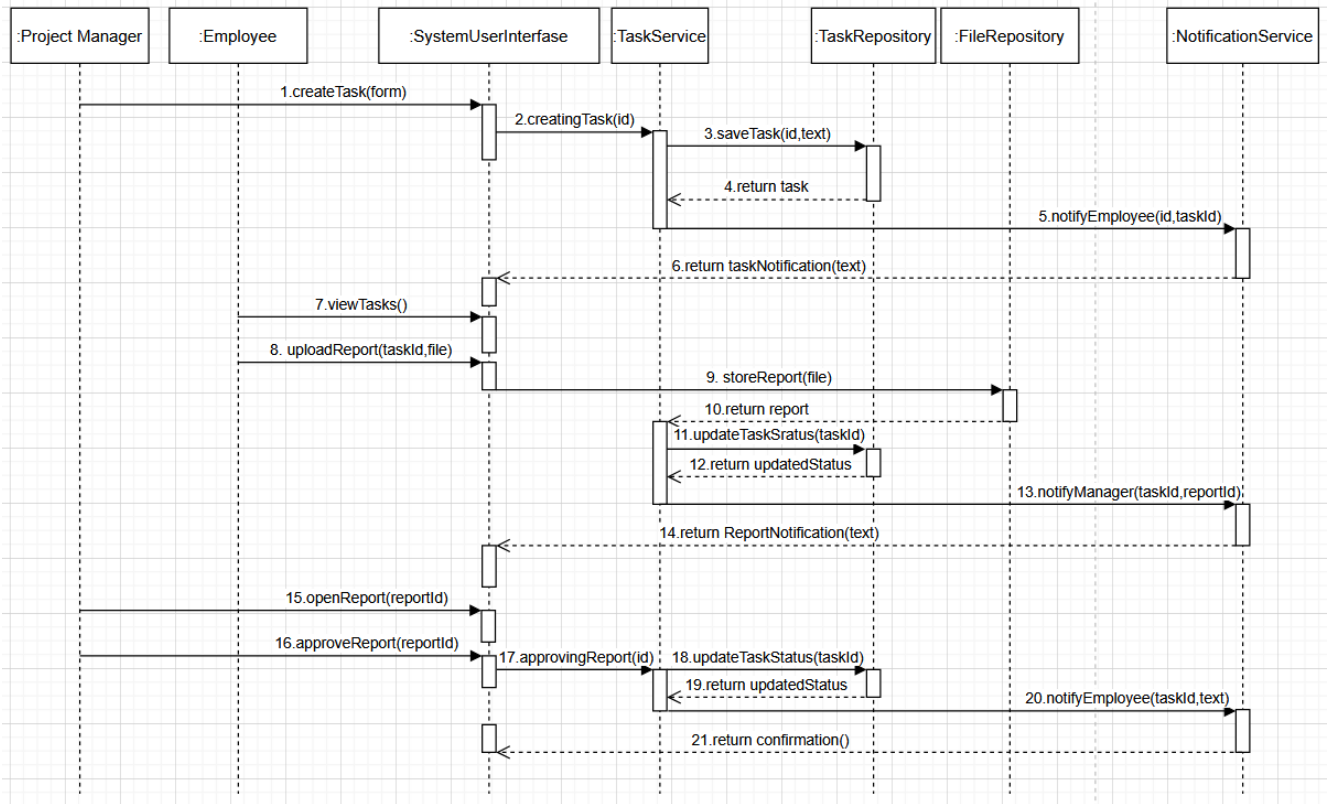


Рисунок 2.4 – Діаграма послідовностей прецеденту «Task Execution Process»

UML-діаграма послідовностей прецеденту «Task Execution Process» відображає послідовність взаємодій між акторами (Project Manager, Employee) та компонентами системи (SystemUserInterfase, TaskService, TaskRepository, FileRepository, NotificationService) під час виконання основного сценарію керування задачами.

Менеджер створює завдання через інтерфейс користувача, після чого система передає дані до сервісного рівня та зберігає їх у репозиторії завдань. Далі відбувається сповіщення працівника про нове завдання. Працівник переглядає його, виконує і завантажує звіт, який зберігається у файловому репозиторії. Після успішного збереження TaskService оновлює статус завдання та повідомляє менеджера про наявність нового звіту. Менеджер переглядає звіт, затверджує його, після чого система оновлює статус завдання і надсилає сповіщення працівнику про успішне завершення процесу.

## 2.4 Постановка завдання на розроблення CRM-системи

Постановка завдання на розробку веб-орієнтованої CRM-системи для виробничого підприємства базується на результатах аналізу предметної області, архітектурних рішень та побудованих UML-діаграм, які відображають логіку взаємодії користувачів із системою.

Основними проблемами, які необхідно вирішити за допомогою розроблення веб-орієнтованої CRM-системи, є автоматизація бізнес-процесів підприємства, підвищення ефективності управління виробничими операціями, а також забезпечити зручну взаємодію користувачів із системою через веб-інтерфейс.

Система повинна забезпечувати реєстрацію та авторизацію користувачів, створення й редагування завдань, фіксацію статусів виконання, управління клієнтами та замовленнями, а також формування звітів. Важливою вимогою є забезпечення безпеки даних і можливості подальшого масштабування системи при зростанні кількості користувачів чи обсягу інформації. Для реалізації проєкту обрано стек технологій, що включає PHP 8.0, сервер Apache 2.4 та систему керування базами даних MySQL 8.0, що забезпечує сумісність, стабільність і продуктивність веб-застосунку.

Отже, розроблення веб-орієнтованої CRM-системи для виробничого підприємства передбачає створення інтегрованого середовища для управління клієнтськими відносинами, контролю завдань і підвищення ефективності бізнес-процесів. Запропонована архітектура та побудовані UML-діаграми стали основою для формування технічного завдання та подальшої реалізації програмного продукту, який сприятиме підвищенню конкурентоспроможності підприємства та оптимізації його управлінських процесів.

## 2.5 Проектування схеми бази даних (ER-діаграма)

Проектування логічної моделі бази даних для CRM-системи було реалізовано в середовищі MySQL Workbench (рис. 2.5).

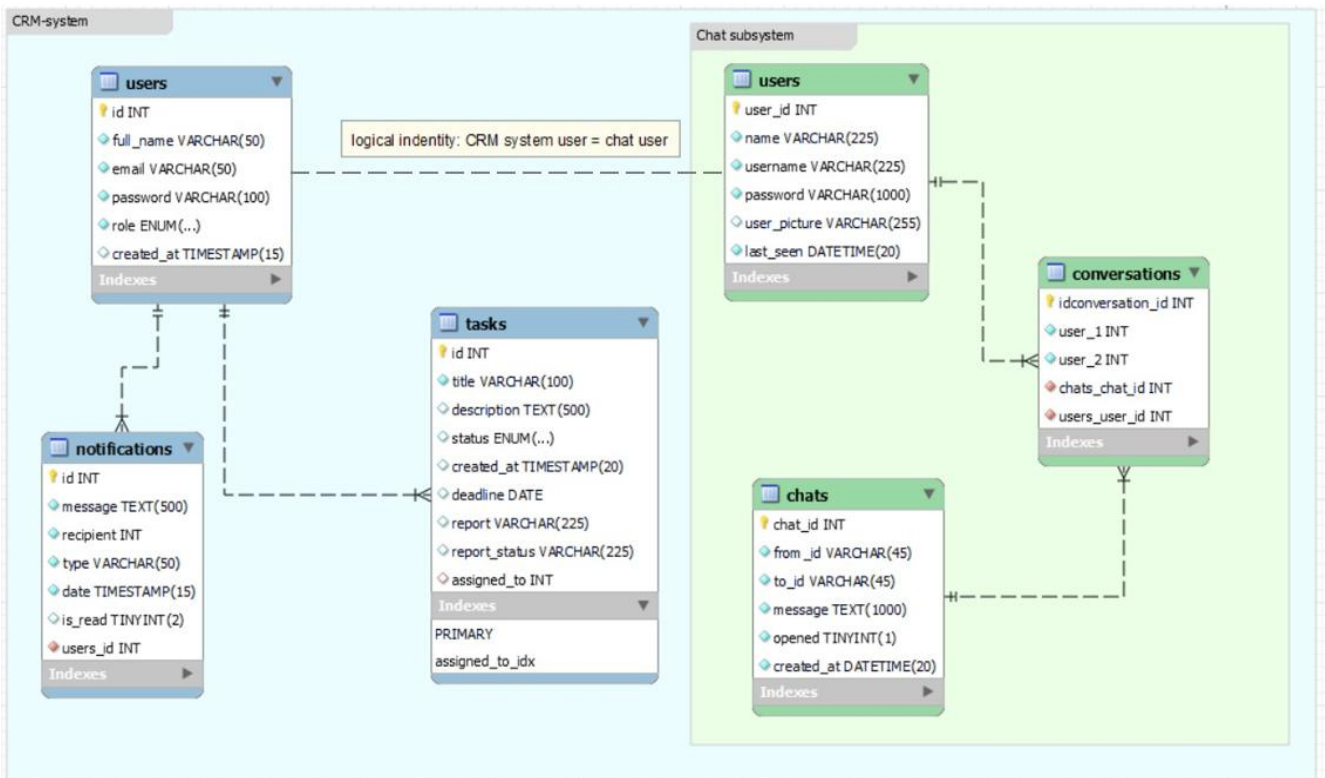


Рисунок 2.5 – Логічна ER-модель бази даних для CRM-системи

Проаналізуємо розроблену логічну модель бази даних. На діаграмі зображено таблиці, які представляють сутності та зв'язки, наприклад, один-до-багатьох та багато-до-одного. У кожній таблиці є рядки, які представляють сутності. Кожна сутність має унікальну назву, тип і розмір. Таблиці також містять первинні ключі (PK) та зовнішні ключі (FK). Деякі атрибути мають вимогу NOT NULL, у такі рядки обов'язково повинні містити дані. Але деякі атрибути повинні за логікою залишатися порожніми, у такому випадку доцільно щоб вони були типу NULL. Система складається з двох тісно пов'язаних між собою компонентів: основної CRM-системи та підсистеми корпоративного чату, яка інтегрована у функціональну

логіку CRM та використовується для внутрішньої комунікації між співробітниками. Незважаючи на те, що фізично обидві підсистеми використовують окремі бази даних та незалежні механізми автентифікації і управління сесіями, на рівні предметної області вони взаємодіють та поділяють спільні сутності. Саме тому логічна модель даних об'єднує їх у єдиний концептуальний простір.

Основа CRM-підсистеми становить сутність «users», яка містить дані про співробітників підприємства, їх персональні відомості, контактні дані, облікові записи та ролі доступу. З цією сутністю пов'язані інші ключові таблиці CRM-системи, зокрема таблиця «tasks», яка використовується для фіксації доручень, їхнього змісту, виконавця та поточного статусу. Ця таблиця містить назву завдання, опис завдання, його статус, тобто на якій стадії життєвого шляху зараз знаходиться завдання, час коли було створено завдання, крайній термін виконання завдання, шлях до файлу звіту, статус звіту та кому призначено завдання. У CRM-системі також є таблиця «Notifications», яка забезпечує механізм внутрішніх повідомлень для співробітників та слугує інструментом інформування про події.

Більш детально розглянемо зв'язки між таблицями. Між таблицями «users» та «tasks» застосовується зв'язок один-до-багатьох (One-to-Many). Кожен користувач (user) може мати декілька завдань (tasks), але кожне завдання відноситься до одного конкретного користувача. У таблиці «tasks» є поле «assigned\_to», воно є зовнішнім ключем (FK) що посилається на id у таблиці users.

Між таблицями «users» та «notifications» застосовується зв'язок один-до-багатьох (One-to-Many). Кожен користувач (user) може мати декілька сповіщень (notifications), але кожне сповіщення відноситься до одного конкретного користувача.

Оскільки корпоративний чат є функціональною частиною CRM, але реалізований як окрема підсистема з власною базою даних, у ньому також присутня таблиця «users», яка містить інформацію про учасників чату. Незважаючи на технічну автономність, на концептуальному рівні ця сутність відображає ті самі

реальні об'єкти, що й таблиця користувачів CRM. Тому у логічній моделі між цими двома таблицями встановлено логічний нефізичний пунктирний зв'язок типу «logical identity», який позначає їхню відповідність. Такий зв'язок не є реляційним і не відображається у вигляді зовнішнього ключа, проте він фіксує факт того, що користувачі обох підсистем представляють одну й ту саму сутність у предметній області. Усі операції з обліковими записами у чаті мають зміст лише у випадку, якщо відповідний користувач існує у CRM-системі, а доступ до чату можливий виключно після авторизації у CRM.

Крім сутності «user» у модель бази даних чату також входять сутності «chats» та «conversations», які формують структуру комунікаційних процесів між користувачами. Таблиця «conversations» визначає канали взаємодії між різними користувачами, тоді як таблиця «chats» містить вміст текстових повідомлень, інформацію про їхніх авторів та часові мітки

Розглянемо більш детально зв'язки між сутностями моделі бази даних чату. Між таблицями «users» та «conversations» застосовується зв'язок один-до-багатьох (One-to-Many). Кожен користувач (user) може мати декілька листувань з іншими користувачами (conversations), але кожне листування розпочинається тільки одним користувачем (user). Щоб зробити такий зв'язок, в таблиці «conversations», було створено додатковий рядок для зв'язку з таблицею «users», яке має назву users\_user\_id. Це поле є зовнішнім ключем, що посилається на id у таблиці users.

Між таблицями «chats» та «conversations» застосовується зв'язок один-до-багатьох (One-to-Many). У кожному чаті може міститися багато повідомлень (conversations) але кожне повідомлення належить тільки одному чату (chats). Щоб зробити такий зв'язок, в таблиці «conversations», було створено додатковий рядок для зв'язку з таблицею «chats», яке має назву chats\_chat\_id. Це поле є зовнішнім ключем, що посилається на id у таблиці chats.

Об'єднання обох підсистем у єдину ER-діаграму з використанням кольорового групування та розмежувальних рамок дозволяє наочно відобразити внутрішню

структуру системи та показати підпорядкованість чату як частини CRM-системи. У рамках логічної моделі таке групування підкреслює цілісність функціональної архітектури та демонструє, що хоча на фізичному рівні компоненти розмежовані, на концептуальному вони утворюють єдину систему.

## 2.6 Розробка запитів до бази даних за допомогою мови програмування PHP

Щоб забезпечити взаємодію з сервером даних, для оброблення та збереження інформації про співробітників, завдання та внутрішні виробничі процеси підприємства було обрано мову програмування PHP, як інструмент для керування базою даних.

PHP надає декілька способів підключення до СУБД MySQL, серед яких найпоширенішими є використання розширення PDO (PHP Data Objects) та процедурного інтерфейсу MySQLi. Кожен із підходів має свої особливості, які впливають як на архітектуру застосунку, так і на безпеку та масштабованість роботи з даними.

Розглянемо розширення MySQLi. Воно забезпечує можливість роботи як у процедурному, так і в об'єктно-орієнтованому стилі, підтримує підготовлені запити та дозволяє виконувати основні операції з даними. Проте MySQLi орієнтоване виключно на роботу з MySQL, що знижує гнучкість застосунку у разі необхідності переходу на іншу СУБД.

У свою чергу PDO є універсальним інтерфейсом доступу до даних і підтримує велику кількість драйверів, зокрема MySQL, PostgreSQL, SQLite та інші. Завдяки цьому розробник отримує можливість змінювати тип бази даних без суттєвого переписування логіки взаємодії з нею. Ще однією важливою перевагою використання PDO у CRM-системі є покращена безпека. Механізм підготовлених запитів дозволяє ефективно запобігати SQL-ін'єкціям, оскільки параметри передаються окремо від структури SQL-команди. Крім того, PDO забезпечує

однаковий підхід до оброблення помилок завдяки використанню виключень (exceptions), що полегшує налагодження та підвищує стабільність роботи проекту.

Саме ці фактори стали вирішальними при виборі способу підключення до бази даних. Розроблювальна система повинна мати можливість легко масштабуватись та для цього не виключена необхідність переходу на іншу СУБД. Приймаючи до уваги все сказане вище, було вирішено реалізувати підключення до бази даних через PDO. Розглянемо, яким чином відбувається керування базою даних за допомогою обраного підходу. Нижче наведено код, за допомогою якого реалізовано підключення застосунку CRM-системи до бази даних.

```
<?php
$driver = 'mysql';
$host = 'localhost';
$db_name = 'crm';
$db_user = 'root';
$db_pass = 'qw187h67';
$charset = 'utf8';
$options = [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC];
try{
    $conn = new PDO
(
        "$driver:host=$host;dbname=$db_name;charset=$charset", $db_user, $db_pass,
        $options);
} catch (PDOException $i){
    die("Error connecting to database");
}
```

Після встановлення з'єднання з базою даних, на основі PDO реалізовано набір функцій, які забезпечують виконання CRUD-операцій над таблицями користувачів, завдань та сповіщень. Такі функції інкапсулюють SQL-запити, використовуючи підготовлені оператори з передачею параметрів для мінімізації ризику SQL-ін'єкцій, і повертають або результат запити, або відповідний індикатор, що дозволяє контролювати логіку виконання операцій. Завдяки цьому, реалізація функцій доступу до даних формує простий і водночас безпечний механізм взаємодії із сутністями у межах CRM-системи. Розглянемо, як у нашій системі реалізовано такі запити.

Функція `insert_task()` реалізує додавання нового завдання в базу даних за допомогою параметризованого INSERT-запити. Передача параметрів через масив дозволяє безпечно обробляти текстові поля (назву, опис), а також атрибути виконавця і дату завершення. Використання підготовленого виразу усуває ризик SQL-ін'єкцій та спрощує подальшу підтримку коду. Нижче наведено код, за допомогою якого реалізовано логіку додавання нового завдання в базу даних.

```
function insert_task($conn, $data){  
    $sql = "INSERT INTO tasks (title, description, assigned_to, deadline)  
VALUES(?,?,?,?)";  
    $stmt = $conn->prepare($sql);  
    $stmt->execute($data);  
}
```

Функція `get_all_tasks_due_today()` виконує вибірку завдань із дедлайном на поточну дату та статусом, відмінним від «completed». Логіка відбору реалізована безпосередньо у SQL-запиті, що дозволяє делегувати фільтрацію на рівень СУБД та мінімізувати післяобробку. Функція повертає асоціативний масив невиконаних

завдань, що потребують оперативного реагування. Нижче наведено код, за допомогою якого реалізовано функцію виконання вибірки завдань із дедлайном.

```
function get_all_tasks_due_today($conn){
    $sql = "SELECT * FROM tasks WHERE deadline = CURDATE() AND status !=
'completed' ORDER BY id DESC";
    $stmt = $conn->prepare($sql);
    $stmt->execute(["employee"]);
    if($stmt->rowCount() > 0){
        $tasks = $stmt->fetchAll();
    }else $tasks = 0;
    return $tasks;
}
```

Функція `get_all_tasks_overdue()` повертає всі прострочені завдання, використовуючи порівняння дати дедлайну з поточною датою (`deadline < CURDATE()`) і відсіює всі завдання зі статусом «completed». Такий механізм дозволяє виявляти завдання зі встановленим терміном виконання та може використовуватися для формування автоматичних нагадувань або аналітичних звітів. Нижче наведено код, за допомогою якого реалізовано функцію виконання вибірки завдань із дедлайном.

```
function get_all_tasks_overdue($conn){
    $sql = "SELECT * FROM tasks WHERE deadline < CURDATE() AND status !=
'completed' ORDER BY id DESC";
    $stmt = $conn->prepare($sql);
    $stmt->execute(["employee"]);
    if($stmt->rowCount() > 0){
```

```

    $tasks = $stmt->fetchAll();
} else $tasks = 0;
return $tasks;
}

```

Функція `update_task_status()` виконує оновлення статусу завдання із можливістю зміни дедлайну. Вона інкапсулює логіку переходів між станами життєвого циклу завдання (наприклад, «pending», «in\_progress», «completed»), що дозволяє контролювати зміни у процесі виконання завдання. Нижче наведено код, за допомогою якого реалізовано функцію оновлення статусу завдання.

```

function update_task_status($conn, $data){
    $sql = "UPDATE tasks SET status=?, deadline=? WHERE id=?";
    $stmt = $conn->prepare($sql);
    $stmt->execute($data);
}

```

Функція `get_all_tasks_by_id()` повертає всі завдання, закріплені за конкретним користувачем. Запит виконується як підготовлений оператор з параметром, що дозволяє фільтрувати записи та реалізувати відображення завдань для окремого співробітника. Нижче наведено код, за допомогою якого реалізовано функцію вибірки завдань, які закріплені за користувачем.

```

function get_all_tasks_by_id($conn, $id){
    $sql = "SELECT * FROM tasks WHERE assigned_to=?";
    $stmt = $conn->prepare($sql);
    $stmt->execute([$id]);
    if($stmt->rowCount() > 0){

```

```

    $tasks = $stmt->fetchAll();
} else $tasks = 0;
return $tasks;
}

```

Функція `count_my_tasks_overdue()` виконує підрахунок прострочених завдань, призначених конкретному користувачу. Вона в себе включає перевірку виконавця, оцінку дедлайну та статусу, що дозволяє швидко отримати статистичні показники для моніторингу продуктивності. Нижче наведено код, за допомогою якого реалізовано функцію підрахунку прострочених завдань.

```

function count_my_tasks_overdue($conn, $id){
    $sql = "SELECT id FROM tasks WHERE deadline < CURDATE() AND status !=
'completed' AND assigned_to=?";
    $stmt = $conn->prepare($sql);
    $stmt->execute([$id]);
    return $stmt->rowCount();
}

```

## 2.7 Розроблення логіки взаємодії користувача зі сторінками CRM-системи

У будь-якому застосунку важливою складовою є реалізація програмної логіки, що забезпечує взаємодію користувача з інтерфейсом застосунку. Вона охоплює обробку запитів, відображення відповідних сторінок, виконання CRUD-операцій над даними та динамічне оновлення інформації залежно від дій користувача. Серверна частина CRM-системи реалізована мовою PHP з використанням об'єктних механізмів PDO та модульної організації коду, що забезпечує розмежування відповідальностей між окремими функціональними блоками.

Користувацькі дії ініціюють серверні сценарії, які здійснюють перевірку коректності введених даних, формують SQL-запити, звертаються до бази даних, а також повертають результати у вигляді готових для відображення даних. Завдяки цьому користувачі отримують можливість керувати проєктами, задачами та персоналом, вносити зміни до відповідних сутностей і контролювати їхній поточний стан у реальному часі.

Розглянемо код, за допомогою якого реалізовано логіку взаємодії користувача системи з формою авторизації та забезпечує перевірку коректності вхідних даних перед здійсненням доступу до системи.

На початку виконується ініціалізація сесії та перевірка наявності переданих користувачем параметрів електронної пошти та пароля. Отримані дані проходять етап валідації, під час якого здійснюється усунення зайвих пробілів, екранування небезпечних символів та видалення елементів, що потенційно можуть призвести до некоректної обробки даних або виникнення вразливостей. У випадку, якщо один із обов'язкових параметрів відсутній, користувач перенаправляється назад на сторінку входу із відповідним повідомленням про помилку. Якщо обидва параметри передані, здійснюється пошук запису у базі даних за вказаною електронною адресою. У разі успішного знаходження, виконується перевірка відповідності введеного пароля за допомогою механізму хешування (функція `password_verify`). У разі успішної автентифікації система ідентифікує роль користувача, зберігає необхідні параметри в сесії та перенаправляє його до головної сторінки системи. Якщо ж автентифікація не проходить, користувач отримує повідомлення про некоректні дані. Тим самим, даний модуль забезпечує базову автентифікацію, обробку помилок та контроль доступу на основі ролей, що є необхідним для управління безпечним доступом до CRM-системи. Нижче наведено код, за допомогою якого реалізовано логіку авторизації користувача при вході в CRM-систему.

```
<?php
session_start();
if (isset($_POST['email']) && isset($_POST['password'])) {
    include "../DB_connection.php";
    function validate_input($data) {
        $data = trim($data);
        $data = stripslashes ($data);
        $data = htmlspecialchars($data);
        return $data;}
    $email = validate_input($_POST['email']);
    $password = validate_input($_POST['password']);
    if (empty($email)) {
        $em = "Email is required";
        header("Location: ../login.php?error=$em");
        exit();
    } else if (empty($password)) {
        $em = "Password is required";
        header("Location: ../login.php?error=$em");
        exit();
    } else {
        $sql = "SELECT * FROM users WHERE email = ?";
        $stmt = $conn->prepare($sql);
        $stmt->execute([$email]);
        if ($stmt->rowCount() == 1) {
            $user = $stmt->fetch();
            $emailDb = $user['email'];
            $passwordDb = $user['password'];
            $role = $user['role'];
```

```
$id = $user['id'];
if ($email === $emailDb) {
    if (password_verify($password, $passwordDb)) {
        if ($role == "admin") {
            $_SESSION['role'] =$role;
            $_SESSION['id'] =$id;
            $_SESSION['email'] =$emailDb;
            header("Location: ../index.php");
        }else if ($role == 'employee') {
            $_SESSION['role'] =$role;
            $_SESSION['id'] =$id;
            $_SESSION['email'] =$emailDb;
            header("Location: ../index.php");
        }else if($role == 'pr_manager') {
            $_SESSION['role'] =$role;
            $_SESSION['id'] =$id;
            $_SESSION['email'] =$emailDb;
            header("Location: ../index.php");
        }else {
            $em = "Incorrect email or password";
            header("Location: ../login.php?error=$em");
            exit();}
    }else {
        $em = "Incorrect email or password";
        header("Location: ../login.php?error=$em");
        exit();
    }
}
}else {
```

```
        $sem = "Incorrect email or password";  
        header("Location: ../login.php?error=$sem");  
        exit();  
    } } }  
}else {  
    $sem = "unknown error occurred";  
    header("Location: ../login.php?error=$sem");  
    exit();}  
?>
```

Наступним розглянемо логіку додавання, нового користувача. На першому етапі виконується перевірка ролі поточного користувача, що дозволяє обмежити доступ до створення акаунтів лише адміністраторам. Після цього обробник отримує дані з HTML-форми надсилання, зокрема адресу електронної пошти, пароль та повне ім'я нового користувача, які проходять попередню фільтрацію та очищення з метою запобігання потенційним ін'єкціям та некоректним символам. Далі виконується валідація обов'язкових полів, що дозволяє гарантувати коректність введених даних та надати користувачеві повідомлення про помилки у випадку, якщо вони є.

У разі успішного проходження перевірок система здійснює хешування пароля за допомогою алгоритму `password_hash`, що забезпечує безпечне зберігання паролів для входу в систему. Після цього формується масив параметрів із хешованим паролем, адресою електронної пошти, повним ім'ям та попередньо визначеною роллю нового користувача, після чого ці дані передаються до функції моделі, що відповідає за створення запису у базі даних. У результаті виконання запиту система повідомляє адміністратора про успішне створення користувача або про можливу помилку, перенаправляючи його на відповідну сторінку інтерфейсу. Нижче наведено код, за допомогою якого реалізовано логіку додавання нового користувача.

```
<?php
session_start();
if (isset($_SESSION['role']) && isset($_SESSION['id'])) {
if (isset($_POST['email']) && isset($_POST['password']) && isset($_POST['full_name'])
&& $_SESSION['role'] == 'admin') {
    include "../DB_connection.php";
    function validate_input($data) {
        $data = trim($data);
        $data = stripslashes ($data);
        $data = htmlspecialchars($data);
        return $data;}
    $email = validate_input($_POST['email']);
    $password = validate_input($_POST['password']);
    $full_name = validate_input($_POST['full_name']);
    if (empty($email)) {
        $em = "Email is required";
        header("Location: ../add-user.php?error=$em");
        exit();
    } else if (empty($password)) {
        $em = "Password is required";
        header("Location: ../add-user.php?error=$em");
        exit();
    } else if (empty($full_name)) {
        $em = "Full name is required";
        header("Location: ../add-user.php?error=$em");
        exit();
    } else {
        include "Model/User.php";
```

```

$password = password_hash($password, PASSWORD_DEFAULT);
$data = array($full_name, $email, $password, "employee");
insert_user($conn, $data);
$em = "User created successfully";
header("Location: ../add-user.php?success=$em");
exit();}
}else {
    $em = "unknown error occurred";
    header("Location: ../add-user.php?error=$em");
    exit();}
}else{
    $em = "First login";
    header("Location: ../add-user.php?error=$em");
    exit();}
?>

```

Також розглянемо логіку отримання сповіщень користувачами. На початку здійснюється перевірка наявності активної сесії, що дозволяє гарантувати доступ до функціоналу лише для зареєстрованих користувачів, і забезпечує отримання ідентифікатора користувача, на основі якого формуються подальші запити до бази даних. Після підключення до бази даних та моделі сповіщень виконується виклик функції, що отримує всі сповіщення, пов'язані з конкретним користувачем. Розроблена логіка враховує можливість порожньої вибірки даних, та у разі відсутності записів у базі система формує відповідне повідомлення інтерфейсу.

Якщо система отримує список сповіщень, виконуються операції з їх відображення у вигляді посилань на сторінки перегляду, причому додатково враховується статус прочитання кожного повідомлення. Для непрочитаних сповіщень застосовується спеціальне виділення, що дозволяє покращити

інформативність та акцентувати увагу користувача на нових подіях. Крім того, кожне сповіщення містить текстове повідомлення, дату створення та тип, що надає короткий опис події. У випадку натискання на відповідне повідомлення про нове завдання статус сповіщення оновлюється з не прочитаного на прочитане. Якщо ж користувач не авторизований, система не повертає жодної інформації, що забезпечує базовий рівень безпеки та захист доступу до даних. Нижче наведено код, за допомогою якого реалізовано логіку отримання сповіщень користувачами.

```
<?php
session_start();
if (isset($_SESSION['role']) && isset($_SESSION['id'])) {
    include "../DB_connection.php";
    include "Model/Notification.php";
    $notifications = get_all_my_notifications($conn, $_SESSION['id']);
    if ($notifications == 0) {
?>
        <li>
            <a href="#"> You have zero notification </a>
        </li>
<?php }else{
foreach ($notifications as $notification) {
// $users = get_all_users($conn);?>
        <li>
            <a href="app/notification-read.php?notification_id=?=$notification['id']?>">
                <?php if ($notification['is_read'] == 0) {
                    echo "<mark>".$notification['type']."</mark>:";
                }else echo $notification['type'].": "?>
                <?=$notification['message']?>
```

```

        &nbsp;&nbsp;<small><?=$notification['date']?></small>
    </a>
</li>
<?php} }
}else{
    echo "";
}??>

```

Розглянемо логіку редагування завдання зі сторони працівника. Спочатку система визначає, чи було завантажено файл у полі report, та перевіряє відсутність стандартних помилок PHP при завантаженні. Після цього виконується низка валідаційних перевірок, які включають у себе контроль максимального розміру файлу, відповідність до дозволеного до завантаження формату документів та зображень, а також аналіз коректності імені файлу.

Після проходження валідації застосунок готує директорію для збереження звітів, а також виконує запит до бази даних для отримання попереднього шляху до файлу, який міг бути завантажений раніше. Це дає змогу коректно видаляти застарілі звіти та уникати накопичення непотрібних файлів на диску. Далі формується нове унікальне ім'я для завантажуваного файлу, після чого він переноситься до службової директорії сервера. Завершальним етапом є оновлення запису в таблиці tasks. У відповідний рядок заноситься новий шлях до файлу звіту та встановлюється статус reportstatus = 'uploaded', що фіксує факт успішного завантаження документа працівником.

У разі наявності попереднього файлу, код додатково перевіряє його фізичне розташування, щоб уникнути видалення сторонніх ресурсів, та прибирає старий файл із диска. Таким чином, цей фрагмент коду забезпечує повний цикл завантаження, перевірки, збереження та оновлення звітів у системі, включаючи

роботу з файловою системою та взаємодію з базою даних. Нижче наведено код, за допомогою якого реалізовано логіку редагування завдання зі сторони працівника.

```

    if (isset($_FILES['report']) && $_FILES['report']['error'] !==
UPLOAD_ERR_NO_FILE) {
    $file = $_FILES['report'];
    $allowed_ext = ['pdf','doc','docx','xls','xlsx','png','jpg','jpeg'];
    $max_size = 5 * 1024 * 1024; // 5 MB
    if ($_FILES['error'] !== UPLOAD_ERR_OK) {
        $sem = "File upload error (code: {$_FILES['error']})";
        header("Location: ../edit-task-employee.php?error=" . urlencode($sem) .
"&id=" . urlencode($id));
        exit();}
    if ($_FILES['size'] > $max_size) {
        $sem = "File is too large. Max 5 MB.";
        header("Location: ../edit-task-employee.php?error=" . urlencode($sem) .
"&id=" . urlencode($id));
        exit();}
    $origName = $file['name'];
    $ext = strtolower(pathinfo($origName, PATHINFO_EXTENSION));
    if (!in_array($ext, $allowed_ext)) {
        $sem = "File type not allowed. Allowed: " . implode(", ", $allowed_ext);
        header("Location: ../edit-task-employee.php?error=" . urlencode($sem) .
"&id=" . urlencode($id));
        exit();}
    $uploadDir = __DIR__ . '/../uploads/reports/';
    if (!is_dir($uploadDir)) {
        mkdir($uploadDir, 0755, true);}

```

```

$stmt0 = $conn->prepare("SELECT report FROM tasks WHERE id = ?");
$stmt0->execute([$id]);
$sold = $stmt0->fetch(PDO::FETCH_ASSOC);
$soldReportPath = $sold ? $sold['report'] : null;
$safeBase = preg_replace('/^[^A-Za-z0-9_\-]/', '_', pathinfo($origName,
PATHINFO_FILENAME));
$newFileName = $safeBase . '_' . time() . '_' . bin2hex(random_bytes(4)) . '.';
$ext;
$targetPath = $uploadDir . $newFileName;
if (!move_uploaded_file($file['tmp_name'], $targetPath)) {
    $em = "Failed to move uploaded file.";
    header("Location: ../edit-task-employee.php?error=" . urlencode($em) .
"&id=" . urlencode($id));
    exit();}
$webPath = 'uploads/reports/' . $newFileName;
$updateSql = "UPDATE tasks SET report = ?, reportstatus = 'downloaded'
WHERE id = ?";
$stmt = $conn->prepare($updateSql);
$stmt->execute([$webPath, $id]);
if (!empty($soldReportPath)) {
    $soldFullPath = __DIR__ . '/../' . $soldReportPath;
    if (file_exists($soldFullPath) && is_file($soldFullPath)) {
        $expectedDir = realpath($uploadDir);
        $realOld = realpath($soldFullPath);
        if ($realOld && strpos($realOld, $expectedDir) === 0)
        { @unlink($soldFullPath); }
    }
}
}

```

## 2.8 Розроблення логіки онлайн чату для crm-системи

У межах розроблення веб-орієнтованої CRM-системи для виробничого підприємства додатково було реалізовано модуль корпоративного чату, який забезпечить оперативну комунікацію між співробітниками. Необхідність створення такого інструменту обумовлена тим, що процеси у виробничому середовищі часто потребують швидкого узгодження дій, обговорення поточних завдань та передачі важливої інформації без використання сторонніх месенджерів. Тому інтеграція чату всередині CRM підвищує зручність роботи користувачів та сприяє більшій ефективності командної взаємодії.

Для реалізації чату було розроблено окремий веб-проект, інтегрований у структуру CRM-системи з власною базою даних та незалежним модулем авторизації. Такий підхід дозволяє спростити архітектуру, уникнути конфліктів у схемах таблиць та забезпечити автономність модулю.

Проаналізуємо реалізацію основних функцій корпоративного чату. Розглянемо логіку отримання списку співрозмовників для поточного авторизованого користувача. На першому етапі формується запит до таблиці `conversations`, який вибирає всі діалоги, де користувач виступає як `user_1` або `user_2`. Результат упорядковується за ідентифікатором розмови у спадному порядку, щоб найновіші діалоги відображалися першими у списку. Далі йде перевірка, чи повернула база даних хоча б один запис. Якщо так, то відбувається обробка кожної розмови. Визначається інший учасник діалогу, після чого виконується другий SQL-запит до таблиці `users`, який отримує повний профіль цього співрозмовника. Тож, для кожного знайденого діалогу завчасно завантажується інформація про іншого користувача, з яким ведеться переписка.

Даний код перевіряє наявність розмов, визначає співрозмовників у кожній із них та формує підсумковий список користувачів, з якими ведеться чат. Нижче

наведено код, за допомогою якого реалізовано логіку отримання списку співрозмовників користувача.

```
<?php
function getConversation($user_id, $conn){
    $sql = "SELECT * FROM conversations
           WHERE user_1=? OR user_2=?
           ORDER BY conversation_id DESC";
    $stmt = $conn->prepare($sql);
    $stmt->execute([$user_id, $user_id]);
    if($stmt->rowCount() > 0){
        $conversations = $stmt->fetchAll();
        $user_data = [];
        # looping through the conversations
        foreach($conversations as $conversation){
            # if conversations user_1 row equal to user_id
            if ($conversation['user_1'] == $user_id) {
                $sql2 = "SELECT *
                       FROM users WHERE user_id=?";
                $stmt2 = $conn->prepare($sql2);
                $stmt2->execute([$conversation['user_2']]);
            }else {
                $sql2 = "SELECT *
                       FROM users WHERE user_id=?";
                $stmt2 = $conn->prepare($sql2);
                $stmt2->execute([$conversation['user_1']]);
            }
            $allConversations = $stmt2->fetchAll();
        }
    }
}
```

```

    # pushing the data into the array
    array_push($user_data, $allConversations[0]);
}
return $user_data;
}else {
    $conversations = [];
    return $conversations;}
}

```

Розглянемо логіку обробки отриманих повідомлень та зміну їх статусу на прочитані. Спершу код перевіряє, чи авторизований користувач, і чи передано ідентифікатор іншого учасника діалогу через POST-запит. Після цього формуються два основні SQL-запити. Перший запит звертається до таблиці chats та вибирає всі повідомлення, які були надіслані від другого користувача до поточного авторизованого користувача. Вибірка повідомлень проводиться в порядку за зростанням ідентифікатора, що дозволяє виводити їх у хронологічному порядку. Після код перевіряє, чи було прочитане кожне повідомлення раніше. Якщо статус opened дорівнює 0, тобто повідомлення ще не відкривалося, формується другий SQL-запит, який оновлює цей статус до 1 у відповідному рядку таблиці. Лише після оновлення статусу повідомлення, виводиться його текст та час відправлення.

Таким чином система отримує та обробляє всі вхідні повідомлення для вибраного діалогу. Нижче наведено код, за допомогою якого реалізовано логіку обробки отриманих повідомлень.

```

<?php
session_start();
if (isset($_SESSION['username'])) {
    if (isset($_POST['id_2'])) {

```

```

include '../db.conn.php';
$Sid_1 = $_SESSION['user_id'];
$Sid_2 = $_POST['id_2'];
$opend = 0;
$sql = "SELECT * FROM chats
      WHERE to_id=?
      AND from_id= ?
      ORDER BY chat_id ASC";
$stmt = $conn->prepare($sql);
$stmt->execute([$Sid_1, $Sid_2]);
if ($stmt->rowCount() > 0) {
    $chats = $stmt->fetchAll();
    foreach ($chats as $chat) {
        if ($chat['opened'] == 0) {
            $opened = 1;
            $chat_id = $chat['chat_id'];
            $sql2 = "UPDATE chats
                  SET opened = ?
                  WHERE chat_id = ?";
            $stmt2 = $conn->prepare($sql2);
            $stmt2->execute([$opened, $chat_id]);
        }
    }
    <p class="ltext border
        rounded p-2 mb-1">
        <?=$chat['message']?>
        <small class="d-block">
            <?=$chat['created_at']?>
        </small>

```

```

    </p>
    <?php
    }}}}else {
header("Location: ../../index.php");
exit;
}

```

Розглянемо логіку відправлення нового повідомлення у чаті. Спочатку йде перевірка чи авторизований користувач, а також чи передано через POST-запит текст повідомлення та ідентифікатор користувача-одержувача. Далі програма отримує вміст повідомлення та ідентифікатор відправника зі змінної сесії та формує перший SQL-запит на вставку нового повідомлення у таблицю chats. Якщо запис успішно створено, система додатково перевіряє, чи існує вже між цими двома користувачами запис про розмову в таблиці conversations. Для цього виконується другий SQL-запит, який шукає наявну розмову незалежно від того, хто був ініціатором. Якщо запису немає, створюється новий запис у таблиці conversations. Такий підхід дозволяє правильно формувати список діалогів у майбутньому. Після цього формується HTML-фрагмент, який одразу відображає щойно відправлене повідомлення разом із позначкою часу. Нижче наведено код, за допомогою якого реалізовано логіку відправлення нового повідомлення у чаті.

```

<?php
session_start();
if (isset($_SESSION['username'])) {
    if (isset($_POST['message']) &&
        isset($_POST['to_id'])) {
        include '../db.conn.php';
        $message = $_POST['message'];

```

```

$to_id = $_POST['to_id'];
$from_id = $_SESSION['user_id'];
$sql = "INSERT INTO
      chats (from_id, to_id, message)
      VALUES (?, ?, ?)";
$stmt = $conn->prepare($sql);
$res = $stmt->execute([$from_id, $to_id, $message]);
if ($res) {
    $sql2 = "SELECT * FROM conversations
           WHERE (user_1=? AND user_2=?)
           OR (user_2=? AND user_1=?)";
    $stmt2 = $conn->prepare($sql2);
    $stmt2->execute([$from_id, $to_id, $from_id, $to_id]);
    define('TIMEZONE', 'Europe/Kyiv');
    date_default_timezone_set(TIMEZONE);
    $time = date("h:i:s a");
    if ($stmt2->rowCount() == 0 ) {
        $sql3 = "INSERT INTO
              conversations(user_1, user_2)
              VALUES (?,?)";
        $stmt3 = $conn->prepare($sql3);
        $stmt3->execute([$from_id, $to_id]);}?>
<p class="rtext align-self-end
      border rounded p-2 mb-1">
    <?=$message?>
    <small class="d-block"><?=$time?></small>
</p>
<?php

```

```

    }
}
}else {
header("Location: ../../index.php");
    exit;
}

```

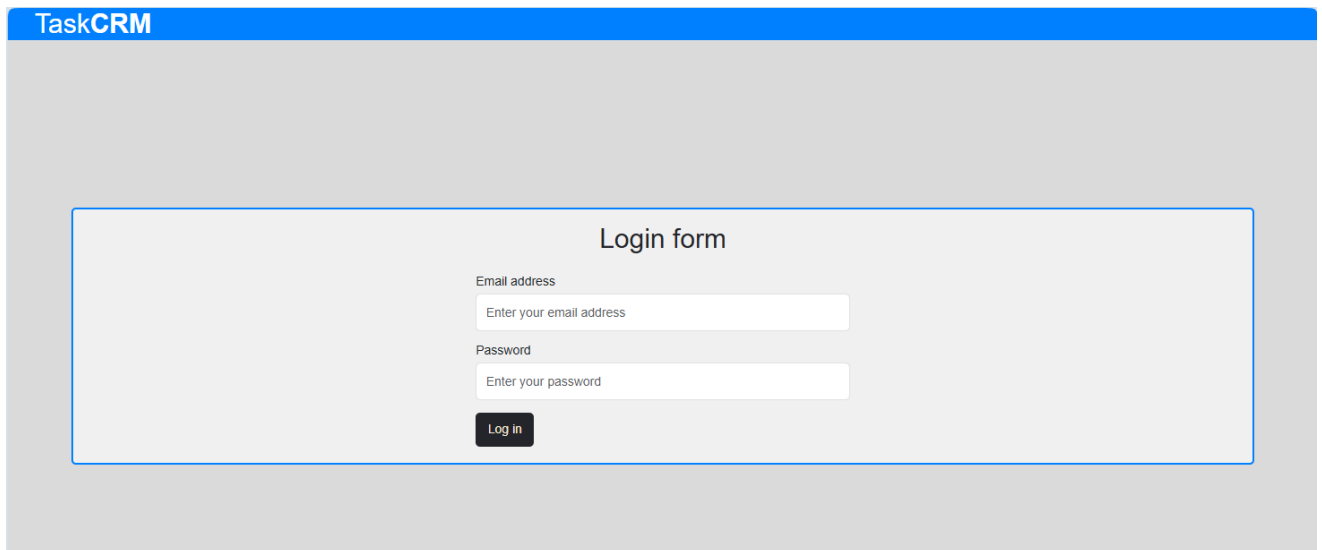
## 2.9 Реалізація інтерфейсу користувача

Реалізація інтерфейсу користувача у веб-орієнтованій CRM-системі є дуже важливим етапом розроблення, оскільки саме інтерфейс визначає зручність взаємодії співробітників із системою та впливає на ефективність виконання щоденних бізнес-процесів. У межах цього проєкту було створено повноцінний веб-інтерфейс, який забезпечує доступ до всіх функціональних можливостей системи та підтримує інтуїтивно зрозумілу навігацію.

Особливу увагу під час реалізації інтерфейсу було приділено простоті його структури, інтуїтивно зрозумілому розташуванню елементів та узгодженості візуальних компонентів. Інтерфейс побудований таким чином, щоб користувач міг швидко знаходити необхідні інструменти, переглядати інформацію про клієнтів, завдання чи внутрішні процеси, а також виконувати операції без складних додаткових дій. Усі сторінки системи мають єдиний стиль оформлення, що забезпечує цілісність та зручність роботи.

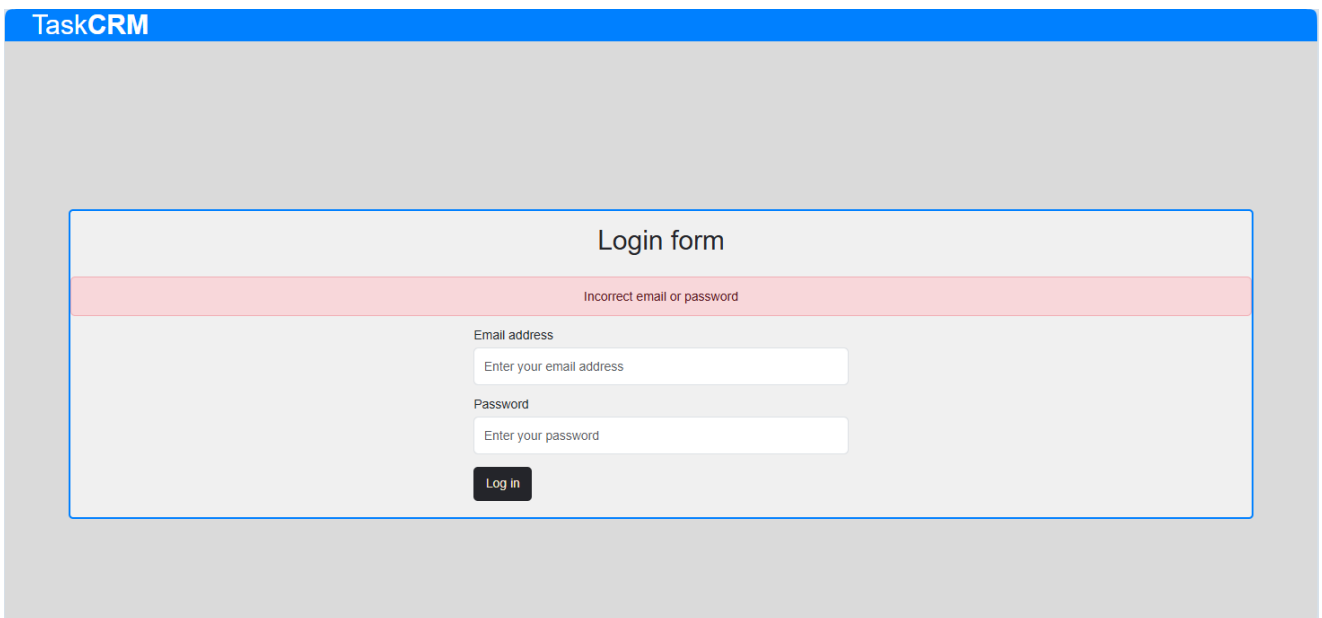
Докладно розглянемо основні сторінки розробленої CRM-системи.

Першою сторінкою, на яку потрапляє користувач є сторінка авторизації з формою входу. Так як реєструвати користувачів може тільки адміністратор системи, то увійти можуть тільки ті, у кого є корпоративна пошта та пароль та вже є зареєстрованими у системі. Це зроблено для того, щоб зайти могли тільки співробітники підприємства (рис. 2.6, рис. 2.7).



The screenshot shows the TaskCRM login form. At the top left, there is a blue header with the text "TaskCRM". Below this, the form is centered on a light gray background. The form is titled "Login form" and contains two input fields: "Email address" with the placeholder text "Enter your email address" and "Password" with the placeholder text "Enter your password". Below the password field is a black "Log in" button.

Рисунок 2.6 – Форма авторизації в CRM-системі



The screenshot shows the TaskCRM login form with an error message. At the top left, there is a blue header with the text "TaskCRM". Below this, the form is centered on a light gray background. The form is titled "Login form". A red horizontal bar spans the width of the form, containing the text "Incorrect email or password". Below the error bar, the form contains two input fields: "Email address" with the placeholder text "Enter your email address" and "Password" with the placeholder text "Enter your password". Below the password field is a black "Log in" button.

Рисунок 2.7 – Форма авторизації при введенні неправильних даних

У разі якщо користувач увів не правильний пароль або адресу електронної пошти, у шапці форми з'явиться червоне повідомлення про невірно введенні дані (рис. 2.7). Після успішного входу користувача перенапрвляє на головну сторінку CRM-системи. Головною сторінкою є панель керування. Така панель є як у адміністратора системи так у проєктного менеджера та звичайного працівника. Але

ця сторінка відрізняється кількістю відображеної інформації залежно від ролі користувача.

У верхній частині на всіх сторінках CRM-системи зображено логотип та кнопка, яка відкриває бічне навігаційне меню, кнопка посилання на корпоративний чат та кнопка, яка відкриває віконце з сповіщеннями. У бічному навігаційному меню наведено електронну пошту з якої користувач зайшов до системи та посилання на різні сторінки застосунку. Посилання на поточну сторінку, на якій зараз знаходиться користувач підсвічується синім кольором. Це спрощує навігацію між сторінками у системі.

На саму адміністративну сторінку виводиться вся статистична інформація щодо всіх користувачів та розпочатих або виконаних проєктів. На цій сторінці адміністратор може побачити кількість співробітників, які наразі працюють на підприємстві, повну кількість завдань, кількість завдань, виконання яких є протермінованим, завдання на виконання, яких не було встановлено дедлайн, всі завдання, що потрібно завершити сьогодні, кількість всіх сповіщень, кількість всіх назначених завдань, кількість всіх завдань, які перебувають у роботі, кількість всіх виконаних завдань, повна кількість завдань, що є активними (зі статусом «pending» або «in\_process»), завантаженість працівників відображається у вигляді кількості завдань, що приходяться на одного користувача. Далі відображається статистична інформація у відсотках. Виведена інформація відображає відсоток завдань без дедлайну, відсоток завдань, що було протерміновано, відсоток активних завдань та відсоток виконаних завдань. Розглянемо адміністративну панель керування (рис. 2.8).

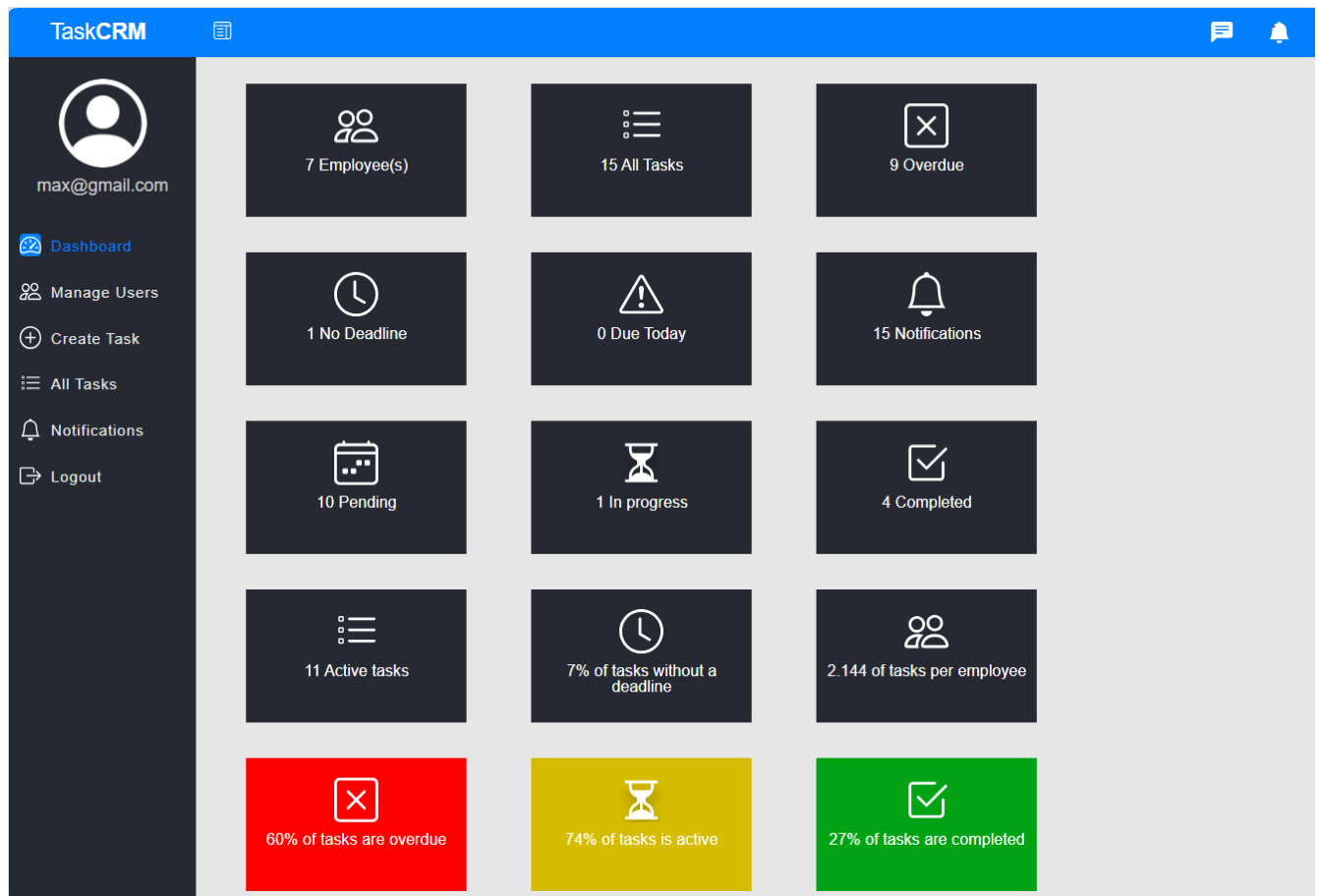


Рисунок 2.8 – Адміністративна панель керування

На сторінці керування користувачами відображається список всіх користувачів, що були зареєстровані в системі. Ще є два активних посилання на сторінку додавання нового користувача та на сторінку редагування інформації про користувача. А також є кнопка за допомогою якої, можна видалити користувача з системи (рис. 2.9).

При натисканні посилання «Add User» адміністратор переходить на сторінку додавання нового користувача. На цій сторінці зображено форму, у якій потрібно заповнити поля «Full Name», «Email» та «Password». Після того, як дані були введені потрібно натиснути кнопку «Add» й новий користувач буде доданий у систему. Також з цієї сторінки можна повернутися назад до списку користувачів за допомогою активного посилання «Back to users» (рис. 2.10).

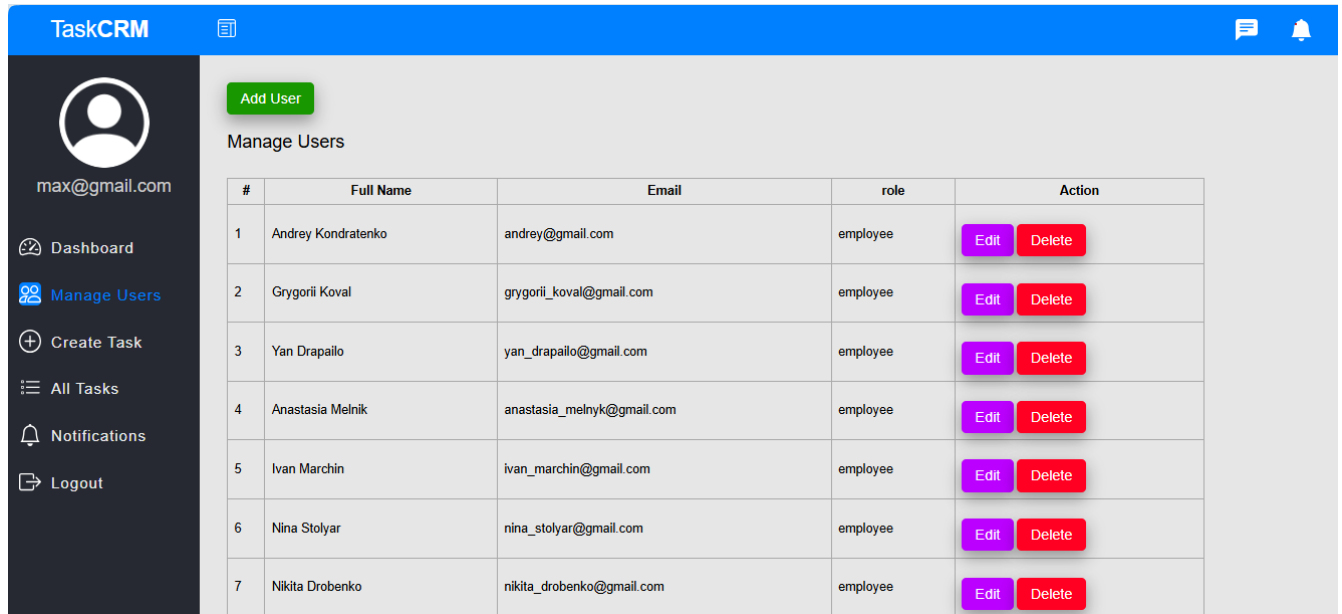


Рисунок 2.9 – Сторінка керування користувачами

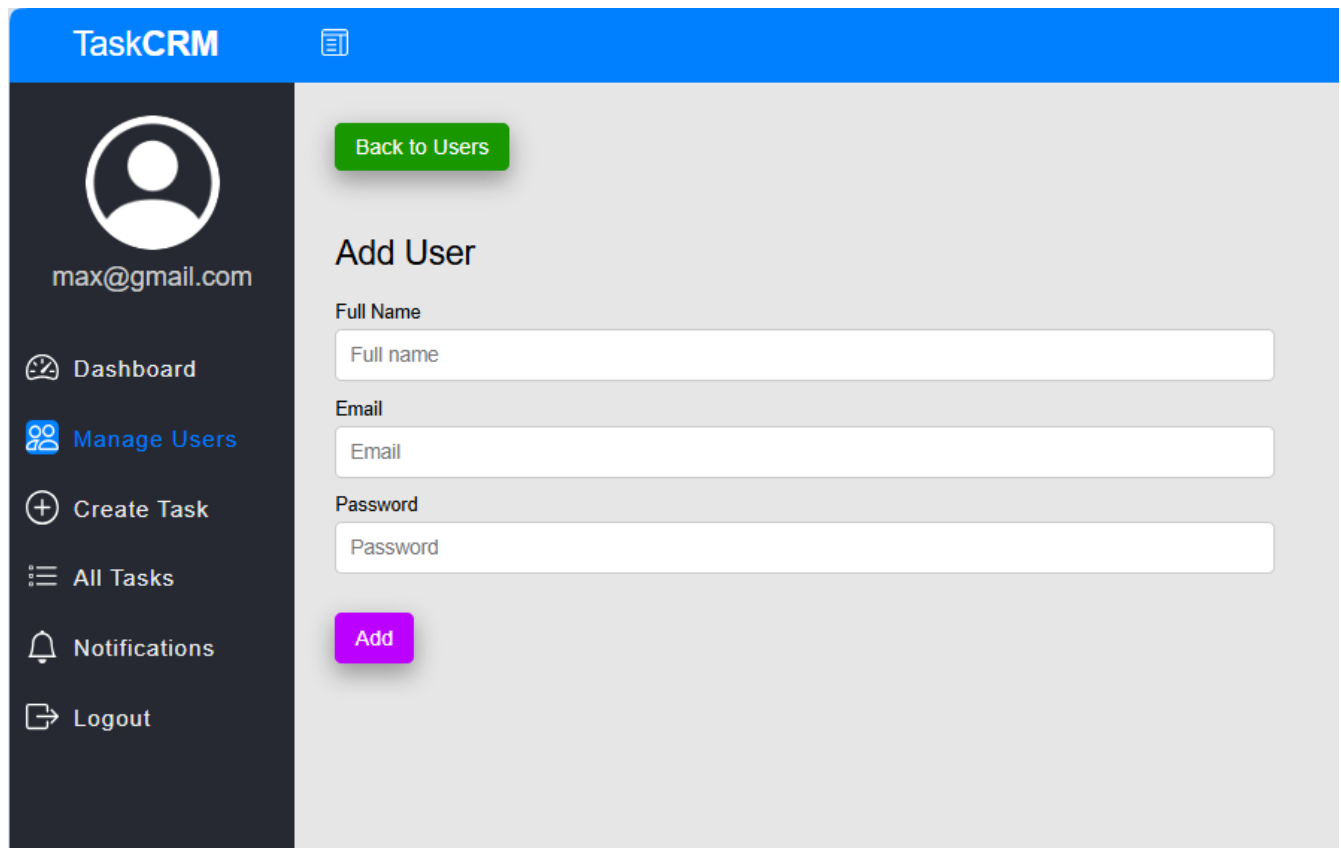


Рисунок 2.10 – Сторінка додавання нового користувача

Сторінка редагування даних користувача зображена на рисунку 2.11.

Рисунок 2.11 – Сторінка редагування даних користувача

На сторінці створення нового завдання потрібно заповнити рядки у формі. Вони в себе включають заголовок (Title), опис завдання (Description), кому призначено завдання (assigned\_to), кінцевий термін виконання (Deadline). Натиснувши на кнопку (Create Task) дані користувача зміняться (рис. 2.12).

Ще однією сторінкою у цьому застосунку є All tasks. На цій сторінці зображено список всіх завдань та наведена вся інформація щодо кожного завдання. Це назва, опис завдання, кому воно назначено, статус завдання, кінцевий термін задачі завдання, статус звіта (завантажено/не завантажено). Також на цій сторінці адміністратор може завантажити звіт з будь-якого завдання. Кнопка завантаження звіту з завдання буде відображатися, якщо користувач, якому воно було назначено його завантажив. В іншому випадку буде світитися статус «No reports». Також адміністратор натиснувши на кнопку «EDIT» може внести зміни у завдання або видалити його за допомогою кнопки «Delete». Для зручності пошуку, можна відфільтрувати завдання за чотирьма критеріями (рис. 2.13).

**TaskCRM**

**Create Task**

Title

Description

Assigned to

Deadline

Create Task

Рисунок 2.12 – Сторінка створення нового завдання

**TaskCRM**

Create Task

Filter tasks: Due Today Overdue No Deadline All tasks

*All tasks (14)*

#	Title	Description	Assigned To	Status	Deadline	Report Status	Action
1	Inspection of Technical Condition of Equipment in Workshop No. 2	It is necessary to carry out a scheduled inspection of the main production equipment in Workshop No. 2. Pay special attention to the condition of drives, cooling systems, and emergency stop buttons. Based on the inspection results, prepare a short report and upload it to the system.	Grygorii Koval	completed	2025-12-04	Download report	Edit Delete
2	Preparation of the Production Report for the Current Month	You need to collect data on production volumes for the current month. The report should include the quantity of manufactured products, downtime, and possible reasons for deviations from the plan. Attach the completed document to the task in the CRM system.	Grygorii Koval	completed	2025-12-05	Download report	Edit Delete
3	Quality Control of Finished Products	Conduct a random quality inspection of finished products in the warehouse. Pay particular attention to packaging and compliance with technical requirements. If any defects are identified, record them in the report.	Grygorii Koval	pending	2025-12-05	No Reports	Edit Delete
4	Update of Occupational Safety Instructions	It is necessary to review the current occupational safety instructions for production employees. If outdated items are identified, prepare proposals for their update and send them to the administrator.	Grygorii Koval	pending	2025-12-05	No Reports	Edit Delete
	Preparation of	You need to prepare the equipment for					Edit

Рисунок 2.13 – Сторінка керування всіма завданнями

На сторінці редагування завдання відображається вся інформація, яка була вказана при його створенні. Адміністратор може змінити будь-які дані у формі й щоб дані були занесені, потрібно натиснути кнопку (Update) (рис. 2.14).

Рисунок 2.14 – Сторінка редагування завдання

У шапці всіх сторінок зображено символ дзвіночка, якщо користувач отримує нове сповіщення то на ньому з'являється червона індикація з числом непрочитаних сповіщень й відкриється вікно для їх перегляду. Розглянемо сторінки, які доступні працівнику. Першою сторінкою на яку попадає працівник після авторизації є панель керування. На цій сторінці наведена статистична інформація, щодо повної кількості завдань, що були назначені працівнику, кількість завдань, виконання яких є протермінованим, завдання без дедлайну, всі завдання, що потрібно завершити сьогодні, кількість всіх назначених завдань, кількість всіх завдань, які перебувають у роботі та кількість всіх виконаних завдань (рис. 2.15).

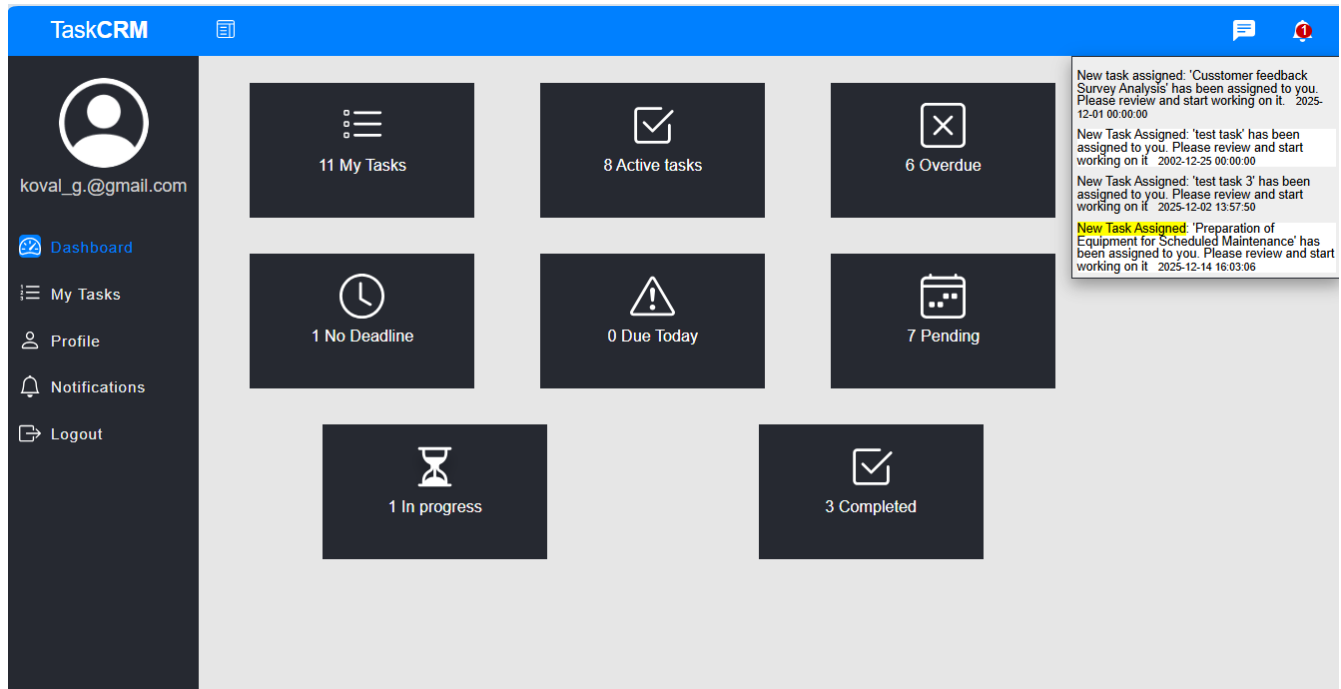


Рисунок 2.15 – Сторінка панелі керування для працівника

На сторінці мої завдання (My Tasks) працівник може переглянути інформацію про всі завдання, що були йому назначені. Також на сторінці відображається інформація щодо завантаження до системи звітів (рис. 2.16).

#	Title	Description	Deadline	Status	Report Status	Action
1	Preparing products for transportation	Prepare all products in the warehouse for loading onto trucks	2025-11-23	completed		Edit
2	Warehouse inventory	Conduct a warehouse inventory and prepare a report.	2025-11-25	pending	uploaded	Edit
3	Monthly Financial Report Preparation	Prepare and review the monthly financial report, including profit and loss statements, balance sheets and cash flow analysis.	2025-11-26	pending	uploaded	Edit
4	Customer Feedback Survey Analysis	Collect and analyze data from the latest customer feedback survey to identify areas for improvement in customer service.	2025-11-27	pending	uploaded	Edit
5	Website Maintenance and Update	Perform regular maintenance on the company website, update content, and ensure all security patches are applied.	2025-11-29	in_progress		Edit
6	Preparation of Equipment for Scheduled Maintenance	You need to prepare the equipment for scheduled technical maintenance. Check the availability of the necessary tools and consumables. After completing the task, update its status in the system.		pending		Edit
7	Update of Occupational Safety Instructions	It is necessary to review the current occupational safety instructions for production employees. If outdated items are identified, prepare proposals for their update and send them to the administrator.	2025-12-05	pending		Edit

Рисунок 2.16 – Сторінка всіх завдань зі сторони працівника

Також працівник може змінити статус завдання та вказати на якій стадії виконання воно зараз перебуває. Ще такий користувач може завантажити звіт з виконання завдання. Для цього йому потрібно перейти зі сторінки «My Tasks», натиснувши на кнопку «Edit». Якщо звіт був вже завантажений у системи, працівник може скачати його на свій пристрій. Також він може змінити звіт на інший. Сторінка для внесення інформації щодо завдань працівником зображена на рисунку 2.17.

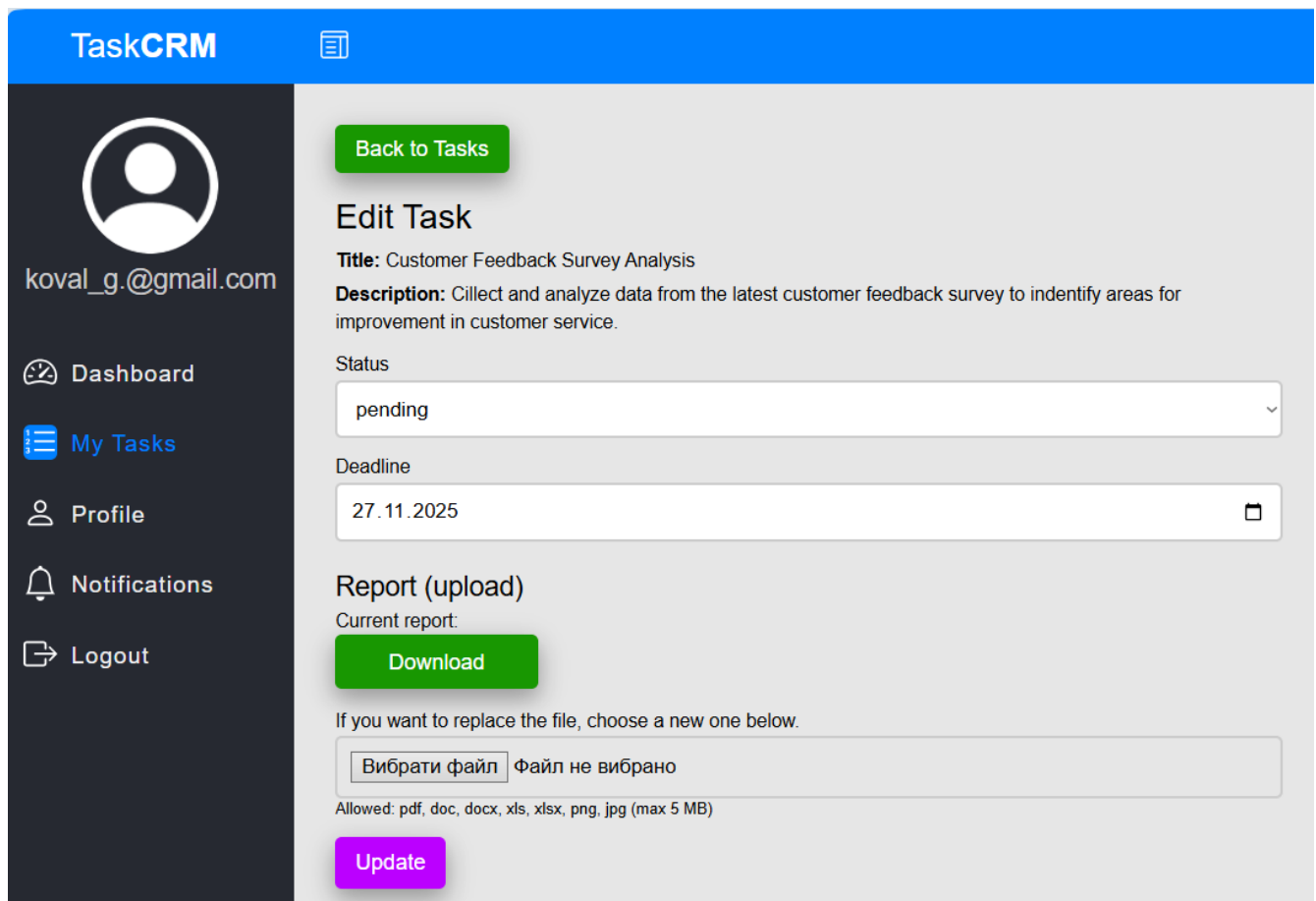


Рисунок 2.17 – Сторінка внесення інформації працівником

Також працівник може змінити свою особисту інформацію перейшовши на сторінку «Профіль». Форма зміни особистої інформації для працівника зображена на рисунку 2.18.

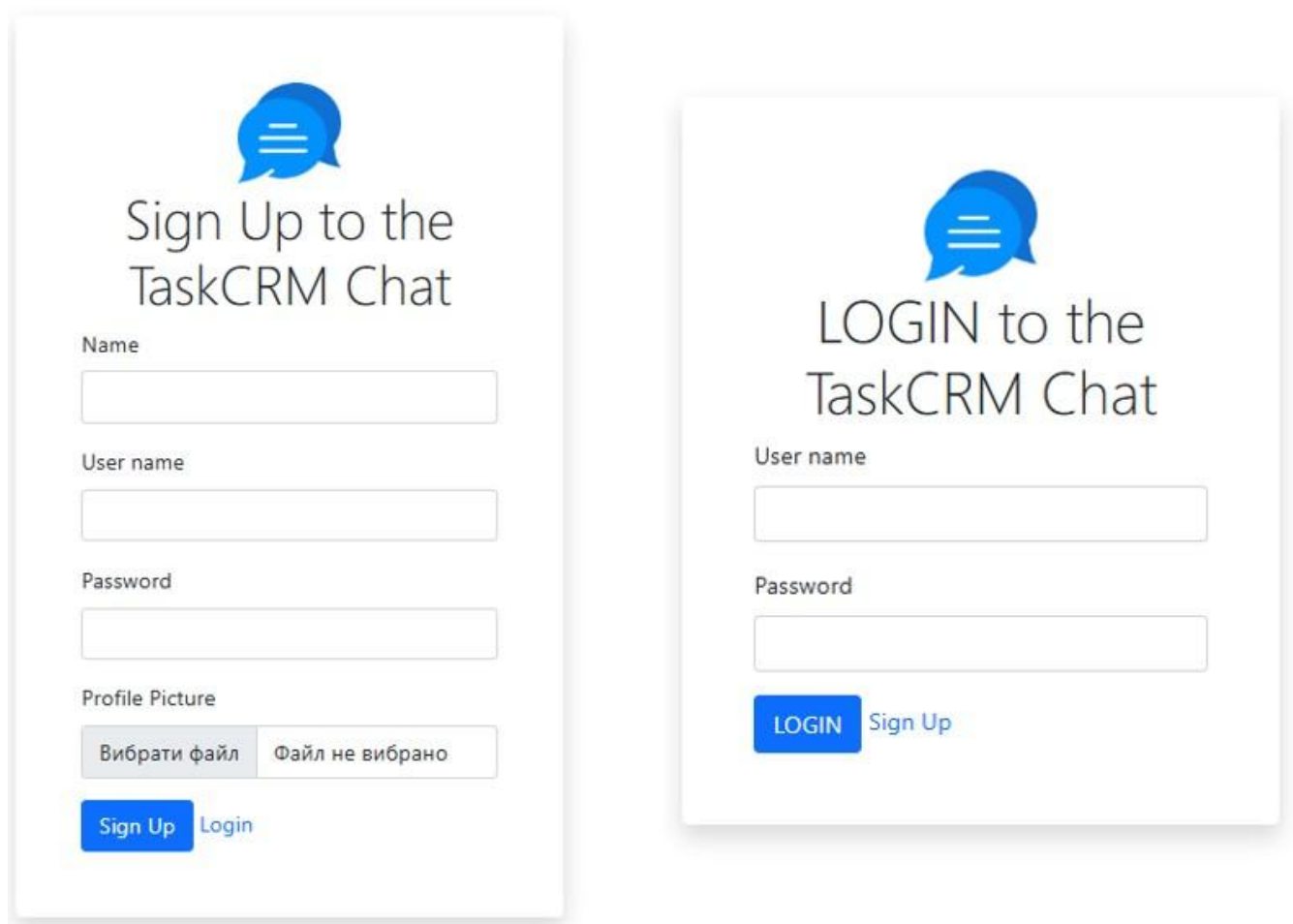
Рисунок 2.18 – Сторінка редагування особистої інформації працівником

На сторінці «Сповіщення» працівник може переглянути зміст, тип і дату та час отримання сповіщення, що надходиди йому (рис .2.19).

#	Message	Type	Date
1	'Customer feedback Survey Analysis' has been assigned to you. Please review and start working on it.	New task assigned	2025-12-01 00:00:00
2	'test task' has been assigned to you. Please review and start working on it	New Task Assigned	2002-12-25 00:00:00
3	'test task 3' has been assigned to you. Please review and start working on it	New Task Assigned	2025-12-02 13:57:50
4	'Preparation of Equipment for Scheduled Maintenance' has been assigned to you. Please review and start working on it	New Task Assigned	2025-12-14 16:03:06

Рисунок 2.19 – Сторінка перегляду інформації щодо сповіщень працівником

З будь-якої сторінки CRM-системи, крім сторінки входу, можна перейти до корпоративного чату. При натисканні на посилання у вигляді повідомлення у шапці сторінки, у браузері відкриється додаткове вікно входу до корпоративного чату (TaskCRM Chat). Для того, щоб користуватися чатом, потрібно додатково зареєструватися у ньому, використовуючи такі самі дані для входу, як й при вході до CRM-системи. Також для реєстрації у чаті потрібно завантажити фото профіля. Сторінки з формама реєстрації та авторизації до чату зображені на рисунку 2.20.



The image displays two side-by-side screenshots of web forms for the TaskCRM Chat. Both forms feature a blue speech bubble icon with three horizontal lines inside, representing a chat. The left form is titled 'Sign Up to the TaskCRM Chat' and includes input fields for 'Name', 'User name', and 'Password'. Below these is a 'Profile Picture' section with a file selection button labeled 'Вибрати файл' and a status indicator 'Файл не вибрано'. At the bottom, there are two buttons: a blue 'Sign Up' button and a grey 'Login' button. The right form is titled 'LOGIN to the TaskCRM Chat' and includes input fields for 'User name' and 'Password'. At the bottom, there are two buttons: a blue 'LOGIN' button and a grey 'Sign Up' button.

Рисунок 2.20 – Сторінки реєстрації та авторизації у корпоративному чаті

Після реєстрації або входу у чат, користувача перенаправляє на сторінку зі списком всіх створених чатів. Якщо ще не було створено чатів, то буде відображатися повідомлення про відсутність листувань (рис. 2.21).

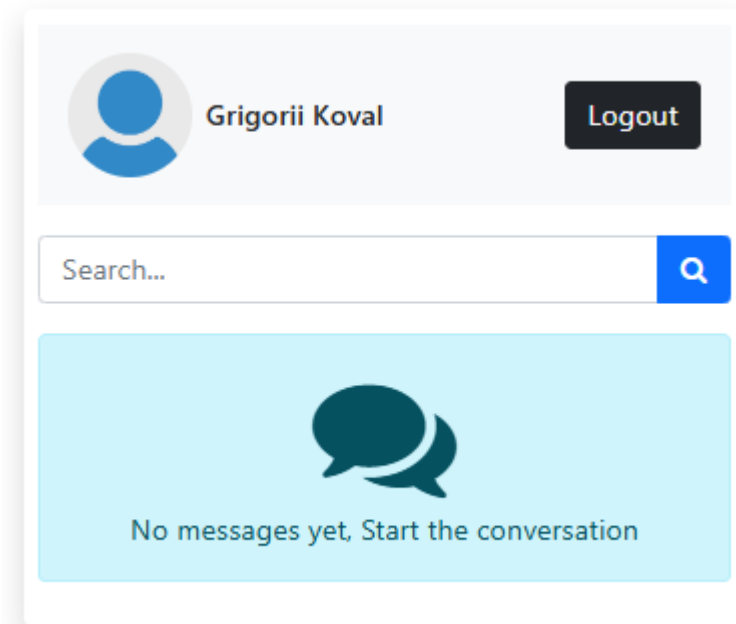


Рисунок 2.21 – Сторінка чату після реєстрації

Після того як користувач був зареєстрований він може у полі пошуку за ім'ям знайти людину, якій хоче написати. При надсиланні людині повідомлення автоматично створюється новий чат (рис. 2.22).

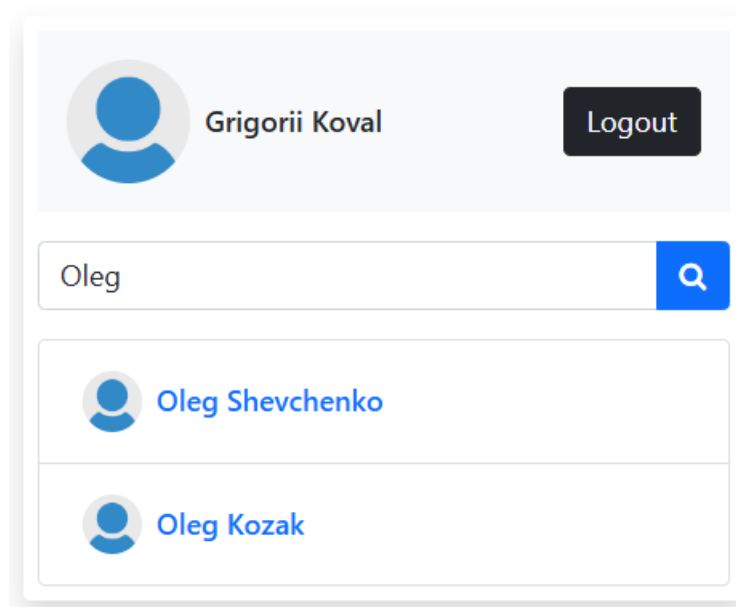


Рисунок 2.22 – Пошук за ім'ям у чаті

Після того, як користувач знайшов потрібну людину, він може написати їй, натиснувши на її ім'я. Після цього створиться чат, натиснувши на який, відкриється сторінка, де буде проводитись листування. На цій сторінці можна побачити ім'я свого співрозмовника, коли він був останній раз у мережі, можна переглянути надіслані повідомлення та відправити нове, ввівши у поле текст та натиснувши на кнопку відправлення. На кожному повідомленні буде відображатися дата та час його надсилання. На рисунку 2.23 зображено демонстраційне листування між користувачем Oleg Kozak та користувачем Grigorii Koval.

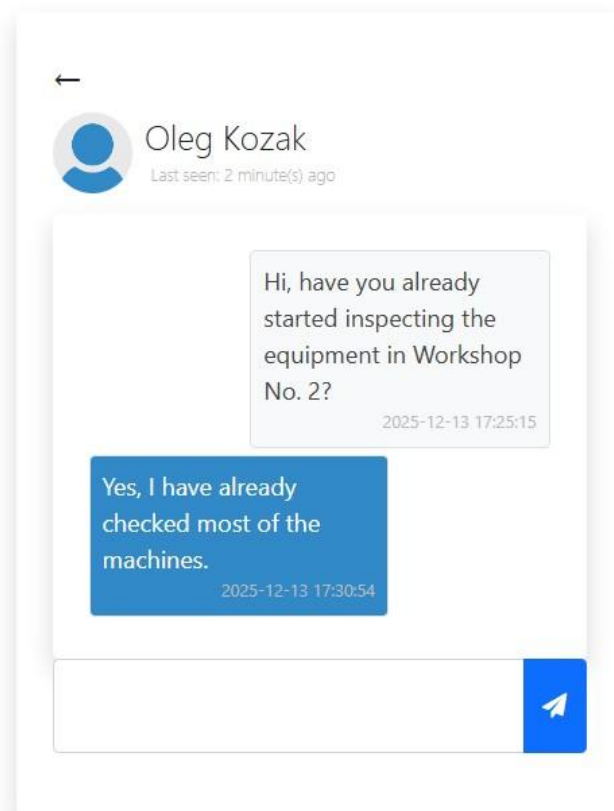


Рисунок 2.23 – Листування між користувачами

Якщо користувачу надійшло нове повідомлення й він його ще не переглянув, то на сторінці списку всіх листувань буде індикація навпроти чату з новим листом (рис. 2.24).

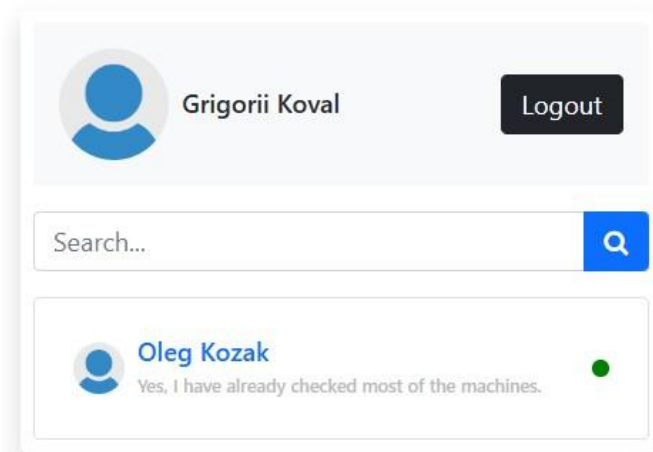


Рисунок 2.24 – Зелений індикатор на чаті з новим непрочитаним повідомленням

Інтеграція корпоративного чату в CRM-систему розширює її функціональні можливості та забезпечує зручний канал оперативної взаємодії між співробітниками в межах єдиного інформаційного середовища. Реалізований чат дозволяє обговорювати робочі питання без використання сторонніх засобів комунікації, що сприяє підвищенню узгодженості дій персоналу, скороченню часу на прийняття рішень та підвищенню ефективності організації виробничих процесів.

### 3 ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Тестування є невід’ємною складовою процесу розроблення програмного забезпечення та спрямоване на перевірку коректності роботи системи, відповідності її функціональних можливостей поставленим вимогам, а також забезпечення стабільності та надійності під час експлуатації.

Основною метою тестування є виявлення можливих помилок у роботі системи, перевірка правильності реалізації основних функцій, а також оцінка зручності використання CRM-системи кінцевими користувачами. Тестування проводилося на етапі завершення розробки та включало перевірку як стандартних сценаріїв використання, так і обробку помилкових та граничних ситуацій.

#### 3.1 Функціональне тестування CRM-системи

Під час функціонального тестування перевіряється робота найважливіших модулів системи, зокрема модулів авторизації, керування користувачами, завданнями, звітами та обміну повідомленнями між співробітниками. Тестування виконувалося шляхом моделювання дій користувачів з різними ролями (адміністратор, співробітник) та перевірки результатів виконання операцій (табл. 3.1).

Таблиця 3.1 – Результати функціонального тестування

№	Модуль	Тестовий сценарій	Очікуваний результат	Результат
1	Авторизація	Вхід з коректними даними	Успішний вхід у систему	Успішно

Продовження таблиці 3.1

№	Модуль	Тестовий сценарій	Очікуваний результат	Результат
2	Авторизація	Вхід з некоректним паролем	Виведення повідомлення про помилку	Успішно
3	Користувачі	Створення нового користувача	Користувач додається до БД	Успішно
4	Користувачі	Зміна інформації щодо користувача	Дані у базі даних оновлюються	Успішно
5	Завдання	Створення нового завдання	Завдання додається до БД	Успішно
6	Завдання	Зміна статусу завдання	Статус оновлюється	Успішно
7	Звіти	Завантаження файлу звіту	Файл зберігається та прив'язується до завдання	Успішно
8	Сповіщення	Надходження сповіщення при створенні завдання для користувача	Сповіщення завантажується у базу даних	Успішно
9	Чат	Відправлення та отримання повідомлень	Повідомлення завантажуються до БД	Успішно

Проаналізувавши результат функціонального тестування можна сказати, що робота найважливіших модулів системи відбувається коректно. Всі операції виконались успішно.

### 3.2 Оцінка надійності CRM-системи на основі статистичної моделі

У процесі тестування та експериментальних досліджень розробленої веб-орієнтованої CRM-системи для виробничого підприємства доцільно виконати оцінку показників її надійності з використанням імовірнісних статистичних моделей. Для опису часу до відмови програмної системи використано експоненціальний закон розподілу, який широко застосовується для оцінювання надійності інформаційних систем.

Ймовірність безвідмовної роботи системи протягом часу визначається за формулою:

$$P(t) = e^{-\lambda t}, \quad (3.1)$$

де  $\lambda$  – інтенсивність відмов, 1/год;

$t$  – тривалість безперервної роботи системи, год.

При експлуатаційному моделюванні CRM-системи було розглянуто два сценарії. Ідеалізований сценарій, за якого протягом тестового періоду не зафіксовано жодної критичної помилки. У такому випадку інтенсивність відмов  $\lambda=0$  і ймовірність безвідмовної роботи дорівнює  $P(t) = 1$ . Але такий сценарій є малореалістичним для реальної експлуатації. Тому було розглянуто реалістичний сценарій, за якого змодельовано одну відмову за 100 годин роботи системи. У цьому випадку інтенсивність відмов буде становити:

$$\lambda = \frac{1}{100} = 0,01 \text{ 1/год.}$$

Проведемо розрахунок ймовірності безвідмовної роботи протягом 24 годин. Підставимо значення інтенсивності відмов у формулу (3.1):

$$P(24) = e^{-0,01 \cdot 24} = e^{-0,24} \approx 0,7866.$$

Отже, ймовірність безвідмовної роботи CRM-системи протягом 24 годин становить приблизно 78,7 %.

Розрахуємо середній час безвідмовної роботи (середній час напрацювання на відмову). Таке значення визначається як обернена величина інтенсивності відмов:

$$T_{\text{ср}} = \frac{1}{\lambda}. \quad (3.2)$$

Підставивши значення  $\lambda = 0,01$  1/год, отримаємо:

$$T_{\text{ср}} = \frac{1}{0,01} = 100 \text{ годин.}$$

Таким чином, середній час безвідмовної роботи CRM-системи становить 100 годин.

Розрахуємо коефіцієнт готовності системи, який розраховується за формулою:

$$K_r = \frac{T_{\text{ср}}}{T_{\text{ср}} + T_{\text{в}}}, \quad (3.3)$$

де  $T_{\text{ср}}$  – середній час напрацювання на відмову;

$T_{\text{в}}$  – середній час відновлення працездатності системи.

Припустимо, що середній час відновлення системи після збою (перезапуск сервісів, діагностика, перевірка логів) становить:

$$T_{\text{в}} = 0,5 \text{ год,}$$

Тоді коефіцієнт готовності дорівнює:

$$K_r = \frac{100}{100 + 0,5} = \frac{100}{100,5} \approx 0,995.$$

Отримане значення коефіцієнта готовності  $K_r = 0,995$  свідчить про високий рівень готовності системи до експлуатації.

Проведені розрахунки показують, що розроблена веб-орієнтована CRM-система характеризується достатньо високими показниками надійності та готовності. Імовірність безвідмовної роботи протягом доби перевищує 78 %, а коефіцієнт готовності наближається до одиниці. Це підтверджує доцільність використання системи в умовах реальної експлуатації на виробничому підприємстві.

### 3.3 Аналіз стійкості розробленої CRM-системи для виробничого підприємства

Розроблена веб-орієнтована CRM-система виконує функції інформаційної автоматизованої системи управління завданнями, проектами та працівниками на виробничому підприємстві. З огляду на архітектуру та функціональну поведінку систему доцільно розглядати з позицій загальних принципів систем автоматичного управління (САУ).

У ролі вхідного збурення виступають дії користувачів системи, зокрема створення або редагування завдань, зміна їх статусів, завантаження звітів та обмін повідомленнями. Об'єктом управління є процес обробки цих подій CRM-системою, що включає виконання серверної логіки, оновлення стану бази даних та відображення актуальної інформації в інтерфейсі користувача.

Зворотний зв'язок реалізується шляхом відображення поточного стану завдань, системних повідомлень, логів активності та сповіщень користувачів. Наявність зворотного зв'язку дозволяє користувачам оцінювати результат своїх дій і, за необхідності, коригувати подальші рішення.

Розглянемо динамічну модель реакції системи. Реакцію CRM-системи на вхідні впливи доцільно змоделювати у вигляді аперіодичної ланки першого порядку, яка адекватно описує інерційну поведінку інформаційних систем без коливальних процесів.

Передавальна функція такої системи має вигляд:

$$W(s) = \frac{1}{Ts+1}, \quad (3.4)$$

де  $W(s)$  – передавальна функція системи в операторній формі;

$T$  – постійна часу, що характеризує інерційність системи;

$S$  – оператор Лапласа.

У припущенні, що середній час реакції CRM-системи на зміну вхідного впливу становить  $T = 5$  хвилин, перехідна характеристика в часовій області описується виразом:

$$y(t) = 1 - e^{-t/T}, \quad (3.5)$$

де  $y(t)$  – відгук системи у момент часу.

Проведемо оцінку перехідного процесу. Для окремих моментів часу отримаємо:

$$y(5) = 1 - e^{-1} \approx 0,63,$$

$$y(10) = 1 - e^{-2} \approx 0,86,$$

$$y(15) = 1 - e^{-3} \approx 0,95.$$

Отримані значення свідчать, що стабілізація реакції системи досягається приблизно за час  $3T = 15$  хвилин, що є типовим для інформаційних систем з серверною обробкою та участю користувача.

Проведемо аналіз стійкості системи. Для аперіодичної ланки першого порядку характеристичне рівняння має вигляд:

$$Ts + 1 = 0.$$

Звідси:

$$s = -\frac{1}{T} = -0,2. \quad (3.6)$$

Від'ємне значення кореня характеристичного рівняння свідчить про асимптотичну стійкість системи, тобто здатність CRM-системи повертатися до сталого стану після дії збурень.

Проведемо оцінку точності та впливу людського фактора. Для системи з одиничним коефіцієнтом підсилення усталена похибка визначається за формулою:

$$e = \frac{1}{1+K}. \quad (3.7)$$

За умови  $K = 1$  отримаємо:

$$e = \frac{1}{2} = 0,5.$$

Похибка 50 % зумовлена тим, що частина управлінських рішень у CRM-системі приймається користувачем (працівником менеджером або адміністратором), а не автоматично. Таким чином, остаточна компенсація збурення здійснюється за рахунок людського фактору.

Також потрібно врахувати затримки у системі. Окрім інерційності програмної обробки, у CRM-системі наявні додаткові затримки, пов'язані з:

Реакцією менеджера, яка дорівнює  $t_m = 2$  хв та часом оновлення інтерфейсу користувача, який дорівнює  $t_i = 1$  хв. Сумарна затримка системи становить:

$$t_3 = T + t_m + t_i = 5 + 2 + 1 = 8 \text{ хв.}$$

Таким чином, динамічна модель CRM-системи для виробничого підприємства у вигляді аперіодичної ланки першого порядку дозволяє кількісно оцінити її реакцію на входні події. Отримані результати свідчать, що система є стійкою, має передбачувану поведінку та здатна адаптуватися до змін умов експлуатації.

Сумарний час реакції, що враховує програмну інерційність, людський фактор та оновлення інтерфейсу, не перевищує 8 хвилин, що є допустимим значенням для інформаційної системи управління виробничими процесами. Це підтверджує доцільність впровадження розробленої CRM-системи в реальних умовах експлуатації.

### 3.4 Охорона праці

У процесі виконання дипломного проєкту з розробки веб-орієнтованої CRM-системи для виробничого підприємства основна діяльність пов'язана з тривалою роботою за персональним комп'ютером. Тому особлива увага приділяється забезпеченню безпечних та комфортних умов праці відповідно до вимог чинних нормативно-правових актів України з охорони праці.

Роботи виконувались у приміщенні з такими характеристиками: площа приміщення – 24 м<sup>2</sup> (4 м × 6 м), висота стелі – 3,3 м, кількість робочих місць із персональними комп'ютерами – 2. Загальний об'єм приміщення становить 79,2 м<sup>3</sup>. Таким чином, на одне робоче місце припадає 12 м<sup>2</sup> площі та 39,6 м<sup>3</sup> об'єму.

Відповідно до вимог Державних санітарних норм і правил ДНАОП 0.00-1.32-01 «Правила охорони праці під час роботи з дисплейними відеотерміналами

електронно-обчислювальних машин», мінімальна площа на одне робоче місце з ПК повинна становити не менше 6 м<sup>2</sup>, а об'єм не менше 20 м<sup>3</sup>. Отже, зазначені вимоги виконуються з достатнім запасом, що забезпечує комфортні умови праці та належну вентиляцію повітря [19].

Приміщення обладнане системою природного та штучного освітлення. Природне освітлення здійснюється через бокові світлові прорізи (вікна), орієнтовані на північний схід, що відповідає рекомендаціям ДБН В.2.5-28:2018 «Природне і штучне освітлення». Коефіцієнт природної освітленості (КПО) для робочих приміщень з ПК повинен бути не менше 1,5 %, що забезпечує достатній рівень денного освітлення без надмірної яскравості та відблисків.

Для оцінки рівня загального штучного освітлення використовуємо метод питомої потужності. Питома потужність освітлення визначається за наступною формулою:

$$W = \frac{W_{\Sigma}}{S}, \quad (3.8)$$

де  $W$  – питома потужність освітлення, Вт/м<sup>2</sup>;

$S$  – площа приміщення, м<sup>2</sup>;

$W_{\Sigma}$  – загальна потужність освітлювальної установки, Вт.

Загальна потужність освітлення визначається за формулою:

$$W_{\Sigma} = W_{\text{св}} \cdot n_{\text{св}}, \quad (3.9)$$

де  $W_{\text{св}}$  – потужність одного світильника, Вт;

$n_{\text{св}}$  – кількість світильників у приміщенні.

У приміщенні встановлено два світлодіодні світильники потужністю 40 Вт кожен. Отже,

$$W_{\Sigma} = 40 \cdot 2 = 80 \text{ Вт},$$

$$W = \frac{80}{24} \approx 3,33 \text{ Вт/м}^2.$$

Згідно з довідковими таблицями для світлодіодних джерел світла, питомій потужності близько 3,3 Вт/м<sup>2</sup> відповідає освітленість приблизно від 350 лк до 400 лк. Відповідно до ДБН В.2.5-28:2018, мінімально допустима освітленість для робіт з використанням ПК становить 300 лк. Таким чином, рівень штучного освітлення відповідає нормативним вимогам та створює сприятливі умови для зорової роботи [20].

Робочі місця обладнані сучасними моніторами з рідкокристалічними екранами, що зменшує електромагнітне випромінювання та навантаження на зір. Відстань від очей користувача до екрана становить від 60 см до 70 см, що відповідає санітарним рекомендаціям. Організація робочого місця забезпечує правильну робочу позу, зменшуючи статичне навантаження на хребет і м'язи.

Таким чином, умови праці під час розробки CRM-системи відповідають чинним нормативним вимогам України у сфері охорони праці, що забезпечує безпеку, збереження здоров'я та високу працездатність користувачів.

## ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розглянуто та проаналізовано сучасні підходи до автоматизації бізнес-процесів виробничих підприємств, а також обґрунтовано доцільність впровадження CRM-системи як інструменту підвищення ефективності управління підприємством та працівниками. Проведений аналіз показав, що використання інформаційних систем такого типу дозволяє зменшити кількість ручних операцій, підвищити прозорість процесів та покращити контроль виконання завдань.

На основі аналізу вже існуючих рішень, у роботі було визначено основні функціональні та нефункціональні можливості CRM-систем для виробничого підприємства, з урахуванням специфіки організації виробничих процесів, керування, персоналом і внутрішніми завданнями. На основі проведених досліджень було сформульовано вимоги та спроектовано структуру системи, а також розроблено UML-діаграми та логічну модель бази даних, що дозволило наочно відобразити архітектуру розроблюваної системи, її основні компоненти, взаємозв'язки між ними, а також процеси обробки даних і взаємодії користувачів із системою.

В результаті, було створено веб-орієнтовану CRM-систему, яка забезпечує автоматизацію бізнес-процесів виробничого підприємства, включаючи управління користувачами, перегляд статистичних показників продуктивності виконання завдань, облік завдань, контроль їх виконання та зберігання структурованої інформації в базі даних. Також, особливістю системи є інтеграція в неї розробленого корпоративного чату для зручної комунікації між працівниками. Реалізоване програмне рішення побудоване з використанням сучасних веб-технологій та відповідає принципам модульності, масштабованості та зручності використання.

При розроблені системи обрано тривірневу клієнт-серверну архітектуру, яка дозволила оптимізувати обмін даними, підвищити ефективність обробки запитів

та забезпечити стабільне функціонування навіть при збільшенні кількості користувачів.

Окрему увагу було приділено розробці користувацького інтерфейсу системи. Всі кнопки та посилання було розміщено в інтуїтивно зрозумілих місцях. Завдяки цьому користувачі витрачають менше часу на пошук потрібних функцій та тим самим прискорюється робота.

Під час тестування розробленої CRM-системи було перевірено коректність роботи основних функціональних модулів, стабільність обробки даних та відповідність результатів очікуваним вимогам. Отримані результати підтвердили працездатність системи та можливість її використання для автоматизації бізнес-процесів у межах виробничого підприємства. Крім того, встановлено, що впровадження такої системи сприяє підвищенню оперативності обміну інформацією та зниженню ризику помилок, пов'язаних із людським фактором.

Таким чином, поставлена мета кваліфікаційної роботи була досягнута, а всі визначені завдання – виконані.

За результатами проведеної роботи є публікації тез на дотичну тему до теми кваліфікаційної роботи, що пройшли апробацію матеріалів у формі доповіді на конференції наведені у додатку А, та безпосередньо пов'язані з Цілями сталого розвитку, зокрема з Ціллю 8 «Гідна праця та економічне зростання» та Ціллю 9 «Індустріалізація, інновації та інфраструктура». Впровадження таких інформаційних систем сприяє підвищенню ефективності виробництва, оптимізації використання ресурсів, цифровій трансформації бізнесу та підвищенню конкурентоспроможності підприємств у глобальному середовищі. Реалізація проєкту спрямована на підтримку інноваційної діяльності та сталого економічного розвитку шляхом автоматизації управлінських процесів і підвищення якості прийняття рішень.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2023. 49 с.

2. Положення про організацію освітнього процесу у ХНУРЕ [електронний ресурс]: Наказ ХНУРЕ від 27 листопада 2020 р. No 400. – Режим доступу: [https://nure.ua/wp-content/uploads/Main\\_Docs\\_NURE/polozhennja-proorganizacioniju-osvitnogo-procesu-v-hnure.pdf](https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-proorganizacioniju-osvitnogo-procesu-v-hnure.pdf).

3. Положення про організацію проведення практики здобувачів вищої освіти Харківського національного університету радіоелектроніки [Електронний ресурс]: Наказ ХНУРЕ від 03 травня 2019 р. No 222. – Режим доступу : <https://nure.ua/wp-content/uploads/222-vid-03.05.2019-pro-vvedennja-vdiju-rishennja-vchenoi-radi-universite-tu.pdf>.

4. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. No 143. – Режим доступу : [https://nure.ua/wp-content/uploads/Main\\_Docs\\_NURE/143-vid-06.05.2021-pro-vvedennja-v-diju-rishennja-vchenoiradi-universitetu.pdf](https://nure.ua/wp-content/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-diju-rishennja-vchenoiradi-universitetu.pdf).

5. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. К.: ДП “УкрНДНЦ”. 2016. 30 с.

6. Стандарт вищої освіти за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» галузі знань 15 «Автоматизація та приладобудування» для другого (магістерського) рівня вищої освіти, затверджений

наказом МОН України No 1022 від 10.08.2020р. «Про затвердження стандарту вищої освіти за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології для другого (магістерського) рівня вищої освіти» Режим доступу: <https://mon.gov.ua/static-objects/mon/sites/1/vishcha-osvita/zatverdzeni%20standarty/2020/08/10/151-avtomatizatsiya-ta-kit-magistr.pdf>

7. Освітньо-професійна програма «Комп'ютерно-інтегровані технологічні процеси і виробництва». – Режим доступу: [https://nure.ua/wpcontent/uploads/Education\\_programs/2021/2021\\_mag\\_151\\_opp\\_kitpv.pdf](https://nure.ua/wpcontent/uploads/Education_programs/2021/2021_mag_151_opp_kitpv.pdf).

8. Положення про академічну доброчесність [Електронний ресурс]: Наказ ХНУРЕ від 02 лютого 2021 р. No 50. – Режим доступу: [https://nure.ua/wpcontent/uploads/Main\\_Docs\\_NURE/polozhennja-pro-akademi-chnu-dobrochesnist.pdf](https://nure.ua/wpcontent/uploads/Main_Docs_NURE/polozhennja-pro-akademi-chnu-dobrochesnist.pdf).

9. Невлюдов І.Ш., Олександров Ю.М., Андрусевич А.О., Чала О.О. Основи наукових досліджень: Навч. посібник. Prague, OKTAN PRINT, 2024 – 468 с. з іл. ISBN 978-80-88618-68-3, e-book ISBN 978-80-88618-60-0, DOI: 10.46489/ONDNP-24-12. Невлюдов І.Ш. Виробничі процеси та обладнання об'єктів автоматизації: Підручник / Кривий Ріг: КК НАУ. – 2017. – 444 с.

10. Невлюдов, І. Ш. Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0. / І. Ш. Невлюдов, та інші. // Монографія. – Харків. – 2023. – 321 с.

11. Nevliudov, I. GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive CyberDesign CPPS Development / I. Nevliudov, I., et al. // Advances in Dynamical Systems and Applications. – 2021. – 16(2). – С. 441-455

12 Невлюдов, І.Ш. Техніко-економічне обґрунтування інтелектуальному виробництві / І.Ш. Невлюдов // Підручник інженерних Кривий Ріг: видавець Чернявський Д.О. – 2024. – 388 с.

13. Rudenko, M. Overview of algorithmic approaches to forecasting in CRM systems / M. Rudenko, S. Sotnik // «Computer-integrated technologies, automation and robotics» CITAR-2025. – 2025. – pp. 101-105

14. Rudenko, M., Overview of approaches to scaling relational databases in development and adaptation of web applications / M. Rudenko, S. Sotnik // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: Тези доповідей XII Міжнародної науково-практичної конференції (10-12 грудня 2024 р., м. Запоріжжя). [Електронний ресурс] /Електрон. дані. – Запоріжжя: НУ «Запорізька політехніка». – 2024. – С. 398-402

15. Sotnik, S. Evaluating relational database scaling strategies in web engineering / S. Sotnik, M. Rudenko // International Conference on Advanced Trends In Radioelectronics and Infocommunications (ATRIC-2025) (May 21–22, 2025), Lviv Polytechnic Publishing House, Lviv, Ukraine. – 2025. – pp. 224-228

16. Rudenko, M., Sotnik, S. Classification of CRM systems / M. Rudenko, S. Sotnik // Manufacturing & Mechatronic Systems 2025: Proceedings of IX st International Conference, Kharkiv, October 25-26, 2025: Theses of Reports. – 2025. – pp. 42-45

17. Polikanov, K. A. Overview of modern technologies for residential automation / K. A. Polikanov, S. V. Sotnik // «Computer-integrated technologies, automation and robotics» CITAR-2025. – 2025. – pp. 85-89

18. Marunich, R.V. Features of IoT application in the security sector / R.V. Marunich, S. V. Sotnik // «Computer-integrated technologies, automation and robotics» CITAR-2025. – 2025. – pp. 80-84

19. ДНАОП 0.00-1.32-01. Правила охорони праці під час роботи з дисплейними відеотерміналами електронно-обчислювальних машин. – Київ: Держнагляд охорони праці України, 2001. – 20 с.

20. ДБН В.2.5-28:2018. Природне і штучне освітлення. [Електронний ресурс]. – Режим доступу:

[https://econstruction.gov.ua/laws\\_detail/3074958732556240833?doc\\_type=2](https://econstruction.gov.ua/laws_detail/3074958732556240833?doc_type=2)