

**Міністерство освіти і науки України**

**Одеський національний технологічний університет**

Кафедра Автоматизації технологічних процесів і робототехнічних систем



**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «Автоматизація управління ресурсозабезпеченням при розробці ІТ-продукту»

(назва кваліфікаційної роботи згідно наказу ОНТУ)

Здобувача **Божко Н. Є.**  
(прізвище, ініціали)

4-го курсу групи Ат-20

Керівник доцент **Жигайло О.М.**  
(посада, прізвище та ініціали)

**Консультанти:**

\_\_\_\_\_  
(посада, прізвище та ініціали)

\_\_\_\_\_  
(посада, прізвище та ініціали)

\_\_\_\_\_  
(посада, прізвище та ініціали)

**Кваліфікаційна робота допускається до захисту**

Рішення кафедри від «\_\_\_» \_\_\_\_\_ 20\_\_ р., протокол № \_\_\_

Завідувач(ка) кафедри АТПіРС \_\_\_\_\_  
(назва кафедри) (підпис) Підпис (ПІБ)

Одеса – 2024 рік

# ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Автоматизації та робототехніки

Кафедра Автоматизації технологічних процесів і робототехнічних систем

Ступінь вищої освіти Бакалавр

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма Комп'ютерні системи та програмна інженерія в автоматизації

**ЗАТВЕРДЖУЮ**

В.о.зав. кафедри АТПіРС

В.М. Левінський

«23» листопада 2023 р.

## **ЗАВДАННЯ**

### **НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

**Здобувач Божко Назар Євгенійович**

1. Тема роботи «Автоматизація управління ресурсозабезпеченням при розробці ІТ-продукту»
2. Керівник кваліфікаційної роботи: Жигайло Олексій Михайлович, к.т.н. доцент  
Пункти 1 і 2 затверджені наказом ОНТУ № 603-03 від 19.10.23 р.
3. Строк подання здобувачем випускної кваліфікаційної роботи «11» червня 2024 р.
4. Вихідні дані до кваліфікаційної роботи: матеріали індивідуальних завдань виробничої і переддипломної практик, курсових і самостійних робіт.
5. Зміст кваліфікаційної роботи (розділи, які потрібно розробити):
  - 1). Загальна характеристика підприємства та управління бізнес процесу, обґрунтування доцільності та шляхів підвищення ефективності його управління.
  - 2). Розробка комплексу моделей бізнес-процесів підприємства, створення системи показників та концептуальної схеми управління. Імітаційне моделювання обраного бізнес-процесу із застосуванням нотації BPMN для його оптимізації.
  - 3) Затвердження меж об'єкта автоматизації, обґрунтування можливих функцій для його системи управління, опис існуючих аналогів автоматизованої системи управління підприємством (АСУП), розробка технічного завдання на створення нового веб-додатку, як основи АСУП.
  - 4) Розробка та створення концептуальної моделі даних та структури бази даних.
  - 5) Розробка макетів для інтерфейсів користувачів та їх дизайну, верстка дизайн-макетів. Створення функціональних модулів програмного продукту.
  - 6). Розробка SCADA для технологічного об'єкту моніторингу і(або) управління, забезпечення її зв'язку із створеним веб-додатком.
  - 7) Вибір технічних засобів, мета використання і реалізація їх зв'язку із створеним веб-додатком.
  - 8) Вибір методів інтелектуального аналізу даних, представлення результатів їх застосування, розробка алгоритмів для автоматизації підтримки прийняття рішень. Опис реалізованої програмної розробки (інструкція користувача).
  - 9) Попереднє обґрунтування економічної доцільності реалізації проекту.

1. Консультанти розділів кваліфікаційної роботи

Номер розділу	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	Світий І.М., доц. каф. АТПіРС		
Розділи 2, 3	Світий І.М., доц. каф. АТПіРС		
Розділ 4	Світий І.М., доц. каф. АТПіРС		
Розділ 5	Левінський В.М., доц. каф. АТПіРС		
Розділ 6	Степанов М.Т., доц. каф. АТПіРС		
Розділ 7	Степанов М.Т., доц. каф. АТПіРС		
Розділ 8	Левінський В.М., доц. каф. АТПіРС		
Розділ 9	Савенко І.І., проф. кафедри МіЛ		

2. Дата видачі завдання 04 грудня 2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

Номер розділу	Строк виконання етапів роботи	Примітка
Розділ 1	«04» березня 2024 р.	
Розділ 2	«18» березня 2024 р.	
Розділ 3	«01» квітня 2024 р.	
Розділ 4	«15» квітня 2024 р.	
Розділ 5	«29» квітня 2024 р.	
Розділ 6	«13» травня 2024 р.	
Розділ 7	«27» травня 2024 р.	
Розділ 8	«03» червня 2024 р.	
Розділ 9	«10» червня 2024 р.	

Здобувач Божко Н.Є.

Керівник роботи Жигайло О.М.

*Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.*

*Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.*

Здобувач Божко В.Є.

Прізвище, ініціали Підпис

## РЕФЕРАТ

Кваліфікаційна робота бакалавра Божко Назара Євгеновійвича «Автоматизація управління ресурсозабезпеченням при розробці ІТ-продукту» викладена на 97 сторінках, кількість таблиць 14, рисунків – 15, додатків - 5, джерел з переліку посилань - 13.

Ключові слова: ресурсозабезпечення, автоматизація, об'єкт керування, система автоматичного керування, ІТ-продукт, імітаційне моделювання, програмований логічний контролер,.

Об'єкт розробки – Автоматизація управління ресурсозабезпеченням при розробці ІТ-продукту  
Мета роботи – підвищення техніко-економічних та екологічних показників функціонування визрівання сирів, що досягаються за рахунок підвищення динамічної точності контурів САР, зниження числа аварійних ситуацій, покращення взаємодії з оператором-технологом.

Методи дослідження – методи теорії автоматичного керування, ідентифікації об'єкту керування, імітаційного моделювання.

*Отримані результати* – розроблені алгоритми керування, які забезпечують підвищення динамічної точності стабілізації регульованих змінних як в перехідних, так і в сталих режимах роботи САР; алгоритми пуску і зупинки обладнання; програми для ПЛК, що реалізують ці алгоритми; розроблені інтерфейси АРМ оператора-технолога і наладчика САР; фрагменти документації технічного забезпечення системи керування. У роботі наведені дослідження процесу визрівання сирів, як об'єкту керування, в результаті котрих отримані математичні моделі основних каналів керування та збурень. На основі отриманих моделей розроблені алгоритми керування процесом в режимах пуску, зупинки та нештатних ситуаціях. Здійснено вибір комплексу технічних засобів відповідно до потрібного ступеню захисту обладнання. Розроблений комплекс програм, що забезпечує реалізацію алгоритмів на контролері фірми Wago, було виконано в середовищі CodeSys. Також в середовищі SCADA-системи WinCC v8.0 розроблено програмне забезпечення АРМ оператора і наладчика. В роботі представлено комплект технічної документації САК на базі контролера фірми Wago.

ВСТУП.....	6
РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПІДПРИЄМСТВА І РІВНЯ АВТОМАТИЗАЦІЇ ЙОГО БІЗНЕС-ПРОЦЕСІВ	
1.1. Опис підприємства і його основних видів діяльності.....	7
1.2. Опис номенклатури продукції, що випускається, сировинних та енергетичних ресурсів.....	9
1.3. Опис програмного забезпечення (ПЗ), що призначено для управління підприємством та підтримки його інформаційних потоків.....	10
1.4. Виявлення і формалізація основних проблем життєдіяльності підприємства.....	11
1.5. Обґрунтування доцільності розробки або вдосконалення автоматизованої системи управління підприємством (АСУП).....	12
РОЗДІЛ 2. РОЗРОБКА КОМПЛЕКСУ МОДЕЛЕЙ БІЗНЕСПРОЦЕСІВ ПІДПРИЄМСТВА	
2.1. Аналіз організаційної структури підприємства та створення її моделі.....	13
2.2. Виділення бізнес-процесів підприємства та створення його процесної моделі.....	15
2.3. Моделювання бізнес-процесів підприємства в нотації eEPC.....	19
2.4. Розробка системи показників та концептуальної схеми управління обраного бізнес-процесу .....	22
2.5. Імітаційне моделювання із застосуванням нотації BPMN для оптимізації бізнес- процесу підприємства.....	25
РОЗДІЛ 3. АНАЛІЗ І ЗАТВЕРДЖЕННЯ ВИМОГ ДО ПРОГРАМНОГО ПРОДУКТУ, ЯК ОСНОВИ АСУП	
3.1. Затвердження меж об'єкта автоматизації та обґрунтування можливих функцій для його системи управління, які треба автоматизувати.....	26
3.2. Опис існуючих аналогів автоматизованої системи управління для підприємства, яке досліджується.....	28

3.3. Розробка та затвердження технічного завдання на створення програмного забезпечення для АСУП.....	32
<b>РОЗДІЛ 4. ПРОЕКТУВАННЯ БАЗИ ДАНИХ (БД)</b>	
4.1. Розробка і створення концептуальної моделі даних.....	41
4.2. Розробка і створення структури таблиць БД.....	44
<b>РОЗДІЛ 5. СТВОРЕННЯ ВЕБ-ДОДАТКУ, ЯК ОСНОВИ РОЗРОБЛЮВАНОЇ АСУП</b>	
5.1. Розробка макетів для інтерфейсів користувачів та їх дизайну.....	48
5.2. Верстка дизайн-макетів.....	50
5.3. Створення функціональних модулів програмного продукту.....	55
5.4. Опис реалізованої програмної розробки (інструкція користувача).....	64
<b>РОЗДІЛ 6. РОЗРОБКА SCADA ДЛЯ ТЕХНОЛОГІЧНОГО ОБ'ЄКТУ МОНІТОРИНГУ І(АБО) УПРАВЛІННЯ</b>	
6.1. Створення моделі системи автоматичного регулювання в програмному середовищі Step7.....	66
6.2. Створення екранної панелі управління в SCADA WinCC flexible.....	70
6.3. Реалізація обміну даними між SCADA та БД MySQL.....	73
<b>РОЗДІЛ 7. ВЗАЄМОДІЯ СИСТЕМИ З ТЕХНІЧНИМИ ЗАСОБАМИ АВТОМАТИЗАЦІЇ</b>	
7.1. Опис технічних засобів та мета їх використання .....	
7.2. Алгоритм і реалізація підключення технічних засобів .....	
<b>РОЗДІЛ 8. ВИБІР І ЗАСТОСУВАННЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ(ІАД) ДЛЯ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ</b>	
8.1. Опис методів ІАД та обґрунтування доцільності їх використання .....	
8.2. Представлення результатів застосування обраного методу.....	
8.3.Розробка концепції алгоритму для автоматизації підтримки прийняття рішень.....	
<b>РОЗДІЛ 9. ОБґРУНТУВАННЯ ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	
<b>ВИСНОВОК.....</b>	
<b>СПИСОК ЛІТЕРАТУРИ.....</b>	

## Вступ

У сучасному світі інформаційних технологій, коли конкуренція на ринку є надзвичайно високою, ефективне управління ресурсами стає ключовим чинником успіху будь-якої ІТ-компанії. Неправильне розподілення ресурсів, повільне реагування на зміни та ручне керування процесами можуть призвести до втрати часу, збільшення витрат та зниження якості продуктів.

Дана дипломна робота присвячена розробці автоматизованої системи управління ресурсами в компанії Real Solutions. Основна мета дослідження полягає в створенні веб-додатку, який оптимізує процеси планування, розподілу та моніторингу ресурсів, забезпечуючи підвищення продуктивності та зниження витрат.

Для досягнення цієї мети було проаналізовано сучасні методи управління ресурсами в ІТ-компаніях, виявлено їхні недоліки та запропоновано шляхи оптимізації. Розроблений веб-додаток допоможе компанії Real Solutions ефективніше керувати своїми ресурсами, швидше реагувати на потреби клієнтів та підвищити загальну ефективність роботи.

У рамках дипломної роботи буде розглянуто наступні питання: аналіз поточного стану управління ресурсами в компанії Real Solutions, вивчення сучасних методів та підходів до автоматизації управління ресурсами, розробка та впровадження веб-додатку для автоматизації процесів управління ресурсами, а також оцінка ефективності впровадженої системи та її вплив на продуктивність і витрати компанії.

Результати дослідження будуть корисними не лише для компанії Real Solutions, але й для інших ІТ-компаній, які прагнуть оптимізувати свої бізнес-процеси та підвищити ефективність управління ресурсами.

# Розділ 1. Загальна характеристика підприємства і рівня автоматизації його бізнес-процесів

## 1.1. Опис підприємства і його основних видів діяльності

Компанія "Real Solutions" - це компанія яка спеціалізується на створенні та впровадженні IT-продуктів для різних галузей, таких як фінанси, охорона здоров'я, освіта, електронна торгівля та інші. Головною метою компанії є розробка вебсайтів, мобільних або комп'ютерних додатків, які відповідають потребам клієнтів.

Метою компанії Real Solutions є надання ефективних, інноваційних IT-рішень для різних галузей і замовників. Компанія прагне надавати високу якість обслуговування, починаючи з аналізу потреб клієнтів і закінчуючи впровадженням і підтримкою програмних продуктів. На всіх етапах співпраці Real Solutions гарантує максимальну продуктивність і задоволеність клієнтів завдяки постійному моніторингу та вдосконаленню своїх рішень.

Компанія являється молодого та існує всього кілька років. Вона має невеликий офіс у Києві, а також працює з віддаленими співробітниками, що дозволяє залучати таланти з різних регіонів та забезпечувати гнучкість у роботі.

Основні сфери діяльності компанії "Real Solutions" у галузі IT включають:

Одним з основних є створення програмного забезпечення. Бекенд-розробка – це створення внутрішнього механізму веб-застосунків. Вона покриває створення серверної складової проекту (додатка/продукту), яка охоплює обробку даних, взаємодію з базами даних і забезпечення безпеки інформації. Використовуються різні мови програмування, такі як Node.js, Python, Java, та бази даних MySQL, PostgreSQL при розробці бекенд частини.

Frontend - це частина веб-додатків, з якою користувачі можуть прямо взаємодіяти. В неї входить створення інтерфейсу користувача і забезпечення його взаємодії з продуктом на стороні клієнта. Технологій частини фронтенд дуже багато, але стов новітніми з них являються такі як JS, React, Angular, HTML і CSS, для розробки

адаптивних та зручних інтерфейсів.

UX/UI дизайн - це напрями дизайну, які чудово доповнюють один одного.

**User Experience дизайн** — це про те, як користувач відчуває та взаємодіє з продуктом. Це включає дослідження того, що користувачам потрібно і чого вони хочуть, а також створення продукту, який буде зручним і приємним у використанні.

**User Interface дизайн** — це про зовнішній вигляд і стиль продукту. Це включає створення елементів, з якими користувачі взаємодіють, таких як кнопки, меню, ікони та шрифти. Компанія застосовує інструменти, такі як Figma, Adobe Photoshop і Illustrator для розробки прототипів, макетів і інтерактивних моделей.

Забезпечення якості шляхом тестування. Тестування — це процес перевірки відповідності заявлених вимог до продукту та його реальної функціональності шляхом спостереження за його роботою в штучно створених умовах і на обмеженому наборі тестів, обраних спеціально. В компанії проводиться проведення перевірки програмного забезпечення на відповідність вимогам, специфікаціям і очікуваним результатам клієнта з метою виявлення помилок і забезпечення якості.

Інформаційне консультування у галузі ІТ, а саме бізнес консалтинг який виступає як допомога в обранні програмного забезпечення та впровадженні новітніх технологій для клієнтів. Компанія досліджує наявні системи в різних галузях, створює рекомендації та надаємо підтримку у впровадженні оптимальних рішень.

Компанія "Real Solutions" прагне забезпечити високий рівень задоволеності клієнтів завдяки індивідуальному підходу, використанню сучасних технологій та багаторічному досвіду в Інформаційних Технологіях.

## 1.2. Опис номенклатури продукції, що випускається, сировинних та енергетичних ресурсів

Компанія Real Solutions спеціалізується на розробці та впровадженні програмних продуктів для різних галузей, включаючи фінансову галузь, охорону здоров'я, освіту, логістику та онлайн торгівля та інші.

Основні інформаційні продукти, що випускається компанією, є:

Веб додатки. Компанія розробляє різноманітні веб-додатки які підтримуються на будь-якому браузері, що забезпечують зручний доступ до інформації та сервісів через інтернет.

CRM-системи. Компанія виробляє програми для управління взаємовідносинами з клієнтами, які дозволяють клієнтам оптимізувати процеси обслуговування клієнтів, маркетингу та продажів.

Крім того, створюються та впроваджуються ERP-системи, які допомагають керувати всіма основними операціями бізнесу, такими як фінанси, людські ресурси, постачання та виробництво.

Виробництво мобільних додатків, але не мобільних ігор. Окрім усього вищезазначеного, також розробляються мобільні додатки.

До всього можна додати також розробку та випуск лендінг сторінок (односторінковий сайт-візитка), дизайн логотипів, графічний дизайн та інше. Сфера є дуже розвиненою на сьогоднішній день та продовжує свій зріст, тому список можна ще продовжувати, якщо вдаватися в деталі.

Для розробки вищеперелічених продуктів компанії необхідно використовувати різноманітні сировинні ресурси, основні з яких є співробітники компанії, а саме їх інтелектуальний потенціал та цінний опит який включає в себе знання, досвід та навички в розробці програмного забезпечення. Другим компонентом є техніка, а

саме комп'ютери або ноутбуки які використовуються для створення програмного коду, розробки дизайну та тестування продуктів.

Також можна віднести маршрутизатори, роутери та інше обладнання для мережі, яке забезпечує зв'язок між працівниками.

Основним енергоресурсом компанії є електроенергія, яка забезпечує життєдіяльність комп'ютерів та інших електронних пристроїв, потрібних для роботи. Рівнозначним є підтримання зручної температури в офісі, що також вимагає використання електроенергії.

### 1.3. Опис програмного забезпечення (ПЗ), що призначено для управління підприємством та підтримки його інформаційних потоків.

Будучи молодою та невеликою компанією, Real Solutions використовує програмне забезпечення MS Project яке є дуже відомим серед компаній.

**Microsoft Project** - це програмне забезпечення, створене Microsoft, яке сприяє керуванню проектами, дозволяючи менеджерам контролювати виконання завдань та оптимально розподіляти ресурси. Навіть досвідченим менеджерам буде важко працювати в ПЗ, яке є відомим та поширеним.

MS Project може бути використаний для різноманітних проєктів, включаючи як невеликі завдання, так і великомасштабні проєкти з багатьма учасниками. Більше того, Microsoft Project дозволяє спільно працювати і обмінюватися даними з іншими користувачами, інтегруватися з іншими програмами та сервісами Microsoft.

MS Project використовується в організації для планування та керування ресурсами та завданнями у межах IT-проєктів. Він дозволяє розподіляти ресурси, встановлювати та контролювати терміни виконання завдань, а також створювати графіки та звіти для звітування та аналізу при потребі.

#### **Недоліки MS Project включають:**

1. Важко зрозуміти та використовувати інтерфейс MS Project, особливо для початківців та недодгдуб[ користувачів, оскільки він дуже схожий на Excel і вимагає відповідні навички. Складність і специфічна структура функцій ускладнювали процес роботи та знижували продуктивність.

2. Microsoft Project є дорогим програмним забезпеченням, що може бути проблематичним для невеликих компаній з обмеженим бюджетом. Крім того,

вартість ліцензії може зрости, якщо потрібно придбати програму для великої кількості користувачів.

3. Для невеликих груп або проєктів програмне забезпечення може виявитися занадто складним через велику кількість функцій, багато з яких можуть залишатися без використання.

Приймаючи до увагу цю інформацію, можна сказати що хоча Microsoft Project має свої переваги,але його недоліки впливають на ефективність та продуктивність робочого процесу, що є причиною для заміни цієї системи або створення власного ПЗ який буде більше підходити під потреби компанії.

#### 1.4. Виявлення і формалізація основних проблем життєдіяльності підприємства

У Real Solutions регулярно проводять анонімні опитування співробітників щодо їх роботи в команді, загального задоволення або скарг. Під час оцінки цих опитувань та діяльності компанії виділено декілька ключових проблем, які так чи інакше мають негативний вплив на продуктивність та вимагають вирішення. Нижче перераховані головні проблеми, з якими зіштовхується компанія.

**Неефективна автоматизація процесів.** Одним з основних питань, яке турбує компанію, є неякісна автоматизація управління проєктами через використання незручного для Real Solutions забезпечення Microsoft Project. Ця програма має складний інтерфейс, який вимагає тривалого навчання для нових співробітників, особливо менеджерів проєктів. Це приводить до неефективного управління проєктами та непорозуміння, що призводить до порушень термінів та вимог виконання завдань та знижує загальну продуктивність компанії.

**Непрозорість процесів.** Іншим важливим аспектом є складність для клієнтів у відстеженні виконання проєктів у Microsoft Project. Ця програма дозволяє відстежувати прогрес проєкту, але не надає достатньої ясності та зручності на цьому ключовому етапі. Часто клієнти висловлюють незадоволення через неясність стану завдань і відсутність зрозумілих звітів про прогрес. Це має негативний вплив на взаємодію з менеджером проєкту, їх задоволеність і довіру до компанії, оскільки вони не можуть ефективно оцінити прогрес проєкту та вносити потрібні зміни вчасно.

Виявлені проблеми підкреслюють, що потрібно поліпшити автоматизацію управління проєктами та внутрішню комунікацію в компанії. Використання власного програмного забезпечення замість Microsoft Project підніме рівень управління проєктами, оскільки воно краще відповідає потребам компанії. Наше нове програмне забезпечення "Менеджер" полегшить спілкування між клієнтами та розробниками завдяки зручному механізму зворотного зв'язку, що допоможе

уникнути недорозумінь та забезпечити вчасне виконання завдань. Це допоможе підвищити загальну продуктивність та удосконалити якість виконання проектів у компанії.

### 1.5. Обґрунтування доцільності розробки або вдосконалення автоматизованої системи управління підприємством (АСУП)

В даний час велика кількість ІТ компаній швидко розвивається, а автоматизація стає необхідною складовою їхньої роботи. Неможливо уявити ефективну ІТ компанію, де всі процеси виконуються без автоматизації. Спосіб управління всіма процесами “вручну” швидко спричинить низьку продуктивність та втрату конкурентоспроможності.

Автоматизація сприяє оптимізації процесів, зменшенню витрат і підвищенню якості виконання завдань, що має велике значення в сучасному світі високих технологій.

Компанія "Real Solutions" застосовує програмне забезпечення Microsoft Project для керування проектами. Цей програмний засіб має свої плюси, включаючи велику кількість різноманітних функцій для керування проектами. Проте це не завжди відповідає вимогам компанії, особливо при урахуванні її масштабів. Складний інтерфейс, висока вартість ліцензій та відсутність прозорості у відстеженні процесів ускладнює виконання проектів.

На основі вказаних проблем потрібно розробити власне програмне забезпечення для ефективної автоматизації управління ресурсами та проектами компанії.

Нижче наведено основні переваги цієї ідеї:

**Зрозумілий інтерфейс.** Програма буде мати простий та зручний інтерфейс, який не потребуватиме тривалого навчання нових співробітників і дозволить їм швидко адаптуватися до роботи з системою.

**Ефективне спілкування.** Програма має надати зручні способи зворотного зв'язку між клієнтами та розробниками. Це буде включати можливість обміну коментарів і документів.

**Прозорість у проведенні процесів.** ПЗ буде забезпечувати прозорість у спостереженні за процесами виконання проектів. Клієнти зможуть відстежувати стан завдань у реальному часі і залишат необхідні коментарі або додавати завдання.

**Оптимізація використання коштів.** Саме розроблене програмне забезпечення зменшить витрати на ліцензії і дозволить оптимально використовувати ресурси компанії.

**Відеоспостереження за продуктивністю.** Програма буде оснащена функцією відеоспостереження, що дозволить керівництву в режимі реального часу стежити за роботою співробітників, оцінювати їх продуктивність і завантаженість протягом робочого дня.

**Чат для спілкування.** Клієнти зможуть швидко та легко спілкуватися з менеджерами за допомогою інтегрованого чата в програмі. Це забезпечить швидкий обмін інформацією, вирішення проблем і зручне обговорення незначних деталей проектів.

Розробка власного софту є важливим і необхідним кроком для підвищення продуктивності компанії "Real Solutions". Фактично, ПЗ дозволяє оптимізувати управління проектами та ресурсами, забезпечує прозорість та зручність комунікації і допомагає знизити витрати на зовнішні інструменти.

## **Розділ 2. Розробка комплексу моделей бізнес-процесів підприємства**

### **2.1. Аналіз організаційної структури підприємства та створення її моделі**

Сучасні ІТ компанії акцентують увагу на створенні ефективної організаційної структури, яка є важливим для успішної реалізації бізнес-процесів та досягнення стратегічних цілей. У межах дипломної роботи розглядається організаційна структура невеликої ІТ компанії.

Управління структурою - це механізм, який забезпечує сполученість усіх бізнес-процесів. Управління підприємством включає в себе розподіл обов'язків, взаємодію та підпорядкування між підрозділами та керівництвом.

На першому рівні знаходиться вище керівництво, зокрема головний виконавчий директор, інакше кажучи - СЕО, тобто «Перша особа» компанії. Разом з досягненням успіху компанії, генеральний директор також відповідає за створення та впровадження стратегій і спілкується з керівниками інших відділів для узгодження їх дій.

Далі в організації знаходяться відділи за різними напрямками роботи. Таким чином, у компанії працюють відділи розробки, продажів, фінансів та управління персоналом. Безумовно, кількість відділів може варіюватися в залежності від розміру компанії та конкретних цілей, які вона ставить перед собою. Крім того, у

протязі існування компанії структура може зазнавати змін через різкі перетворення або концентрацію на конкретних цілях.

**Відділ розробки** складається з групи фахівців, яка займається розробкою програмного забезпечення або інших ІТ-продуктів. До цього підрозділу можуть входити розробники веб-сайтів або мобільних додатків, дизайнери і тестувальники.

Розробники відповідають за створення чистого та якісного коду, розробку функціональності і виконання вимог замовника. Дизайнери розробляють привабливі інтерфейси та оформлення продукту, щоб забезпечити зручність користування для кінцевих користувачів. Тестувальники аналізують створене програмне забезпечення для перевірки відповідності вимогам, виправляють виявлені помилки та допомагають забезпечити якість продукту перед релізом.

**Команда відділу продажів** займається рекламою та реалізацією послуг компанії. Цей відділ може включати менеджерів з продажу, акаунт-менеджерів та лідогенераторів.

Менеджери з продажу забезпечують пошук, залучення нових клієнтів, укладання угод і проведення переговорів. Акаунт-менеджери укладають та підтримують контакти зі своїми клієнтами, вирішують їх проблеми та потреби. Лідогенератори активно шукають потенційних клієнтів, надсилають холодні повідомлення і поповнюють базу даних.

**Фінансовий відділ** - це група, що відповідає за грошові питання компанії, оплата праці, звітність та інше. В цей відділ можуть входити спеціалісти з фінансового аналізу, бухгалтери та фінансові менеджери.

Аналітики у сфері фінансів аналізують діяльність компанії, вивчають фінансові показники та розробляють стратегії управління фінансами. Бухгалтери відповідають за облік та звітність, контроль фінансових операцій та податкові зобов'язання компанії.

Фінансові менеджери готують і реалізують фінансові стратегії і розпоряджаються фінансовими ресурсами, вирішують завдання фінансового планування та управління.

**Команда відділу управління персоналом** відповідає за пошук, найм, управління та стан людей як ресурсом компанії. Сюди можуть входити кадрові менеджери та фахівці зібрання персоналу, або ж рекрутери.

HR менеджер - це особа, яка виступає посередником між працівниками і керівництвом, захищаючи інтереси обох сторін. Фахівці з рекрутингу шукають, відбирають і оцінюють кандидатів на вільні посади.

Безумовно, у компанії може бути значно більше працівників і ролей в різних відділах, різноманіття керівників та підлеглих може бути дуже великим. У цьому разі досліджується лише організаційна структура компанії "Real Solutions". Кожен відділ є ключовим у роботі компанії та сприяє досягненню стратегічних цілей.

## 2.2. Виділення бізнес-процесів підприємства та створення його процесної моделі

Процесний підхід до управління організацією ґрунтується на виділенні в організації бізнес-процесів та управлінні цими процесами. Мережа процесів — це набір пов'язаних між собою дій і процесів, які охоплюють всю діяльність підприємства.

Для розуміння структури бізнес-процесів необхідно звернутися до їх класифікації. Як правило, вони поділяються на три основні категорії: первинні, підтримуючі та розвиваючі.

1. **Первинні процеси** (основні, операційні) - це основні процеси діяльності підприємства, спрямовані на створення основних продуктів або послуг. Вони включають в себе весь шлях від розробки до постачання кінцевого продукту споживачам.
2. **Підтримуючі процеси** (допоміжні) - це процеси, які забезпечують діяльність первинних процесів. Вони включають в себе всі ресурси та діяльність, необхідну для безперебійного функціонування основних процесів.
3. **Розвиваючі процеси** - це процеси управління та розвитку організації. Вони охоплюють стратегії розвитку, управління персоналом, фінансовий менеджмент тощо.

### Виділення бізнес-процесів у ІТ-компанії

Для ІТ-компанії, яка займається розробкою програмного забезпечення, можна виділити наступні бізнес-процеси:

1. Операційні (основні) процеси:
  - Написання коду, створення програмного забезпечення відповідно до вимог та технічної документації.
  - Перевірка якості та функціональності програмного забезпечення. Проектування. Розробка архітектури програмного забезпечення.
  - Створення технічної документації.
  - Дизайн інтерфейсу користувача

2. Підтримуючі процеси:
- Забезпечення необхідними інструментами та технологіями для розробки.
  - Захист даних та забезпечення конфіденційності інформації.
  - Управління внутрішньою документацією.
  - Організація внутрішніх комунікацій.

3. Розвиваючі процеси:

- Визначення довгострокових цілей та напрямків розвитку
- Аналіз ринку та конкурентного середовища
- Розробка стратегій розвитку
- Управління інвестиціями та витратами
- Впровадження нових технологій

**Описання та побудова процесної моделі бізнес-процесу “Автоматизація управління ресурсозабезпеченням”:**

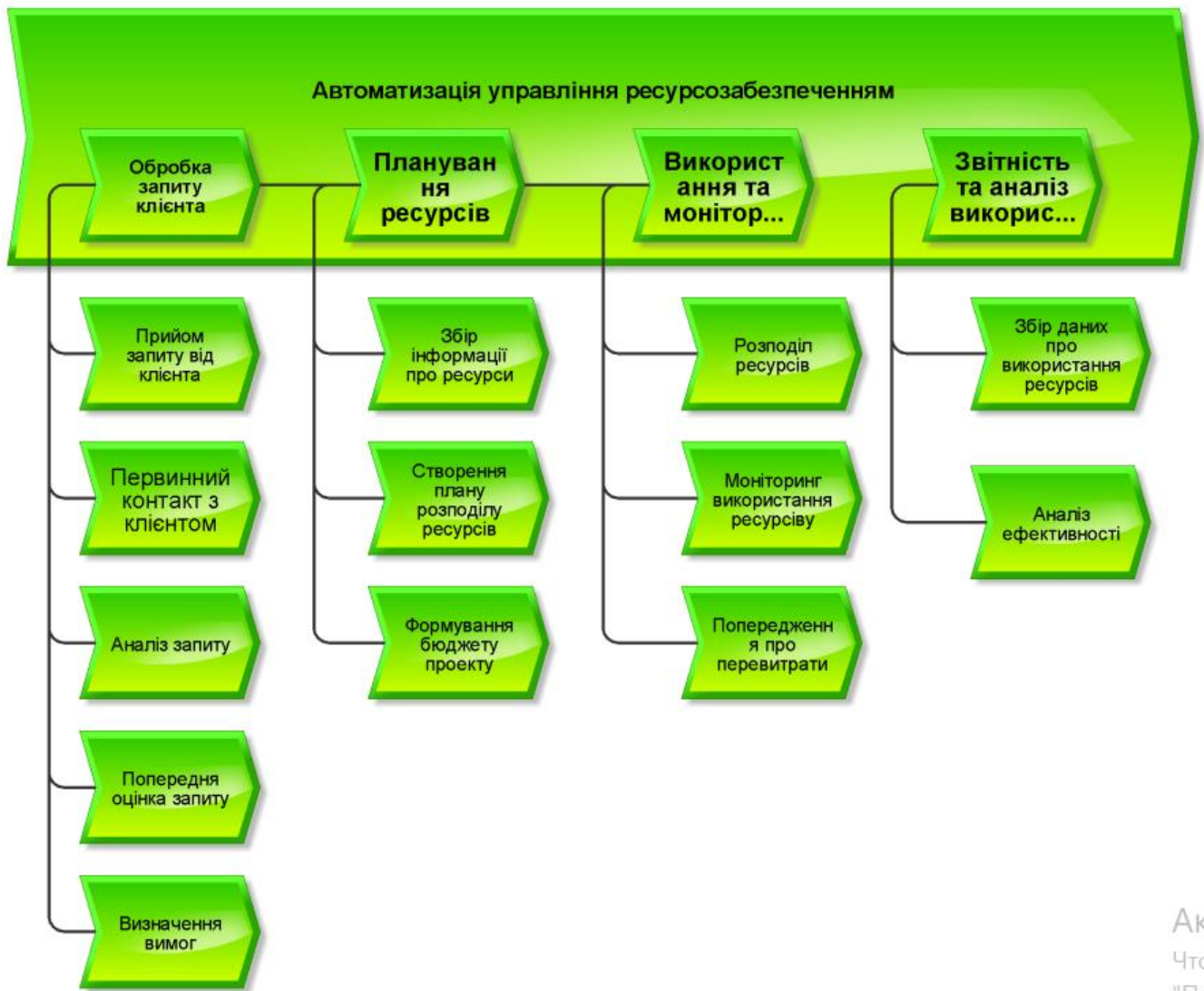
Бізнес-процес "Автоматизація управління ресурсозабезпеченням" включає в себе наступні підпроцеси:

1. Обробка запиту клієнта
2. Планування ресурсів
3. Використання та моніторинг ресурсів
4. Звітність та аналіз використання ресурсів

<b>Автоматизація управління ресурсозабезпеченням</b>		
Назва підпроцесу	Вхід	Вихід
<b>Обробка запиту клієнта</b>		
Прийом запиту від клієнта	Запит клієнта (електронна пошта, телефонний дзвінок, форма на сайті)	Зареєстрований запит у системі
Первинний контакт з клієнтом	Інформація про клієнта, канал зв'язку, початкові вимоги клієнта	Підтверджений запит клієнта, деталі початкових вимог

Аналіз запиту	Підтверджений запит клієнта, початкові вимоги	Деталізований запит
Попередня оцінка запиту	Оцінка можливостей реалізації	Попередній естимейт вартості та часу, технічна оцінка
Визначення вимог	Деталізований запит, додаткові питання до клієнта	Узгоджені вимоги, технічні специфікації
<b>Планування ресурсів</b>		
Збір інформації про ресурси	Запит на розробку продукту, Інформація про наявні ресурси	Зібрані дані про ресурси
Створення плану розподілу ресурсів	Зібрані дані про ресурси	План розподілу ресурсів
Формування бюджету проекту	План розподілу ресурсів	Бюджет проекту
<b>Використання та моніторинг ресурсів</b>		
Розподіл ресурсів	План розподілу ресурсів	Реальні дані про використання ресурсів
Моніторинг використання ресурсів	Реальні дані про використання ресурсів	Оновлені дані про використання ресурсів
Попередження про перевитрати	Оновлені дані про використання ресурсів	Попередження про можливі перевитрати
<b>Звітність та аналіз використання ресурсів</b>		
Збір даних про використання ресурсів	Дані про використання ресурсів	Зібрані дані для звітів
Аналіз ефективності	Зібрані дані для звітів	Звіти про ефективність використання ресурсів

Таблиця 2.1. - Описання підпроцесів, які входять до бізнес-процесу "Автоматизація управління ресурсозабезпеченням"



Ак  
Что  
"Па

Рис. 2.1. – Процесна модель бізнес-процесу "Автоматизація управління ресурсозабезпеченням" *Розроблено в додатку ARIS Express*

Процесна модель бізнес-процесу "Автоматизація управління ресурсозабезпеченням" відображає взаємозв'язок між підпроцесами, їх входами та виходами. Кожен підпроцес має свій вхідний матеріал чи інформацію, що перетворюється у вихідний продукт або послугу, споживану працівниками компанії або зовнішніми клієнтами. Така модель допомагає забезпечити прозорість та ефективність процесу, полегшує управління ресурсами та дозволяє досягати високих результатів у розробці ІТ-продуктів.

### 2.3. Моделювання бізнес-процесів підприємства в нотації eEPC

Визначення функцій і подій бізнес-процесу «Обробка запиту клієнта» є необхідним для побудови моделі в нотації eEPC.

Застосування сучасних підходів до моделювання бізнес-процесів є критично важливим для забезпечення ефективності бізнесу. Сучасні методи моделювання дозволяють організаціям краще прогнозувати події, знижувати ризики та знаходити нові шляхи розвитку.

**Подія** - демонструє стан або зміни в бізнес-процесах. Це може бути початком або кінцем процесу, або важливою подією, яка відбувається в його середині.

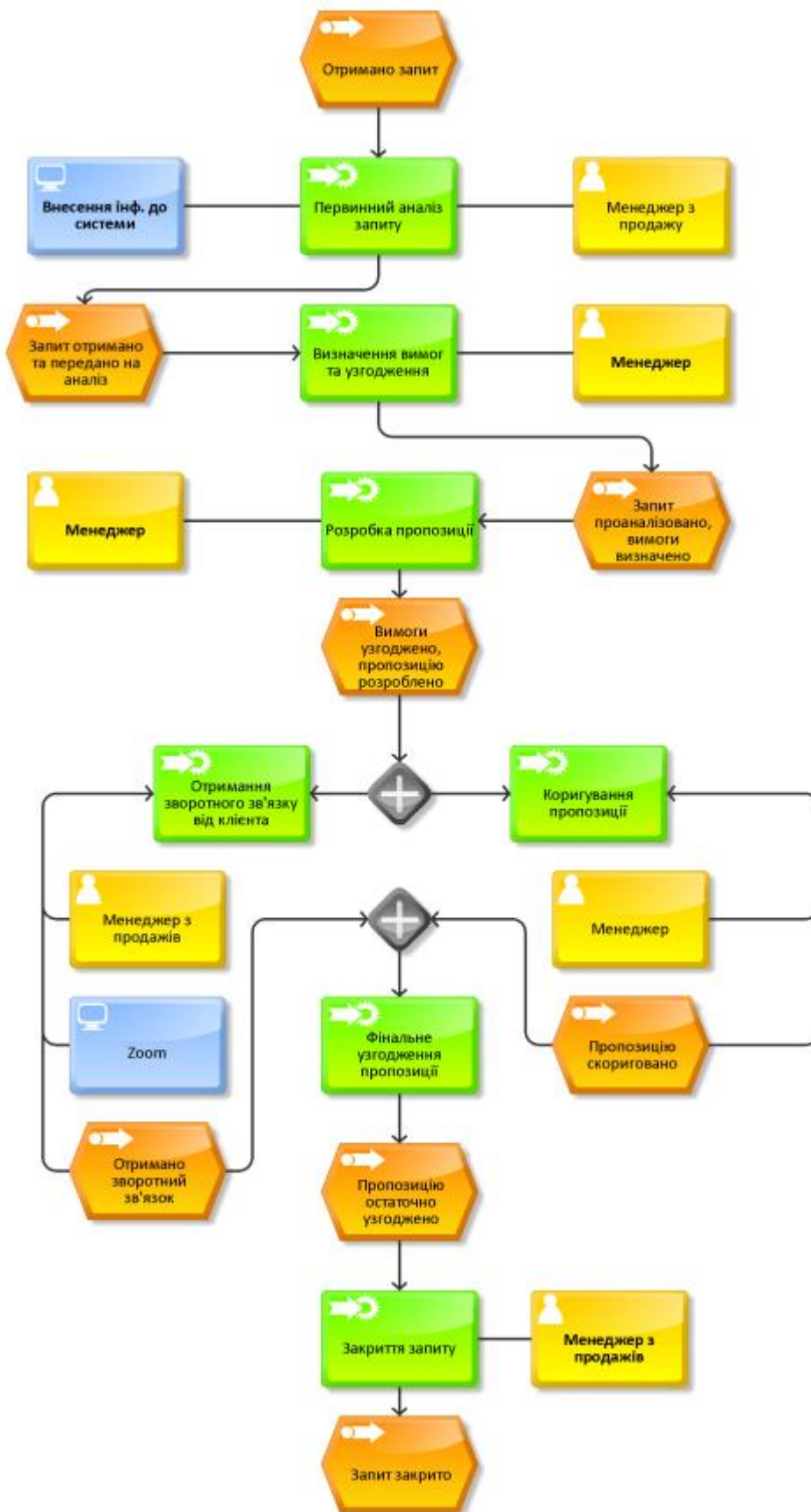
**Функція** — це термін, який описує діяльність або завдання, які виконуються в процесі. Вони переходять з одного стану (події) в інший.

Функції які виконуються для підпроцесу “Обробка запиту клієнта”

- Отримання запиту від клієнта
- Первинний аналіз запиту
- Визначення вимог та узгодження
- Розробка пропозиції
- Відправка пропозиції клієнту
- Отримання зворотного зв'язку від клієнта
- Коригування пропозиції
- Фінальне узгодження пропозиції
- Закриття запиту

Події які виконуються для підпроцесу “Обробка запиту клієнта”

- Отримано запит.
- Запит отримано та передано на аналіз
- Запит проаналізовано, вимоги визначено
- Вимоги узгоджено, пропозицію розроблено
- Пропозицію відправлено клієнту
- Отримано зворотний зв'язок
- Пропозицію скориговано
- Пропозицію остаточно узгоджено
- Запит закрито



2.4. Розробка системи показників та концептуальної схеми управління обраного бізнес-процесу

Система показників — це набір окремих показників, які логічно пов'язані між собою та відображають умови і результати роботи підприємства.

Безперервне поліпшення показників процесу є основною метою управління процесом. Це вимагає використання системи показників, яка включає три основні потоки даних:

а) інформація про якість продукції або послуги, ступінь її відповідності встановленим і прогнозованим вимогам клієнтів, стабільність і відтворюваність параметрів продукту.

б) інформація про якість, ефективність і ресурсоемність процесу, стабільність і відтворюваність параметрів процесу.

в) інформація про рівень задоволеності клієнта процесом, а також про можливість і виконання передбачуваних потреб клієнта

Для визначення показників пропонується використовувати таблицю, що містить вартісні показники, показники часу та технічні показники.

**Показники процесу, продукту і задоволеності клієнтів**

	<b>Вартісні показники (грн.)</b>	<b>Показники часу (t)</b>	<b>Технічні показники (Т)</b>
<b>Показники процесу</b>	сумарні витрати на обсяг виробництва	тривалість циклу обробки заявки клієнта	число співробітників; % невідповідної продукції
<b>Показники продукту</b>	ціна продукту	термін придатності	технічні параметри продукту
<b>Показники задоволеності клієнтів</b>	зростання обсягу продажів по одному клієнту	тривалість використання продукту	число скарг

Основні показники бізнес-процесу «Автоматизація управління ресурсозабезпеченням»:

<b>Обробка</b>	<b>запиту</b>	<b>клієнта:</b>
-	Точність обробки	запиту
-	Тривалість обробки	запиту

- Кількість оброблених запитів на одного співробітника

### **Планування**

**ресурсів:**

- Час на розробку плану розподілу ресурсів
- Точність прогнозування ресурсів
- Відсоток відхилення від запланованого бюджету

### **Використання**

**та**

**моніторинг**

**ресурсів:**

- Відсоток використання ресурсів від планованого
- Кількість попереджень про перевитрати
- Ефективність використання ресурсів

### **Звітність та аналіз використання ресурсів:**

- Кількість підготовлених звітів
- Час на підготовку та аналіз звітів
- Відсоток реалізованих рекомендацій
- Рівень оптимізації використання ресурсів

**Власник бізнес-процесу** – посадова особа, затверджена керівництвом компанії, яке має у своєму розпорядженні ресурси та інформацію, необхідні для виконання бізнес-процесу. Власник має обов'язки по управлінню ходом бізнес-процесу і несе відповідальність за його результати та ефективність. Власником підпроцесів «Планування ресурсів», «Використання та моніторинг ресурсів», «Звітність та аналіз використання ресурсів» є Менеджер проєкту, а підпроцесу «Обробка запиту клієнта» є Менеджер продажів.

**Вищестоящий керівник** – посадова особа, яка планує і несе відповідальність за результати діяльності мережі процесів або її частини, веде аналіз інформації про результати і приймає управлінські рішення для забезпечення максимальної ефективності діяльності.

Власником бізнес-процесу «Автоматизація управління ресурсозабезпеченням», який складається з підпроцесів «Планування ресурсів», «Використання та моніторинг ресурсів», «Звітність та аналіз використання ресурсів» та «Обробка запиту клієнта», є *Директор*.

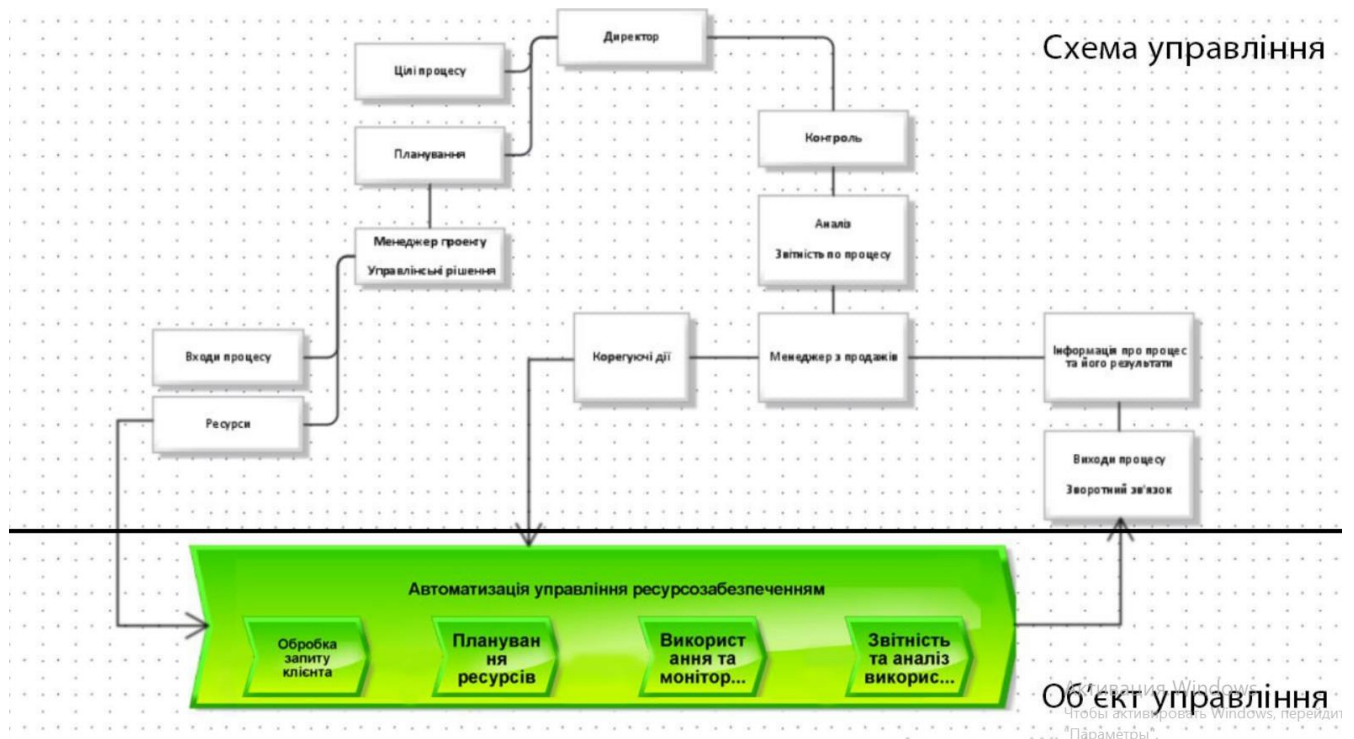


Рис. 2.2 – Концептуальна схема управління бізнес-процесом «**Автоматизація управління ресурсозабезпеченням**»

## 2.5. Імітаційне моделювання із застосуванням нотації BPMN для оптимізації бізнес-процесу підприємства

Створення бізнес-процесів є необхідним для створення будь-яких бізнес-рішень. Будівництво бізнес-процесу не завжди просте завдання, оскільки необхідно не тільки створити схему, але й подумати про те, як це буде працювати на практиці,

керувати процесом, передавати інформацію кожному співробітнику, перевіряти, як все працює на практиці, розглядати проблеми та шляхи їх вирішення.

Імітаційне моделювання — це метод дослідження та аналізу складних систем, процесів або явищ шляхом створення моделей і експериментування з ними. Імітаційне моделювання полягає в тому, щоб використовувати комп'ютерні моделі для відтворення поведінки реальної системи, що дозволяє оцінювати різні сценарії розвитку подій, тестувати гіпотези та приймати обґрунтовані рішення.

В нотації BPMN розглядається бізнес-процес, який називається «Управління технічними ресурсами». Цей процес поділяється на два пули:

### **Відділ управління**

#### **Відділ**

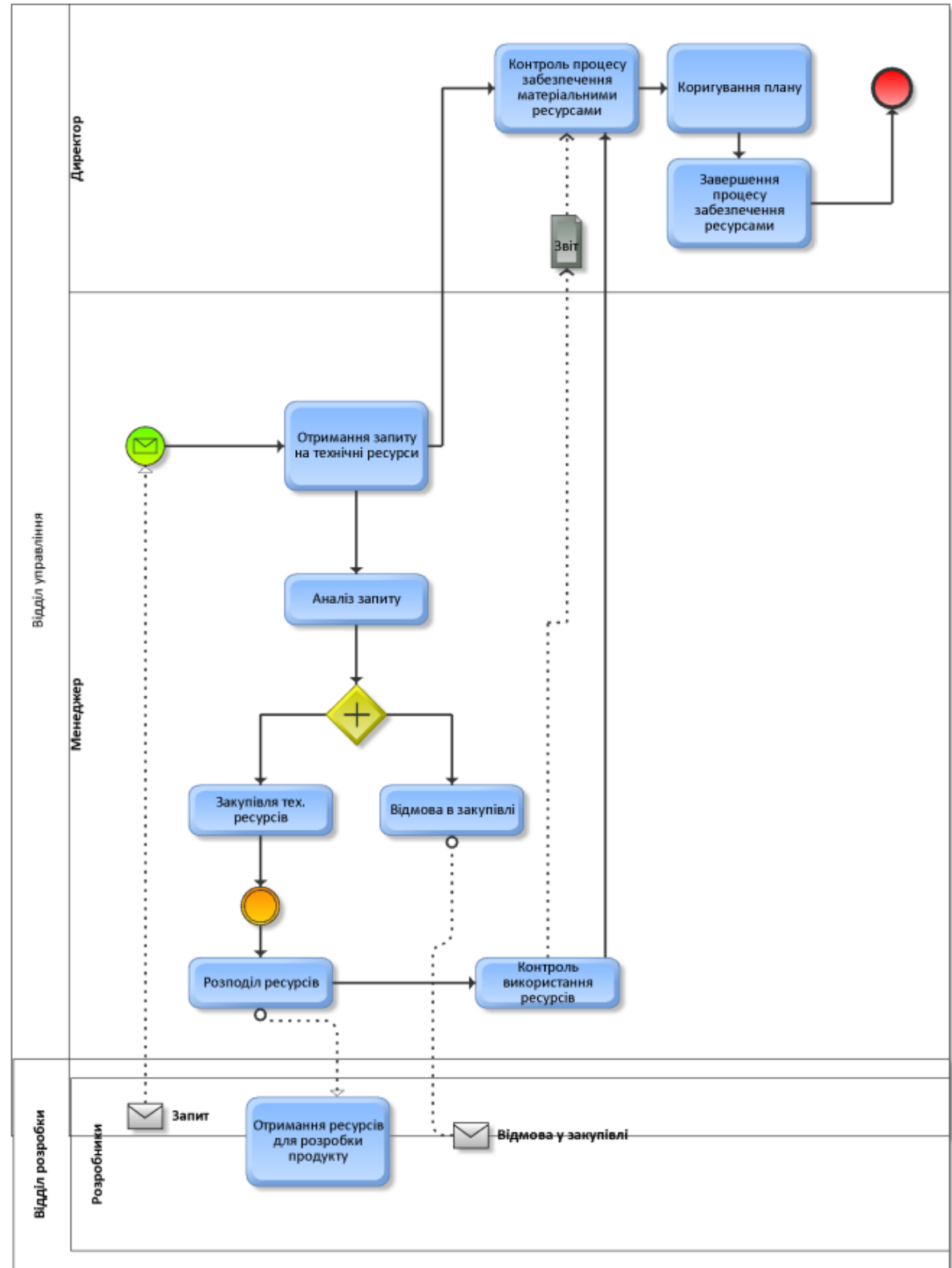
#### **розробки**

Участь в даному процесі беруть 3 учасника: директор, менеджер проекту і розробник.

Початковим етапом процесу є отримання запиту на технічні ресурси від відділу розробки. Після отримання запиту менеджер проводить аналіз наявності матеріальних ресурсів.

Після ретельного аналізу, менеджер закупає необхідні матеріали, після чого вони передаються розробникам для створення продукту.

Директор контролює процес забезпечення матеріальними ресурсами і, при необхідності, вносить корективи до плану. Кінцевою подією є "Завершення процесу забезпечення технічними ресурсами".



Модель бізнес-процесу «Управління технічними ресурсами» в нотатії BPMN

### **Розділ 3. Аналіз і затвердження вимог до програмного продукту, як основи АСУП**

#### **3.1. Затвердження меж об'єкта автоматизації та обґрунтування можливих функцій для його системи управління, які треба автоматизувати**

Бізнес процес "Формування та підготовка запиту від клієнта" охоплює кілька ключових відділів компанії, кожен з яких відіграє важливу роль у забезпеченні ефективності та якості виконання бізнес-процесу. Визначення меж об'єкта автоматизації дозволяє чітко окреслити відповідальність та функції кожного з цих відділів, що є необхідним для подальшої автоматизації процесів.

Нижче перераховані відділи, які охоплені бізнес-процесом **"Формування та підготовка запиту від клієнта"**

- Відділ продажів
- Відділ управління проектами
- Відділ розробки

**Відділ продажів:**

Відділ продажів є першим контактним пунктом для клієнтів. Його основні завдання включають залучення нових клієнтів, обробку вхідних запитів, консультування клієнтів щодо послуг компанії та укладання контрактів. Керівництвом цього відділу займається менеджер з продажів, який відповідає за координацію роботи співробітників та забезпечення високої якості обслуговування клієнтів

Відділ продажів також відповідає за первинну взаємодію з клієнтом, отримання та попередню оцінку запиту, а також за забезпечення своєчасної передачі інформації до відділу управління проектами.

**Відділ управління:**

Цей відділ займається плануванням, організацією та контролем за виконанням проектів. Основні завдання включають аналіз запиту клієнта, розробку проектного плану, координацію роботи всіх учасників проекту та моніторинг прогресу. Керівництвом цього відділу займається менеджер проекту, який забезпечує ефективне управління ресурсами та координацію дій між різними відділами.

Відділ управління проектами несе відповідальність за точний аналіз запиту клієнта, визначення ресурсів, необхідних для виконання проекту, та забезпечення їх своєчасного надання.

### **Відділ**

### **розробки:**

Відділ розробки відповідає за безпосереднє виконання технічних завдань, пов'язаних з розробкою програмного забезпечення. Це включає програмування, тестування, інтеграцію та впровадження рішень. Керівництвом цього відділу займається старший розробник або технічний лідер, який контролює роботу команди розробників та забезпечує відповідність технічних рішень вимогам проекту.

Відділ розробки несе відповідальність за якість та своєчасне виконання технічних завдань, що входять до проекту. Вони також взаємодіють з відділом управління проектами для забезпечення відповідності кінцевого продукту вимогам клієнта.

Крім працівників компанії, до учасників бізнес-процесу належить клієнт, який починає цей процес, роблячи запит. Для подальшого протікання бізнес-процесу між цими учасниками повинні бути встановлені «зв'язки» в програмному забезпеченні, які дозволять їм взаємодіяти та обмінюватися необхідною інформацією.

Для забезпечення відповідного використання та захисту інформації треба створити функцію авторизації окремо для кожного користувача у програмному забезпеченні.

Всі дані, які знаходяться в базі даних і мають бути відображені у програмному забезпеченні, мають бути представлені у зручному та легко сприйнятному вигляді, наприклад у таблицях або випадаючому списку.

### 3.2. Опис існуючих аналогів автоматизованої системи управління для підприємства, яке досліджується

Для вибору оптимальної автоматизованої системи управління, необхідно розглянути існуючі аналоги, що вже використовуються в багатьох компаніях.

Аналоги автоматизованих систем управління для підприємства:

1. **Microsoft Project** — це інструмент для управління проектами, який дозволяє планувати етапи робіт, забезпечувати ресурси, швидко виявляти проблеми, відстежувати хід виконання завдань і складати звіти.

Програма є надійним помічником для бізнесу завдяки великому функціоналу. Її дизайн поєднує практичність і гнучкість, необхідні для професійного виконання завдань. Microsoft Project можна використовувати для багатьох типів проектів, від невеликих до великих проектів із великою кількістю учасників і складними зв'язками.

Крім того, проект Microsoft дозволяє людям працювати разом і ділитися даними, а також інтегруватися з додатками та сервісами Microsoft, такими як Microsoft Teams і SharePoint.

**Основні можливості проекту Microsoft включають:**

- Розподіл основного завдання на складові та встановлення пріоритету, потреби в ресурсах і терміну виконання для кожної складової.
- Синхронізація етапів проекту з календарним планом і графіком, що враховує робочі та неробочі (святкові, вихідні) дні.
- Функція розрахунку критичного шляху, яка передбачає повне обчислення часу, необхідного для завершення проекту, щоб проект був успішним.

- Детальне планування забезпеченості ресурсами, необхідними для проекту, а також їх призначення за завданнями та підзавданнями, щоб було легко контролювати поточні обставини.
- Використання сітки, дошки завдань і шкали часу (графік Ганта) для наочної та зрозумілої візуалізації всіх взаємозалежностей.

### **Недоліки програми:**

- Microsoft Project є дорогим продуктом, що може бути непосильним для малих і середніх підприємств. Вартість ліцензій може значно збільшити загальні витрати на впровадження системи.
- Інтерфейс та функціонал Microsoft Project можуть бути складними для нових користувачів, що вимагає значного часу та зусиль для навчання.
- Система може бути недостатньо гнучкою для адаптації під унікальні вимоги певної компанії або проекту, що може обмежувати її ефективність.
- Для ефективною роботи Microsoft Project необхідні сучасні комп'ютери з високими технічними характеристиками, що може вимагати додаткових витрат на оновлення обладнання.

2. **Notion** — це спеціальна система для створення нотаток і текстових документів, списків справ, баз даних, таблиць, канбан-дошок, баз знань, для ведення проектів та спільної роботи. Notion поєднує в собі всі необхідні функції, дозволяючи не перескакувати між вкладками з багатьма відкритими сервісами, а мати все необхідне в одному місці.

### **Переваги сервісу:**

- Багатофункціональність. Можливість зібрати в одному місці різноманітні види контенту, створювати текстовий контент і мудборди, таблиці, а також працювати разом з командою на канбан-дошках.
- Підходить для кросфункціональної роботи. Можна бачити, хто зайшов на сторінку, де саме знаходиться користувач і які зміни вносить.
- Полегшений імпорт. Завдання з Trello, Jira або Asana, нотатки з Evernote чи статті з Word, навіть цілі бази знань з Confluence. Вся інформація в кілька кліків опиняється у просторі Notion.

### **Недоліки сервісу:**

- Надмір функцій. Багато користувачів заходять у сервіс, губляться у кількості доступних опцій і швидко його залишають. Щоб не загубитися в Notion, на початку можна скористатися готовими шаблонами. Крім того, нижче ми детально розповіли і показали, як можна налаштувати свою роботу в сервісі.
- Відсутність україномовного інтерфейсу. Створювати свої листи в цьому багатофункціональному блокноті можна українською мовою, але щоб розібратися в інтерфейсі, знадобиться хоча б елементарна англійська.
- За командну роботу треба платити. Спільна робота неможлива без оплати. Якщо з текстовими документами можна працювати і в безкоштовній версії, то за коментування і управління на канбан-дошках доведеться заплатити.

**3. Smartsheet** — це сучасна платформа для управління проектами, програмами та бізнес-процесами, спеціально розроблена для корпоративного сектору. Цей потужний інструмент пропонує широкий функціонал, необхідний для ефективного управління робочими завданнями та проектами. Сьогодні ми детально розглянемо, як Smartsheet може покращити вашу роботу і підвищити продуктивність вашої команди.

Smartsheet надає рішення для різноманітних завдань, від управління проектами до аналізу даних, і є надійною основою для ефективного управління завданнями, програмами та процесами, які можна масштабувати відповідно до ваших потреб.

**За допомогою цієї платформи можна:**

- Координувати роботу, співробітників та технологічні процеси підприємства. Ця централізована хмарна платформа дозволяє об'єднати ваших співробітників, бізнес-процеси та технології для більш ефективного виконання завдань підприємства.
- Налаштовувати робочі процеси відповідно до потреб компанії. Платформа дозволяє адаптувати робочі процеси, що робить її гнучкою для різних видів бізнесу.

- Об'єднувати системи звітності та додатки на одній платформі. Це підвищує продуктивність і зручність використання.

**Недоліки цієї платформи:**

- Складність для новачків. Для нових користувачів, особливо без досвіду використання подібних інструментів, Smartsheet може здатися складним на перших етапах.
- Необхідність навчання. Через різноманітність функцій платформи може знадобитися певний час і навчання, щоб використовувати її на повну потужність.
- Вартість. Деякі з більш просунутих тарифних планів Smartsheet можуть бути дорогими для малих компаній або початківців.
- Не підходить для всіх видів бізнесу. У деяких галузях або типах бізнесу використання Smartsheet може бути менш придатним у порівнянні з іншими інструментами.

Аналіз наявних аналогів автоматизованих систем управління для компаній свідчить про те, що сучасні програмні забезпечення мають багато спільних особливостей та недоліків.

Ці системи пропонують широкі можливості для планування, організації роботи та інтеграції з іншими послугами. Однак, вони часто є складними для нових користувачів і вимагають значного часу на навчання. Крім того, вартість деяких систем може бути високою, що робить їх менш доступними для малих підприємств.

Впровадження власного програмного забезпечення, спеціально розробленого для потреб компанії дозволить подолати ці недоліки. Це програмне забезпечення буде більш гнучким, адаптованим до специфічних вимог і зручним для співробітників, що беруть участь у процесах. Такий підхід забезпечить оптимізацію робочих процесів, підвищить ефективність управління проектами та ресурсами, і створить

систему, яка максимально відповідає потребам в автоматизації управлінських завдань.

### 3.3. Розробка та затвердження технічного завдання на створення програмного забезпечення для АСУП

Розробка програмного забезпечення для автоматизованої системи управління підприємством (АСУП) є складним та багатофазним процесом, який вимагає ретельного планування та чіткого визначення вимог.

Технічне завдання слугує основою для всього процесу розробки, оскільки воно встановлює вимоги до функціональності, продуктивності, безпеки та інтерфейсів системи.

Розробка технічного завдання починається з аналізу поточних бізнес-процесів та вивчення потреб користувачів. На основі цього аналізу визначаються основні цілі системи, її функціональні можливості та обмеження.

Метою технічного завдання є забезпечення чіткості, однозначності та повноти вимог до програмного забезпечення. Це допомагає уникнути непорозумінь, мінімізувати ризики та забезпечити успішну розробку та впровадження системи. В результаті, добре підготовлене технічне завдання стає запорукою ефективного та своєчасного виконання проекту, що в кінцевому підсумку сприяє ефективності діяльності підприємства.

**Повна назва системи:** Автоматизація управління ресурсозабезпеченням при розробці ІТ-продукту.

**Назва розробника та замовника АС:**

Розробник:

Студент Одеського Національного Технологічного Університету

Денної форми навчання

Факультету Автоматизації та робототехніки

Групи Ат-20

Божко Назар Євгенійович

Замовник: ТОВ “Real Solutions”

### **Підстави для створення АС**

Учбовий план кафедри АТПіРС

### **Терміни початку і закінчення створення АС**

Плановий термін початку створення: 15.03.2023

Плановий термін закінчення створення: 30.05.2024

### **Відомості про джерела та порядок фінансування створення АС**

Фінансування створення АС не проводиться.

### **Порядок оформлення і подання замовнику результатів робіт зі створення системи**

Результатом створення АС є дипломна робота, оформлена відповідно до «Методичні вказівки до ДР бакалавра АУП».

### **Призначення та цілі створення АС**

Автоматизована система управління ресурсозабезпеченням в компанії Real Solutions призначена для:

- Забезпечення ефективного розподілу та використання матеріальних і нематеріальних ресурсів компанії.
- Підвищення точності та швидкості обробки запитів, зменшуючи ймовірність помилок.
- Покращення комунікації між відділами, забезпечуючи зручну та прозору платформу для обміну інформацією.

- Підвищення рівня обслуговування клієнтів, швидко та точно обробляючи нові запити.
- Покращення контролю за виконанням завдань, дозволяючи керівникам оперативно реагувати на відхилення від плану.

## **Мета створення АС**

- Приріст нових клієнтів на 10% за рахунок підвищення ефективності обслуговування та задоволеності клієнтів.
- Зменшення середнього часу обробки запитів на 20%, що дозволить швидше реагувати на потреби проектів.
- Зниження прямих витрат на управління ресурсами на 8%, включаючи витрати на покупку ліцензій на програмне забезпечення.
- Скорочення часу розробки продукту на 5%;

Автоматизована система управління ресурсозабезпеченням в ІТ компанії спрямована на поліпшення управління ресурсами, збільшення точності та швидкості обробки запитів, а також покращення взаємодії та звітності між відділами. Після запровадження системи очікується значне скорочення витрат, збільшення клієнтської бази та підвищення загальної продуктивності компанії.

## **Відомості про об'єкт автоматизації**

Об'єктом автоматизації являється бізнес-процес «управління ресурсозабезпеченням при розробці ІТ-продукту»

## **Вимоги до системи**

### **Вимоги до структури та функціонуванню системи**

Вимоги, які необхідно враховувати при виділенні процесів, у великій мірі залежать від самої організації, її розміру, способу управління.

Автоматизована система управління ресурсозабезпеченням повинна забезпечувати високу продуктивність, масштабованість та надійність. Вона має бути багатофункціональною, інтегрованою та гнучкою, з можливістю розширення відповідно до потреб підприємства. Система повинна забезпечувати безперебійну роботу з мінімальними витратами на обслуговування та підтримку. Основною вимогою до структури є наявність чітко визначених функціональних підсистем, які включають управління проектами, управління ресурсами, фінансове управління та аналітику.

Система управління проектами повинна забезпечувати можливість планування та відстеження виконання проектів, управління завданнями та ресурсами, контроль термінів та етапів виконання, а також формування звітів про статус проектів. Це дозволить підвищити ефективність управління проектами та своєчасно виявляти і вирішувати проблеми, що виникають.

Також система повинна мати інтуїтивно зрозумілий інтерфейс користувача, що забезпечує легкість у використанні та мінімальні витрати на навчання персоналу.

Підтримка режиму з декількома користувачами необхідна для спільної роботи над проектами та забезпечення доступу до системи для всіх учасників. Система повинна також забезпечувати надійний захист даних та доступу до системи, включаючи механізми аутентифікації та авторизації користувачів.

Ці вимоги до структури та функціонування системи забезпечать ефективну та надійну роботу автоматизованої системи управління ресурсозабезпеченням, сприятимуть підвищенню продуктивності компанії та оптимізації використання ресурсів.

### **Вимоги до чисельності та кваліфікації персоналу системи і режиму його роботи**

До складу персоналу, необхідного для експлуатації системи, необхідне виділення наступних відповідальних співробітників:

- Адміністратор системи
- Менеджер
- Розробник програмного забезпечення

**Далі наведені основні обов'язки співробітників необхідних для експлуатації системи:**

- Адміністратор системи забезпечує безперебійну роботу системи, налаштовує параметри безпеки, проводить регулярне резервне копіювання даних та моніторинг продуктивності.
- Менеджер проекту планує та координує діяльність команди, розподіляє завдання, контролює виконання та взаємодіє з іншими відділами.
- Розробник програмного забезпечення займається розробкою, тестуванням та впровадженням нових функцій, а також усуненням технічних проблем

**Для співробітників можна визначити наступні кваліфікаційні вимоги:**

**Адміністратор** повинен мати вищу освіту в галузі інформаційних технологій або суміжній сфері. Мати досвід роботи з адміністрування інформаційних систем. Також є важливим знання баз даних, та вміння працювати з відповідними програмними засобами та інструментами.

**Менеджер** впевнено користуватися комп'ютером, мати досвід роботи з управління проектами та командами, включаючи навички планування, координації та моніторингу виконання завдань. Важливим є знання методологій управління проектами, таких як Agile або Scrum, а також вміння користуватися відповідними програмними засобами, такими як Jira. Також необхідні впевнені навички користування комп'ютером.

**Розробник** повинен мати досвід роботи у сфері розробки програмного забезпечення, включаючи навички написання коду, тестування та відлагодження. Важливим є знання різних мов програмування, а також вміння працювати з інструментами розробки, такими як Git.

**Показники**

**призначення**

До показників призначення автоматизованої системи управління ресурсозабезпеченням належать наступні значення параметрів, які характеризують ступінь відповідності системи її призначенню:

Точність планування ресурсів - відсоток точності прогнозування потреб у ресурсах відповідно до фактичного їх використання.

Час на виконання проєкту - середній час, необхідний для завершення проєктів, порівняно з запланованим часом.

Рівень автоматизації - відсоток автоматизованих рутинних операцій та процесів у системі.

Задоволеність користувачів - відсоток користувачів, які задоволені роботою системи, згідно з проведеними опитуваннями або зворотним зв'язком.

### **Вимоги до надійності**

Автоматизована система управління ресурсами повинна бути надзвичайно надійною. При нормальній роботі технічних засобів система повинна нормально функціонувати в режимі реального часу.

### **Вимоги до безпеки**

Вимоги до безпеки системи повинні відповідати стандартам захисту персональних комп'ютерів та серверів, на яких ця система буде встановлена. Необхідно реалізувати надійні механізми автентифікації та авторизації, а також регулярно проводити перевірки системи на вразливості та оперативно їх усувати.

### **Вимоги до експлуатації, технічного обслуговування, ремонту та зберігання компонентів системи**

База даних системи повинна розміщуватися на сервері, що забезпечує стандартні можливості доступу та редагування даних. Технічне обслуговування системи має включати регулярне оновлення програмного забезпечення, моніторинг працездатності та швидке усунення збоїв.

### **Вимоги щодо захисту даних від несанкціонованого доступу**

- Система повинна забезпечувати:
- Автентифікацію користувачів для перевірки їхніх повноважень та доступу до даних.
  - Шифрування даних під час передачі та зберігання для захисту від перехоплення та несанкціонованого доступу.
  - Журналювання дій користувачів для відстеження та контролю доступу до конфіденційної інформації.

### **Вимоги до схоронності даних при аваріях**

У разі виходу з ладу технічного обладнання, дані залишаються неушкодженими та збереженими завдяки розміщенню бази даних на надійному сервері, який забезпечує захист від аварійних ситуацій. Сервер повинен мати систему резервного копіювання та відновлення, щоб запобігти втраті даних під час технічних збоїв.

### **Вимоги до функцій системи**

В системі повинні бути реалізовані наступні функції:

- Обробка та управління запитам
- Додавання нових користувачів в систему
- Автентифікація та управління доступом користувачів
- Моніторинг та контроль виконання завдань в режимі реального часу.
- Генерація звітів та аналітика проектів в реальному часі
- Виставлення пріоритетності на задачах
- Можливість спілкування в системі
- Візуалізація даних та показників ефективності.
- Відео-спостереження
- Відображення даних із БД в зручному вигляді для сприйняття (випадаючий список або таблиця).

Всі функції повинні працювати без збоїв, виконуватися за зазначений час.

### **Вимоги до інформаційного забезпечення**

Дані, які зберігаються в системі, повинні зберігатися в одній базі даних, яка структурована у вигляді таблиць. Обмін інформацією між різними частинами та компонентами системи здійснюється через одну базу даних. Нові дані повинні вводитися в базу даних за допомогою електронних форм. Вихідна інформація

повинна надаватися у вигляді звітів або таблиць, щоб користувачі системи могли легко її зрозуміти.

### **Вимоги до лінгвістичного забезпечення**

При реалізації системи повинні використовуватися такі мови програмування, як :HTML, PHP, SQL, Javascript,CSS.

### **Вимоги до програмного забезпечення**

Програмне забезпечення повинно забезпечувати стабільну та безперебійну роботу системи, підтримуючи необхідні функції для користувачів. Воно також повинно бути сумісним з операційними системами, які використовуються в організації

Система повинна функціонувати в операційній системі Windows XP, або macOS 10.14 та новіше.

### **Вимоги до технічного забезпечення**

Система повинна працювати на надійній та продуктивній техніці з встановленими на них Windows XP/macOS 10.14 або новіше, яка забезпечує безперебійну роботу та високу швидкість обробки даних.

Серверне обладнання повинно мати достатню потужність для обробки великої кількості запитів і підтримки одночасної роботи багатьох користувачів.

### **Вимоги до метрологічного забезпечення**

Вимоги в системі до метрологічного забезпечення не пред'являються.

### **Вимоги до організаційного забезпечення**

Користувачами системи будуть менеджери, адміністратори, розробники та клієнти. Для забезпечення ефективної роботи та безпеки системи необхідно:

- Впровадити функції аутентифікації та авторизації, щоб гарантувати, що доступ до системи мають лише уповноважені користувачі.
- Встановити чіткі права доступу та ролі для кожного типу користувачів, щоб уникнути несанкціонованих змін або видалення даних.
- Розробити докладні інструкції з використання системи для всіх категорій користувачів, включаючи описи основних функцій та процедур безпеки

### **Склад і зміст робіт по створенню системи**

- Створення дизайну та інтерфейсів користувачів (12.02.23 – 11.03.23);
- Розробка програмної частини системи (12.03.23 – 24.04.2023);
- Тестування правильності праці системи, допрацювання недоліків та створення інструкції користувача (25.04.2023 – 15.05.2023).

### **Порядок прийому та контролю системи**

Контроль виконання робіт по створенню системи виконується дипломним керівником.

Прийом системи виконується в ході захисту дипломної роботи перед комісією.

### **Вимоги до складу і вмісту робіт по підготовці об'єкта автоматизації до введення системи**

Для успішного впровадження автоматизованої системи управління ресурсозабезпеченням необхідно виконати наступні кроки: Якщо є необхідність, компанія може організувати курси підвищення кваліфікації для працівників, щоб забезпечити належний рівень знань та навичок для роботи з новою системою, а саме:

- Забезпечити контроль за процесом, що автоматизується, щоб переконатися, що всі етапи відповідають вимогам для ефективного функціонування системи.
- Організувати налаштування робочих місць, включаючи встановлення необхідного програмного забезпечення та апаратного забезпечення, якщо це необхідно для роботи з новою системою.

## РОЗДІЛ 4. ПРОЕКТУВАННЯ БАЗИ ДАНИХ (БД)

### 4.1. Розробка і створення концептуальної моделі даних

На початковому етапі розробки моделі створюється інформаційна модель, яка може мати різні знакові форми. На завершальному етапі ці форми перетворюються на комп'ютерну модель. Таким чином, інформаційна модель зазвичай представлена в конкретній знаковій формі. Наприклад, таблиця є одним з прикладів таких знакових моделей.

Таблиця "users" призначена для зберігання даних про користувачів і може використовуватись у системах управління базами даних, які підтримують SQL.

1. id - це унікальний ідентифікатор користувача. Він автоматично збільшується при додаванні нового запису.
2. firstname - це текстове поле, що містить ім'я користувача.
3. lastname - це текстове поле, що містить прізвище користувача.
4. email - це текстове поле, що містить адресу електронної пошти користувача.
5. role – текстове поле в якому зберігається роль користувача
6. tel - це текстове поле, що містить номер телефону користувача.
7. password - це текстове поле, в якому зберігається зашифрований пароль користувача.
8. remember\_token - це текстове поле, яке використовується для запам'ятовування токена авторизації користувача.
9. created\_at - поле, що відображає час створення запису про користувача.

Таблиця "projects" містить наступні стовпці:

1. id: це унікальний ідентифікатор проекту.
2. title: це текстове поле, що містить назву проекту.
3. consultant\_id: це числове поле, що зберігає ід менеджера проекту.
4. client\_id: це числове поле, що зберігає ід клієнта проекту.
5. back: це числове поле, що зберігає ід бек-енд розробника проекту.

6. `front`: це числове поле, що зберігає ід фронт-енд розробника проекту.
7. `tester`: це числове поле, що зберігає ід тестера проекту.
8. `designer`: це числове поле, що зберігає ід дизайнера проекту.
9. `file`: це текстове поле, що містить посилання або шлях до файлу проекту.
10. `description`: це текстове поле, що містить опис проекту.
11. `slug`: це текстове поле, що відображає код або ідентифікатор проекту.
12. `created_at`: це поле дати і часу, що відображає час створення проекту.

Таблиця "messages" призначена для зберігання інформації про повідомлення користувачів. Кожен запис в цій таблиці відповідає окремому повідомленню і містить наступні дані:

1. `id`: це унікальний ідентифікатор повідомлення.
2. `project_id`: це зовнішній ключ, який посилається на ідентифікатор проекту.
3. `user_id`: це зовнішній ключ, який посилається на ідентифікатор користувача, що створив повідомлення.
4. `time`: це поле зберігає час створення повідомлення.
5. `content`: це текстове поле, яке містить зміст повідомлення.

Таблиця "projects\_tasks" призначена для зберігання інформації про різні задачі в рамках проектів. Кожен запис в цій таблиці відповідає окремій задачі і містить наступні дані:

1. `id`: унікальний ідентифікатор задачі, автоматично збільшується.
2. `name`: назва задачі.
3. `project_id`: зовнішній ключ, що посилається на ідентифікатор проекту, може бути NULL, значення за замовчуванням 0.
4. `dev_time`: час, відведений на розробку задачі.
5. `consultant_id`: ідентифікатор консультанта, значення за замовчуванням 0.
6. `status`: поточний статус задачі, значення за замовчуванням "start".
7. `priority`: пріоритет задачі, значення за замовчуванням "normal".
8. `content`: текстовий опис задачі.

9. time: час створення запису про задачу.

10.deleted: позначає, чи була задача видалена. Можливі значення: 'no' (не видалено) або 'yes' (видалено), значення за замовчуванням 'no'.

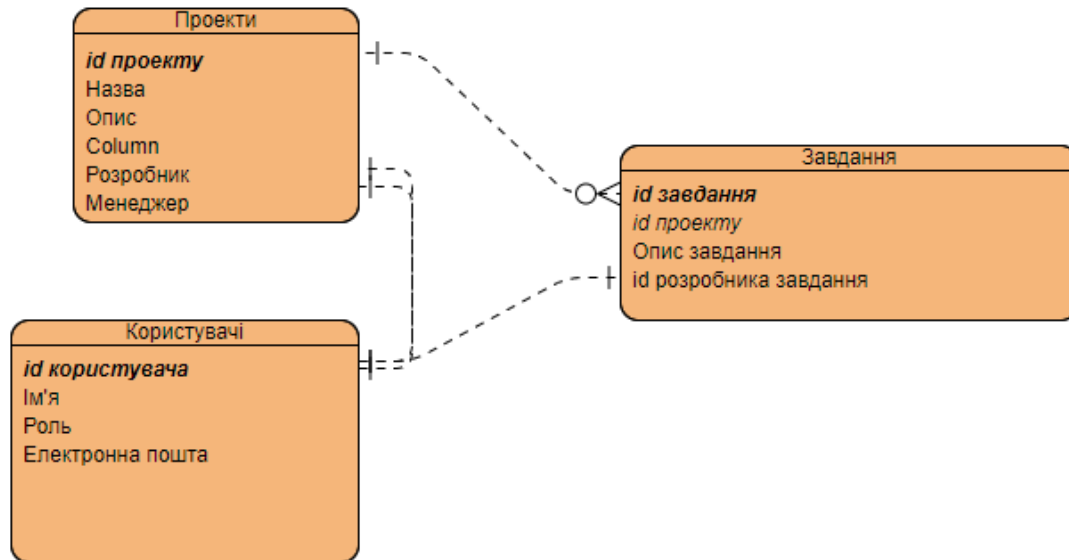


Рис. 4.1 – Концептуальна модель даних бізнес процесу «Управління ресурсозабезпеченням при розробці ІТ-продукту»

#### 4.2. Розробка і створення структури таблиць БД

Кожна таблиця містить поля, які мають свій тип даних. Типи даних бувають:

Таблиця 4.1 – Типи даних

Група	Типи даних	Опис
Цілі числа	INT, BIGINT, SMALLINT, TINYINT	Цілі числа з різною розрядністю
Дійсні числа	DECIMAL, FLOAT, DOUBLE	Дійсні числа з плаваючою комою

Група	Типи даних	Опис
Рядки	VARCHAR, TEXT	Рядкові дані різної довжини
Дата та час	DATE, TIMESTAMP	Дата та/або час
Логічні	BOOLEAN	Логічне значення (TRUE або FALSE)
Бітові	BIT	Одиночний біт (0 або 1)
Байтові	BINARY, VARBINARY	Масив байтів з фіксованою або змінною довжиною
Геометричні	GEOMETRY	Геометричні дані (точки, лінії, полігони та ін.)
JSON	JSON	Дані в форматі JSON
XML	XML	Дані в форматі XML
Статичні	ENUM, SET	Набір значень або перерахування значень

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int		UNSIGNED	No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 email	varchar(200)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 password	varchar(80)	utf8mb4_unicode_ci		Yes				<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 token	varchar(50)	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	5 restore_token	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	6 role	enum('unconfirmed','user','moder','front','bac...	utf8mb4_unicode_ci		Yes	unconfirmed			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	7 firstname	varchar(50)	utf8mb4_unicode_ci		Yes				<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	8 lastname	varchar(50)	utf8mb4_unicode_ci		Yes				<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	9 description	text	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	10 tel	varchar(30)	utf8mb4_unicode_ci		No				<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Рис. 4.2 – Структура таблиці «users»

Таблиця "users" містить наступні поля з відповідними типами даних:

1. id: int (ціле число), UNSIGNED, AUTO\_INCREMENT, PRIMARY KEY
2. email: varchar(200) (рядок з обмеженням у 200 символів), індексоване поле
3. password: varchar(60) (рядок з обмеженням у 60 символів), дозволяє NULL значення
4. token: varchar(50) (рядок з обмеженням у 50 символів), дозволяє NULL значення, за замовчуванням NULL

5. `restore_token`: `varchar(255)` (рядок з обмеженням у 255 символів), дозволяє NULL значення, за замовчуванням NULL
6. `role`: `enum('unconfirmed', 'user', 'moder', 'front', 'back')` (перечислювальний тип), дозволяє NULL значення, за замовчуванням 'unconfirmed'
7. `firstname`: `varchar(50)` (рядок з обмеженням у 50 символів), дозволяє NULL значення
8. `lastname`: `varchar(50)` (рядок з обмеженням у 50 символів), дозволяє NULL значення
9. `description`: `text` (текстове поле), дозволяє NULL значення, за замовчуванням NULL
10. `tel`: `varchar(30)` (рядок з обмеженням у 30 символів)

Ці типи даних використовуються для відповідного збереження інформації про користувачів в базі даних.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	<b>id</b> 🔑	int		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>name</b>	varchar(200)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/> 3	<b>blog_id</b>	int			Yes	0		
<input type="checkbox"/> 4	<b>dev_time</b>	varchar(250)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/> 5	<b>consultant_id</b>	int			Yes	0		
<input type="checkbox"/> 6	<b>status</b>	varchar(250)	utf8mb4_unicode_ci		Yes	start		
<input type="checkbox"/> 7	<b>priority</b>	varchar(250)	utf8mb4_unicode_ci		Yes	normal		
<input type="checkbox"/> 8	<b>deadline</b>	varchar(250)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/> 9	<b>content</b>	text	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/> 10	<b>time</b>	varchar(250)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/> 11	<b>deleted</b>	enum('no', 'yes')	utf8mb4_unicode_ci		Yes	no		

Рис. 4.3 – Структура таблиці «**projects\_tasks**»

Таблиця "projects\_tasks" містить наступні поля з відповідними типами даних:

1. id: int (ціле число), UNSIGNED, AUTO\_INCREMENT, PRIMARY KEY
2. name: varchar(200) (рядок з обмеженням у 200 символів), не дозволяє NULL значення
3. project\_id: int (ціле число), дозволяє NULL значення.
4. dev\_time: varchar(250) (рядок з обмеженням у 250 символів), дозволяє NULL значення, за замовчуванням NULL
5. consultant\_id: int (ціле число), дозволяє NULL значення, за замовчуванням 0
6. status: varchar(250) (рядок з обмеженням у 250 символів), дозволяє NULL значення, за замовчуванням 'start'
7. priority: varchar(250) (рядок з обмеженням у 250 символів), дозволяє NULL значення, за замовчуванням 'normal'
8. deadline: varchar(250) (рядок з обмеженням у 250 символів), дозволяє NULL значення, за замовчуванням NULL
9. content: text (текстове поле), дозволяє NULL значення, за замовчуванням NULL
10. time: varchar(250) (рядок з обмеженням у 250 символів), дозволяє NULL значення, за замовчуванням NULL
11. deleted: enum('no', 'yes') (перечислювальний тип), дозволяє NULL значення, за замовчуванням 'no'


Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 	bigint		UNSIGNED	No	None		AUTO_INCREMENT
name	varchar(255)	utf8mb4_unicode_ci		No	None		
description	varchar(255)	utf8mb4_unicode_ci		No	None		
created_at	timestamp			Yes	NULL		
updated_at	timestamp			Yes	NULL		

Рис. 4.5 – Структура таблиці «documents»

Таблиця "documents" містить наступні поля з відповідними типами даних:

1. id: bigint (ціле число, PRIMARY KEY)
2. name: varchar(255) (рядок з обмеженням у 255 символів)
3. description: varchar(255) (рядок з обмеженням у 255 символів)
4. created\_at: timestamp (дата та час створення запису)
5. updated\_at: timestamp (дата та час оновлення запису)

Ця таблиця використовується для зберігання інформації про документи. Вона містить унікальний ідентифікатор кожного документа, його назву, опис, а також дати створення та оновлення запису.

## РОЗДІЛ 5. СТВОРЕННЯ ВЕБ-ДОДАТКУ, ЯК ОСНОВИ РОЗРОБЛЮВАНОЇ АСУП

### 5.1. Розробка макетів для інтерфейсів користувачів та їх дизайну

Макет інтерфейсу користувача – це візуальне представлення дизайну веб-сторінки або додатку, яке демонструє його структуру, розташування елементів і загальний вигляд. Це прототип або схема, яка ілюструє, як користувач бачитиме інтерфейс.

*Головний екран:*

- Бокове меню з логотипом та навігаційними посиланнями, такими як "Головна", "Вихід", "Проекти", "Створити проект".

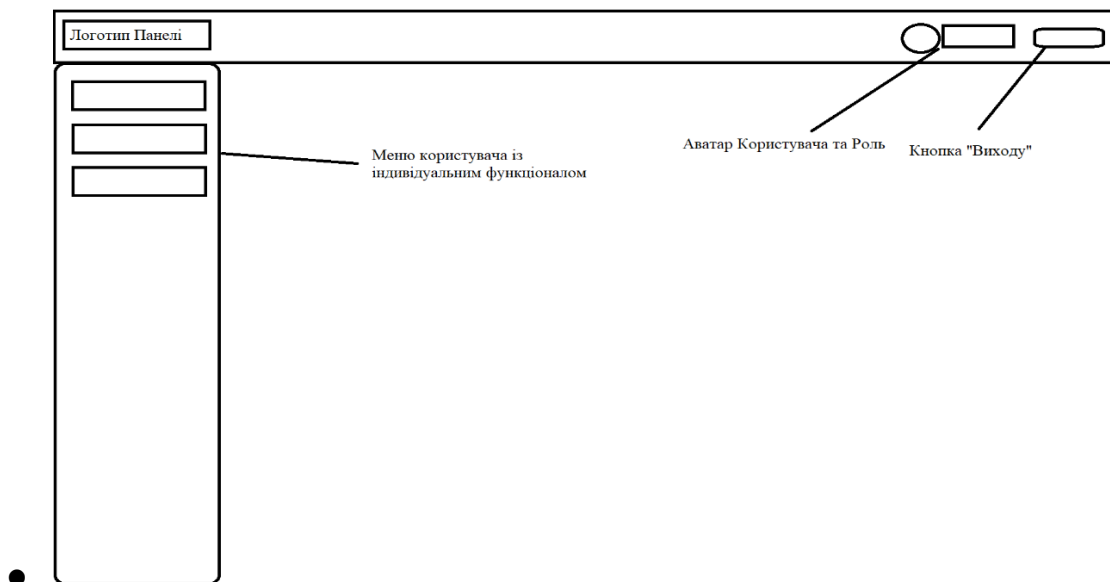


Рис. 5.1 – Макет інтерфейсу користувача.

*Сторінка проекту:*

- Відображення поточного проекту та завдань до нього.
- Відображення деталей про завдання.
- Чат із підтримкою менеджера проекту.

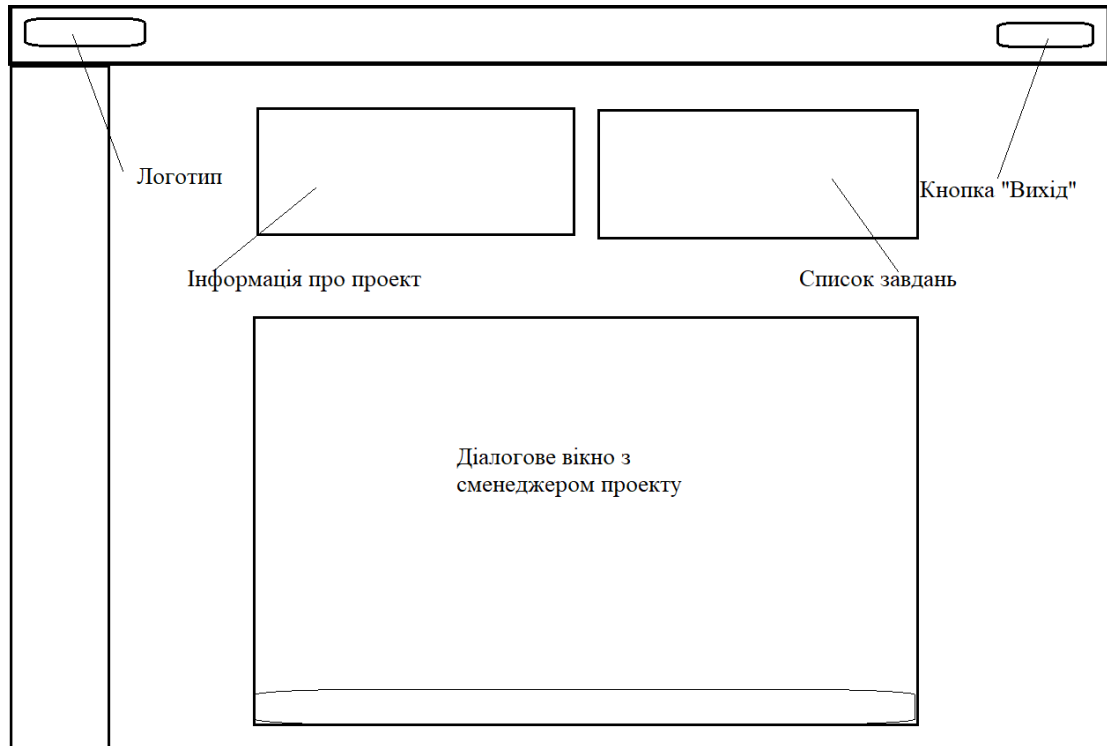


Рис. 5.2 – Макет сторінки «Перегляд проекту»

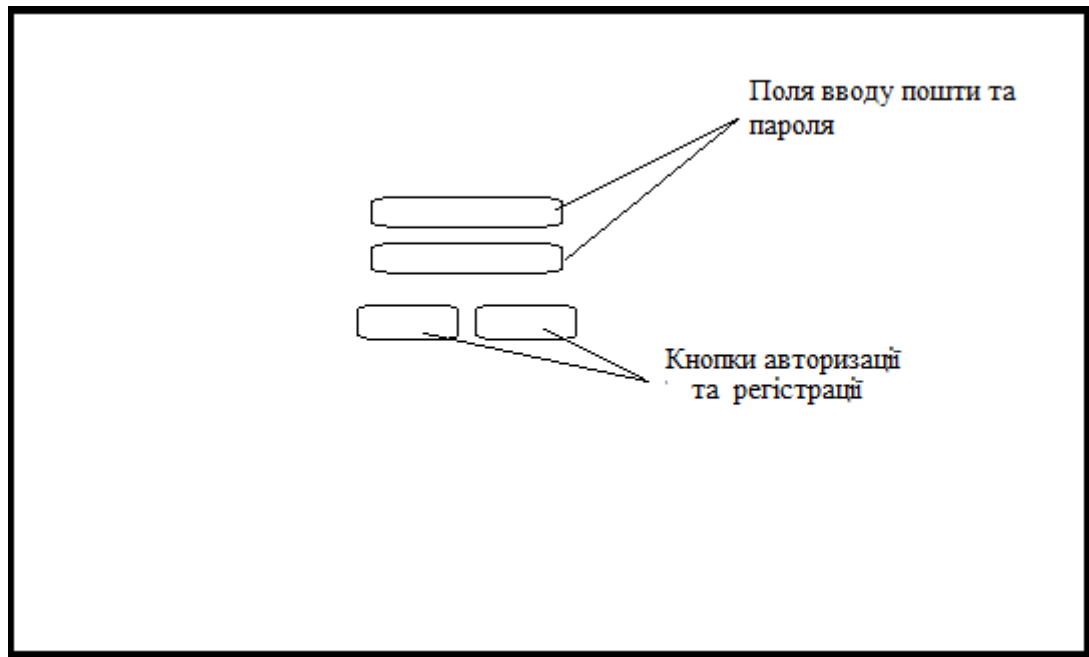


Рис. 5.3 – Макет інтерфейсу авторизації.

## 5.2. Верстка дизайн-макетів

При розробці макетів розробнику слід враховувати численні аспекти, такі як естетика, взаємодія з користувачем, зручність використання і брендування. Це передбачає використання кольорів, типографіки, графічних елементів і компонентів для створення візуально привабливих інтерфейсів.

Верстка веб-сайту здійснюється за допомогою різних інструментів і технологій, що сприяють формуванню структури, макету та зовнішнього вигляду сторінок. Основні інструменти для верстки включають:

HTML (HyperText Markup Language) – основна мова розмітки, яка використовується для створення структури і визначення елементів на веб-сторінці. Вона застосовується для визначення заголовків, параграфів, списків, таблиць, форм та інших елементів.

CSS (Cascading Style Sheets) – мова, що використовується для стилізації та оформлення веб-сторінок. Вона дозволяє налаштовувати розміри, кольори, шрифти, відступи, рамки, фонові зображення та інші візуальні атрибути елементів HTML.

JavaScript – мова програмування, що дозволяє створювати інтерактивність і динаміку на веб-сторінках. Вона використовується для реалізації різних функцій,

таких як анімація, валідація форм, взаємодія з користувачем, завантаження даних з сервера тощо.

Фреймворки, такі як Bootstrap, Foundation, або бібліотеки, такі як jQuery, спрощують і прискорюють процес верстки. Вони надають готові компоненти, шаблони, стилі та функціонал для створення веб-інтерфейсу.

Ці інструменти і технології працюють разом, дозволяючи створювати зручні, естетичні та інтерактивні веб-сайти, що відповідають потребам користувачів.

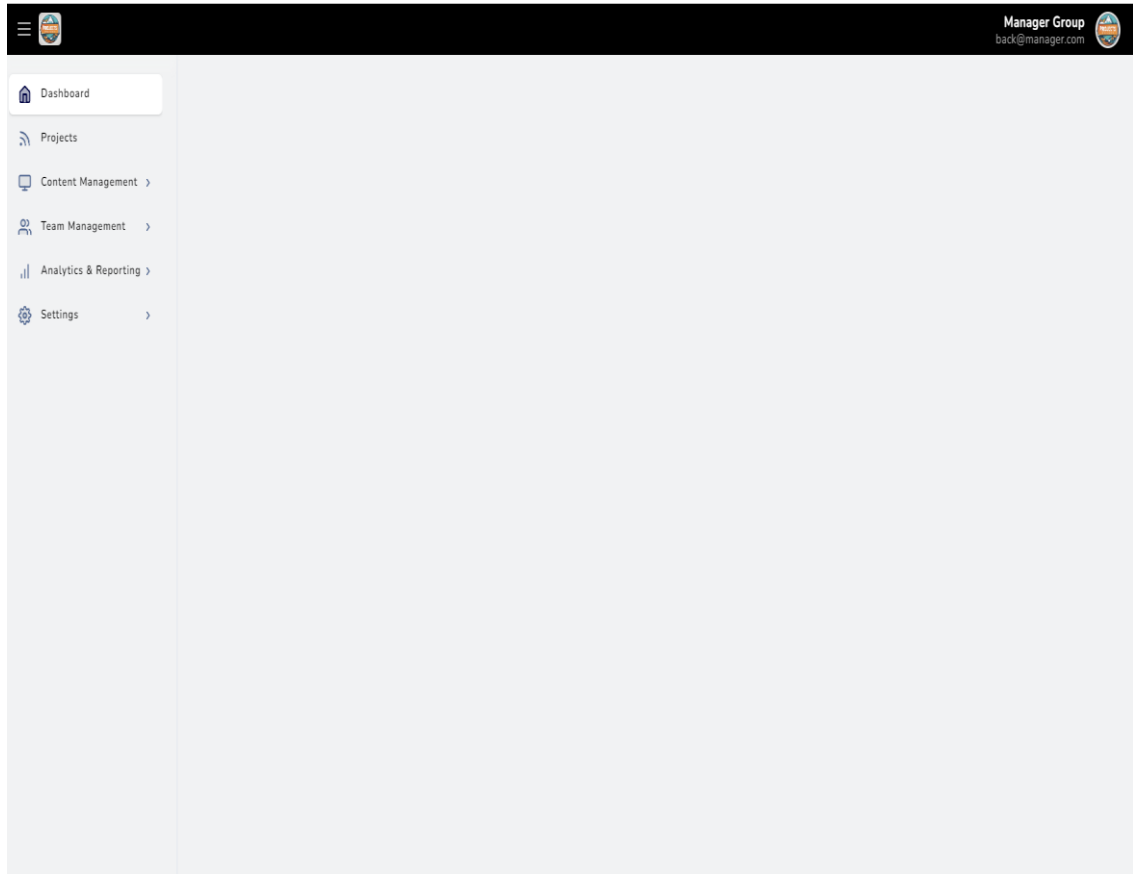


Рис. 5.4 – Головна сторінка.

```

public function indexAction()
{

    Model::import(model: 'panel/blog');
    $this->view->new_projects = BlogModel::getAll( where: " AND (
('consultant_id' IS NULL OR 'consultant_id' = 0)
OR ('front' IS NULL OR 'front' = 0)
OR ('back' IS NULL OR 'back' = 0)
OR ('tester' IS NULL OR 'tester' = 0)
OR ('designer' IS NULL OR 'designer' = 0))*");

    Model::import(model: 'panel/blog/categories');
    $this->view->sectors = CategoriesModel::getAll();

    Model::import(model: 'panel/team');
    $this->view->users = TeamModel::getUsersWhere( where: "role IN ('moder', 'admin', 'front', 'back', 'tester', 'designer')");

    if (User::checkRole(['front', 'back', 'designer', 'tester'])) {
        $this->view->my_projects = BlogModel::getAll( where: " AND " . User::get( property: 'role') . " = " . User::get( property: 'id') . " ");
    } else if (User::checkRole(['moder'])) {
        $this->view->my_projects = BlogModel::getAll( where: " AND 'consultant_id' = " . User::get( property: 'id') . " ");
    }

    Request::setTitle( title: 'Dashboard');
}

```

Рис. 5.5 – Функція головної сторінки.

Рис. 5.6 – Форма авторизації.

```

<form class="text-left" method="post" action="{URL:panel/login}">
  <div class="form">
    <?php if (isset($this->errors) && $this->errors) { ?>
      <div class="alert alert-danger mb-4" role="alert">
        <button type="button" class="close" data-dismiss="alert" aria-label="Close"></button>
        <?php echo is_array($this->errors) ? implode( separator: "<br/>", $this->errors) : $this->errors; ?>
      </div>
    <?php } ?>

    <div id="username-field" class="field-wrapper input">
      <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none"
        stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
        class="feather feather-user">
        <path d="M20 21v-2a4 4 0 0 0-4-4H8a4 4 0 0 0-4 4v2"></path>
        <circle cx="12" cy="7" r="4"></circle>
      </svg>
      <input id="username" name="email" type="email" class="form-control" placeholder="Email Address"
        value="{?= post( name: 'email'); ?}">
    </div>

    <div id="password-field" class="field-wrapper input mb-2">
      <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none"
        stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
        class="feather feather-lock">
        <rect x="3" y="11" width="18" height="11" rx="2" ry="2"></rect>
        <path d="M7 11V7a5 5 0 0 1 10 0v4"></path>
      </svg>
      <input id="password" name="password" type="password" class="form-control" placeholder="Password"
        value="{?= post( name: 'password'); ?}">
    </div>

    <div class="d-sm-flex justify-content-between align-items-center">
      <div class="field-wrapper">
        <button type="submit" class="btn btn-primary" value="">Log In</button>
      </div>
      <div class="field-wrapper d-sm-flex justify-content-center align-items-center">
        <a href="{URL:panel/register}" class="forgot-pass-link">Register</a>
        <a href="{URL:panel/restore_password}" class="forgot-pass-link">Forgot Password?</a>
      </div>
    </div>
  </form>


```

Рис. 5.7 – Верстка форми авторизації.

# Registration

 **Firstname**

 **Lastname**

 **Email Address**

 **Password**

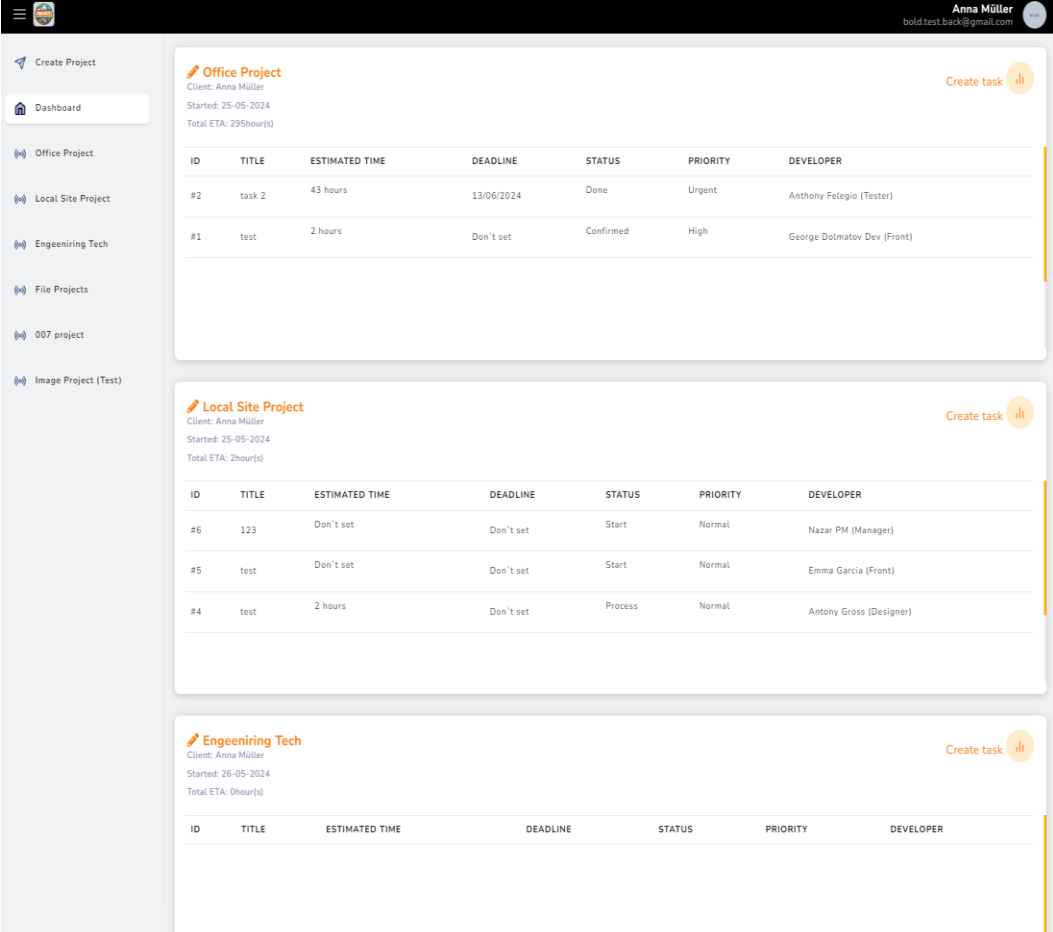
Register

[Log In](#)

[Forgot Password?](#)

© 2024 Manager Diplom Ltd. All rights reserved.

Рис. 5.8 – Форма реєстрації.



The screenshot shows a user dashboard for 'Anna Müller' (bold.test.back@gmail.com). The dashboard is divided into three main project sections, each with a 'Create task' button:

- Office Project**  
Client: Anna Müller  
Started: 25-05-2024  
Total ETA: 295hour(s)
- Local Site Project**  
Client: Anna Müller  
Started: 25-05-2024  
Total ETA: 2hour(s)
- Engeeniring Tech**  
Client: Anna Müller  
Started: 26-05-2024  
Total ETA: 0hour(s)

Each project section contains a table with the following columns: ID, TITLE, ESTIMATED TIME, DEADLINE, STATUS, PRIORITY, and DEVELOPER.

ID	TITLE	ESTIMATED TIME	DEADLINE	STATUS	PRIORITY	DEVELOPER
#2	task 2	43 hours	13/06/2024	Done	Urgent	Anthony Felegio (Tester)
#1	test	2 hours	Don't set	Confirmed	High	George Dolmatov Dev (Front)

ID	TITLE	ESTIMATED TIME	DEADLINE	STATUS	PRIORITY	DEVELOPER
#6	123	Don't set	Don't set	Start	Normal	Nazar PM (Manager)
#5	test	Don't set	Don't set	Start	Normal	Emma Garcia (Front)
#4	test	2 hours	Don't set	Process	Normal	Antony Gross (Designer)

ID	TITLE	ESTIMATED TIME	DEADLINE	STATUS	PRIORITY	DEVELOPER
----	-------	----------------	----------	--------	----------	-----------

Рис. 5.9 – Сторінка системного користувача «Клієнт».

```

<?php if ($this->my_projects) {
    foreach ($this->my_projects as $project) { ?>
        <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12 layout-spacing">

            <div class="widget-two">
                <div class="widget-content">
                    <div class="w-numeric-value">
                        <div class="w-content">

                            <span class="w-value">
                                <a href="{LINK:edit-project/<?= $project->id ?>}" class="dropdown-toggle">
                                    <i class="fa fa-pencil-alt"></i>
                                </a>
                                <a href="{URL:client-project/<?= $project->id ?>}"><?= $project->title ?></a></span>
                            <span
                                class="w-numeric-title">Client: <?= $project->client->firstname . ' ' . $project->client->lastname ?></span><br />
                            <span class="w-numeric-title">Started: <?= date( format: 'd-m-Y', $project->time) ?></span><br />
                            <span class="w-numeric-title">Total ETA: <?= $project->total_eta ?>hour(s)</span><br />
                        </div>
                    </div>

                    <a href="" onClick="load('page/create_task/<?= $project->id ?>?reload=false'); return false;"
                        class="dropdown-toggle">
                        <span>Create task</span>
                    </a>
                    <div class="w-icon">
                        <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none"...>
                    </div>
                </div>
            </div>

            <div class="widget_scroll_box widget_mm">
                <div class="table-responsive">
                    <table class="table">
                        <thead>
                            <tr...>
                        </thead>
                        <tbody id="vacancies_result">

                            <?php if ($project->tasks) {
                                foreach ($project->tasks as $task) { ?>
                                    <tr...>
                                    <script...>
                                    <?php
                                }
                            } ?>

                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    } ?>
} ?>

```

Рис. 5.10 – Верстка системної сторінки Клієнта.

### 5.3. Створення функціональних модулів програмного продукту

При створенні функціональних модулів програмного забезпечення важливо врахувати кілька критеріїв, щоб забезпечити їх якість та ефективність:

1. Модуль повинен виконувати свою основну функцію та досягати поставленої мети. Він має вміти обробляти вхідні дані або події, виконувати необхідні обчислення чи операції та повертати очікувані результати або зміни стану.
2. Модуль повинен працювати стабільно та надійно. Він має бути здатний витримувати навантаження, бути стійким до помилок, виявляти та обробляти виняткові ситуації, а також відновлюватися після збоїв.
3. Модуль повинен бути добре структурованим і розділеним на логічні компоненти, що полегшує розробку, тестування та підтримку.
4. Модуль повинен бути легким для розуміння та управління.
5. Модуль повинен бути захищеним від потенційних загроз безпеці. Це включає валідацію вхідних даних, обмеження доступу до конфіденційної інформації, захист від зловмисних атак та забезпечення цілісності даних.

Визначимо функціональні модулі кожної ролі.

*Адміністратор:*

- Проекти.
- Аналітика.
- Користувачі.
- Налаштування.
- Відео-спостереження.

*Менеджер:*

- Проекти.

*Клієнт:*

- Проекти.
- Створення проекту.

*Розробники (фронт-енд, бек-енд, дизайнер, тестер):*

- Проекти.

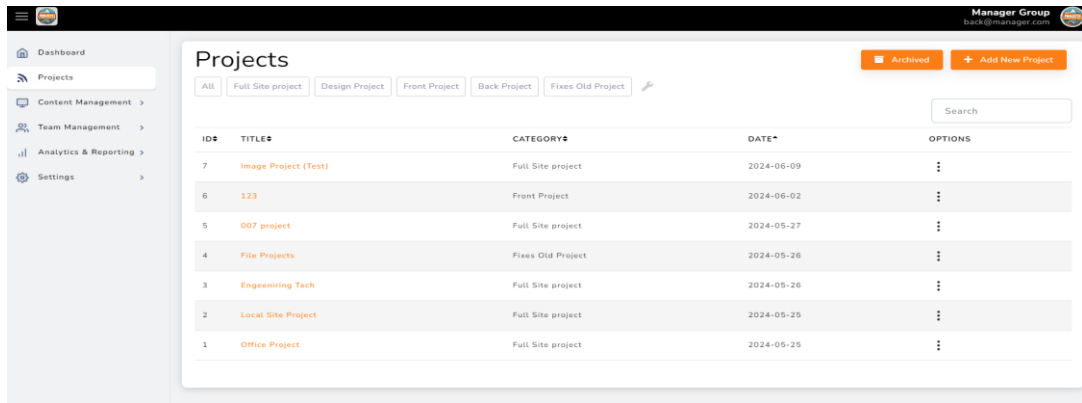


Рис. 5.11 – Модуль «Проекти» головна сторінка (вигляд Адміністратора)

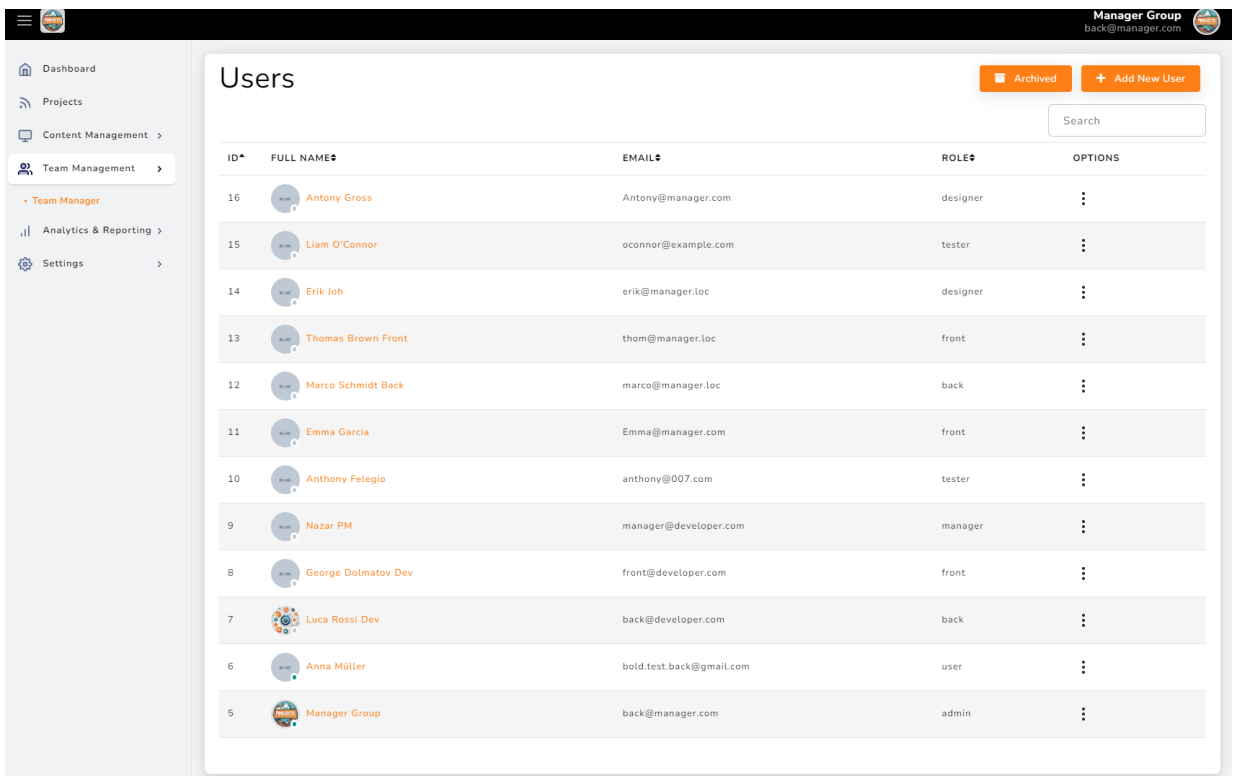


Рис. 5.12 – Модуль «Користувачі» (вигляд Адміністратора)

```

public function indexAction()
{
    $this->view->count = $count = BlogModel::countActive();
    Pagination::calculate(post( name: 'page', mode: 'int', default: 1), self::$numElement, $count);

    //search
    $this->view->list = BlogModel::getActive( where: false, categoryId: false, orderBy: 'id', sort: 'DESC', Pagination::$start, Pagination::$end);
    $this->view->elemPerPage = self::$numElement;

    Model::import( model: 'panel/blog/categories');
    $this->view->categories = CategoriesModel::getAll();

    Request::setTitle( title: 'Projects');
}

```

Рис. 5.13 – Функція головної сторінки модуля «Проекти»

Рис. 5.14 – Модуль «Проекти» вікно редагування проекту (вигляд Адміністратора)

```

public function editAction()
{
    $id = intval(Request::getUri( num: 0));
    $this->view->blog = BlogModel::get($id);

    if (!$this->view->blog)
        redirect(url('panel/blog'));

    Model::import( model: 'panel/blog/categories');
    $this->view->sectors = CategoriesModel::getAll();

    Model::import( model: 'panel/team');
    $this->view->team = TeamModel::getUsersWhere( where: "role IN ('moder', 'admin', 'user', 'back', 'tester', 'designer', 'front')");

    if ($this->startValidation()) {
        $this->validatePost( key: 'title', name: 'Project Name', rules: 'required|trim|min_length[1]|max_length[200]');
        $this->validatePost( key: 'consultant_id', name: 'Manager', rules: 'required|trim');
        $this->validatePost( key: 'client_id', name: 'Client', rules: 'required|trim');
        $this->validatePost( key: 'content', name: 'Description', rules: 'required|trim|min_length[0]');
        $this->validatePost( key: 'sector', name: 'Category', rules: 'required|trim|min_length[1]');
        $this->validatePost( key: 'time', name: 'Date Published', rules: 'trim|min_length[1]|max_length[50]');
        $this->validatePost( key: 'posted', name: 'Posted', rules: 'trim|min_length[1]|max_length[50]');

        // Times comparing/checking
        $intTime = convertStringTimeToInt(post( name: 'time'));
        $checkTime = date( format: "d/m/Y", $intTime);

        if ($checkTime != post( name: 'time')) {...}
        $intTimeDeadline = convertStringTimeToInt(post( name: 'deadline'));
        $checkTimeDeadline = date( format: "d/m/Y", $intTimeDeadline);

        if ($checkTimeDeadline != post( name: 'deadline')) {...}

        if ($this->isValid()) {
            $data = array(
                'title' => post( name: 'title'),
                'consultant_id' => post( name: 'consultant_id', mode: 'int', default: 0),
                'client_id' => post( name: 'client_id', mode: 'int', default: 0),
                'back' => post( name: 'back', mode: 'int', default: 0),
                'front' => post( name: 'front', mode: 'int', default: 0),
                'tester' => post( name: 'tester', mode: 'int', default: 0),
                'designer' => post( name: 'designer', mode: 'int', default: 0),
                'image' => post( name: 'image'),
                'dev_time' => post( name: 'dev_time', mode: 'int', default: 0),
                'content' => post( name: 'content'),
                'sector' => post( name: 'sector'),
                'status' => post( name: 'status'),
                'file' => post( name: 'file'),
                'slug' => Model::createIdentifier( table: 'blog', makeSlug( post( name: 'title'), fieldName: 'slug', $this->view->blog->id),
                'posted' => post( name: 'posted') ? 'yes',
                'time' => $intTime,
                'deadline' => $intTimeDeadline,
            );

            // Copy and remove image
            if ($this->view->blog->image != $data['image'] && $data['image']) {...}
            // Copy and remove file
            if ($data['file'] != $this->view->blog->file && $data['file']) {...}
        }
    }
}

```

Рис. 5.15 – Функція редагування проекту

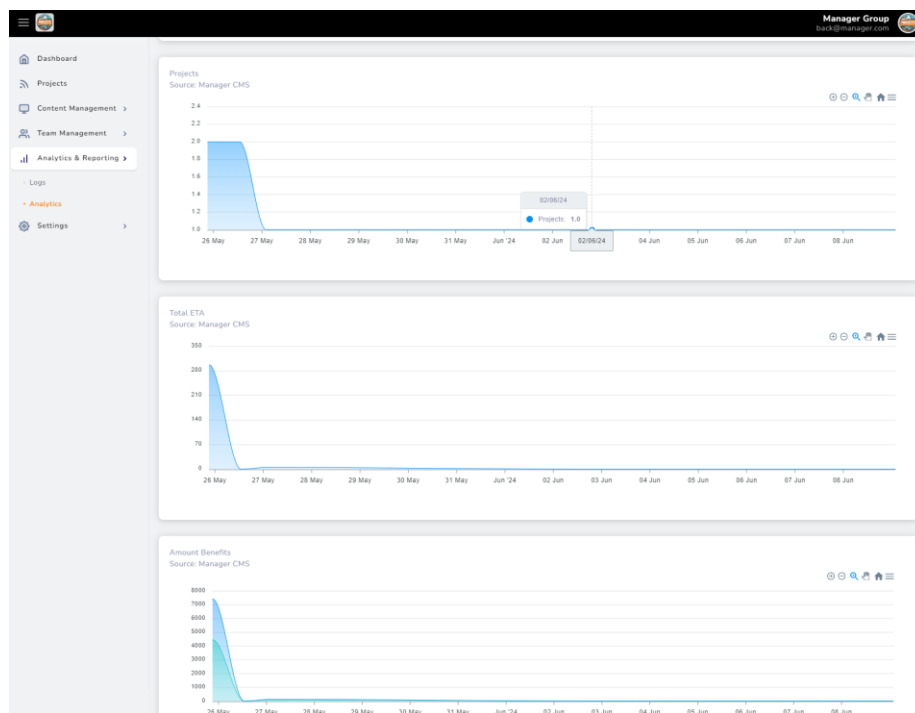


Рис. 5.16 – Модуль «Аналітика» головна сторінка (вигляд Адміністратора)

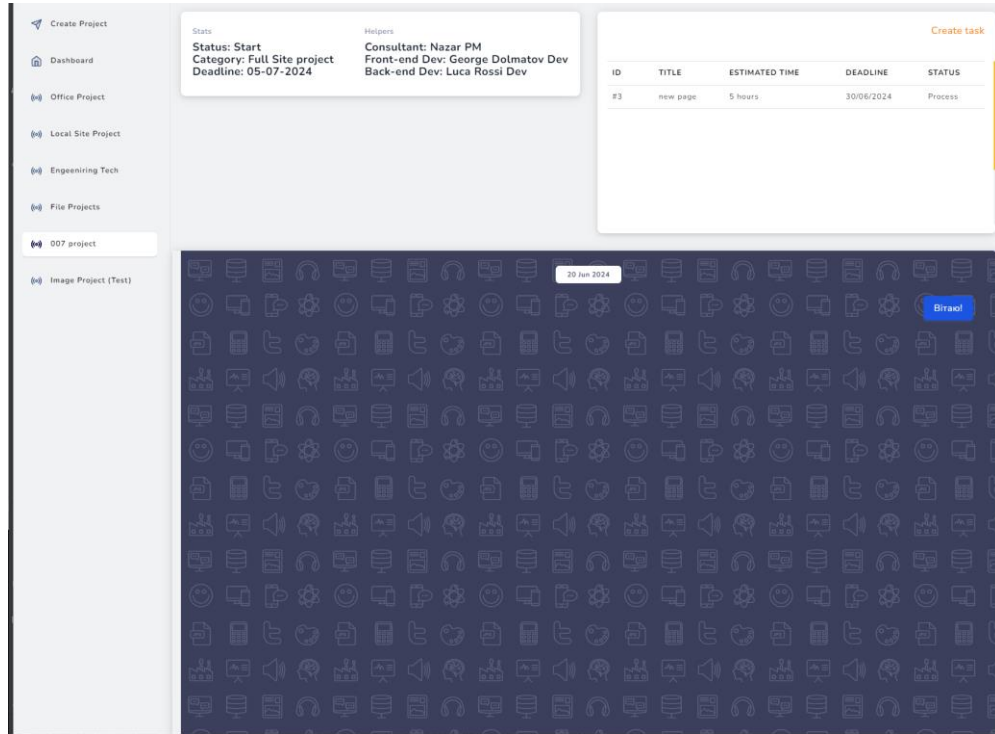


Рис. 5.17 – Модуль «Проекты» вікно перегляду проекту

```

<!-- Left col -->
<div class="col-xl-6 col-lg-12 col-md-12 col-sm-12 col-12" style="...">
  <!-- Stats -->
  <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12 layout-spacing">
    <div class="widget-two">
      <div class="widget-content">
        <div class="w-numeric-value">
          <div class="w-content">
            <span class="w-numeric-title">Stats</span>
            <span class="w-value">Status: <?> ucfirst($this->project->status) ?></span>
            <span class="w-value">Category: <?> ucfirst($this->sectors[$this->project->sector]->name) ?></span>
            <span class="w-value">Deadline: <?> date( format: 'd-m-Y', $this->project->deadline) ?></span>
            <?php if ($this->project->timeLine) { ?>
              <span class="w-value">TimeLine: <?> $this->project->timeLine ?> hours</span>
            <?php } ?>
          </div>
          <div class="w-content">
            <?php if ($this->project->consultant_id) { ?>
              <span class="w-numeric-title">Helpers</span>
            <?php } ?>
            <?php if ($this->project->consultant_id) { ?>
              <span class="w-value">Consultant: <?> $this->team[$this->project->consultant_id]->firstname . ' ' . $this->team[$this->project->consultant_id]->lastname ?></span>
            <?php } ?>
            <?php if ($this->project->front) { ?>
              <span class="w-value">Front-end Dev: <?> $this->team[$this->project->front]->firstname . ' ' . $this->team[$this->project->front]->lastname ?></span>
            <?php } ?>
            <?php if ($this->project->back) { ?>
              <span class="w-value">Back-end Dev: <?> $this->team[$this->project->back]->firstname . ' ' . $this->team[$this->project->back]->lastname ?></span>
            <?php } ?>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<!-- Right col -->
<div class="col-xl-6 col-lg-12 col-md-12 col-sm-12 col-12" style="...">
  <!-- Latest Vacancies -->
  <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12 layout-spacing">
    <div class="widget-two">
      <div class="widget-content">
        <div class="w-numeric-value">
          <div class="w-content">
            <span class="w-numeric-title">Source: Website</span>
          </div>
          <a href="" onClick="load('page/create_task/<?> $this->project->id ?>'); return false;" class="dropdown-toggle">
            <span>Create task</span>
          </a>
        </div>
        <div class="widget__scroll_box widget_mm">
          <div class="table-responsive">
            <table class="table">
              <thead>
                <tr>
                  <th><div class="th-content">id</div></th>
                  <th><div class="th-content">Title</div></th>
                  <th><div class="th-content">Estimated Time</div></th>
                  <th><div class="th-content">Deadline</div></th>
                  <th><div class="th-content">Status</div></th>
                </tr>
              </thead>
              <tbody id="vacancies_result">
                <?php if ($this->project->tasks) {
                  foreach ($this->project->tasks as $task) { ?>
                    <tr>
                      <td class="max_w_80" title="#<?> $task->ref; ?>"><div class="td-content customer-name mini_size">#<?> $task->id; ?></div></td>
                      <td>
                        <div class="td-content product-brand">
                          <?> $task->name; ?>
                        </div>
                      </td>
                      <td><div class="td-content"><?> $task->dev_time ? $task->dev_time . ' hours' : 'Don't set'; ?></div></td>
                      <td><div class="td-content"><?> $task->deadline ? date( format: 'd/m/Y', $task->deadline) : 'Don't set'; ?></div></td>
                      <td><div class="td-content"><?> ucfirst($task->status); ?></div></td>
                    </tr>
                  } ?>
                </tbody>
              </table>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</div>

```

Рис. 5.18 – Верстка сторінки редагування замовлення



Рис. 5.19 – Модуль «Відео»

```
<script>
document.addEventListener("DOMContentLoaded", () => {
  var but = document.getElementById("but");
  var video = document.getElementById("vid");
  var mediaDevices = navigator.mediaDevices;
  video.muted = true;
  but.addEventListener("click", () => {

    // Accessing the user camera and video.
    mediaDevices
      .getUserMedia({
        video: true,
      })
      .then((stream) => {...})
      .catch(alert);
  });

  function stop(e) {
    var stream = video.srcObject;
    var tracks = stream.getTracks();

    for (var i = 0; i < tracks.length; i++) {
      var track = tracks[i];
      track.stop();
    }

    video.srcObject = null;
  }
});
</script>
<style>
#container {
  margin: 0px auto;
  width: 500px;
  height: 375px;
  border: 1px #333 solid;
  background-color: #a2a2a2;
}
video {
  width: 500px;
  height: 375px;
}
</style>
```

Рис. 5.20 – Скрипт реалізації відеоспостереження та його стилі



## CREATE TASKS

Task Name

Description

 Create a new task

Рис. 5.21 – Модуль «Створення задач»

```
public function create_taskAction()
{
    Request::ajaxPart();

    Model::import('model/panel/blog');

    $id = intval(Request::getUrl('name/0'));
    $this->view->project = BlogModel::get($id);

    if (!$this->view->project)
        redirect(url('clients-dashboard'));

    Model::import('model/panel/blog/categories');
    $this->view->sectors = CategoriesModel::getAll();

    Model::import('model/panel/team');
    $this->view->team = TeamModel::getUsersWhere('where' => ['role' IN ('moder', 'admin', 'user')]);

    if ($this->startValidation()) {
        $this->validatePost(['key' => 'name', 'name' => 'Task Name', 'min' => 'required|trim|min_length[1]|max_length[200]'];
        $this->validatePost(['key' => 'description', 'name' => 'Description', 'min' => 'required|trim|min_length[0]'];

        if ($this->isValid()) {
            $data = array(
                'name' => post('name'),
                'blog_id' => $id,
                'content' => post('description'),
                'time' => time(),
            );

            $result = Model::insert('table/blog_tasks', $data); // Insert row
            $insertID = Model::insertID();

            if (!$result && $insertID) {
                $this->makeVersion($insertID, 'table/blog_tasks', 'type' => 'add');

                Model::insert('table/actions_logs', ['user_id' => User::get('property' => 'id'), 'action' => 'add', 'entity' => 'blog/' . $insertID, 'time' => time()]);

                Request::addResponse('action' => 'func', 'key' => 'closePopup');
                if (get('name' => 'reload') != 'false')
                    Request::addResponse('action' => 'redirect', 'key' => 'false', url('client-project', $id));
                Request::endAjax();
            } else {
                Request::returnError('error' => 'Database error');
            }
        } else {
            if (Request::isAjax())
                Request::returnErrors($this->validationErrors);
        }
    }

    Request::addResponse('action' => 'html', 'key' => 'popup', $this->getView());
    Request::endAjax();
}
```

Рис. 5.22 – Функція валідації та створення задачі

```

<?php Popupp::head( title: 'Create Tasks', classHelper: 'old-popup-styles full-height'); ?>

<form id="task_form" class="pop_form" style="...">
  <div style="...">
    <div class="form-row">
      <div class="form-group col-md-12">
        <label for="name">Task Name</label>
        <input type="text" class="form-control" name="name" id="name">
      </div>
    </div>
    <div class="form-row">
      <div class="form-group col-md-12">
        <label for="description">Description</label><br/>
        <textarea name="description" id="description" style="..." rows="10"></textarea>
      </div>
    </div>

    <a class="btn btn-success"
      onclick="load('page/create_task/<?=$this->project->id; ?>?reload=<?=$get( name: 'reload') ?>', 'form:#task_form'); return false;"
      <i class="fas fa-crop-alt"></i>Create a new task
    </a>
  </form>

<?php Popupp::foot(); ?>
<?php Popupp::closeListener(); ?>

```

Рис. 5.23 – Верстка вікна «Створення задачі»

**Office Project**  
 Client: Anna Müller  
 Started: 25-05-2024  
 Project ETA: 250hour(s)  
 Total ETA: 295hour(s)

ID	TITLE	ESTIMATED TIME	DEADLINE	STATUS	PRIORITY	DEVELOPER
#2	task 2	<input type="text" value="43 hours"/>	<input type="text" value="13/06/2024"/>	<input type="text" value="Done"/>	<input type="text" value="Urgent"/>	-- Select Dev --
#1	test	<input type="text" value="2 hours"/>	<input type="text" value="20/06/2024"/>	<input type="text" value="Confirmed"/>	<input type="text" value="High"/>	George Dolmatov C

Рис. 5.24 – Вікно проекту для Менеджера з редагуванням атрибутів задач

```

public function change_devsAction()
{
  Request::ajaxPart();

  Model::import( model: 'panel/blog');
  $id = post( name: 'project', mode: 'int');

  $result = Model::update( table: 'blog', [post( name: 'type') => post( name: 'id')], where: 'id' = '$id'); // Update row

  if ($result) {
    Request::addResponse( action: 'func', key: 'noticeSuccess', value: 'Saved');

    $this->view->project = BlogModel::get($id);

    if ($this->view->project->back && $this->view->project->front && $this->view->project->tester && $this->view->project->designer && $this->view->project->consultant_id)
      Request::addResponse( action: 'remove', key: '#item_' . $id);
  }

  Request::endAjax();
}

```

Рис. 5.25 – Функція редагування

#### 5.4. Опис реалізованої програмної розробки (інструкція користувача)

##### **Клієнт:**

- Створити проект:
    - Використовуйте опцію " Створити проект " для заповнення форми створення проекту.
  - Проекти:
    - Використовуйте опцію "Проекти" для перегляду своїх проектів.
    - Здійснюйте пошук проектів на головній сторінці "Дашборд".
    - Ознайомтесь з детальною інформацією про кожен проект, натиснувши на назву проекту.
    - Редагуйте інформацію про проект, натиснувши на іконку олівця.
- Бухгалтер:
- Розрахунок витрат на розробку:
    - Біля кожного проекту відображається орієнтовна ціна на всі затрачені години проекту.
    - Біля кожного проекту відображається орієнтовна кількість годин на всі затрачені завдання проекту.
  - Розрахунок прибутку:
    - Розрахунок здійснюється автоматично.  
(Кількість год. \* Ціну за год.' ) – (Кількість год. \* Ціна за год. розробки')
    - ‘ - встановлюється адміністратором в адмін-панелі.

##### **Адміністратор:**

- Відеоспостереження:
  - Використовуйте опцію "Відеоспостереження" для перегляду відео в реальному часі з камер спостереження.
- Проекти:
  - Використовуйте опцію "Проекти" для перегляду всіх проектів.
  - Здійснюйте пошук проектів використовуючи фільтр праворуч над списком.

- Ознайомтесь з детальною інформацією про кожен проект, натиснувши на назву проекту або зображення олівця.
- Формування відсотка зарплатні ролям:
- Використовуйте опцію "Ціна клієнтів" у розділі генеральних налаштувань.
- Кількість грошей в год.
- Перегляньте та збережіть налаштування перед застосуванням.

### **Розробники (бек-енд, фронт-енд, дизайнер, тестер):**

- Головне меню:
  - Використовуйте головне меню як таблицю активних проектів та їх завдань.
  - Редагуйте статуси завдань та затрачену кількість годин на відповідне завдання.
  - Орієнтуйтеся на
  - Не враховуйте час який витрачається на пошук рішення завдання, а враховуйте лише той час який знадобився на реалізацію рішення.
  - актуальної інформації про залишок продукції на складі.

Кожна роль повинна авторизуватися в системі використовуючи свій індивідуальний обліковий запис. Будьте обережні зі збереженням своїх облікових даних та не розголошуйте їх іншим користувачам.

## **6. РОЗРОБКА SCADA ДЛЯ ТЕХНОЛОГІЧНОГО ОБ'ЄКТУ МОНІТОРИНГУ І УПРАВЛІННЯ**

### **6.1. Створення моделі системи автоматичного регулювання в програмному середовищі Step7**

У програмі SIMATIC Manager створюємо новий проект для контролера S7-313-2DP, скориставшись підказками помічника Wizard. Потім запускаємо програму конфігурації обладнання HW Config.

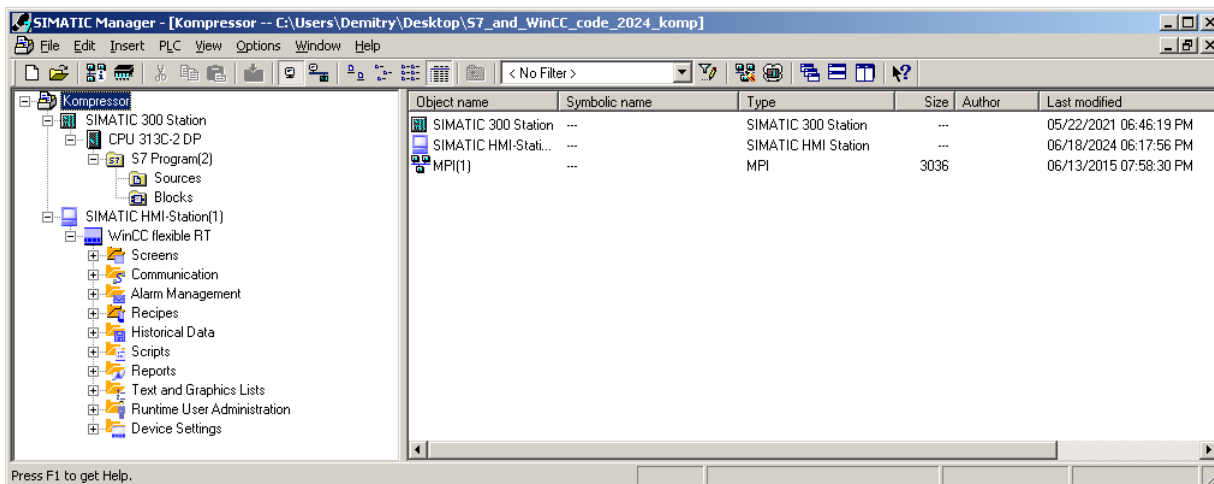


Рис. 6.1 – Створений проект в програмі SIMATIC Manager

Далі в програмі HW Config додаємо мережу та виконуємо конфігурацію обладнання.

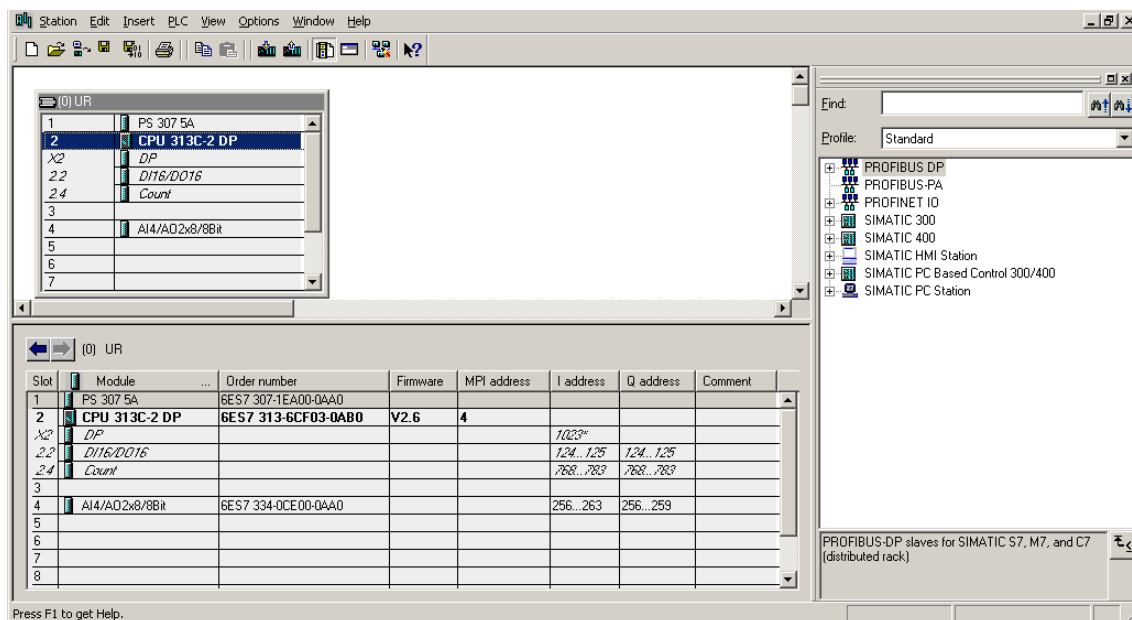


Рис. 6.2 – Додане і зконфігуроване обладнання

В якості зв'язку між контролером і SCADA-системою будемо використовувати MPI-мережу на основі інтерфейсу RS - 485.

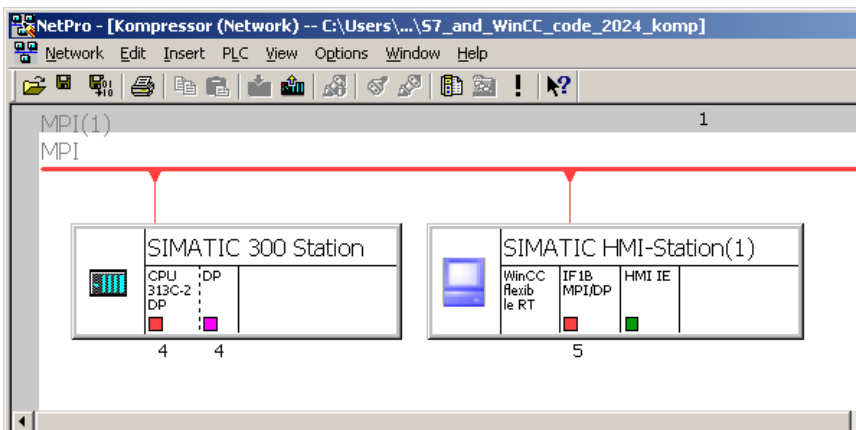


Рис. 6.3 – Зконфігурована мережа MPI

### 6.1.1 Створення програми імітації роботи САР.

Створимо блок DB2 для HMI-тегів для подальшого зберігання, використання в програмі контролера і зв'язку SCADA-системи з контролером.

Address	Name	Type	Initial value	Actual value	Comment
0.0	PID_Tk.MAN_ON	BOOL	FALSE	FALSE	сигнал перемикання
2.0	PID_Tk.MAN	REAL	0.000000e+000	0.000000e+000	сигнал керування в
6.0	PID_Tk.SP_INT	REAL	4.000000e+001	2.500000e+001	задане значення
10.0	PID_Tk.PV	REAL	0.000000e+000	0.000000e+000	поточне значення
14.0	PID_Tk.U	REAL	0.000000e+000	0.000000e+000	керуваний вплив
18.0	PID_Tk.GAIN	REAL	7.390000e+000	-1.800000e+000	коефіцієнт передачі
22.0	PID_Tk.TI	TIME	T#233340MS	T#203340MS	час ізодрому
26.0	PID_Tk.TD	TIME	T#188900MS	T#108900MS	час предварення
30.0	PID_Tk.TM_LAG	TIME	T#33780MS	T#33780MS	час інерційності реє
34.0	PID_Mk.MAN_ON	BOOL	FALSE	FALSE	сигнал перемикання
36.0	PID_Mk.MAN	REAL	0.000000e+000	0.000000e+000	сигнал керування в
40.0	PID_Mk.SP_INT	REAL	7.500000e+001	4.000000e+000	задане значення
44.0	PID_Mk.PV	REAL	0.000000e+000	0.000000e+000	поточне значення
48.0	PID_Mk.U	REAL	0.000000e+000	0.000000e+000	керуваний вплив
52.0	PID_Mk.GAIN	REAL	3.560000e+000	1.200000e+001	коефіцієнт передачі
56.0	PID_Mk.TI	TIME	T#118600MS	T#158600MS	час ізодрому
60.0	PID_Mk.TD	TIME	T#48500MS	T#48500MS	час предварення
64.0	PID_Mk.TM_LAG	TIME	T#900MS	T#13	час інерційності реє
68.0	PUSK	BOOL	FALSE	FALSE	Команда пуск
68.1	STOP	BOOL	FALSE	FALSE	Команда стоп
70.0	Time_ZS	S5TIME	S5T#10S	S5T#10S	Час звукової сигналі
72.0	V1_start	S5TIME	S5T#10S	S5T#10S	Затримка часу при в
74.0	M1_start	S5TIME	S5T#10S	S5T#10S	Затримка часу при в
76.0	T_START	BOOL	FALSE	FALSE	Ввімкнення контуру
76.1	TP_START	BOOL	FALSE	FALSE	Ввімкнення контуру
78.0	M2_start	S5TIME	S5T#10S	S5T#10S	Затримка часу при в
80.0	M2_stop	S5TIME	S5T#5S	S5T#5S	Затримка часу на в
82.0	M1_stop	S5TIME	S5T#15S	S5T#15S	Затримка часу на в
84.0	V1_stop	S5TIME	S5T#10S	S5T#10S	Затримка часу на в

Рис. 6.4 – Створений блок HMI-тегів

Створимо модель об'єкта керування. Фрагмент програми моделі ОК наведено на рис. 6.5.

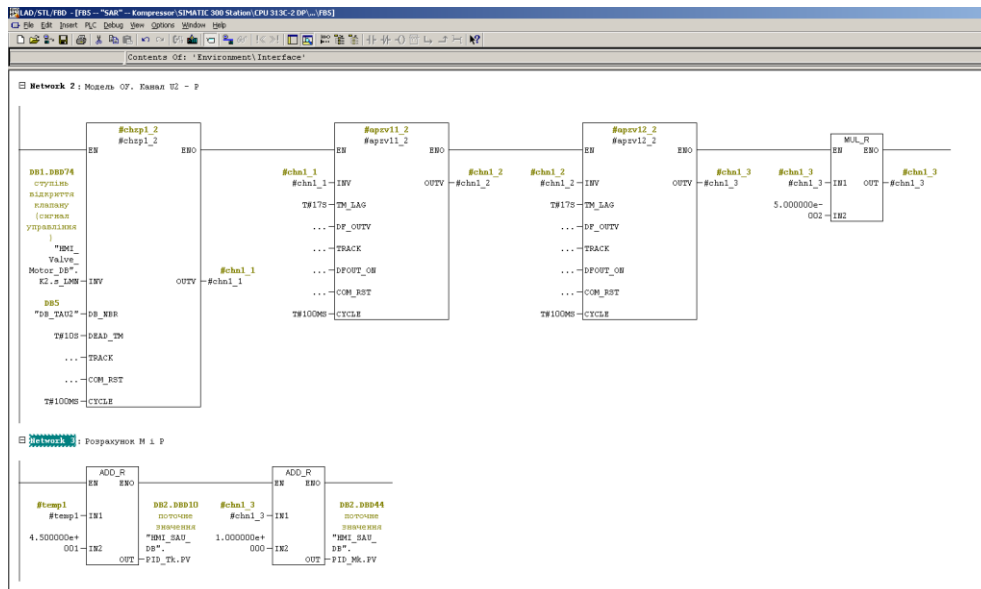


Рис. 6.5 – Фрагмент програми з моделлю ОК за каналами керування відносною вологості і тиску повітря на виході установки яка створена в Step7

Створимо регулятор відносної вологості повітря на виході з повітряної компресорної станції (рис. 6.6).

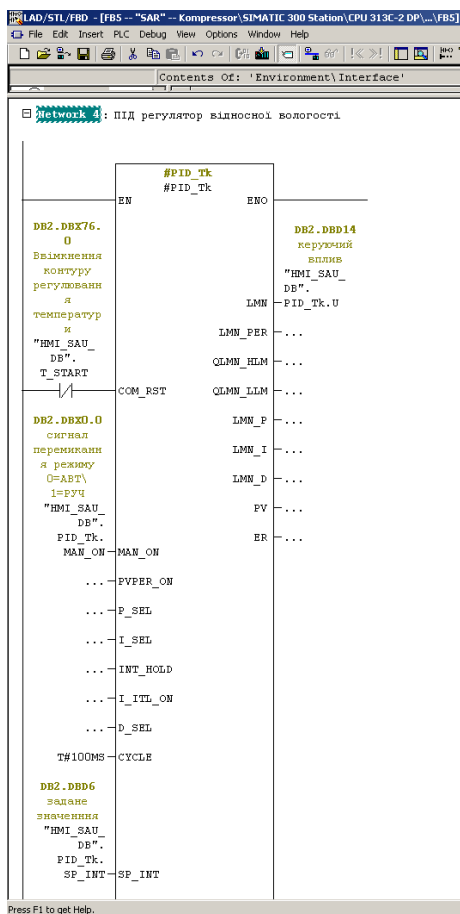


Рис. 6.6 – Фрагмент програми з моделлю ПІД-регулятора відносної вологості повітря яка створена в Step7

## 6.1.2 Виконаємо зв'язок SCADA-системи і проекту SIMATIC Manager.

За допомогою SCADA-системи ми матимемо можливість змінювати параметри регулятора і значення заданої відносної вологості повітря і його тиску на виході з повітряної компресорної станції.

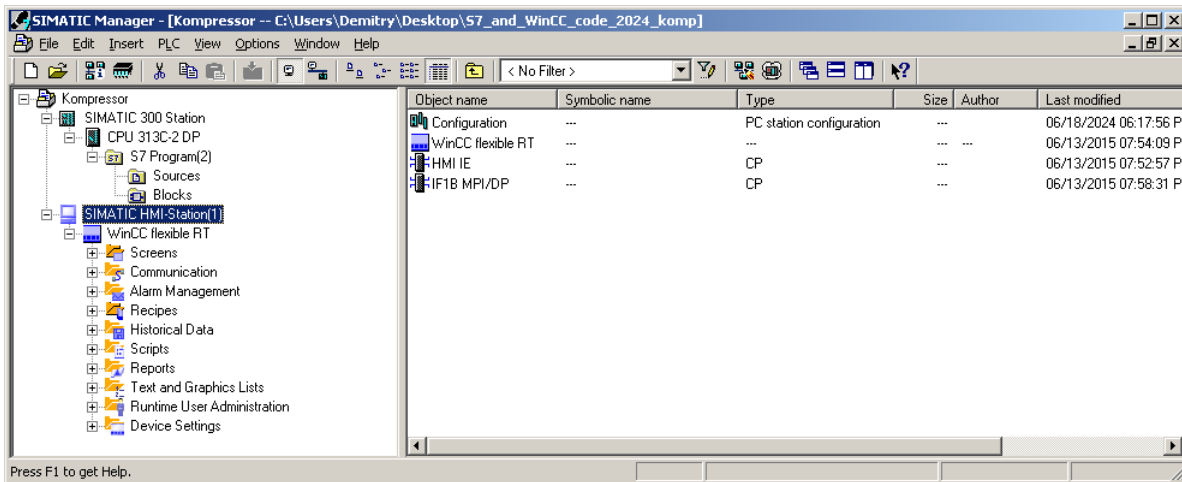


Рис. 6.7

– Загальний вигляд вікна Simatic Manager при інтеграції SCADA-системи і проекту контролера Step7

## 6.2. Створення екранної панелі управління в SCADA WinCCflexible

6.2.1. Створимо екранну панель користувача для керування процесом у повітряній компресорній станції

Наша панель управління передбачатиме:

- відображення поточного стану обладнання повітряної компресорної станції;
- можливість перемикання стану обладнання повітряної компресорної станції;
- можливість ручного дистанційного керування;
- відображення регульованих змінних повітряної компресорної станції;
- відображення повідомлень про відхилення в технологічному процесі;
- виконання налаштування САР;
- запис даних технологічного процесу в БД MySQL;
- авторизований доступ користувачів до системи керування.

Параметри мережевої взаємодії між контролером та SCADA налаштовуються у розділі «Connection» (рис. 6.8).

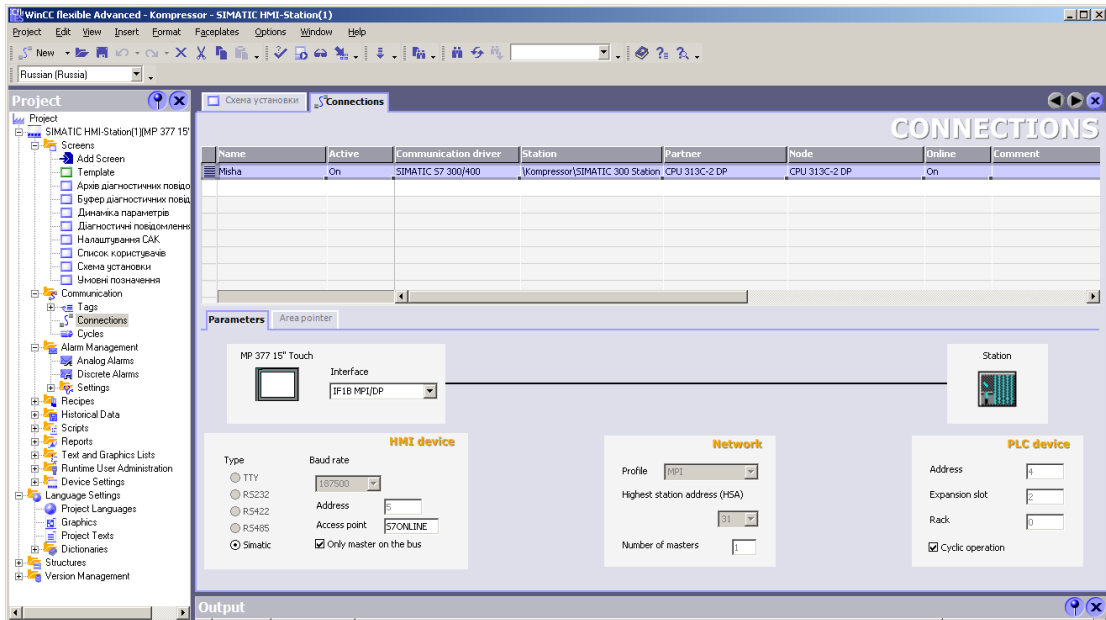


Рис. 6.8 – Вікно налаштування мережевого зв'язку між контролером та SCADA

Далі створимо список тегів, які будуть використовуватися в ході роботи. Фрагмент вікна з тегами ПД регулятора відносної вологості повітря на виході з повітряної компресорної станції на рис. 6.9.

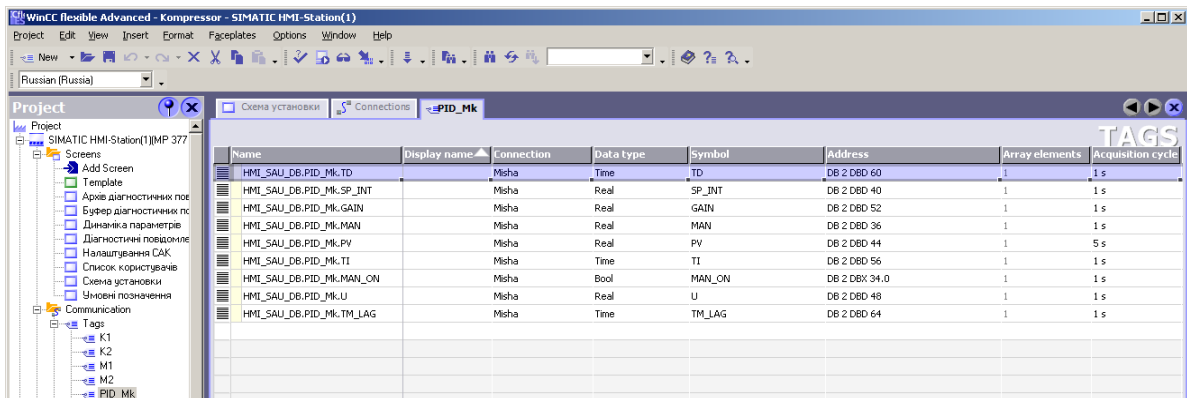


Рис. 6.9 - Список тегів для ПД регулятора відносної вологості повітря на виході установки

Далі створюємо декілька екранних форм для оператора. Створимо екранну форму «Схема установки» (рис. 6.10). Це головна екранна форма з зображенням мнемосхеми повітряної компресорної станції, стану обладнання, значень основних параметрів процесу, динаміки зміни відносної вологості та тиску повітря.

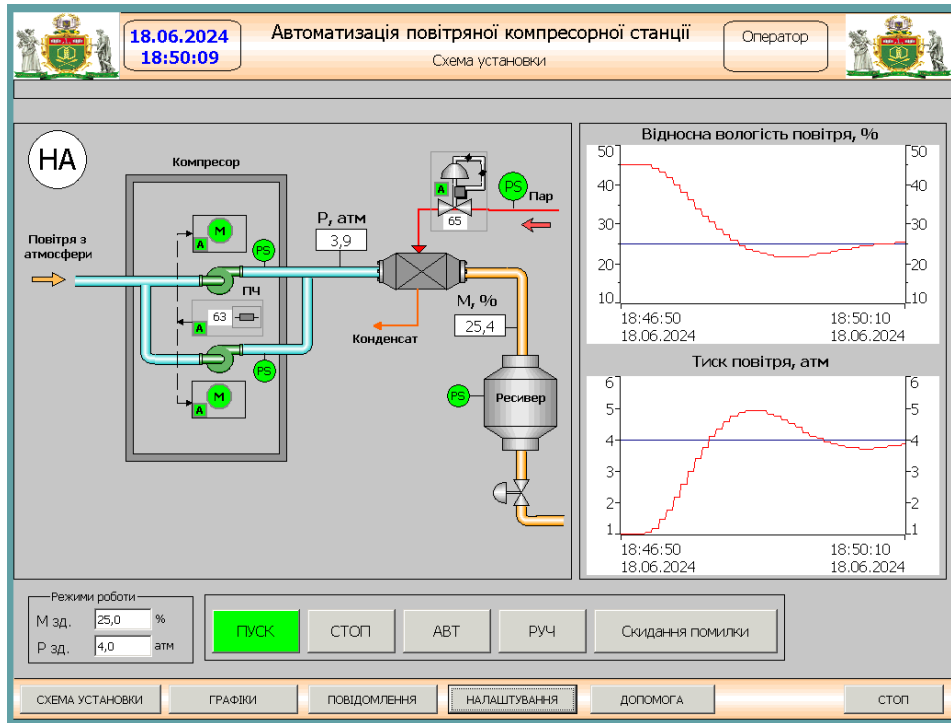


Рис. 6.10 – Екранна форма «Схема установки»

Нижче на рисунках в якості прикладу наведено зображення екранних форм «Повідомлення» та «Налаштування».

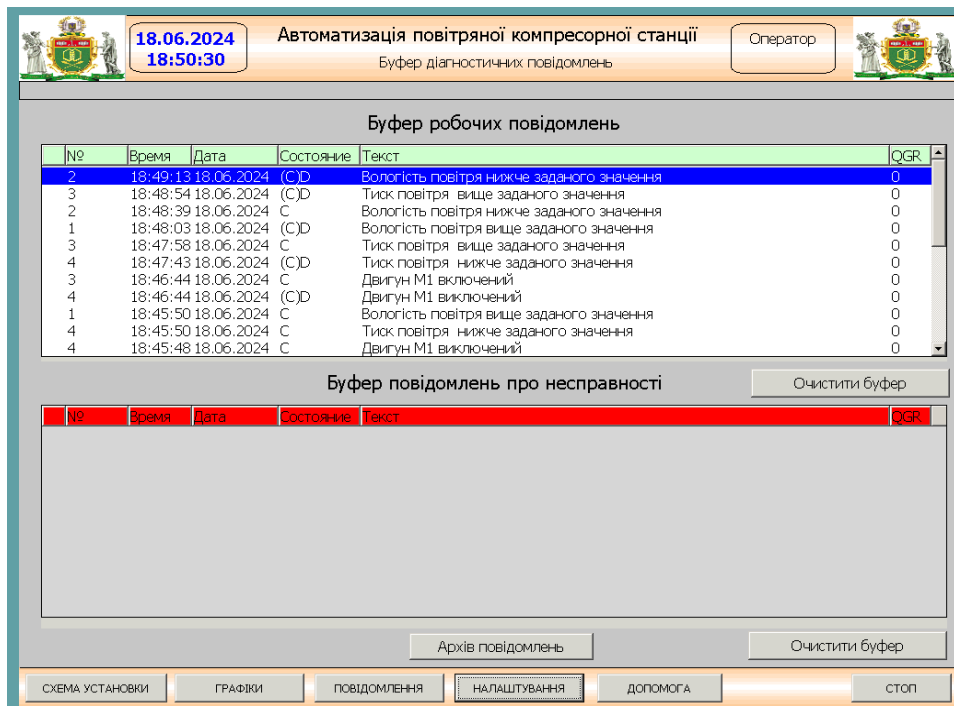


Рис. 6.11 – Екранна форма «Повідомлення»

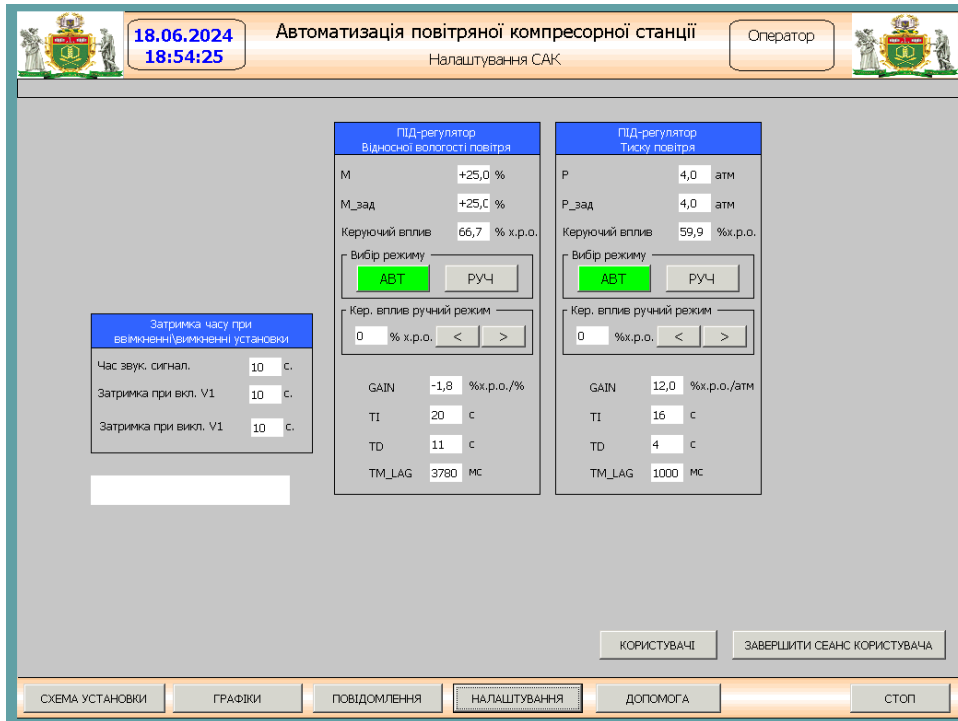


Рис. 6.12 – Екранна форма «Налаштування»

### 6.3. Реалізація обміну даними між SCADA та БД MySQL.

6.3.1. Організація обміну даними між SCADA-системою WINCCflexible і БД MySQL/MariaDB.

Для початку роботи дізнаємося яка БД використовується сервером (рис. 6.13).

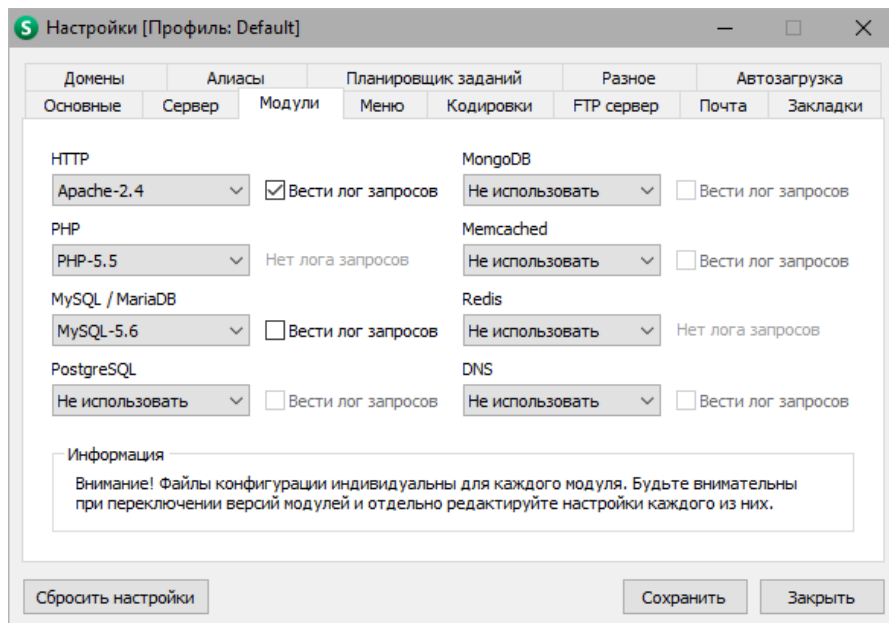


Рис. 6.13 – Налаштування веб-сервера и БД

Створимо БД для зберігання даних.

В нашому випадку буде використовуватися БД веб-додатку розробленого під час виконання курсової роботи.

Створимо таблицю для зберігання даних технологічного процесу, які буде надсилати SCADA-система (рис. 6.14).

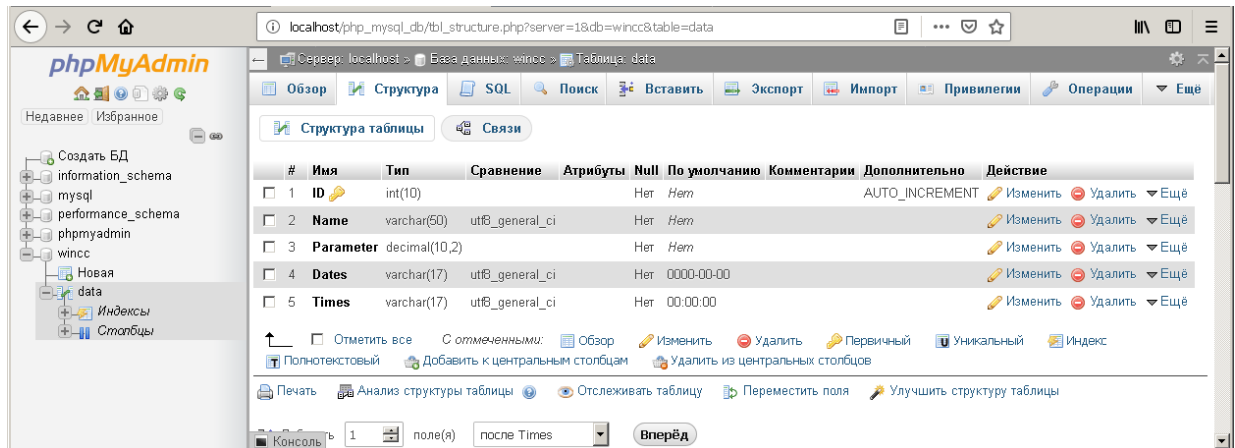


Рис. 6.14 – Створена таблиця “DATA”

Створимо об’єкт джерела даних odbs для обміну даними SCADA системи та веб-додатком (рис. 6.15, 6.16).

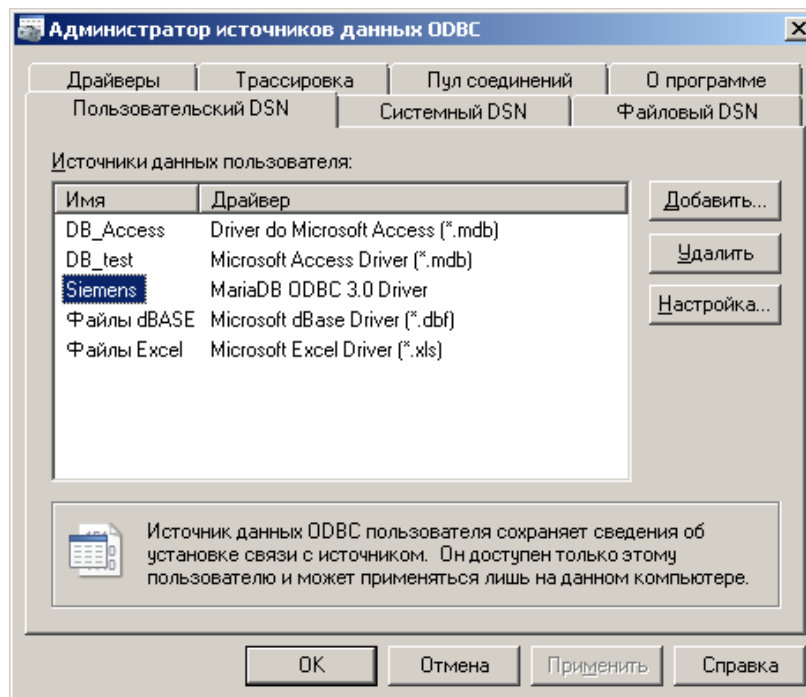


Рис. 6.15 - Створений об'єкт джерела даних ODBC

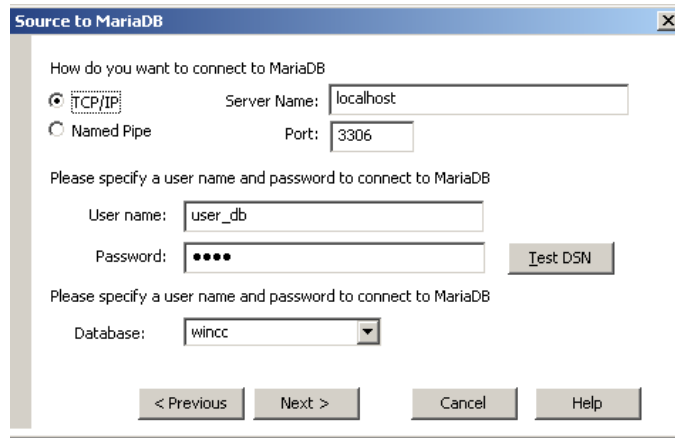


Рис. 6.16 - Створений об'єкт джерела даних ODBC. Підключення до бази даних Wincc

Для запису даних зі SCADA системи в базу даних необхідно створити програмний скрипт. Цей скрипт буде записувати необхідні данні у базу через створене джерело даних ODBC .

Створимо скрипт на запис даних в БД (рис. 6.17, 6.18).

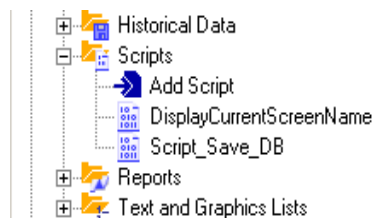


Рис. 6.17 - Список всіх створених скриптів використаних в проекті

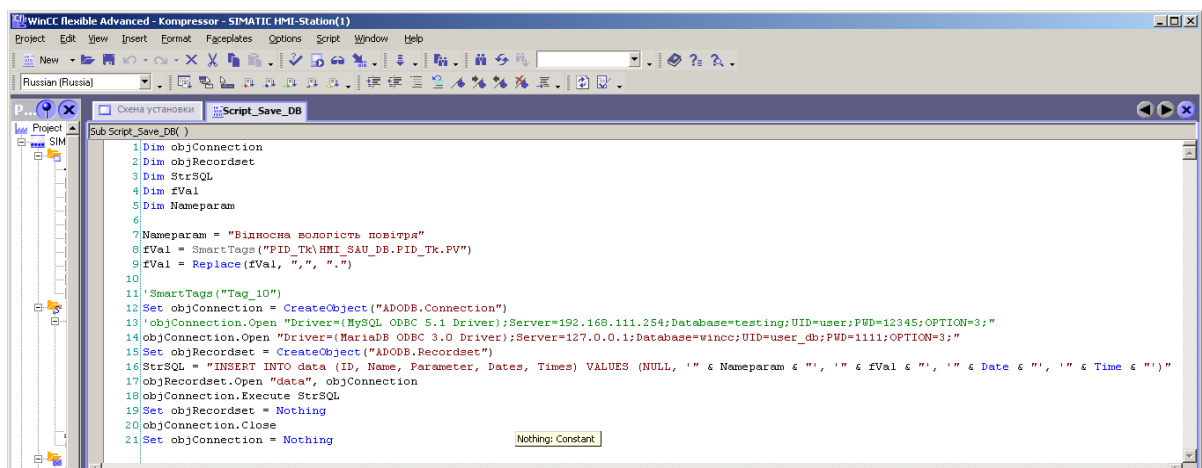


Рис. 6.18 - Скрипт для запису даних в БД

Налаштуємо наш проект і зробимо так що запис у базу даних буде проходити при кожній зміні тегу. Опитування тегу будемо проводити з кроком 5 секунд щоб зменшити обсяг даних що потраплять у базу.

В моєму випадку SCADA-система буде надсилати данні в БД про вихідну величину об'єкту керування.

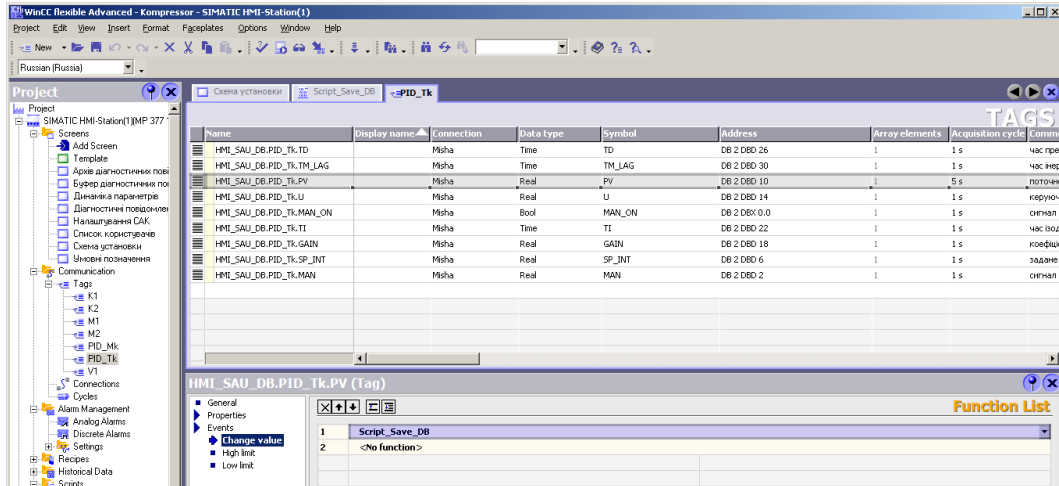


Рис. 6.19 – Фрагмент вікна прив'язки розробленого скрипта до тегу відносної вологості повітря на виході з повітряної компресорної станції

Перевіримо роботу нашого скрипта (рис. 6.20).

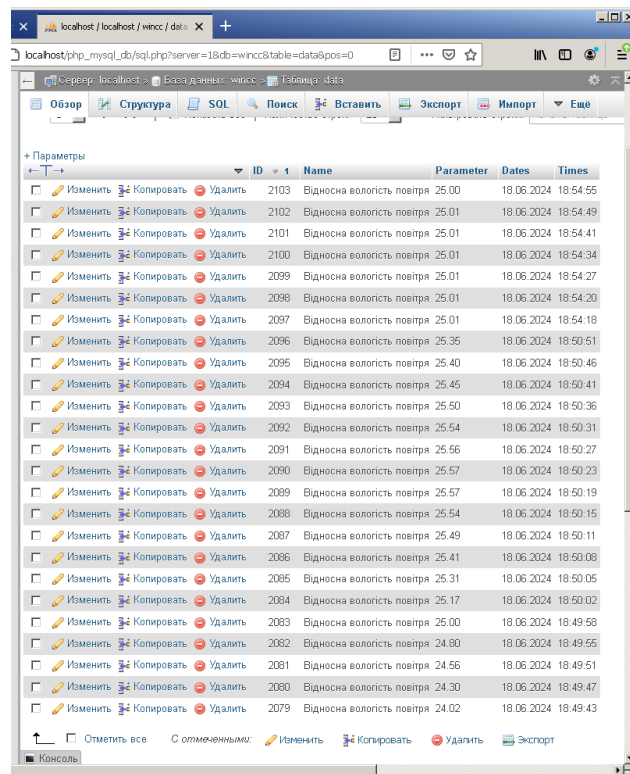


Рис. 6.20 - Результат запису даних в БД

6.3.2. Створимо в нашому веб-додатку сторінку для відображення параметрів ТП.

На сторінці з параметрами ТП передбачимо вивід інформації з БД в табличному і в графічному вигляді.

*Скрипт сторінки для відображення даних ТП (див. додаток 1, ст.21).*

*Скрипт модулю побудови графіка (див. додаток 2, ст.23).*

Перевіримо роботу нашого веб-додатку та SCADA-системи.

Завантажуємо в пам'ять написані програми для контролера та запускаємо симуляцію його роботи.

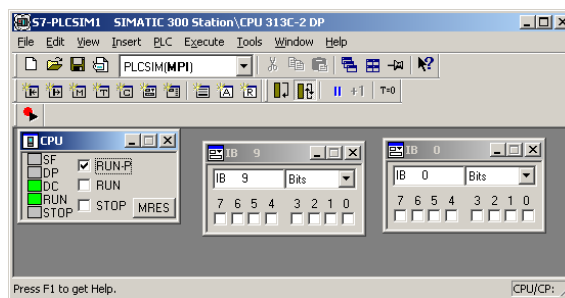


Рис.6.22 – Вікно симулятора контролера

Запускаємо SCADA-систему. Запускаємо систему керування повітряної компресорної станції і бачимо що значення відносної вологості повітря записуються у базу даних а потім відображуються у табличному вигляді на сторінці веб додатку.

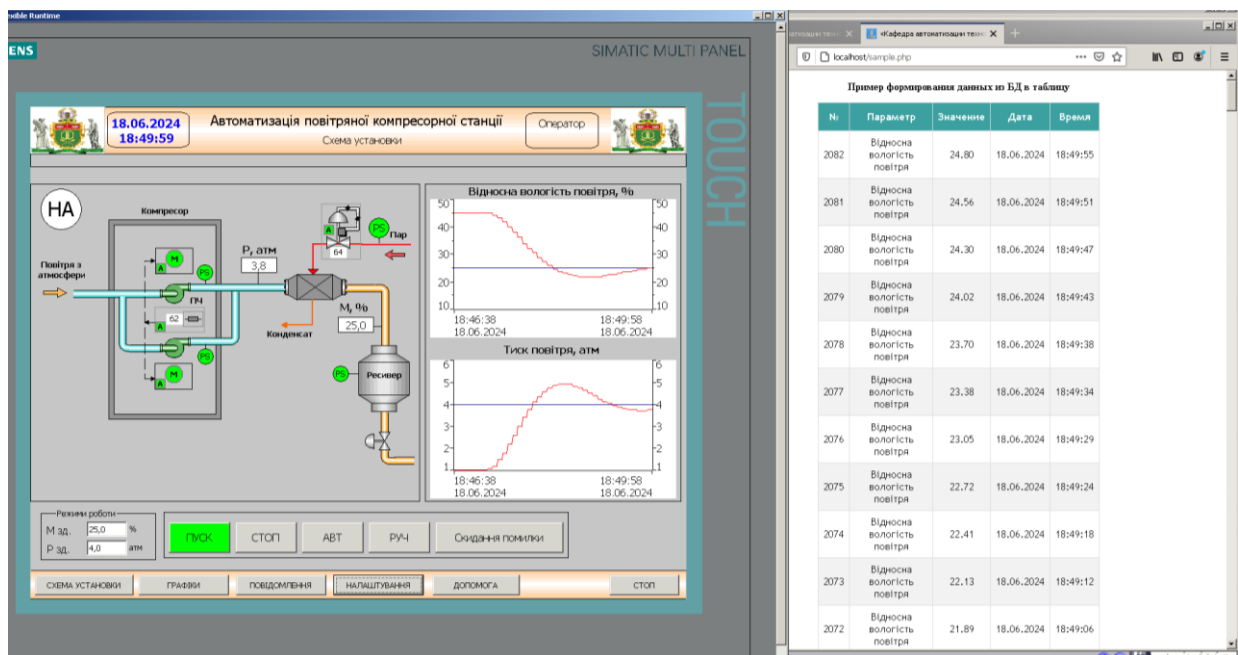


Рис.6.23 – Вікно SCADA-системи та сторінки веб-додатку

Після порівняння отриманих даних з веб-додатку та SCADA-системи, ми можемо зробити висновок що, вони ідентичні.

## **Розділ 7. Взаємодія системи з технічними засобами автоматизації**

### **7.1. Опис технічних засобів та мета їх використання**

В додатку передбачено використання одного технічного засобу автоматизації, а саме: відеоспостереження. Але в розділі також розглядається система обліку часу, так як вона може бути впроваджена в майбутньому, коли компанія буде зростати. Ці засоби допомагають забезпечити безпеку, контроль та ефективність роботи співробітників.

Відеоспостереження в офісі використовується для моніторингу роботи співробітників, трекінгу їх завантаженості протягом дня та забезпечення безпеки. Важливо враховувати наступні аспекти при виборі та встановленні системи відеоспостереження:

- Існують різні типи камер відеоспостереження, такі як аналогові та цифрові камери.
- За конструкцією камери бувають: купольні, корпусні, поворотні, мікрокамери.

При виборі системи відеоспостереження, потрібно враховувати місце, де будуть встановлені камери. Вони повинні бути встановлені у таких місцях, де можна забезпечити максимальне покриття зони спостереження. Так як приміщенням виступає офіс, то використовується бездротова камера для внутрішнього використання. Функція відео-спостереження розроблена в інтерфейсі «Менеджер».



Рис. - Внутрішня IP-камера спостереження TP-LINK Tapo C200

Камера спостереження TP-LINK Tapo C200 - важлива складова системи безпеки та контролю роботи співробітників.

Технічні характеристики TP-Link Tapo C200-

IP-камера

Бездротова

Кількість пікселів: 2 Мп

Кут огляду: 360° (горизонтально), 114° (вертикально)

Фокусна відстань: 4 мм

Дальність ІЧ підсвітки: 10 м

Зображення: 1920x1080, 15 кадрів/с, H.264

Підтримувані карти пам'яті: microSD до 128 ГБ

Нічна зйомка: є

Система виявлення руху: є

Робоча температура: від 0° до 40°C

Карткова система входу в офіс — це технологія, яка використовує безконтактні карти для ідентифікації працівників для управління доступом і обліку їхнього робочого часу. Альтернативою картковій системі є ручний облік часу приходу і відходу співробітників, що вимагає додаткових ресурсів, таких як адміністратори, які будуть вести журнал обліку.

Бувають різні види систем обліку часу. Якщо говорити про види таких систем, то вони **поділяються на:**

- Біометричні системи (дані зчитуються за геометрією обличчя або відбитком пальця)
- Системи за безконтактною картою (Якщо казати простими словами, то RFID-мітка складається з чипа, який зберігає інформацію, і антени, яка використовується для передачі даних. Після того, як мітку підносять до зчитувача, радіохвилі активують чип, що дозволяє зчитувачу переглядати інформацію, збережену на мітці.)
- Комбіновані системи (поєднують біометрію і карткові методи)

**За способом застосування ці системи поділяють на:**

- Автономні системи, які працюють самостійно без підключення до мережі.
- Мережеві системи, які підключені до локальної мережі або інтернету для централізованого управління та моніторингу.
- Хмарні системи, які використовують хмарні технології для зберігання даних і управління доступом.

**За способом підключення до ПК систему можна поділити на провідні і непровідні, детальніше описано нижче:**

Використання USB або кабельного підключення. Це найпоширеніші способи

підключення систем обліку робочого часу до комп'ютерів або серверів.

Підключення через контролери доступу і зчитувачі карт. Контролери доступу — це пристрої, які управляють роботою зчитувачів карт. Зчитувачі карт з'єднуються з контролерами, які обробляють дані і передають їх на сервер або комп'ютер для подальшого аналізу та зберігання.

Кожного дня працівники компанії приходять до офісу. Задля точного обліку робочого часу та безпеки використовується система обліку часу. Для компанії система забезпечує безпеку офісу, запобігає доступу сторонніх осіб і логування робочого часу співробітників. Кожному співробітнику видається персональна карта, яку він використовує для входу в офіс. Інформація про час приходу і відходу співробітника передається в систему обліку робочого часу, що дозволяє керівництву бачити точний час роботи кожного співробітника.



Рис. - Безконтактний термінал ZKTeco S300

#### **Технічні характеристики системи:**

Вбудований зчитувач: 125kHz RFID Em-Marine

База RFID карт: 30,000 карт

Відстань зчитування RFID карт: 5-10 см

Пам'ять подій проходів: 50,000 подій

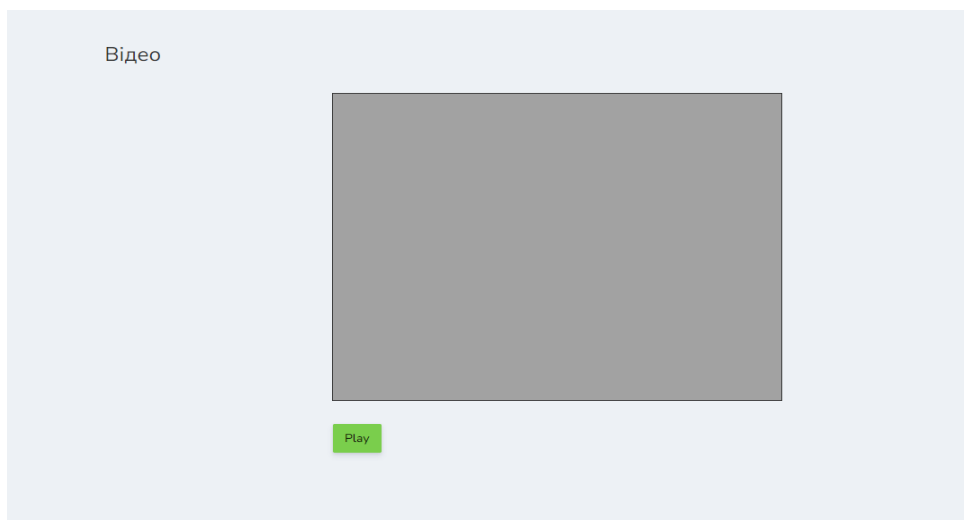
Розміри: 180 x 125 x 55 мм

## 7.2. Алгоритм і реалізація підключення технічних засобів

### *Відео-спостереження*

Для стабільного з'єднання камери підключаються до локальної мережі за допомогою кабелів. Після підключення необхідно налаштувати програмне забезпечення TP-Link для віддаленого моніторингу та інтеграції з основною системою управління. Це дозволяє отримувати відео в режимі реального часу та аналізувати завантаженість співробітників.

При відкритті сторінки відеоспостереження менеджеру необхідно натиснути кнопку "Play", щоб почати відтворення відео.



```

<script>
  document.addEventListener("DOMContentLoaded", () => {
    var but = document.getElementById("but");
    var video = document.getElementById("vid");
    var mediaDevices = navigator.mediaDevices;
    var muted = true;
    but.addEventListener("click", () => {

      // Accessing the user camera and video.
      mediaDevices
        .getUserMedia({
          video: true,
        })
        .then((stream) => {...})
        .catch(alert);
    });

    function stop(e) {
      var stream = video.srcObject;
      var tracks = stream.getTracks();

      for (var i = 0; i < tracks.length; i++) {
        var track = tracks[i];
        track.stop();
      }

      video.srcObject = null;
    }
  });
</script>
<style>
  #container {
    margin: 0px auto;
    width: 500px;
    height: 375px;
    border: 1px #333 solid;
    background-color: #a2a2a2;
  }
  video {
    width: 500px;
    height: 375px;
  }
</style>

```

Рис. – Скрипт реалізації відеоспостереження

Якщо казати про карткову систему обліку часу, то при її використанні впровадження та підключення може виглядати наступним чином. Зчитувачі підключаються до локальної мережі офісу для збереження даних про входи та виходи. Після встановлення зчитувачів необхідно налаштувати програмне забезпечення для управління доступом, яке автоматично зберігатиме дані про присутність співробітників. Інтеграція цієї системи з програмним забезпеченням для управління ресурсозабезпеченням дозволяє вести точний облік робочого часу та оптимізувати управління ресурсами.

## **Розділ 8. Вибір і застосування методів інтелектуального аналізу даних(ІАД) для управління бізнес-процесами**

### 8.1. Опис методів ІАД та обґрунтування доцільності їх використання

ІАД (інтелектуальний аналіз даних) - це галузь, яка поєднує методи машинного навчання, статистики, інформаційної технології та дослідження даних для виявлення прихованих закономірностей, витягування корисної інформації та зроблення прогнозів на основі великого обсягу даних. ІАД використовується в багатьох сферах, включаючи бізнес, медицину, фінанси, маркетинг та багато інших.

IBM SPSS (Statistical Package for the Social Sciences) є одним з провідних програмних забезпечень для статистичного аналізу даних та інтелектуального аналізу даних (ІАД). Він надає широкий спектр методів для обробки та аналізу даних. Нижче описані деякі з них та обґрунтування доцільності їх використання.

**Дескриптивна статистика:** SPSS дозволяє проводити різноманітний дескриптивний аналіз даних, такий як обчислення середнього, медіани, моди, дисперсії, кореляції та інших статистичних характеристик. Це допомагає отримати загальну картину про вхідні дані та їх розподіл.

**Регресійний аналіз:** SPSS надає можливості для проведення регресійного аналізу, що дозволяє виявити залежність між залежною та незалежними змінними. Цей метод корисний для прогнозування та встановлення статистичних зв'язків між змінними.

**Кластерний аналіз:** SPSS дозволяє виконувати кластерний аналіз для групування схожих об'єктів разом на основі схожості їх характеристик. Це дозволяє виявити групи аналогічних об'єктів або популяцій.

**Факторний аналіз:** SPSS має функціонал для проведення факторного аналізу, який допомагає виявити основні фактори або конструкти, що пояснюють варіативність у

наборі змінних. Це дозволяє зменшити розмірність даних та зрозуміти основні фактори, які впливають на спостережувані явища.

У нашому випадку, ми будемо використовувати кластерний аналіз для управління бізнес-процесами. Кластерний аналіз дозволяє групувати схожі об'єкти разом на основі схожості їх характеристик, що є корисним для виявлення груп аналогічних процесів або ресурсів у компанії.

Метод "k-means" є одним з найпопулярніших алгоритмів кластеризації. Він розділяє набір даних на  $k$  кластерів, де  $k$  - це заздалегідь визначена кількість. Алгоритм намагається мінімізувати внутрішньокластерну варіацію шляхом знаходження оптимальних центроїдів для кожного кластеру. "k-means" є швидким і простим у реалізації, але вимагає заздалегідь відомої кількості кластерів.

Метод "two-step" є альтернативою для "k-means" і використовує два кроки для кластеризації. Перший крок включає обчислення ієрархічної кластерної структури, де кожен вузол представляє окремий кластер. Другий крок полягає у поступовому об'єднанні близьких кластерів на основі певного критерію схожості. "two-step" може бути корисним для даних зі складною структурою, оскільки він може виявити кластери різної форми та розміру.

Мережа Кохонена, також відома як нейронна мережа самоорганізації, є іншим методом кластеризації, який використовується для навчання без вчителя. Мережа Кохонена здатна виявити складні залежності та структури в даних. Вона створює мережу нейронів, де кожен нейрон представляє потенційний центроїд кластеру. Під час навчання мережа виробляє сигнали відгуку, що вказують на найбільш подібний центроїд для кожного вхідного зразка.

## 8.2. Представлення результатів застосування обраного методу

Аналіз ефективності розподілу ресурсів для проектів

Для правильної роботи програми було додано більше даних.

	 VAR00 001	 VAR00 002	 VAR00 003	 VAR00 004	 VAR00 005	 VAR00 006
1	1,00	2,00	10000,00	20,00	30,00	85,00
2	2,00	3,00	25000,00	25,00	75,00	70,00
3	3,00	3,00	15000,00	15,00	60,00	80,00
4	4,00	1,00	7000,00	10,00	80,00	75,00
5	5,00	2,00	10000,00	20,00	30,00	85,00
6	6,00	3,00	25000,00	25,00	75,00	70,00
7	7,00	3,00	15000,00	15,00	60,00	80,00
8	8,00	1,00	7000,00	10,00	80,00	75,00
9	9,00	2,00	10000,00	20,00	30,00	85,00
10	10,00	3,00	25000,00	25,00	75,00	70,00
11	11,00	3,00	15000,00	15,00	60,00	80,00
12	12,00	1,00	7000,00	10,00	80,00	75,00

Рис. 8.1 – Документ з послугами для аналізу

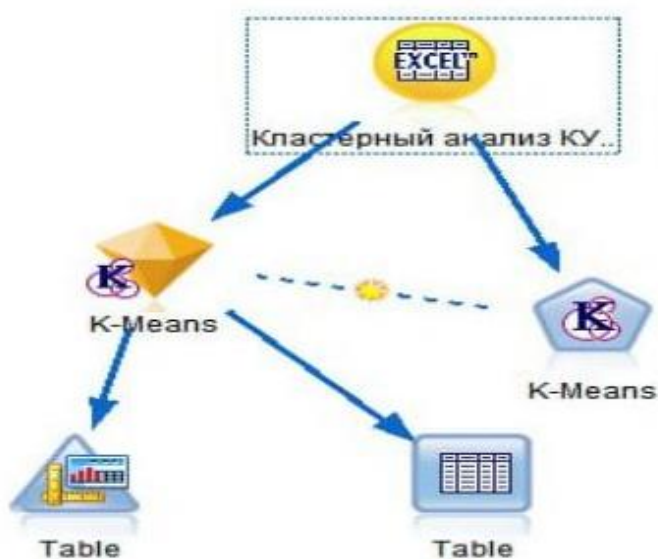


Рис. 8.2 – Схема моделювання аналізу для методу K-means (k-середніх)

Для демонстрації результату обрано табличний метод (блок table).

Метод K-means не може автоматично визначити кількість кластерів, обираємо їх

кількість самі:

- Проекти з низькими витратами та високою ефективністю
- Проекти з високими витратами та низькою ефективністю
- Проекти з середніми витратами та середньою ефективністю
- Проекти з різним співвідношенням витрат та ефективності

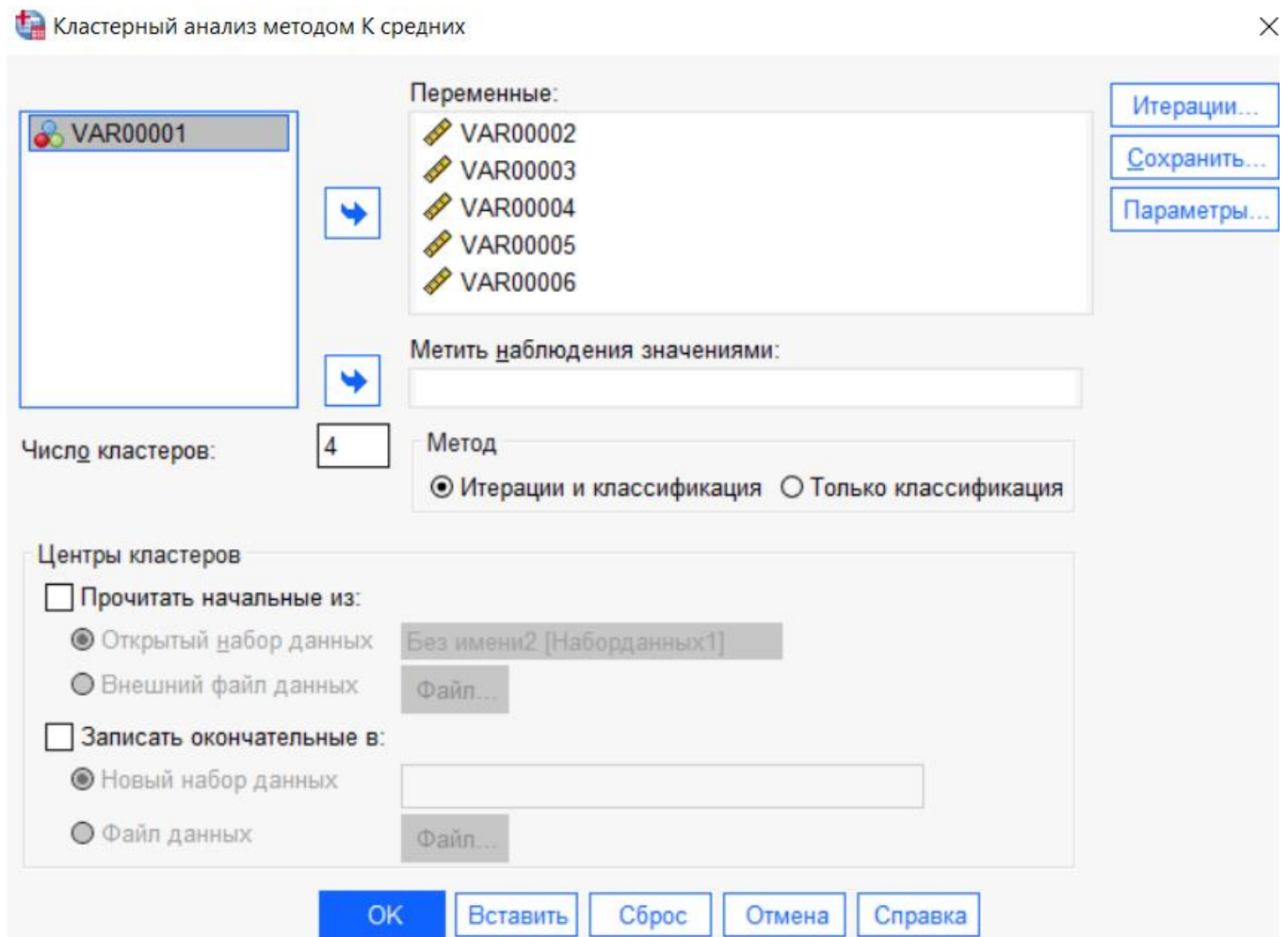


Рис. 8.3 – Вікно налаштувань блоку K-Means

Це кластеризація допомагає виявити групи проектів, які мають схожі характеристики, та дозволяє компанії оптимізувати використання ресурсів, покращити управління проектами та підвищити загальну ефективність діяльності.

### 8.3. Розробка концепції алгоритму для автоматизації підтримки прийняття рішень

Розробка концепції алгоритму для автоматизації підтримки прийняття рішень - це процес визначення загальної ідеї та підходу до створення алгоритму, який забезпечує автоматизовану підтримку в процесі прийняття рішень. Це включає в себе визначення критеріїв, які будуть використовуватися для оцінки рішень, аналіз даних, розробку математичних моделей та визначення логіки алгоритму.

Використання кластерного аналізу, зокрема методів "k-means", "two-step" або "Kohonen network", може надати наступні переваги:

Групування даних: Кластерний аналіз дозволяє групувати схожі дані разом у відповідні кластери. Це допомагає зрозуміти структуру та організацію даних, виявити схожі зразки або групи об'єктів.

Виявлення патернів: Кластерний аналіз може допомогти виявити патерни або характеристики, які спільні для об'єктів у межах одного кластера. Це може допомогти виявити внутрішні зв'язки та залежності між об'єктами даних.

Спрощення складних даних: Кластерний аналіз може допомогти скоротити складність даних, розбиваючи їх на більш прості групи або кластери. Це полегшує подальший аналіз та розуміння даних.

Візуалізація даних: Кластерний аналіз може бути використаний для візуалізації даних. Результати кластеризації можуть бути відображені у вигляді графіків або діаграм, що допомагає легше сприймати та аналізувати дані.

Прийняття рішень: Кластерний аналіз може бути використаний для підтримки прийняття рішень. Наприклад, він може допомогти ідентифікувати групи клієнтів або покупців зі схожими характеристиками, що може вплинути на стратегію маркетингу або асортимент товарів.

k-means: Цей метод є одним з найпоширеніших і простих у реалізації методів кластерного аналізу. Він розбиває дані на k кластерів шляхом мінімізації суми квадратів відстаней між об'єктами та центрами їхніх кластерів. K-means ефективно працює з числовими даними, зручний у використанні та має широкий спектр застосувань.

two-step: Цей метод поєднує групування даних з використанням ієрархічного кластерного аналізу та методів центроїдів. Спочатку він використовує ієрархічний аналіз для визначення початкових кластерів, а потім застосовує метод центроїдів для поліпшення цих кластерів. Two-step може бути корисним у випадках, коли кластери не мають сферичної форми або коли дані мають велику кількість шуму.

Kohonen network: Цей метод базується на нейромережах і використовується для безнаглядного навчання та виявлення патернів у даних. Він може використовуватися для визначення топології кластерів без заздалегідь відомих кількостей кластерів. Кохоненські мережі корисні для виявлення складних візуальних патернів та мають застосування в обробці зображень, розпізнаванні мови та інших сферах.

## **РОЗДІЛ 9. ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **Ціль розробки ПЗ**

Автоматизація управління ресурсозабезпеченням при розробці ІТ-продукту з метою створення зручного менеджера для реалізації ідей клієнтів.

Задачі, реалізація яких допоможе досягнути поставленої цілі:

- Уникнення втрат інформації;
- Реалізація швидкого створення документів, необхідних для протікання бізнес-процесів;
- Контроль протікання процесів;
- Збереження та обробка даних.

Впровадження автоматизованої системи управління ресурсозабезпеченням призведе до змін основних показників діяльності підприємства. До впровадження система приносить приблизно 1 200 000 грн/рік, адміністративні витрати становлять 100 000 грн, інші витрати – 15 000 грн. Після впровадження автоматизації прогнозується зменшення адміністративних витрат на 20%, інших витрат на 10%, збільшення

прибутку на 15%.

### 9.1. Розрахунок інвестиційних витрат на модернізацію системи автоматизації

Капітальні інвестиції по базовому варіанті включають:

- Вартість техніки за оптовими цінами – 250 000 грн Цопт;
- Витрати на супутні товари – 18 000 грн Цт;
- Заготівельно-складські видатки – 20 000 грн Цз-з;
- Витрати на монтаж устаткування – 30 000 грн Цм;
- Транспортні витрати – 12 000 грн Цтв.

Розрахунок інвестиційних витрат на модернізацію системи автоматизації:

- Потужність комп'ютера - 100 Вт/год (Р);
- Тариф на електроенергію - 6 грн за 1кВт/год (Цее);
- Час необхідний на розробку ПЗ - 200 год (Т);
- Заробітна плата програміста РНР - 300 грн/год (Зг);

Розрахунок витрат на заробітну плату РНР-програмісту:

$$B_{зп} = Zг * Т + Зг * Т * Н_{есв} = Zг * Т * (1 + Н_{есв}) = 300 * 200 * 1,22 = 73200 \text{ грн}$$

де  $N_{есв}$  – єдиний соціальний внесок на з/п РНР-програміста, 22%.

Розрахунок витрат за електроенергію:

$$B_{ее} = P * C_{ее} * T = 0,1 * 6 * 200 = 120 \text{ грн}$$

де  $B_{ее}$  – витрати на електроенергію;

$P$  – потужність комп'ютера;

$C_{ее}$  – ціна 1 кВт електроенергії за годину;

$T$  – час праці комп'ютера.

Розрахунок вартості розробки ПЗ:

$$I C_p = (B_{зп} + B_p + B_{ее}) + (B_{зп} + B_p + B_{ее}) * ПДВ =$$

$$= (B_{\text{зп}} + B_p + B_{\text{се}}) * (1 + \text{ПДВ}) = (73200 + 150 + 120) * 1,2 = 88740 \text{ грн}$$

ПДВ – податок на додану вартість, 20%.

Визначаємо вартість комплексу засобів автоматизації:

Сканер штрих-коду MS-3100: 899 грн (без ПДВ) (Цск).

Камера відео-спостереження Hikvision DS-2CE56D0T-IRMMF: 1038 грн  
(без ПДВ) (Цвк).

$$Ц_{\text{за}} = (Ц_{\text{ск}} + Ц_{\text{вк}}) + (Ц_{\text{ск}} + Ц_{\text{вк}}) * \text{ПДВ} = (Ц_{\text{ск}} + Ц_{\text{вк}}) * (1 + \text{ПДВ}) =$$

$$= (899 + 1038) * 1,2 = 2324,4 \text{ грн}$$

де  $Ц_{\text{за}}$  – ціна засобів автоматизації;

**Інвестиційні витрати на модернізацію системи управління  
технологічним процесом**

№	Статті витрат	Сума, грн.
1	Вартість комплексів засобів автоматизації	250 000
2	Витрати для виконання робіт з проектування	1 267 286,8
3	Витрати для модернізації системи управління	88 740
4	Всього інвестиційні витрати на модернізацію системи управління процесом ( $\square IC_n$ )	1 606 026,8

Розрахунок інвестицій на комплекс технічних засобів:

$$IC_{ктз} = (Цза + Цза * K_m + Цза * H_T + 100 + Цза * H_{тр}) * 100 = (3240 + 3240 * 0,25 + 3240 * 32,5 + 3240 * 1,2) * 100 = 1587286,8 \text{ грн}$$

де  $K_m$  – коефіцієнт відрахувань на монтажні роботи, 0,1%;  $H_T$  – норматив відрахувань на тару та упакування, 0,25%;  $H_{тр}$  – норматив відрахувань на транспортні витрати, 5%;  
 $H_{зс}$  – норматив відрахувань на заготівельно-складські витрати, 1,2%.

Розрахунок загальних інвестиційних витрат:

$$IC_3 = IC_p + IC_{ктз} = 88740 + 1587286,8 = 1676026,8 \text{ грн}$$

Визначимо величину амортизаційних відрахувань :

$$\Delta A = IC_3 * H_{aIVгр} = 1676026,8 * 0,5 = 838013,4 \text{ грн}$$

де  $H_{aIVгр}$  – річна норма амортизаційних відрахувань для IV групи основних фондів.

**Розрахунок змін основних показників діяльності підприємства**

Зменшення адміністративних витрат на 20%:

$$\Delta B_{адм} = 100000 * (-0,20) = -20000 \text{ грн}$$

Зменшення інших витрат на 10%:

$$\Delta B_{інш} = 15000 * (-0,10) = -1500 \text{ грн}$$

Збільшення прибутку на 15 % за рахунок впровадження автоматизації:

$$\Delta П_1 = 1200000 * 0,15 = 180000 \text{ грн}$$

Розрахунок річної зміни основних показників діяльності за рахунок впровадження ПЗ:

$$\Delta C = \Delta B_{адм} + \Delta B_{інш} + \Delta A = -20000 - 1500 + 838013,4 = 816513,4 \text{ грн}$$

Розрахунок приросту прибутку підприємства за рахунок впровадження ПЗ:

$$\Delta П = \Delta C - \Delta П_1 = 816513,4 - 180000 = 636513,4 \text{ грн}$$

Розрахунок приросту чистого прибутку:

$$\Delta\text{ЧП} = (\Delta\text{П} - \Delta\text{А}) * (1 - 0,18) = (636513,4 - 838013,4) * 0,82 =$$
$$164786,15 \text{ грн}$$

Розрахунок чистого грошового потоку:

$$\text{ЧГП} = \Delta\text{ЧП} + \Delta\text{А} = 164786,15 + 838013,4 = 1002799,55 \text{ грн}$$

Інвестиції в проект доцільні, оскільки приріст чистого прибутку після впровадження автоматизації складає 164 786,15 грн, чистий грошовий потік більше за нуль та складає 1 002 799,55 грн..

*Показники виробничої діяльності підприємства:*

<b>№</b>	<b>Показник</b>	<b>Позначення</b>	<b>Базова</b>	<b>Модернізована</b>	<b>Зміна статей витрат (+) зростання; (-) зниження</b>
1	Обсяг реалізації продуктів та завдань, нат. од. (шт)	ОВ	5000	6500	+1500
2	Дохід від реалізації, тис. грн	ТП	2 400	3 000	+600
3	Повна собівартість продукту, тис. грн	С	600	480	-120
4	Прибуток, тис. грн	П	1 200	1 380	+180
5	Чистий прибуток, тис. грн	ЧП	960	1 165	+205
6	Рентабельність %	Рп	50	58,25	+8,25

## ВИСНОВОК

У результаті виконання дипломної роботи на тему "Автоматизація управління ресурсозабезпеченням при розробці ІТ-продукту" було проведено детальне дослідження та розробка системи автоматизації, спрямованої на оптимізацію процесу управління ресурсами в компанії Real Solutions.

Під час дослідження було проведено аналіз поточних процесів управління ресурсами в компанії та виявлено існуючі проблеми, пов'язані з неефективним розподілом ресурсів, неточностями в обліку та затримками у комунікаціях між відділами. На основі цього аналізу були сформульовані основні вимоги до системи автоматизації.

Для вирішення виявлених проблем була розроблена та впроваджена система, яка включає модуль планування ресурсів, систему автоматизованого контролю за використанням ресурсів та модуль аналітики. Ця система дозволяє автоматизувати процеси розподілу та обліку ресурсів, точно відстежувати їх рух та забезпечувати своєчасну інформацію для прийняття управлінських рішень.

В результаті виконання дипломної роботи було показано, що впровадження розробленої системи автоматизації може суттєво підвищити ефективність управління ресурсами в ІТ-компанії. Вона дозволяє скоротити час, необхідний на обробку даних та прийняття рішень, підвищити точність обліку ресурсів та забезпечити кращий контроль за процесом управління.

Результати дослідження можуть бути корисні для ІТ-компаній, що прагнуть вдосконалити процеси управління ресурсами та підвищити їх ефективність.

Загалом, дипломна робота відзначається високою якістю виконання, систематичним підходом до дослідження та глибоким аналізом проблеми. Рекомендується до захисту для отримання кваліфікації бакалавра.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «Методичні вказівки до виконання курсових робіт за дисципліною «Основи Керування Бізнес-Процесами»» - О.М. Жигайло, Д.В. Дец - Одеса: ОНАХТ, 2018, 19 с
2. «Конспект лекцій з курсу «Основи Керування Бізнес-Процесами»» - О.М. Жигайло, Д.В. Дец - Одеса: ОНАХТ, 2018, 99 с
3. «Методичні вказівки до виконання лабораторних робіт за дисципліною «Основи Керування Бізнес-Процесами»» - О.М. Жигайло, Д.В. Дец - Одеса: ОНАХТ, 2018, 59 с
4. «Методичні вказівки до виконання лабораторних робіт за дисципліною «Програмні Засоби Автоматизації Керування Бізнес-Процесами»» - Д.В. Дец - Одеса: ОНАХТ, 2018, 59 с
5. «Методичні вказівки до виконання кваліфікаційної роботи бакалавра» - Хобін В.А., доц. Левінський В.М., доц. Скаковський Ю.М., доц. Степанов М.Т - Одеса: ОНТУ, 2022, 35 с
6. Автоматизація управління проектами [Електронний ресурс]. – Режим доступу. – <https://kamala-soft.com/uk/avtomatizatsiya-biznesa/avtomatizatsiya-upravleniya-proektami/>
7. Посібник з javascript [Електронний ресурс]. – Режим доступу. – <https://uk.javascript.info/>
8. Що таке PHP [Електронний ресурс]. – Режим доступу. – <https://w3schoolsua.github.io/php/index.html#gsc.tab=0>
9. Посібник з CSS – Режим доступу. – <https://w3schoolsua.github.io/css/index.html#gsc.tab=0>
10. Система обліку робочого часу [Електронний ресурс]. – Режим доступу. – <https://smartel.ua/sistemy-po-beskontaktnoy-karte/>
11. Система відеоспостереження [Електронний ресурс]. – Режим доступу. <https://smartel.ua/ua/videonablyudenie-ofisa/>
12. Бази Даних та SQL – Режим доступу. – <https://acode.com.ua/sql-intro/>