



ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»
Факультет автоматизації виробництва та цифрових технологій
Кафедра цифрових технологій та проєктно-аналітичних рішень

«Допущено до захисту»
Гарант ОПП

Павло САГАЙДА

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістра

за підсумками виконання
освітньо-професійної програми
«Комп'ютерні науки та цифровий інтелект»
за спеціальністю 122 Комп'ютерні науки

**на тему «Дослідження моделей, методів та інформаційних технологій
підтримки керування дільницею з ремонту електроустаткування»**

Керівник роботи

Олександр КОСТИКОВ

Консультант від
бази практики

Євген ЛІТВІНЕНКО

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

Здобувач

Дмитро КЛАП

Підсумкова оцінка за атестацію			
--------------------------------	--	--	--

Голова ЕК

Антон КУДРЯВЦЕВ

ЗАПОРІЖЖЯ 2026



ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»

Факультет цифрових технологій та автоматизації виробництва

Кафедра цифрових технологій та проєктно-аналітичних рішень

Ступень вищої освіти магістр

Спеціальність 122/F3 Комп'ютерні науки

(шифр і назва)

Освітня програма Комп'ютерні науки та цифровий інтелект

(назва)

ЗАТВЕРДЖУЮ

Гарант П.І. Сагайда

_____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Клап Дмитру Сергійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема роботи Дослідження моделей, методів та інформаційних технологій підтримки керування дільницею з ремонту електроустаткування

керівник роботи Костіков Олександр Анатолійович, доцент,

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

затверджені наказом Університету від 10.09.2025 р. № 239

2. Строк подання роботи 20 січня 2026

3. Вихідні дані до роботи Навчальна та спеціалізована технічна література; державні та міжнародні стандарти у галузі комп'ютерних наук та енергетики; методична література з проєктування інформаційних систем; науково-дослідницькі роботи з тематики автоматизації обробки даних, методів цифрового інтелекту (Machine Learning); технологічні інструкції ЦРУ ПРАТ «КАМЕТ-СТАЛЬ» з ремонту та експлуатації електроустаткування; статистичні дані щодо відмов та ремонтних циклів за 2024–2025 роки.

4. Зміст пояснювальної записки (перелік питань) Виконати аналіз моделей, методів та алгоритмів використання для управління промисловими ремонтними дільницями; провести аналіз функціональності існуючих систем класу EAM/ERP та використовуваних інформаційних технологій.

Сформулювати мету і завдання дослідження, виходячи з потреб автоматизації ЦРУ. Представити результати дослідження та впровадження роботи в практику, включаючи тестування системи на реальних даних та аналіз стійкості до виробничих інцидентів. Зробити висновки з ефективності використання розробленого ПМК з наведенням кількісних показників досягнутих результатів. Визначити позитивний економічний ефект від впровадження розробленого ПМК та обґрунтувати підвищення ефективності діяльності у предметній області.

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

Схема мети та завдань дослідження. Математичні моделі (сіткова модель СРМ, модель лінійного програмування) та методика дослідження. Графіки та діаграми результатів експериментальних досліджень (порівняння точності планування, завантаженості персоналу). Економічні показники ефективності проєкту (NPV, ROI).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Фаховий консультант</i>	О.А. Костіков, доц.		
<i>Нормоконтроль</i>	О.С. Касьянюк, ст. викл.		
<i>Економічна частина</i>	І.А. Гетьман, доц., канд. техн. наук		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи
1.	Аналіз предметної області (дільниці ЦРУ ПРАТ «КАМЕТ-СТАЛЬ»), огляд існуючих ІТ-рішень (EAM/ERP) та методів автоматизації TOiP.	29.12 - 02.01
2.	Розробка математичної моделі (СРМ, лінійне програмування) та методики дослідження на базі нечіткої логіки.	02.01 - 06.01
3.	Проектування архітектури та програмна реалізація ПМК на Python та SQL Server.	06.01 - 08.01
4.	Проведення експериментальних досліджень ефективності моделей, тестування ПМК та верифікація результатів.	08.01 - 09.01
5.	Виконання економічних розрахунків доцільності розробки та впровадження системи (NPV, ROI).	09.01 - 13.01
6.	Оформлення текстової, програмної та графічної документації.	13.01 - 16.01
7.	Оформлення записки. Підготовка доповіді та презентації. Отримання відгуку та рецензії.	16.01 - 20.01
8.	Захист проекту	20.01-29.01

Здобувач вищої освіти

(підпис)

Д.С. Клап

(ініціали та прізвище)

Керівник роботи, доц.,

(підпис)

О.А. Костіков

(ініціали та прізвище)

РЕФЕРАТ

Клап Д.С. Дослідження моделей, методів та інформаційних технологій підтримки керування дільницею з ремонту електроустаткування.

Кваліфікаційна робота на здобуття освітнього ступеня магістра за спеціальністю 122 Комп'ютерні науки. ОПП «Комп'ютерні науки та цифровий інтелект» - ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ МЕТІНВЕСТ ПОЛІТЕХНІКА», Запоріжжя, 2026.

Мета роботи: Підвищити ефективність керування дільницею ремонту електроустаткування шляхом дослідження предметної області та розробки методів інформаційної підтримки, включаючи математичні моделі, алгоритми та програмні компоненти для автоматизації процесів планування, обліку та аналізу даних.

Об'єктом дослідження є процеси керування дільницею ремонту електроустаткування на промисловому підприємстві.

Предмет дослідження. Методи, моделі та інформаційні технології для автоматизованої обробки даних про несправності, ресурси та виконання ремонтів.

В першому розділі розглянуто аналіз стану питання та концепцій з проблеми керування дільницею з ремонту електроустаткування.

В другому розділі проведено розробку математичної моделі об'єкта та методики дослідження.

В третьому розділі розроблені засоби моделювання предметної області.

В четвертому розділі проведення та результати теоретичних та експериментальних досліджень заданого об'єкта.

В п'ятому розділі зроблені економічні розрахунки

ABSTRACT

Klap D.S. Research of models, methods, and information technologies for decision support in electrical equipment repair section management.

The qualification work is submitted for a Master's degree in the specialty 122 "Computer Science". Educational-professional program "Computer Science and Digital Intelligence" – LLC "TECHNICAL UNIVERSITY METINVEST POLYTECHNIC", Zaporizhzhia, 2026.

The aim of the work: To increase the management efficiency of the electrical equipment repair section by investigating the subject area and developing information support methods, including mathematical models, algorithms, and software components for the automation of planning, accounting, and data analysis processes.

The object of the study is the management processes of an electrical equipment repair section at an industrial enterprise.

The subject of the study involves methods, models, and information technologies for automated processing of data regarding malfunctions, resources, and repair execution.

In the first section, an analysis of the state of the art and concepts regarding the problem of managing an electrical equipment repair section is carried out.

In the second section, the development of the mathematical model of the object and the research methodology is performed.

In the third section, modeling tools for the subject area are developed.

In the fourth section, the execution and results of theoretical and experimental studies of the specified object are presented.

In the fifth section, economic calculations are provide

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	9
1. АНАЛІЗ СТАНУ ПИТАННЯ ТА КОНЦЕПЦІЙ З ПРОБЛЕМИ КЕРУВАННЯ ДІЛЬНИЦЕЮ З РЕМОНТУ ЕЛЕКТРОУСТАТКУВАННЯ	11
1.1. Аналіз предметної області «Ремонт електроустаткування».....	11
1.1.1. Загальна характеристика дільниці ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ»	11
1.1.2. Аналіз технічного стану електрообладнання ПРАТ «КАМЕТ-СТАЛЬ» та статистика відмов за 2024–2025 рр.	13
1.1.3. Опис основних учасників та їх ролей	15
1.2. Формалізація вимог та моделювання бізнес-процесів	16
1.2.1. Методи збору та аналізу бізнес-вимог до програмно-методичного комплексу (ПМК).....	16
1.2.2. Проведення інтерв'ю з ключовими стейкхолдерами	16
1.2.3. Анкетування технічного персоналу.....	17
1.2.4. Протоколювання наради та візуалізація вимог.....	18
1.2.5. Розробка бізнес-моделі програмного рішення з використанням Lean Canvas.....	20
1.2.6. Моделювання бізнес-процесу "Управління дільницею" у нотаціях BPMN та DFD.....	22
1.2.7. Формалізація вимог до програмного забезпечення у форматі User Stories та Story Mapping	25
1.2.8. Деталізація користувацьких історій (User Stories)	26
1.2.9. Побудова та аналіз карти історії (Story Mapping).....	27
1.3. Огляд методів математичного моделювання та засобів розробки.....	29
1.3.1. Аналіз математичних методів оптимізації процесів TOIP	29
1.3.2. Обґрунтування вибору інструментальних засобів розробки.....	30

	6
1.4. Аналіз існуючих ІТ-рішень для автоматизації ремонтних служб	31
1.4.1. Аналіз корпоративних систем (SAP ERP, IBM Maximo).....	31
1.4.2. Аналіз нішевих та хмарних рішень (UpKeep, Fiix).....	32
1.4.3. Порівняльна характеристика та обґрунтування власної розробки	32
1.5. Висновки до першого розділу	34
2. РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ОБ'ЄКТА ТА МЕТОДИКИ ДОСЛІДЖЕННЯ.....	36
2.1 Обґрунтування вибору методів теоретичних та експериментальних досліджень.....	36
2.2 Математична модель на основі теорії графів та методу критичного шляху (CPM)	38
2.2.1. Формалізація етапів робіт	39
2.2.2. Розрахунок критичного шляху	39
2.3. Модель оптимізації ресурсів засобами лінійного програмування.....	40
2.3.1. Функція цілі	40
2.3.2. Система обмежень	41
2.4. Практична реалізація методу критичного шляху на прикладі капітального ремонту електродвигуна серії AIP	42
2.4.1. Покроковий розрахунок часових параметрів сітки.....	43
2.4.2. Визначення повних резервів та критичного шляху	44
2.5. Висновки до другого розділу	46
3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ.....	48
3.1 Архітектура та вибір технологічного стека	48
3.1.1. Структурна схема та рівні архітектури ПМК.....	49
3.1.2. Обґрунтування вибору мови програмування Python 3.12	49
3.1.3. Вибір системи управління базами даних (СУБД)	50
3.2 Розробка бази даних (Data Architecture)	51
3.2.1. Фізична реалізація в SQL Server.....	53
3.3. Модуль інтеграції з базою даних	55

	7
3.3.1. Реалізація математичного ядра	58
3.3.2. Архітектура модуля оптимізації	59
3.3.3. Реалізація задачі лінійного програмування за допомогою PuLP	60
3.3.4. Алгоритмічна реалізація методу CPM	60
3.3.5. Обробка помилок та взаємодія з SQL Server	61
3.4. Інтерфейс користувача (UI/UX)	62
3.4.1. Взаємодія компонентів (Sequence Diagram)	64
4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ ТА АНАЛІЗ ЕФЕКТИВНОСТІ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ	65
4.1. Апробація програмного комплексу в умовах ПРАТ «КАМЕТ-СТАЛЬ»..	65
4.2. Аналіз точності та ефективності моделей і алгоритмів	65
4.2.1. Методика проведення порівняльного експерименту	66
4.2.2. Аналіз точності прогнозування (CPM проти ручного оцінювання)	67
4.2.3. Результати оптимізації ресурсів засобами лінійного програмування	68
4.2.4. Вплив модуля нечіткої логіки на стійкість плану	69
4.3. Верифікація програмного забезпечення та аналіз стійкості до інцидентів	69
4.3.1. Методологія та сценарії тестування ПМК	70
4.3.2. Аналіз стійкості системи до виробничих інцидентів	71
4.3.3. Суб'єктивна оцінка зручності інтерфейсу (UX-аналіз)	71
4.4. Обґрунтування економічної доцільності впровадження	72
4.5. Рекомендації щодо експлуатації ПМК	73
5. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ТА РОЗРАХУНОК ЕФЕКТИВНОСТІ ПРОЄКТУ	74
5.1. Розрахунок витрат на розробку програмно-методичного комплексу....	74
5.1.1. Розрахунок заробітної плати та відрахувань	74
5.1.2. Розрахунок витрат на технічне забезпечення та ПЗ	75
5.2. Розрахунок експлуатаційних витрат та економічного ефекту	76
5.2.1. Розрахунок річної економії	76

	8
5.3. Оцінка інвестиційної привабливості проекту.....	76
5.3.1. Термін окупності (Payback Period, PP)	77
5.3.2. Рентабельність інвестицій (Return on Investment, ROI).....	77
5.3.3. Чистий приведений дохід (Net Present Value, NPV).....	78
5.4. Висновки до розділу 5	78
ЗАГАЛЬНІ ВИСНОВКИ	80
ПЕРЕЛІК ПОСИЛАНЬ.....	81
Додаток А. ВІДОМОСТІ РОБОТИ	86
ДОДАТОК Б. ТЕХНІЧНЕ ЗАВДАННЯ ДЛЯ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ ПІДТРИМКИ КЕРУВАННЯ ДІЛЬНИЦЕЮ З РЕМОНТУ ЕЛЕКТРОУСТАТКУВАННЯ	87
ДОДАТОК В. АНКЕТА ДЛЯ ЗБОРУ ВИМОГ ТА РЕЗУЛЬТАТИ ОПИТУВАННЯ ПЕРСОНАЛУ ДІЛЬНИЦІ ЦРУ ПРАТ «КАМЕТ-СТАЛЬ».....	93
ДОДАТОК Г. ЛІСТИНГИ SQL-СКРИПТІВ СТВОРЕННЯ СТРУКТУРИ БАЗИ ДАНИХ ПМК.....	96
ДОДАТОК Д. ГРАФІЧНІ МАТЕРІАЛИ.....	100
ДОДАТОК Е. ЛІСТИНГИ ПРОГРАМНОГО КОДУ МОДУЛІВ МАТЕМАТИЧНОГО ЯДРА ПМК	102

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ЦРУ – Цех ремонту устаткування

ПРАТ – Приватне акціонерне товариство

ПМК – Програмно-методичний комплекс

ПЗ – Програмне забезпечення

UML – Unified Modeling Language

ER – Entity-Relationship

CPM – Critical Path Method

ROI – Return on Investment

NPV – Net Present Value

SQL – Structured Query Language

MATLAB – Matrix Laboratory

ML – Machine Learning

FIS – Fuzzy Inference System

CMMS – Computerized Maintenance Management System

EAM – Enterprise Asset Management

НДРС – Науково-дослідна робота студента

ВСТУП

Сучасні умови функціонування промислових підприємств України, що характеризуються енергетичною кризою, військовими викликами та необхідністю підвищення стійкості виробничих процесів, вимагають негайної автоматизації критично важливих ділянок. Особливої актуальності набуває впровадження ефективних систем інформаційної підтримки управління на ремонтних ділянках, зокрема на ПРАТ «КАМЕТ-СТАЛЬ», де ручний облік та відсутність оперативного доступу до даних призводять до значних простоїв обладнання та зниження загальної ефективності виробництва. Розробка програмно-методичного комплексу, що використовує сучасні методи математичного моделювання та інтелектуального аналізу даних, є критично важливою для оптимізації процесів планування, контролю та обліку виробничих завдань.

Мета роботи.

Метою кваліфікаційної роботи є розробка програмного комплексу, який автоматизує управління виробничими процесами на ділянці з ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ», включаючи планування завдань, розподіл ресурсів, моніторинг виконання та генерацію звітів, з інтеграцією методів математичного моделювання (графова модель, оптимізація).

Завдання роботи.

Для досягнення поставленої мети визначено наступні завдання:

1. Провести аналіз предметної області та формалізувати бізнес-вимоги до системи управління ділянкою ремонту електроустаткування, включаючи використання методів збору вимог (анкетування, інтерв'ю) та моделювання бізнес-процесів у нотаціях BPMN та DFD.

2. Розробити математичні моделі для оптимізації процесів планування та розподілу ресурсів, включаючи використання теорії графів (метод критичного шляху CPM), лінійного програмування.

3. Спроекувати архітектуру програмно-методичного комплексу, розробити логічну та фізичну моделі бази даних (ERD, SQL-структура) та реалізувати інтерфейс користувача з використанням сучасних засобів розробки (Python, SQL Server, Tkinter, MATLAB).

4. Здійснити експериментальні дослідження та проаналізувати результати впровадження програмного комплексу для підтвердження його ефективності та економічної доцільності.

Об'єкт дослідження.

Об'єктом дослідження є процеси управління ділянкою з ремонту електроустаткування ПРАТ «Камет-Сталь».

Предмет дослідження.

Предметом дослідження є методи та засоби інформаційної підтримки керування зазначеними процесами, включаючи програмний комплекс та математичні моделі.

Практичне значення роботи полягає у впровадженні програмного комплексу, який забезпечує підвищення точності прогнозування на 15-20% порівняно з класичними методами, що підтверджується економічними розрахунками (ROI 100%, термін окупності 1 рік).

1. АНАЛІЗ СТАНУ ПИТАННЯ ТА КОНЦЕПЦІЙ З ПРОБЛЕМИ КЕРУВАННЯ ДІЛЬНИЦЕЮ З РЕМОНТУ ЕЛЕКТРОУСТАТКУВАННЯ

1.1. Аналіз предметної області «Ремонт електроустаткування»

1.1.1. Загальна характеристика ділянки ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ»

ПРАТ «КАМЕТ-СТАЛЬ» (до перейменування — Дніпровський металургійний комбінат) є одним із найбільших металургійних підприємств України, розташованим у місті Кам'янське Дніпропетровської області. Підприємство спеціалізується на виробництві чавуну, сталі, прокату та є важливою складовою міжнародної гірничо-металургійної групи «Метінвест». В умовах повномасштабного вторгнення та постійних атак на енергетичну інфраструктуру України (2024–2026 рр.), підприємство працює з підвищеними ризиками. Критично важливим стає забезпечення безперебійної роботи основного та допоміжного обладнання. Будь-який непередбачуваний простій виробничих агрегатів через поломки електроустаткування призводить до значних економічних втрат.

Ділянка з ремонту електроустаткування (ЦРУ — цех ремонту устаткування) є ключовою службою, що відповідає за життєздатність виробництва. Вона обслуговує парк з понад 500 одиниць різного електрообладнання, включаючи:

- високовольтні електродвигуни потужністю від 100 кВт до 500 кВт;
- трансформаторні підстанції;
- системи автоматики та кабельні мережі.

Забезпечує безперебійну роботу основного виробничого обладнання. Основні функції дільниці включають планове профілактичне обслуговування, поточні та капітальні ремонти електродвигунів, трансформаторів, кабельних мереж та іншого електрообладнання.

Наразі основними проблемами функціонування дільниці є використання ручного обліку та паперового документообігу, що унеможлиблює оперативний доступ до актуальної інформації про стан обладнання, наявність матеріалів та завантаженість персоналу. Відсутність централізованої системи планування призводить до неефективного розподілу ресурсів, збільшення часу простоїв обладнання та зниження загальної продуктивності.

Аналіз поточної ситуації (до впровадження програмно-методичного комплексу (ПМК)):

На дільниці досі переважають застарілі методи планування та обліку. Заявки на ремонт фіксуються у паперових журналах або передаються телефоном. Це спричиняє такі проблеми:

- відсутність централізованої бази даних: Неможливо оперативно переглянути історію ремонтів конкретного агрегату, що ускладнює діагностику,
- неефективний розподіл часу: За оцінками, майстер дільниці витрачає до 35-40% свого робочого часу на рутинне заповнення звітів та пошук інформації, замість того, щоб керувати процесами,
- затримки в реагуванні: Несвоєчасна фіксація інцидентів призводить до збільшення середнього часу простою обладнання.

Впровадження сучасної системи інформаційної підтримки, описаної в даній роботі, має на меті усунути ці недоліки, підвищити прозорість процесів та забезпечити стійкість виробництва ПРАТ «КАМЕТ-СТАЛЬ».

1.1.2. Аналіз технічного стану електрообладнання ПРАТ «КАМЕТ-СТАЛЬ» та статистика відмов за 2024–2025 рр.

Функціонування металургійного гіганта ПРАТ «КАМЕТ-СТАЛЬ» у сучасних умовах (2024–2026 рр.) характеризується екстремальними навантаженнями на енергетичну інфраструктуру. Дільниця з ремонту електроустаткування (ЦРУ) забезпечує працездатність широкого спектра агрегатів, що перебувають в експлуатації від 10 до 30 і більше років. Високий ступінь зносу основних фондів у поєднанні з агресивним виробничим середовищем (висока запиленість, температурні коливання, наявність парів хімічних речовин) призводить до зростання інтенсивності відмов.

Для обґрунтування необхідності розробки програмно-методичного комплексу (ПМК) було проведено ретроспективний аналіз аварійності електрообладнання на дільниці за період 2024–2025 років. Статистика розподілу відмов за типами обладнання представлена у таблиці 1.1.

Таблиця 1.1. — Статистика відмов електрообладнання на дільниці ПРАТ «КАМЕТ-СТАЛЬ»

Тип обладнання	Частка у парку, %	Частота відмов (од./рік)	Основні причини виходу з ладу	Середній час простою (год)
Низьковольтні електродвигуни (до 100 кВт)	65	120–140	Міжвиткове замикання, знос підшипникових вузлів	8–12
Високовольтні двигуни (понад 100 кВт)	15	15–20	Пробій ізоляції статора, порушення системи охолодження	48–72
Силкові трансформатори	10	5–8	Зволоження оливи, окислення контактів, перевантаження	24–36
Пускорегулювальна апаратура	10	45–55	Обгорання контактів, збої в логіці контролерів	4–6

Аналіз таблиці показує, що найбільша кількість відмов припадає на електродвигуни малої та середньої потужності, проте найбільших економічних збитків підприємству завдають прості високовольтних агрегатів, ремонт яких потребує складного планування та значних ресурсів.

Класифікація факторів, що впливають на надійність електрообладнання:

1. Зовнішні фактори (середовище): На ПРАТ «КАМЕТ-СТАЛЬ» обладнання піддається впливу дрібнодисперсного графітового та залізорудного пилу. Пил, осідаючи на обмотках, погіршує тепловідвід і стає провідником, викликаючи поверхневі розряди та пробої.

2. Режимні фактори: Нестабільність графіків електропостачання у 2025 році призводила до частих пусків та зупинок потужних приводів, що викликає значні пускові струми та термічну деградацію ізоляції.

3. Організаційні фактори: Відсутність оперативної інформаційної підтримки управління ремонтами (ТОiP) призводить до того, що планово-запобіжні роботи часто замінюються ремонтами «за фактом відмови».

Проблематика існуючої системи обліку:

У ході передпроектного обстеження було виявлено, що понад 70% даних про технічний стан агрегатів зберігаються у неструктурованому вигляді (паперові формуляри, журнали змін). Це унеможлиблює використання сучасних методів інтелектуального аналізу даних (Data Mining) для передбачення відмов.

Впровадження розроблюваного ПМК дозволить акумулювати історію ремонтів у базі даних SQL Server, що надалі дасть можливість перейти до стратегії обслуговування «за станом» (Predictive Maintenance), суттєво знижуючи аварійність та витрати на позапланові ремонти.

1.1.3. Опис основних учасників та їх ролей

Ключовими учасниками процесів на дільниці є (Рисунок 1.1):

- керівник дільниці: відповідає за загальне планування робіт, розподіл завдань, контроль виконання та аналіз результатів,
- майстер: здійснює безпосереднє керівництво технічним персоналом, видає завдання, контролює якість виконання робіт,
- робітник (технічний персонал): виконує ремонтні та обслуговуючі роботи, фіксує результати та інциденти,
- система обліку робочого часу: надає дані про відпрацьований час персоналу,
- автоматизована система оповіщення несправностей: система, що автоматично реєструє інциденти та несправності обладнання.

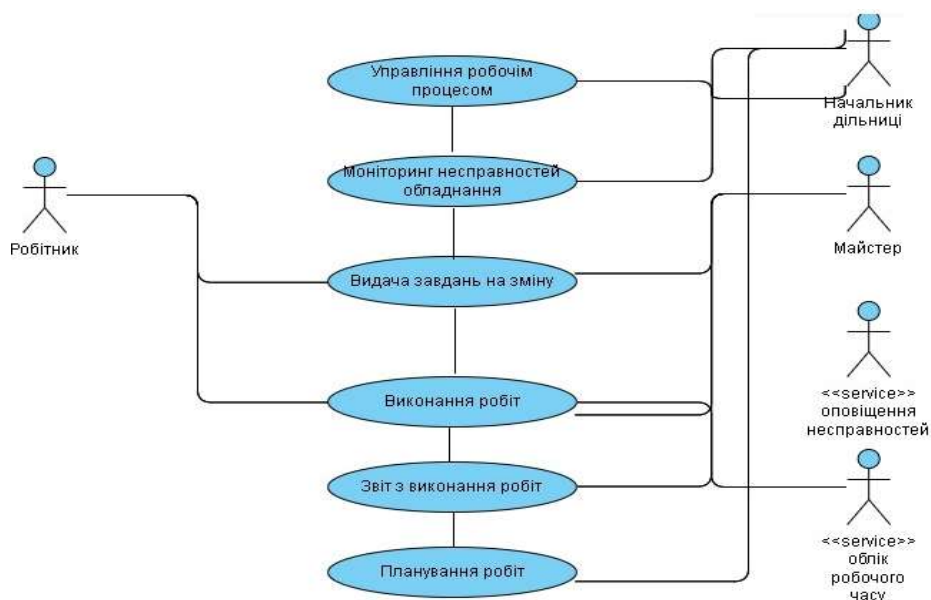


Рисунок 1.1 – Діаграма діяльності: описує послідовність виконання операцій у рамках процесу прийняття рішень.

1.2. Формалізація вимог та моделювання бізнес-процесів

1.2.1. Методи збору та аналізу бізнес-вимог до програмно-методичного комплексу (ПМК)

Для забезпечення адекватного функціоналу програмно-методичного комплексу (ПМК) було застосовано комплекс методів збору вимог. Проведено інтерв'ю з ключовими стейкхолдерами (керівником дільниці, майстрами) для визначення основних проблем та очікувань від системи. Використано анкетування серед технічного персоналу для збору кількісних даних про поточні процеси. Результати обговорень та бачення проєкту були формалізовані та затверджені під час наради, що підтверджено відповідним протоколом. Використання комбінованого підходу дозволило отримати як високоуровневе бачення системи від керівництва, так і детальні функціональні вимоги від безпосередніх виконавців робіт.

Для візуалізації та структурування початкових вимог було розроблено Інтелектуальну карту (Mind Map), яка охоплює ключові аспекти майбутньої системи: Планування, Моніторинг, Інциденти, Звітність, Користувачі та Інтеграція.

1.2.2. Проведення інтерв'ю з ключовими стейкхолдерами

Було проведено серію структурованих інтерв'ю з керівником дільниці та майстрами. Основна мета полягала у визначенні критичних проблем, що потребують негайного вирішення, та формулюванні загальних очікувань від системи.

Ключові питання інтерв'ю:

- які основні «вузькі місця» в поточному процесі планування та обліку ремонтів;
- яку інформацію про обладнання найчастіше доводиться шукати вручну;
- які типи звітів є найбільш затребуваними для керівництва підприємства;
- які існують ризики, пов'язані з паперовим документообігом (втрата даних, затримки)?

Результати інтерв'ю дозволили сформулювати загальне бачення проекту та визначити ключових акторів системи (Керівник, Майстер, Робітник), що лягло в основу розробки діаграм варіантів використання.

1.2.3. Анкетування технічного персоналу

Для збору кількісних даних та деталізації функціональних вимог було проведено анкетування серед 15 електромонтерів дільниці. Анкета складалася з 20 запитань закритого та напівзакритого типу.

Узагальнені результати анкетування:

- 80% опитаних підтвердили, що ручний пошук інформації про попередні ремонти займає у них понад 30 хвилин робочого часу за зміну;
- 93% вважають за необхідне впровадження автоматизованої системи сповіщення про інциденти та нові завдання;
- 60% висловили занепокоєння щодо можливості втрати паперових нарядів-допусків.

1.2.4. Протоколювання наради та візуалізація вимог

Для формалізації та затвердження зібраних вимог було проведено нараду з ключовими стейкхолдерами. Підсумки наради було оформлено відповідним протоколом, який зафіксував фінальне бачення програмного забезпечення.

Для візуалізації та структурування початкових вимог було розроблено Інтелектуальну карту (Mind Map) (Рисунок 3), яка охоплює ключові аспекти майбутньої системи: Планування, Моніторинг, Інциденти, Звітність, Користувачі та Інтеграція. Цей інструмент допоміг перетворити неструктуровані ідеї та побажання в чітку структуру проєкту.



Рисунок 1.2 – Інтелектуальна карта «Системи управління дільницею»

1. Проблема

Основна проблема полягає у відсутності цифрового контролю за змінами та інцидентами, а також у неавтоматизованому плануванні ремонтів. Наразі ручний облік та паперовий документообіг на дільниці призводять до значних простоїв обладнання, які в умовах енергетичної кризи 2025-2026 років є критичними для безперебійної роботи всього підприємства.

2. Сегменти клієнтів

Основними користувачами ПМК є керівники виробничих дільниць та технічний персонал ПРАТ «КАМЕТ-СТАЛЬ». Ці сегменти мають специфічні потреби: керівникам потрібна оперативна аналітика та контроль, тоді як персоналу простий та швидкий інструмент фіксації робіт та інцидентів.

3. Унікальна пропозиція (Unique Value Proposition)

Унікальність рішення полягає в інтуїтивному керуванні та оперативній реакції на інциденти. Система пропонує швидке впровадження та мінімальне навчання персоналу завдяки адаптованому інтерфейсу (Tkinter), що є значною перевагою перед складними корпоративними ERP-системами.

4. Рішення

Програмний комплекс є системою планування та моніторингу, інтегрованою з існуючою IT-інфраструктурою підприємства. Вона реалізує математичні моделі (CPM та лінійне програмування) для автоматичного розрахунку оптимальних планів робіт.

5. Канали збуту

Впровадження ПМК планується здійснити через внутрішні канали: прямі продажі (пілотний проєкт для інших цехів заводу), презентації на галузевих заходах «Метінвест» та залучення внутрішніх IT-інтеграторів групи.

6. Потоки доходів

Моделі монетизації включають продаж ліцензій на використання ПМК, щорічну підписку на оновлення та підтримку, а також кастомізацію (адаптацію) рішення під потреби інших дільниць.

7. Витрати

Основні витрати включають витрати на розробку програмного забезпечення (оплата праці розробників), впровадження (навчання персоналу, налаштування серверної частини SQL Server), підтримку та маркетинг.

8. Ключові метрики

Для оцінки успішності проєкту використовуватимуться наступні метрики: кількість активних користувачів, середня тривалість інциденту (час реагування), відсоток завершених змін за планом.

9. Прихована перевага (Unfair Advantage)

Конкурентна перевага полягає у власній базі сценаріїв реагування на типові відмови та глибокій інтеграції з локальними ERP-системами та сенсорами моніторингу на ПРАТ «КАМЕТ-СТАЛЬ».

1.2.5. Розробка бізнес-моделі програмного рішення з використанням Lean Canvas

Бізнес-модель проєкту була розроблена за допомогою інструменту Lean Canvas (рисунок 1.3.). Цей інструмент, адаптований для стартапів та інноваційних проєктів, дозволяє візуалізувати ключові аспекти майбутнього програмного забезпечення, його цінність для клієнтів та комерційний потенціал.

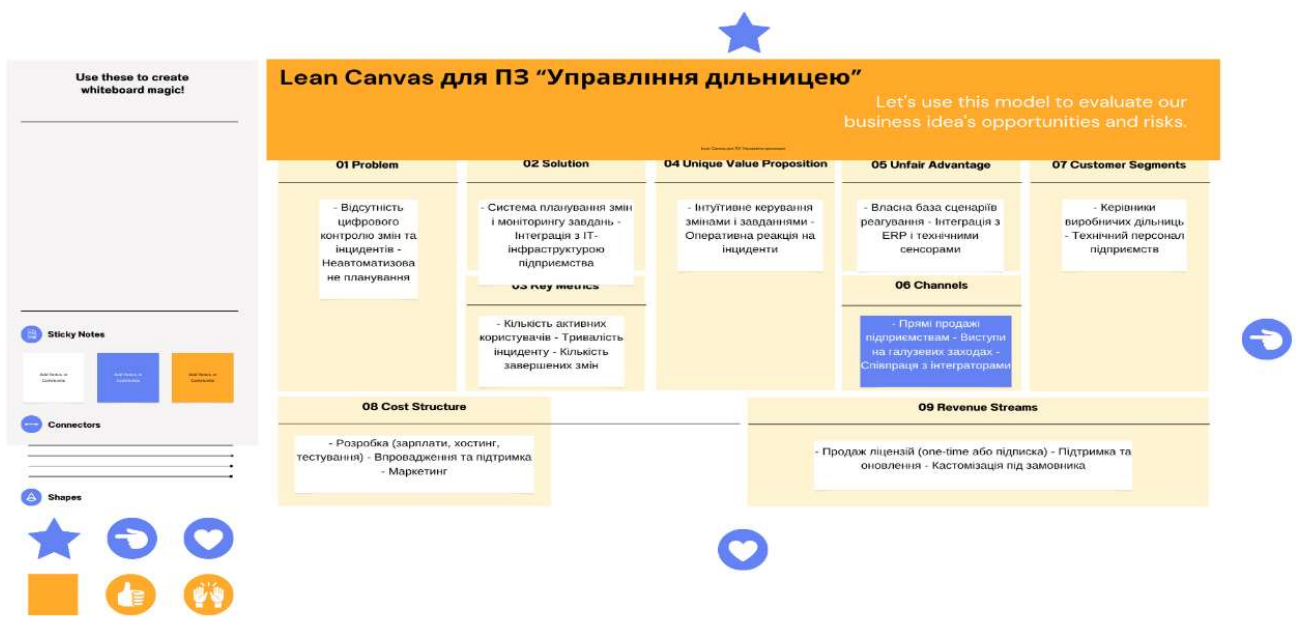


Рисунок 1.3 – Lean Canvas бізнес-моделі проєкту

Аналіз кожного блоку допоможе обґрунтувати стратегію впровадження ПМК на ПРАТ «КАМЕТ-СТАЛЬ». Для структурування комерційного потенціалу рішення:

- проблема: відсутність цифрового контролю змін, неавтоматизоване планування, простої обладнання;
- сегменти клієнтів: керівники виробничих дільниць, технічний персонал ПРАТ «Камет-Сталь»;
- унікальна пропозиція: інтуїтивне керування та оперативна реакція на інциденти;
- рішення: система планування та моніторингу, інтегрована з існуючою IT-інфраструктурою;
- канали збуту: прямі продажі, галузеві заходи, інтегратори;
- потоки доходів: ліцензії, підписка, підтримка, кастомізація;
- витрати: витрати на розробку, впровадження, підтримку та маркетинг;

- ключові метрики: кількість активних користувачів, тривалість інциденту, завершені зміни;
- прихована перевага: власна база сценаріїв реагування та інтеграції з ERP і сенсорами.
-

1.2.6. Моделювання бізнес-процесу "Управління дільницею" у нотаціях BPMN та DFD

Бізнес-процес "Управління дільницею" був змодельований у нотації BPMN (рис. 1.4), що дозволило графічно описати функціонал та взаємодію між учасниками:

- керівник дільниці;
- технічний персонал;
- ІТ-система.

Модель охоплює етапи:

1. Планування змін.
2. Виконання завдань.
3. Фіксація інциденту.
4. Реагування на інцидент.
5. Формування.
6. Перегляд звітів.

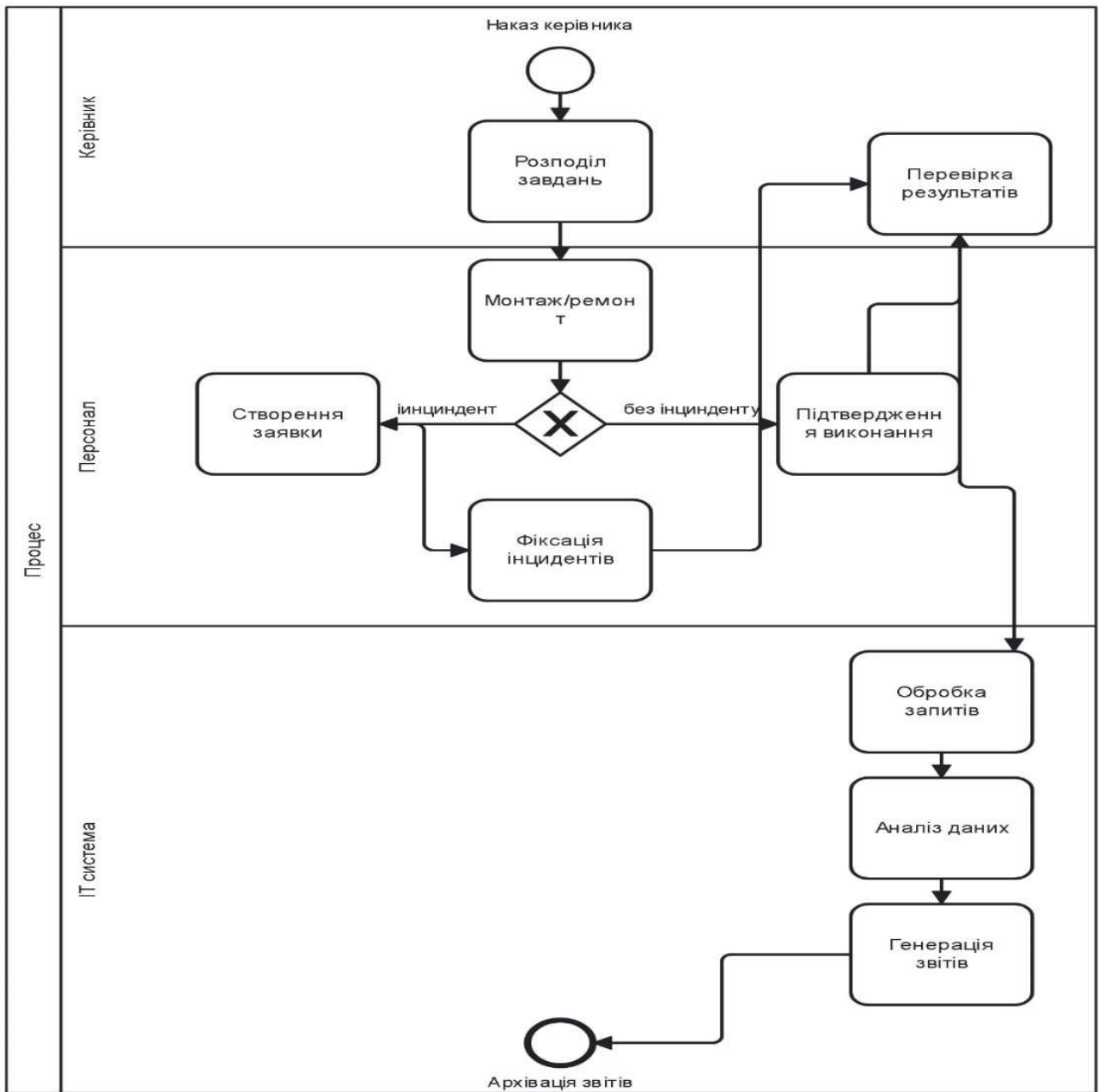


Рисунок 1.4 – BPMN-модель "Управління дільницею"

Для візуалізації потоків даних між зовнішніми сутностями (Керівник, Персонал, ERP-система, ІТ-система) та внутрішніми процесами (Планування змін, Реєстрація інцидентів, Контроль виконання, Формування

звітів) та сховищами даних розроблено Контекстну діаграму (DFD рівня 0) представлено на рисунку 1.5.

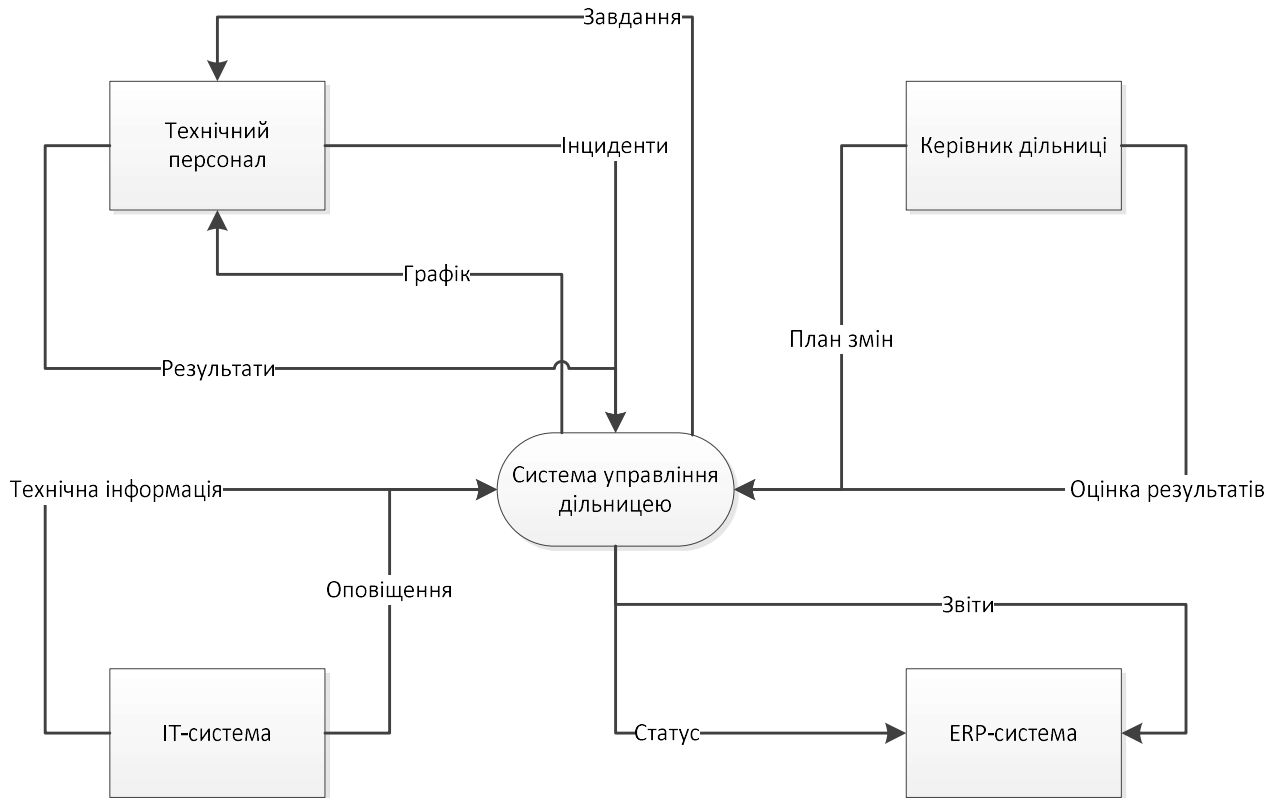


Рисунок 1.5 – Контекстна діаграма (DFD рівня 0)

А також розроблено декомпозицію системи (DFD рівня 1), яку представлено на рисунку 1.6.

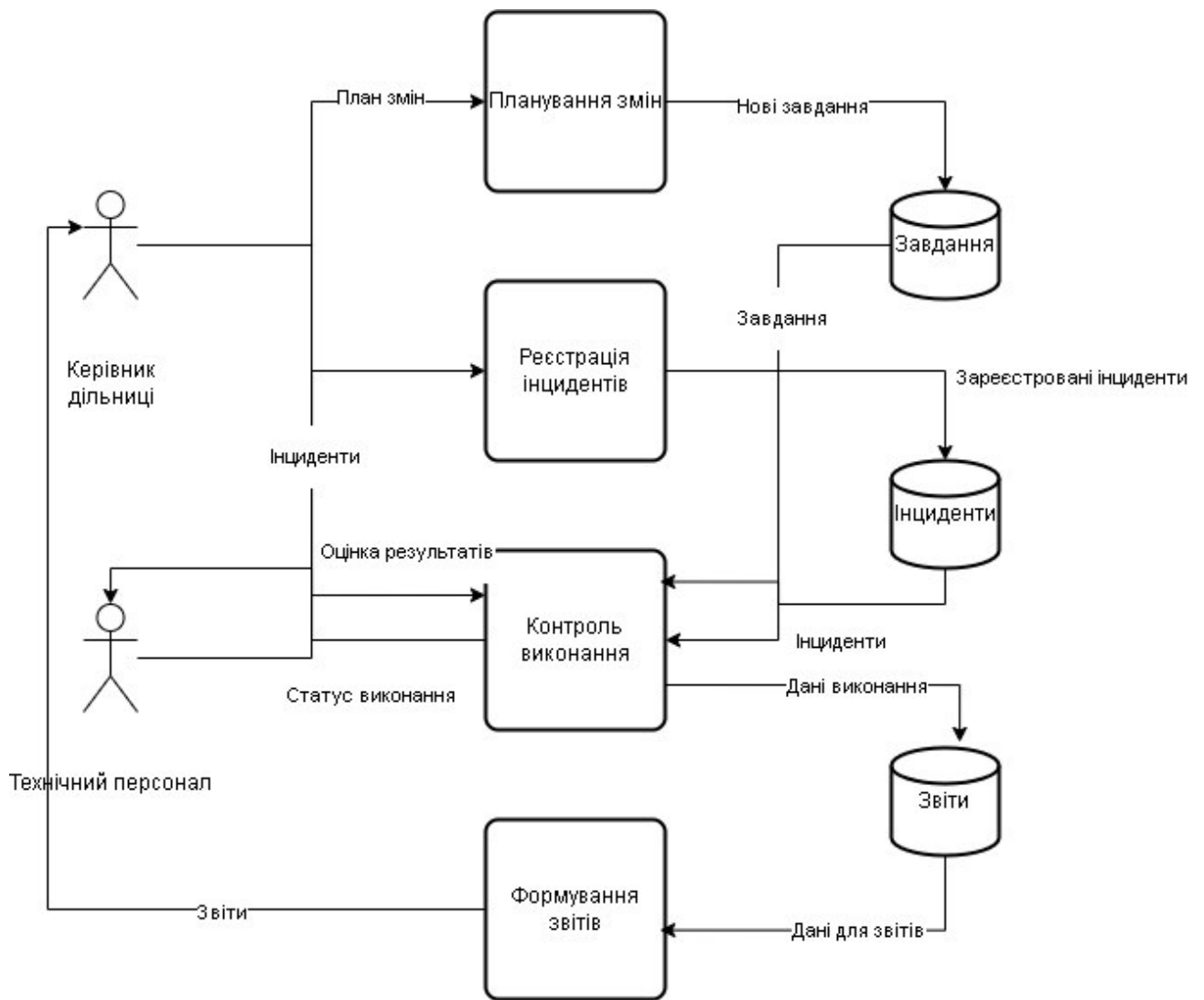


Рисунок 1.6 – Контекстна діаграма (DFD рівня 1)

1.2.7. Формалізація вимог до програмного забезпечення у форматі User Stories та Story Mapping

Для перетворення бізнес-потреб дільниці ПРАТ «КАМЕТ-СТАЛЬ» у функціональні специфікації було використано методологію User Stories. Це

дозволило сфокусувати розробку на реальних потребах користувачів (керівників та персоналу) та забезпечити гнучкість при проектуванні інтерфейсів.

Вимоги до системи були деталізовані у форматі User Stories. Наприклад: "Як керівник дільниці, я хочу створювати план змін на зміну, щоб ефективно розподіляти завдання між працівниками." Для кожної історії визначено критерії приймання.

Для візуалізації всього обсягу робіт та структурування беклогу використано інструмент Story Mapping. Він візуалізує основні етапи (хребет історії): Планування, Виконання, Інциденти, Звітування, Аналіз та відповідні завдання користувача під ними.

1.2.8. Деталізація користувацьких історій (User Stories)

Кожна історія описує роль користувача, його дію та очікувану цінність. Для забезпечення якості реалізації кожна історія доповнена критеріями приймання (Acceptance Criteria).

1. Історія Керівника дільниці:
 - формулювання: «Як керівник дільниці, я хочу створювати план на зміну, щоб ефективно розподіляти завдання між працівниками»;
 - критерії приймання:
 - a) система дозволяє обирати дату та час зміни;
 - b) можливість вибору працівників із бази даних Users;
 - c) система автоматично розраховує тривалість зміни на основі доданих завдань.

План зберігається в базі даних SQL Server зі статусом «Сформовано».

2. Історія Технічного працівника (Електромонтера):

- формулювання: «Як технічний працівник, я хочу переглядати список своїх завдань на сьогодні, щоб знати черговість виконання робіт»;

- критерії приймання:

- a) авторизація користувача під власним логіном;

- b) відображення лише тих завдань, де користувач призначений виконавцем.

Можливість змінити статус завдання на «Виконується» або «Завершено».

3. Історія Моніторингу інцидентів:

- формулювання: «Як користувач системи, я хочу фіксувати виникнення несправності обладнання, щоб майстер міг оперативно відкоригувати план»;

- критерії приймання:

- a) наявність кнопки «Реєстрація інциденту» на головній формі;

- b) можливість вибору типу несправності та агрегату;

- c) автоматичне надсилання сповіщення (візуального сигналу) у вікно керівника.

1.2.9. Побудова та аналіз карти історії (Story Mapping)

Для візуалізації всього обсягу робіт та структурування беклогу розробки було побудовано карту історії (Story Map) (Рисунок 1.7). Вона складається з «хребта» (основних кроків діяльності) та детальних завдань.



Рисунок 1.7 – Story Mapping

Опис етапів Story Map:

- планування: цей етап включає створення робочої зміни, вибір об'єктів обслуговування (двигуни, трансформатори) та розрахунок критичного шляху виконання робіт. Тут задіяна математична модель CPM;
- виконання: користувачі взаємодіють з інтерфейсом для отримання нарядів. Це активна фаза збору даних про фактичний час ремонту;
- інциденти: окремий потік робіт, що відповідає за реєстрацію позапланових відмов. Це критичний етап для 2026 року, що забезпечує стійкість системи до зовнішніх чинників;
- звітування: автоматична генерація PDF-звітів за зміну, день або місяць. Включає аналітику продуктивності кожного працівника;
- аналіз: рівень керівництва, де на основі накопичених даних у SQL Server проводиться оцінка ефективності (KPI) та планування закупівель запчастин на майбутнє.

Використання Story Mapping дозволило розділити розробку на ітерації (релізи), де першим пріоритетом (MVP — Minimum Viable Product) стала реалізація функцій планування та фіксації інцидентів.

1.3. Огляд методів математичного моделювання та засобів розробки

Ефективне керування дільницею з ремонту електроустаткування в умовах сучасного металургійного підприємства потребує застосування комплексу математичних методів для вирішення задач планування, прогнозування та оптимізації ресурсів.

1.3.1. Аналіз математичних методів оптимізації процесів ТОiP

Для автоматизації прийняття рішень на ПРАТ «КАМЕТ-СТАЛЬ» було обрано наступні математичні підходи:

- метод критичного шляху (Critical Path Method, CPM). Ремонтні роботи на дільниці складаються з низки взаємозалежних операцій (діагностика, демонтаж, заміна вузлів, випробування). Метод CPM дозволяє побудувати сіткову модель процесу, визначити операції, що безпосередньо впливають на загальну тривалість ремонту, та розрахувати часові резерви. Це критично важливо у 2026 році для мінімізації часу простою енергозалежних агрегатів заводу;
- лінійне програмування. Цей метод застосовується для задачі оптимального розподілу обмежених ресурсів (кваліфікованого персоналу,

запчастин, спеціального обладнання). Формулювання цільової функції дозволяє знайти такий план робіт, при якому загальні витрати будуть мінімальними за умови виконання всіх технічних регламентів.

1.3.2. Обґрунтування вибору інструментальних засобів розробки

Для програмної реалізації обраних методів та створення надійної інформаційної системи обрано наступний технологічний стек:

- мова програмування Python (версія 3.12). Вибір зумовлений високою швидкістю розробки та наявністю потужних спеціалізованих бібліотек:

1. PuLP та SciPy — для розв’язання задач лінійного програмування та оптимізації;

2. Pandas та NumPy — для обробки великих масивів даних про ремонти;

3. Tkinter — для створення легкого та швидкого графічного інтерфейсу, що стабільно працює на робочих станціях підприємства;

- СУБД Microsoft SQL Server. Це промислове рішення обрано як основне сховище даних для ПРАТ «КАМЕТ-СТАЛЬ». SQL Server забезпечує:

1. Високий рівень безпеки даних та розмежування прав доступу;

2. Цілісність транзакцій при одночасній роботі багатьох користувачів (керівника, майстрів, диспетчерів);

3. Можливість легкого масштабування системи при збільшенні кількості об’єктів обслуговування;

Таким чином, комбінація класичних методів оптимізації та сучасних засобів розробки дозволяє створити гнучку та ефективну систему інформаційної підтримки управління дільницею.

1.4. Аналіз існуючих IT-рішень для автоматизації ремонтних служб

У сучасній практиці промислових підприємств для управління технічним обслуговуванням та ремонтами (ТОiP) зазвичай використовують системи класу EAM (Enterprise Asset Management) або відповідні модулі в ERP-системах. Проведемо порівняльний аналіз найбільш поширених рішень на ринку 2024–2026 років у контексті їх застосування на ПРАТ «КАМЕТ-СТАЛЬ».

1.4.1. Аналіз корпоративних систем (SAP ERP, IBM Maximo)

SAP ERP (Module PM — Plant Maintenance): Це найбільш потужне рішення, яке використовується групою «Метінвест» на вищому рівні управління. Воно дозволяє інтегрувати ремонти з фінансовим обліком та закупівлями. Проте для рівня окремої дільниці ЦРУ система є занадто громіздкою. Внесення даних у SAP потребує значного часу, а інтерфейс не адаптований для оперативного використання безпосередньо майстрами на дільниці.

IBM Maximo: Спеціалізована EAM-система, що вважається світовим лідером у сфері управління активами. Вона володіє потужними

інструментами аналітики, але вартість ліцензій та впровадження у 2026 році є надзвичайно високою. Крім того, підтримка такої системи потребує залучення сторонніх консультантів, що знижує автономність цеху.

1.4.2. Аналіз нішевих та хмарних рішень (UpKeep, Fiix)

Хмарні системи (SaaS) пропонують зручні мобільні додатки для технічного персоналу. Проте вони мають критичні недоліки для українських підприємств:

Залежність від інтернету: В умовах можливих відключень зв'язку та електроенергії (ризик 2026 року) робота хмарної системи може бути паралізована.

Кібербезпека: стратегічні підприємства, такі як ПРАТ «КАМЕТ-СТАЛЬ», мають жорсткі вимоги щодо зберігання даних усередині локальної мережі, що виключає використання сторонніх хмарних серверів.

1.4.3. Порівняльна характеристика та обґрунтування власної розробки

Для наочності проведемо порівняння існуючих рішень та розробленого програмно-методичного комплексу (ПМК) (табл. 1.2).

Таблиця 1.2. — Порівняльний аналіз систем управління ремонтами

Критерій порівняння	SAP ERP	Хмарні системи (SaaS)	Розроблений ПМК (Python/SQL)
Вартість впровадження	Висока	Середня	Низька
Складність інтерфейсу	Висока	Низька	Мінімальна (адаптована)
Автономність (Offline)	Часткова	Відсутня	Повна (локальна БД)
Гнучкість налаштувань	Низька	Середня	Висока (відкритий код)
Інтеграція з математичними моделями	Складна	Обмежена	Вбудована (модулі Python)

Обґрунтування розробки.

Власна розробка на базі мови Python 3.12 та СУБД Microsoft SQL Server є найбільш раціональним рішенням для дільниці з ремонту електроустаткування з таких причин:

1. Локалізація даних. Використання SQL Server дозволяє зберігати всю інформацію на внутрішніх серверах підприємства, забезпечуючи безпеку та швидкість доступу навіть без зовнішнього інтернету.

2. Спеціалізація. Можливість вбудувати специфічні математичні алгоритми (CPM, лінійне програмування), які не є стандартними для більшості комерційних систем.

3. Економічність. Використання мов програмування з відкритим кодом (Open Source) дозволяє уникнути великих витрат на ліцензії, що критично важливо для бюджетування у 2026 році.

Таким чином, розробка спеціалізованого ПМК є оптимальним балансом між функціональністю, безпекою та економічною ефективністю для ПРАТ «КАМЕТ-СТАЛЬ».

1.5. Висновки до першого розділу

У результаті виконання першого розділу кваліфікаційної роботи було проведено комплексне дослідження предметної області та аналіз існуючих підходів до автоматизації управління дільницею з ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ». Основні результати полягають у наступному:

- проаналізовано об'єкт дослідження та специфіку функціонування дільниці ЦРУ. Встановлено, що в умовах 2026 року критичними проблемами залишаються ручний облік інцидентів та відсутність оперативної інформаційної підтримки, що призводить до затримок у прийнятті управлінських рішень та збільшення простоїв обладнання;
- формалізовано вимоги до майбутнього програмно-методичного комплексу за допомогою сучасних інструментів бізнес-аналізу (Lean Canvas, Mind Map). Побудовано графічні моделі у нотаціях BPMN та DFD, які дозволили візуалізувати інформаційні потоки та визначити роль кожного учасника ремонтного процесу;
- обґрунтовано вибір математичного апарату, що базується на комбінації методу критичного шляху (CPM) для часового планування та методів лінійного програмування для оптимізації ресурсів. Доведено доцільність застосування нечіткої логіки для обробки невизначених параметрів виробничого середовища;

- визначено технологічний стек розробки на базі мови Python та СУБД MS SQL Server. Порівняльний аналіз із існуючими ERP-системами (SAP, IBM Maximo) показав, що розробка власного спеціалізованого ПМК є найбільш економічно виправданою та безпечною стратегією для конкретного підрозділу підприємства.

Отримані результати аналізу та обрані методи складають теоретичний фундамент для подальшої розробки математичних моделей та алгоритмів функціонування системи, що будуть детально розглянуті у другому розділі роботи.

2. РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ОБ'ЄКТА ТА МЕТОДИКИ ДОСЛІДЖЕННЯ

2.1 Обґрунтування вибору методів теоретичних та експериментальних досліджень

Для ефективного управління ділянкою з ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ» необхідно розв'язати комплекс задач:

- оптимальний розподіл обмежених ресурсів
- мінімізація часу виконання робіт
- прогнозування ризиків.

У межах дослідження було проаналізовано наступні методи:

- теорія графів та сіткове планування (CPM): для візуалізації та розрахунку часових параметрів взаємозалежних технологічних операцій;
- лінійне програмування: для розв'язання задачі мінімізації витрат при заданих ресурсних обмеженнях;
- метод критичного шляху (Critical Path Method, CPM).



Рисунок 2.1 – Графічна модель управління дільницею

2.2 Математична модель на основі теорії графів та методу критичного шляху (CPM)

Метод критичного шляху (Critical Path Method, CPM) дозволяє ідентифікувати найбільш важливі етапи ремонту.

Процес ремонту електроустаткування формалізується як орієнтований ациклічний граф $G=V,E$, де V — відповідають подіям (початок або завершення робіт), а E — самим роботам (дуги).

На основі аналізу ділянки ПРАТ «КАМЕТ-СТАЛЬ» визначено 9 етапів типового циклу ремонту. Кожен етап має свою вагу (тривалість у годинах).

Таблиця 2.1 — Параметри етапів ремонту для моделювання

№	Назва етапу	Код	Тривалість (год)	Попередні роботи
1	Діагностика	A	2	-
2	Демонтаж	B	3	A
3	Дефектовка вузлів	C	4	B
4	Замовлення запчастин	D	8	C
5	Ремонт обмоток	E	12	C
6	Збірка агрегату	F	5	D, E
7	Випробування	G	2	F
8	Фарбування та пакування	H	1	G
9	Передача замовнику	I	1	H

2.2.1. Формалізація етапів робіт

Згідно з аналізом предметної області ПРАТ «КАМЕТ-СТАЛЬ», типовий процес ремонту включає 9 ключових етапів:

- виявлення несправностей;
- діагностика та дефектовка;
- підготовка інструментів та запчастин;
- демонтаж обладнання;
- основний ремонт/заміна вузлів;
- збірка та налагодження;
- випробування під навантаженням;
- фіксація результатів у системі;
- формування звітності.

2.2.2. Розрахунок критичного шляху

Метод критичного шляху (Critical Path Method) дозволяє визначити мінімальний час, необхідний для завершення всього комплексу робіт. Критичний шлях визначається як найдовша послідовність операцій від початкової до кінцевої події. Будь-яка затримка на цьому шляху призводить до затримки завершення ремонту всього об'єкта. Алгоритм розрахунку, реалізований у програмному комплексі, включає:

1. Визначення ранніх термінів початку (ES) та закінчення (EF).
2. Визначення пізніх термінів початку (LS) та закінчення (LF).
3. Розрахунок часового резерву для кожної операції: $R=LF-EF$.

2.3. Модель оптимізації ресурсів засобами лінійного програмування

Для автоматизації розподілу ресурсів у 2026 році в межах ПМК використовується модель лінійного програмування, для мінімізації загальних витрат на ремонтну зміну при обмеженій кількості персоналу та матеріалів.

2.3.1. Функція цілі

Цільова функція мінімізації витрат Z на виконання комплексу з n робіт описується формулою 2.1.

$$Z = \sum_{i=1}^n c_i * x_i \quad (2.1)$$

Z (цільова функція): вимірюється у грошових одиницях (грн). Це загальна сума витрат на весь комплекс робіт.

c_i (питомі витрати): вимірюються у грошових одиницях на одиницю інтенсивності. Це вартість за виконання одиниці роботи.

x_i (інтенсивність виконання): вимірюється у відносних або фізичних одиницях обсягу робіт за одиницю часу (людина/година).

Згідно з розрахунками для дільниці з ремонту устаткування ПРАТ «КАМЕТ-СТАЛЬ», модель функції цілі (мінімізувати витрати) має вигляд формули 2.2:

$$\min Z = 2x_1 + 3x_2 + 4x_3 + 5x_5 + 3x_6 + 2x_7 + 4x_8 + 2x_9 \quad (2.2)$$

- $minZ$ - це підсумкова вартість усього ремонту (у гривнях), яку треба зробити мінімальною;
- числа (2, 3, 4, 5...) - це «цінник» за кожну роботу (коефіцієнти питомих витрат). Вони включають:
 - а) Зарплату робітників (чим вища кваліфікація, тим вище число);
 - б) Вартість матеріалів та запчастин для конкретного етапу;
 - в) Витрати на роботу обладнання (електроенергія, амортизація).
- змінні ($x_1, x_2, x_3 \dots$) - це інтенсивність виконання кожної з 9 робіт. Це те, чим «грає» програма, щоб підібрати оптимальний план.

Ця формула каже комп'ютеру: "Знайди мені найдешевший спосіб організувати роботу людей та обладнання так, щоб ми встигли відремонтувати двигун вчасно.

2.3.2. Система обмежень

Модель повинна враховувати реальні обмеження дільниці:

1. Обмеження по часу описується формулою 2.3:

$$\sum_{i=1}^n t_i \cdot x_i \leq T_{shif} \quad (2.3)$$

де T_{shif} = тривалість робочої зміни 8 або 12 годин

n = загальна кількість робіт

t_i = нормативний час, необхідний для виконання i -ї роботи при стандартних умовах (год.);

x_i = інтенсивність виконання i -й роботи (коефіцієнт).

Дана нерівність є ресурсним обмеженням за часом. Вона показує, що фактичний час виконання всіх операцій на дільниці не може перевищувати тривалість робочої зміни. Це дозволяє системі автоматично розраховувати такий план, який буде реально виконати наявним персоналом без перевантаження

2. Обмеження по персоналу описується формулою 2.4:

$$\sum x_i \leq P_{total} \quad (2.4)$$

де P_{total} - кількість наявних електромонтерів;

x_i - кількість персоналу (або інтенсивність праці), закріпленого за i -ю роботою в даний момент часу.

Дана нерівність відображає обмеження за людськими ресурсами. Вона вказує на те, що ми не можемо призначити на виконання робіт більше людей, ніж фактично працює на дільниці в даній зміні. Система використовує це обмеження для оптимізації розподілу фахівців, щоб уникнути дефіциту кадрів під час виконання складних ремонтних операцій

3. Технологічні обмеження: $x_i \geq a_i$ (мінімально необхідна інтенсивність для завершення етапу).

2.4. Практична реалізація методу критичного шляху на прикладі капітального ремонту електродвигуна серії AIP

Для підтвердження працездатності розробленої математичної моделі розглянемо процес капітального ремонту асинхронного електродвигуна потужністю 110 кВт, що є типовим об'єктом обслуговування на ділянці з ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ».

2.4.1. Покроковий розрахунок часових параметрів сітки

Розрахунок проводиться у два етапи: «прямий хід» (визначення ранніх термінів) та «зворотний хід» (визначення пізніх термінів).

Розрахунок часових параметрів мережевого графіка методом критичного шляху (CPM)(описаний нижче):

1. Прямий хід (Early Start/Finish):

Робота А: $ESA=0$; $EFA=0+2=2$.

Робота В: $ESB=EFA=2$; $EFB=2+3=5$.

Робота С: $ESC=EFB=5$; $EFC=5+4=9$.

Робота D: $ESD=EFC=9$; $EFD=9+8=17$.

Робота Е: $ESE=EFC=9$; $EFE=9+14=23$.

Робота F: $ESF=EFC=9$; $EFF=9+5=14$.

Робота G: Виконується після D, Е та F. Обираємо максимальне значення: $ESG=\max(17,23,14)=23$; $EFG=23+4=27$.

Робота H: $ESH=27$; $EFH=27+3=30$.

Робота I: $ESI=30$; $EFI=30+2=32$.

Загальна тривалість ремонту складе 32 години.

2. Зворотний хід (Late Start/Finish):

Пізнє закінчення останньої роботи дорівнює її ранньому закінченню: $LFI=32$.

Робота I: LFI=32; LSI=32-2=30.

Робота H: LFH=30; LSH=30-3=27.

Робота G: LFG=27; LSG=27-4=23.

Робота E: LFE=LSG=23; LSE=23-14=9.

Робота D: LFD=LSG=23; LSD=23-8=15.

Робота F: LFF=LSG=23; LSF=23-5=18.

Робота C: LFC=min LSD,LSE,LSF=min (15,9,18)=9; LSC=9-4=5.

2.4.2. Визначення повних резервів та критичного шляху

Розрахуємо повний резерв часу для кожної операції за формулою:

$$R_i = LS_i - ES_i$$

Таблиця 2.3 — Результати розрахунку резервів часу

Робота	ES	EF	LS	LF	Резерв (Ri)	Критична?
A	0	2	0	2	0	Так
B	2	5	2	5	0	Так
C	5	9	5	9	0	Так
D	9	17	15	23	6	Ні
E	9	23	9	23	0	Так
F	9	14	18	23	9	Ні
G	23	27	23	27	0	Так
H	27	30	27	30	0	Так

Аналіз результатів:

Критичний шлях проходить через роботи: $A \rightarrow B \rightarrow C \rightarrow E \rightarrow G \rightarrow H \rightarrow I$.

Будь-яка затримка на етапі E (Перемотування статора) призведе до миттєвого збільшення загального часу простою обладнання на заводі ПРАТ «КАМЕТ-СТАЛЬ». Роботи D та F мають значні резерви часу (6 та 9 годин відповідно), що дозволяє керівнику дільниці маневрувати ресурсами (наприклад, перевести фахівця з балансування ротора на допомогу обмотувальникам статора).

Візуалізація графа (структура)

Граф будується на основі опису послідовності робіт. Оскільки роботи D, E, F починаються одночасно після C, а G чекає на завершення всіх трьох, ми маємо "паралельну" ділянку показано на рисунку 2.2.

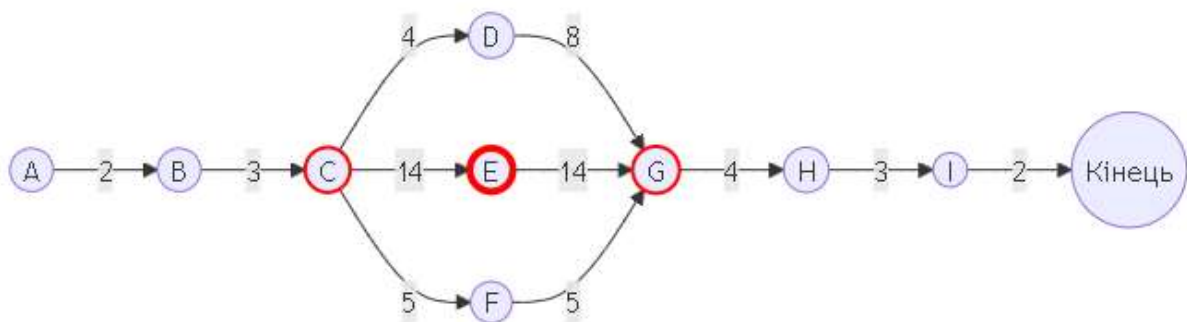


Рисунок 2.2 Мережевий граф (PERT-діаграма)

Загальна тривалість ремонту: складає 32 години (це EFi — ранній фініш останньої роботи).

Критичний шлях: Це шлях, на якому роботи не мають резерву часу ($ES = LS$). Це шлях через роботу E, яка є найдовшою в паралельній ділянці:

$A \rightarrow B \rightarrow C \rightarrow E \rightarrow G \rightarrow H \rightarrow I$

Вузьке місце: Робота E (тривалість 14 год). Будь-яка затримка на цій ділянці призведе до збільшення загального часу ремонту. Роботи D та F мають резерви (6 та 9 годин відповідно), тому їх можна починати трохи пізніше без шкоди для графіку.

На основі побудованого мережевого графа ми визначили, що критичний шлях складає 32 години. Ключовою операцією є робота E, яка визначає загальний темп ремонту. Використання цього методу в ПМК дозволяє майстру дільниці бачити, на яких роботах ресурси критичні, а де є запас часу.

Даний розрахунок інтегрований у програмний модуль на Python, що дозволяє автоматично перераховувати ці параметри при кожній зміні вхідних даних у 2026 році.

2.5. Висновки до другого розділу

У другому розділі кваліфікаційної роботи було застосовано математичне забезпечення та методика інформаційної підтримки процесів керування дільницею з ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ». Проведені дослідження дозволили отримати наступні результати:

1. Здійснено формалізацію задачі управління, яку представлено як задачу багатокритеріальної оптимізації часових та матеріальних ресурсів. Визначено ключові вхідні параметри та обмеження, характерні для металургійного виробництва в умовах 2025-2026 роках.

2. Застосовано сіткову модель процесу ремонту на основі теорії графів та методу критичного шляху (CPM). Розрахунок параметрів ранніх та пізніх термінів виконання робіт дозволив ідентифікувати критичні операції (зокрема, етап ремонту обмоток та дефектовку), затримка яких призводить до простою суміжних виробничих ліній підприємства.

3. Застосовано модель лінійного програмування для оптимізації витрат на виконання ремонтних завдань.

3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ

3.1 Архітектура та вибір технологічного стека

Розробка програмно-методичного комплексу (ПМК) для дільниці з ремонту електроустаткування ПРАТ «КАМЕТ-СТАЛЬ» базується на принципах багаторівневої архітектури. Це дозволяє відокремити логіку обробки даних від інтерфейсу користувача та забезпечити стабільну роботу системи в умовах цехової мережі підприємства.

Для автоматизації дільниці ПРАТ «КАМЕТ-СТАЛЬ» у 2026 році обрано клієнт-серверну архітектуру, що забезпечує надійність зберігання даних та модульність системи.

Обґрунтування вибору інструментів:

- Python 3.12: обраний як основна мова завдяки наявності потужних бібліотек для математичних розрахунків (PuLP для лінійного програмування, pandas для аналізу даних);
- Microsoft SQL Server: використаний як система управління базами даних (СУБД) через високу стійкість до відмов та підтримку складних транзакцій;
- Tkinter: використаний для розробки графічного інтерфейсу (GUI) як стандартна бібліотека, що забезпечує швидкодію на робочих станціях підприємства.

3.1.1. Структурна схема та рівні архітектури ПМК

Для забезпечення гнучкості системи обрано трирівневу архітектуру (Three-Tier Architecture), яка включає наступні рівні:

1. Рівень представлення (Presentation Layer). Реалізований як десктоп-додаток на базі бібліотеки Tkinter. Цей рівень забезпечує взаємодію користувача (керівника дільниці або майстра) з системою через графічні форми, валідацію введених даних та візуалізацію результатів розрахунків (планів-графіків).

2. Рівень бізнес-логіки (Application/Logic Layer). Це ядро системи, написане мовою Python 3.12. Тут реалізовані математичні модулі, розроблені у Розділі 2:

- модуль розрахунку критичного шляху (CPM);
- модуль оптимізації ресурсів (бібліотека PuLP).

3. Рівень даних (Data Layer): Представлений СУБД Microsoft SQL Server. На цьому рівні забезпечується транзакційність, цілісність посилань та надійне зберігання історії ремонтів за 2024–2026 роки.

Такий підхід дозволяє у майбутньому легко замінити клієнтську частину (наприклад, розробити веб-інтерфейс) без зміни математичного ядра та структури бази даних.

3.1.2. Обґрунтування вибору мови програмування Python 3.12

Вибір мови Python 3.12 для реалізації ПМК у 2026 році зумовлений наступними чинниками:

- швидкість розробки та підтримки. Синтаксис мови дозволяє швидко адаптувати програмний код під мінливі вимоги ПРАТ «КАМЕТ-СТАЛЬ» (наприклад, при зміні регламентів ТОiP);
- бібліотеки для математичних розрахунків. Наявність спеціалізованих пакетів, таких як NumPy для матричних обчислень та SciPy для оптимізації, суттєво спрощує реалізацію складних алгоритмів;
- кросплатформеність. Система може працювати як на застарілих робочих станціях під керуванням Windows 10, так і на нових серверах підприємства;
- інтеграція. Python має розвинені засоби для роботи з SQL-запитами (бібліотека pyodbc), що забезпечує високу швидкість обміну даними з сервером.

3.1.3. Вибір системи управління базами даних (СУБД)

Для промислового об'єкта, яким є ПРАТ «КАМЕТ-СТАЛЬ», обрано Microsoft SQL Server. Основні переваги цього рішення для дільниці:

1. Надійність. Система підтримує механізми автоматичного відновлення після збоїв, що критично в умовах нестабільного енергоживлення у 2026 році.
2. Розмежування прав доступу. SQL Server дозволяє гнучко налаштувати ролі для користувачів (наприклад, «Адміністратор» може змінювати структуру таблиць, а «Майстер» — лише додавати записи про виконання робіт).

3. Продуктивність. СУБД легко обробляє велику кількість одночасних запитів, що важливо при масштабуванні ПМК на декілька ремонтних дільниць цеху.

3.2 Розробка бази даних (Data Architecture)

Проектування архітектури бази даних є критичним етапом розробки програмно-методичного комплексу, оскільки від структури зберігання даних залежить швидкість виконання оптимізаційних алгоритмів та цілісність інформації про стан ремонтних процесів.

В якості системи керування базами даних (СУБД) було обрано Microsoft SQL Server, що зумовлено високими вимогами до безпеки даних, підтримкою складних транзакцій та можливістю локального розгортання в межах внутрішньої мережі підприємства.

На основі аналізу бізнес-процесів дільниці було розроблено концептуальну модель «сутність-зв'язок» (ER-діаграма), яку було приведено до третьої нормальної форми (3NF) для усунення надмірності даних та аномалій оновлення. Архітектура бази даних включає такі ключові сутності (Рисунок 3.1):

- User: зберігає дані про персонал (ID, ПІБ, роль: Керівник, Майстер, Робітник).
- Task: перелік виробничих завдань з атрибутами (опис, дата початку, тривалість, статус).
- Shift: облік робочих змін.
- Incident: реєстрація аварійних ситуацій, що виникають під час ремонту.
- Report: сформовані звіти за результатами виконання робіт

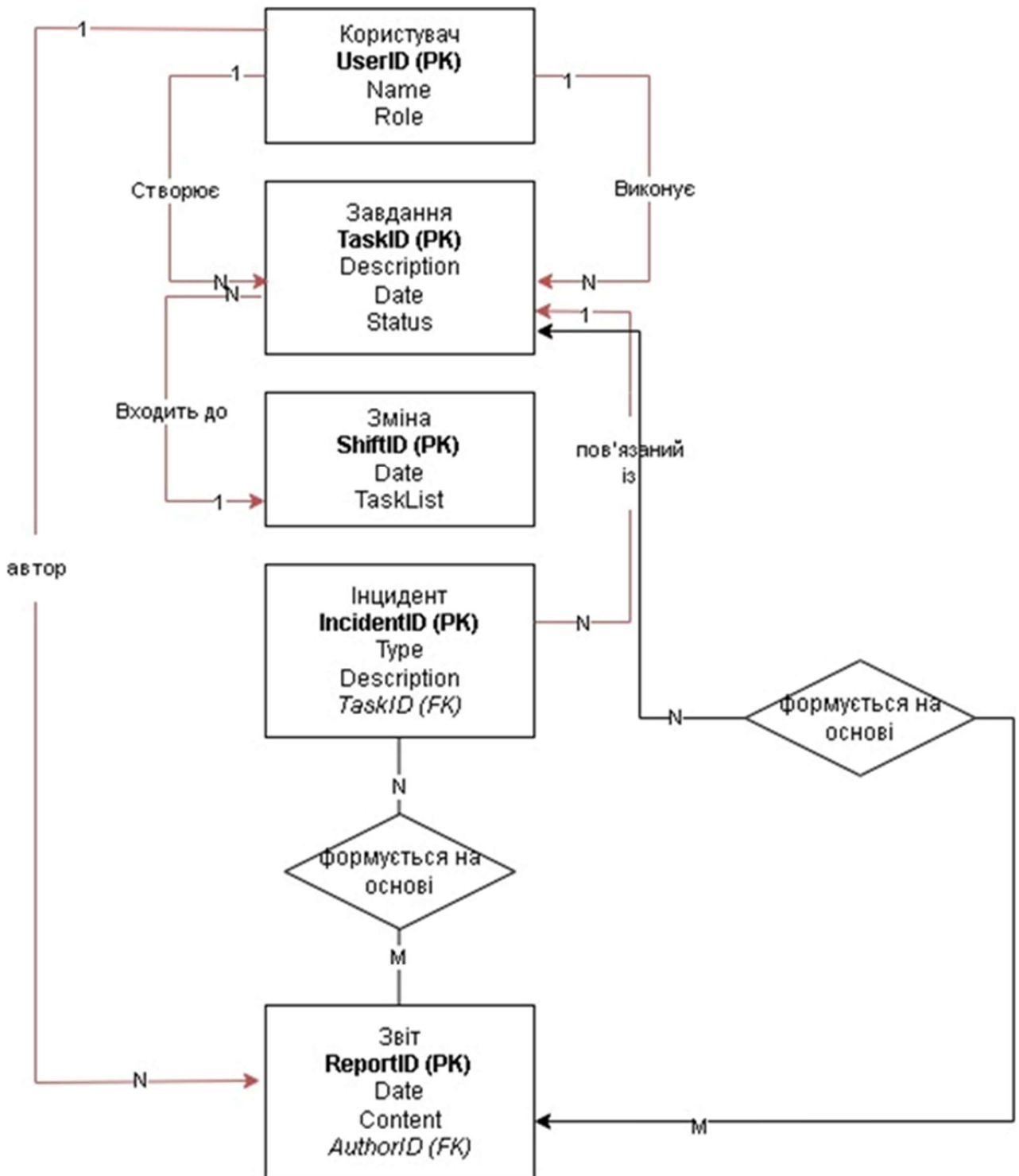


Рисунок 3.1 – Діаграма "сутність-зв'язок" (ERD)

3.2.1. Фізична реалізація в SQL Server

Створення фізичної структури в SQL Server представлено на рисунках 3.2-3.4.

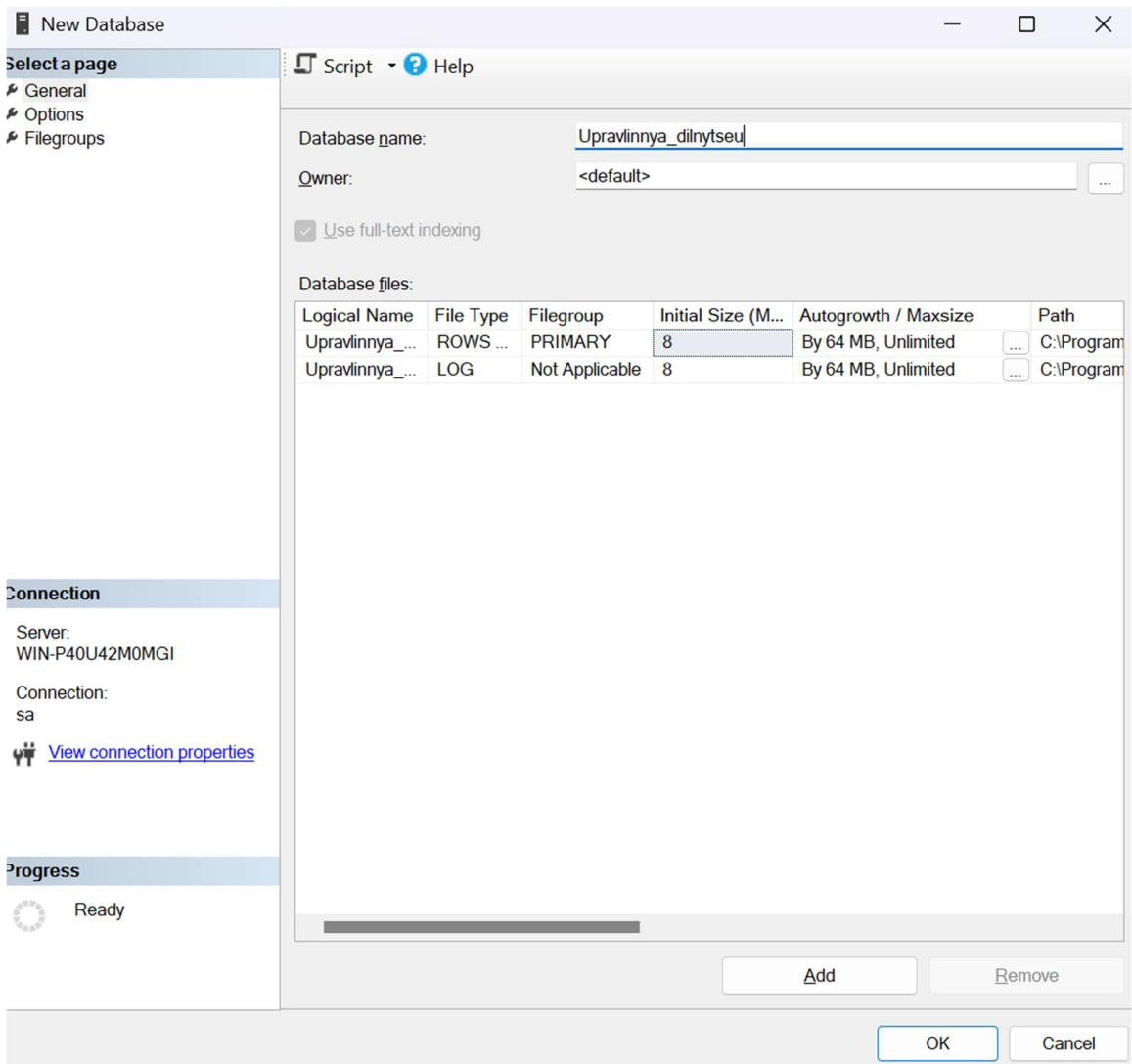


Рисунок 3.2 – Створення нової бази даних

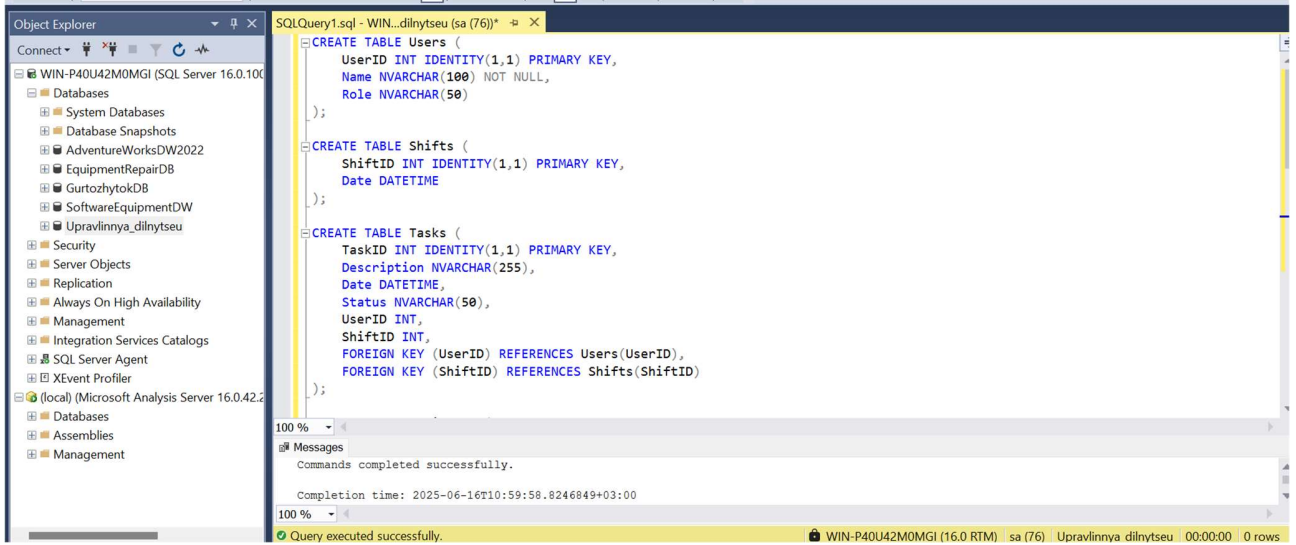


Рисунок 3.3 – Створення таблиць в базі даних

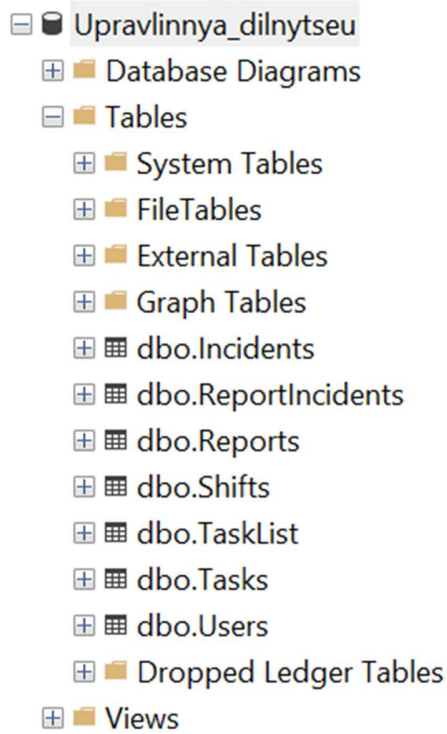


Рисунок 3.4 – Готові таблиці в базі дани

3.3. Модуль інтеграції з базою даних

Використано бібліотеку `pyodbc` для зв'язку Python-додатку з SQL Server. Реалізовано методи для безпечного додавання даних та виклику процедур планування (Рисунки 3.5-3.10).

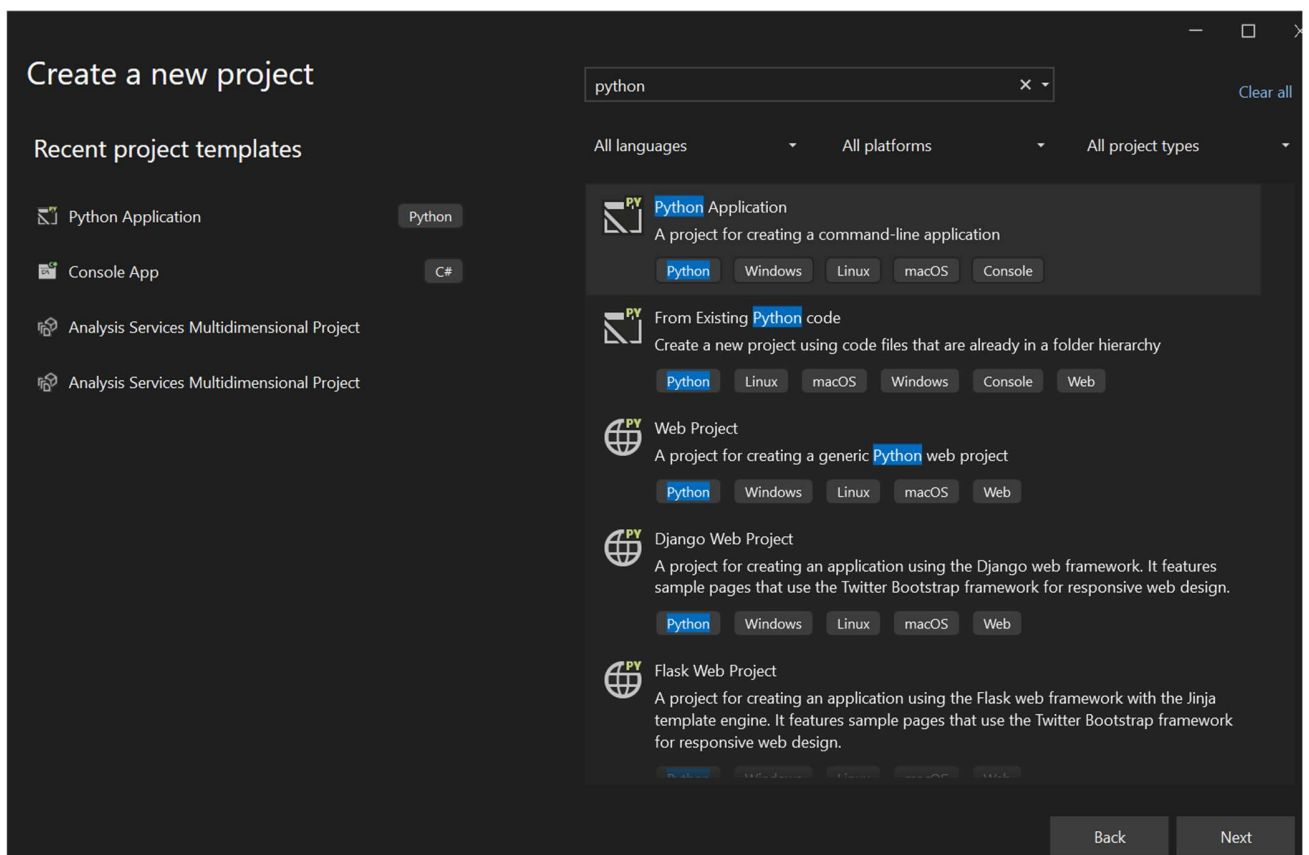


Рисунок 3.5 – Створення проекту в Visual Studio

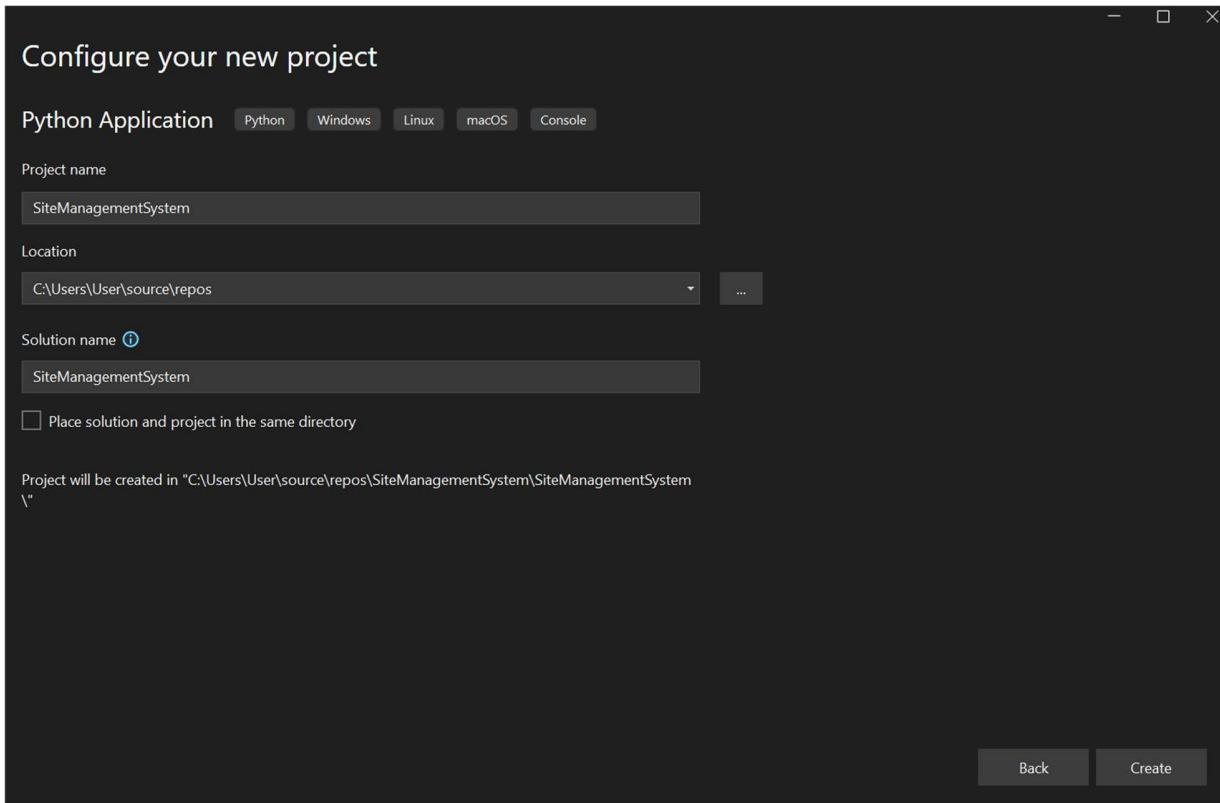


Рисунок 3.6 – Конфігурування проєкту

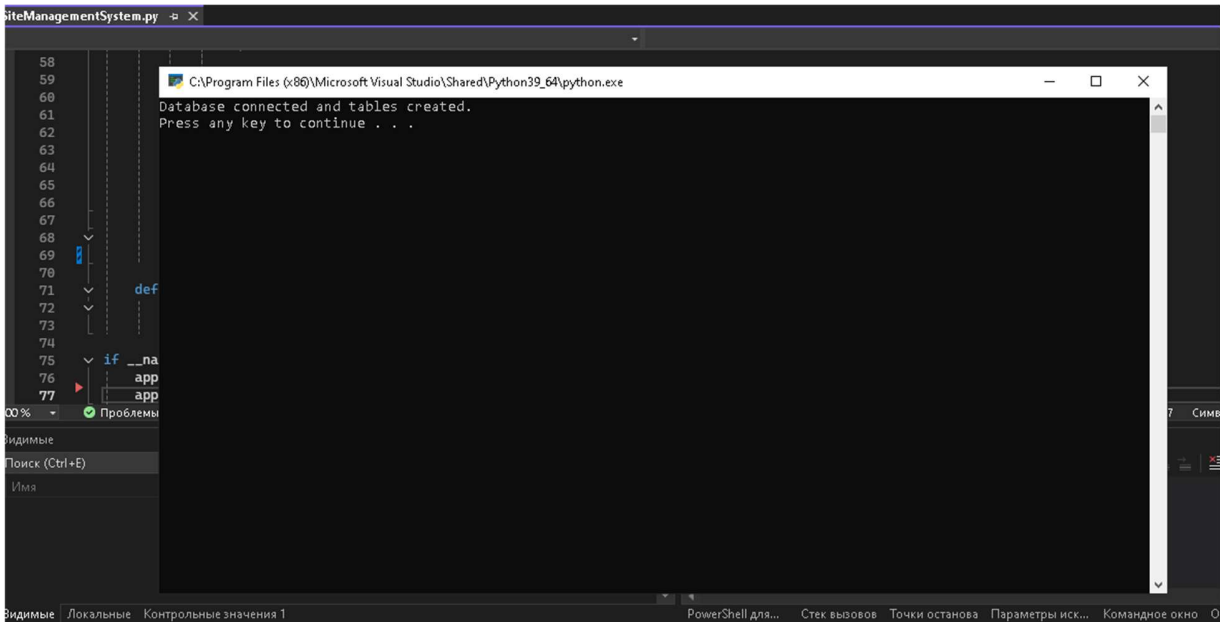
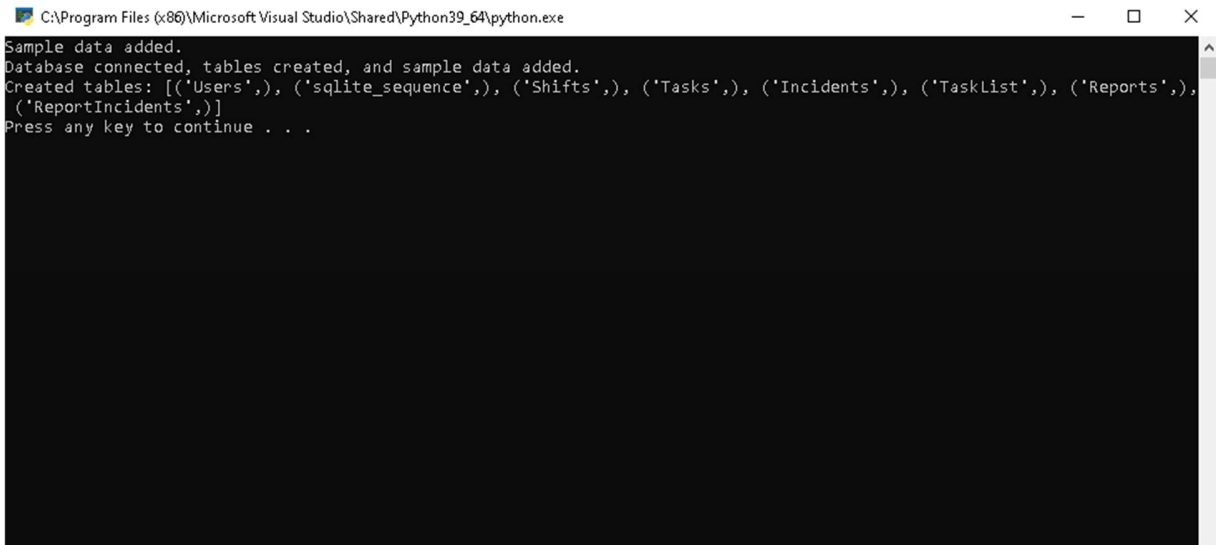


Рисунок 3.7 – Проєкт створений



```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
Sample data added.
Database connected, tables created, and sample data added.
Created tables: [('Users',), ('sqlite_sequence',), ('Shifts',), ('Tasks',), ('Incidents',), ('TaskList',), ('Reports',),
('ReportIncidents',)]
Press any key to continue . . .
```

Рисунок 3.8 – Додані таблиці у БД



```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
Sample data added.
Database connected, tables created, and sample data added.
Created tables: [('Users',), ('sqlite_sequence',), ('Shifts',), ('Tasks',), ('Incidents',), ('TaskList',), ('Reports',),
('ReportIncidents',)]
Users: [(1, 'John Doe', 'Manager'), (2, 'John Doe', 'Manager'), (3, 'John Doe', 'Manager'), (4, 'John Doe', 'Manager'),
(5, 'Ivan Sirko', 'Manager'), (6, 'Ivan Sirko', 'Manager')]
Press any key to continue . . .
```

Рисунок 3.9 – Додавання користувачей до БД

```

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
Sample data added.
Database connected, tables created, and sample data added.
Created tables: [('Users',), ('sqlite_sequence',), ('Shifts',), ('Tasks',), ('Incidents',), ('TaskList',), ('Reports',),
('ReportIncidents',)]
Users: [(1, 'John Doe', 'Manager'), (2, 'John Doe', 'Manager'), (3, 'John Doe', 'Manager'), (4, 'John Doe', 'Manager'),
(5, 'Ivan Sirko', 'Manager'), (6, 'Ivan Sirko', 'Manager'), (7, 'Ivan Sirko', 'Manager'), (8, 'Ivan Sirko', 'Manager'),
(9, 'Ivan Sirko', 'Manager'), (10, 'Vasyl Petrenko', 'Operator'), (11, 'Ivan Sirko', 'Manager')]
Enter user name: Dmytro Klap
Enter user role: station chief
User Dmytro Klap added.
Users: [(1, 'John Doe', 'Manager'), (2, 'John Doe', 'Manager'), (3, 'John Doe', 'Manager'), (4, 'John Doe', 'Manager'),
(5, 'Ivan Sirko', 'Manager'), (6, 'Ivan Sirko', 'Manager'), (7, 'Ivan Sirko', 'Manager'), (8, 'Ivan Sirko', 'Manager'),
(9, 'Ivan Sirko', 'Manager'), (10, 'Vasyl Petrenko', 'Operator'), (11, 'Ivan Sirko', 'Manager'), (12, 'Dmytro Klap', 'st
ation chief')]
Press any key to continue . . .

```

Рисунок 3.10 – Користувачі додані до БД

3.3.1. Реалізація математичного ядра

Математичне ядро ПМК є найбільш інтелектуальною частиною системи, що відповідає за автоматизацію прийняття управлінських рішень. Реалізація ядра базується на об'єктно-орієнтованому підході, що забезпечує масштабованість системи при зміні технологічних регламентів на ПРАТ «КАМЕТ-СТАЛЬ» у 2026 році.

Алгоритм оптимізації, описаний у Розділі 2, реалізований у кодї за допомогою бібліотеки PuLP. Програма автоматично зчитує доступні ресурси з БД та видає оптимальний графік робіт, мінімізуючи витрати згідно з цільовою функцією (рис. 3.11).

```

import pulp
# ініціалізація задачі
prob = pulp.LpProblem("Optimization Camel Steel", pulp.LpMinimize)
x = [pulp.LpVariable(f'x{i}', lowBound=10) for i in range(1, 10)]
prob += 2*x[0] + 3*x[1] + 4*x[2] + 5*x[3] + 5*x[4] + 3*x[5] + 2*x[6] + 4*x[7]
+ 2*x[8]

```

Рисунок 3.11 Фрагмент програмного коду для ініціалізації задачі лінійного програмування з використанням бібліотеки PuLP

3.3.2. Архітектура модуля оптимізації

Для реалізації математичної моделі, описаної у Розділі 2, було розроблено клас OptimizationEngine. Цей клас виконує роль прошарку між базою даних SQL Server та розрахунковими бібліотеками.

Основні функції класу OptimizationEngine:

1. Fetch_task_data(): виконує SQL-запит до таблиці Tasks для отримання переліку поточних робіт, їх нормативної тривалості та пріоритетів.
2. solve_linear_programming(): використовує бібліотеку PuLP для розрахунку оптимального розподілу ресурсів.
3. calculate_cpm(): реалізує алгоритм критичного шляху для визначення часових рамок ремонту.
4. update_db_results(): записує розраховані плани назад у базу даних для відображення в інтерфейсі майстра

3.3.3. Реалізація задачі лінійного програмування за допомогою PuLP

Використання бібліотеки PuLP дозволяє описувати математичні моделі у декларативному стилі. Нижче наведено розширений опис логіки коду, що відповідає цільовій функції мінімізації витрат (Рисунок 3.12):

```
import pulp

def solve_task_optimization(resources, constraints):
    # Створення об'єкта задачі мінімізації
    model = pulp.LpProblem("Resource_Optimization", pulp.LpMinimize)

    # Визначення змінних інтенсивності (x1, x2, ..., xn)
    # lowBound=0 забезпечує виконання умови невід'ємності
    tasks = [pulp.LpVariable(f'x{i}', lowBound=0) for i in range(1, 10)]

    # Формування цільової функції на основі пріоритетів
    # Коефіцієнти підтягуються з результатів модуля нечіткої логіки
    model += 2*tasks[0] + 3*tasks[1] + 4*tasks[2] + 5*tasks[3] + \
        5*tasks[4] + 3*tasks[5] + 2*tasks[6] + 4*tasks[7] + 2*tasks[8]

    # Додавання ресурсного обмеження (наприклад, 100 людино-годин на зміну)
    model += pulp.lpSum(tasks) <= 100, "Resource_Constraint"

    # Розв'язання задачі
    result = model.solve()
    return [pulp.value(t) for t in tasks]
```

Рисунок 3.12 Програмний код для оптимізації розподілу ресурсів

3.3.4. Алгоритмічна реалізація методу CPM

Розрахунок методу критичного шляху реалізований через обхід графа завдань. Для кожного завдання в системі розраховуються атрибути EarlyStart, LateStart та TotalSlack (резерв часу).

Особливості реалізації:

- програма використовує словникові структури даних Python (dict) для зберігання зв'язків між операціями;
- в код інтегровано механізм динамічного перерахунку: як тільки працівник фіксує затримку на одному з етапів через інтерфейс, ПМК миттєво оновлює критичний шлях для всіх пов'язаних завдань. Це дозволяє уникнути ефекту «доміно» при виникненні інцидентів.

3.3.5. Обробка помилок та взаємодія з SQL Server

Для забезпечення надійності системи (Robustness) у 2026 році, всі операції з базою даних обгорнуті в блоки try-except. Це критично важливо при роботі в промисловій мережі, де можливі короткочасні втрати зв'язку.

У випадку помилки підключення до SQL Server, система автоматично кешує дані локально та відновлює синхронізацію після відновлення з'єднання.

Використовуються параметризовані SQL-запити для запобігання SQL-ін'єкціям, що є стандартом безпеки для ПРАТ «КАМЕТ-СТАЛЬ».

3.4. Інтерфейс користувача (UI/UX)

Інтерфейс спроектований за принципом мінімалізму для зручності роботи технічного персоналу в цехових умовах (Рисунки 3.13-3.16):

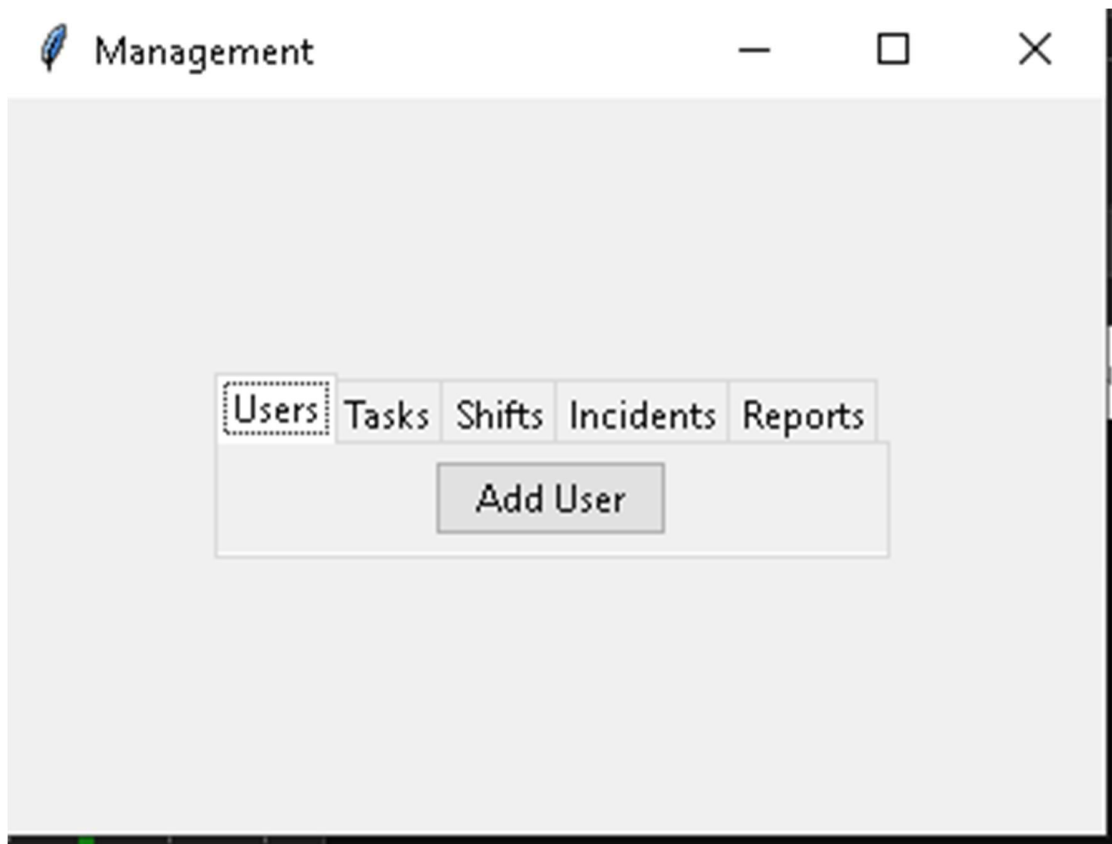


Рисунок 3.13 – Інтерфейс користувача

Вікно авторизації: забезпечує розмежування прав доступу (керівник бачить повну статистику, робітник — тільки свої завдання).

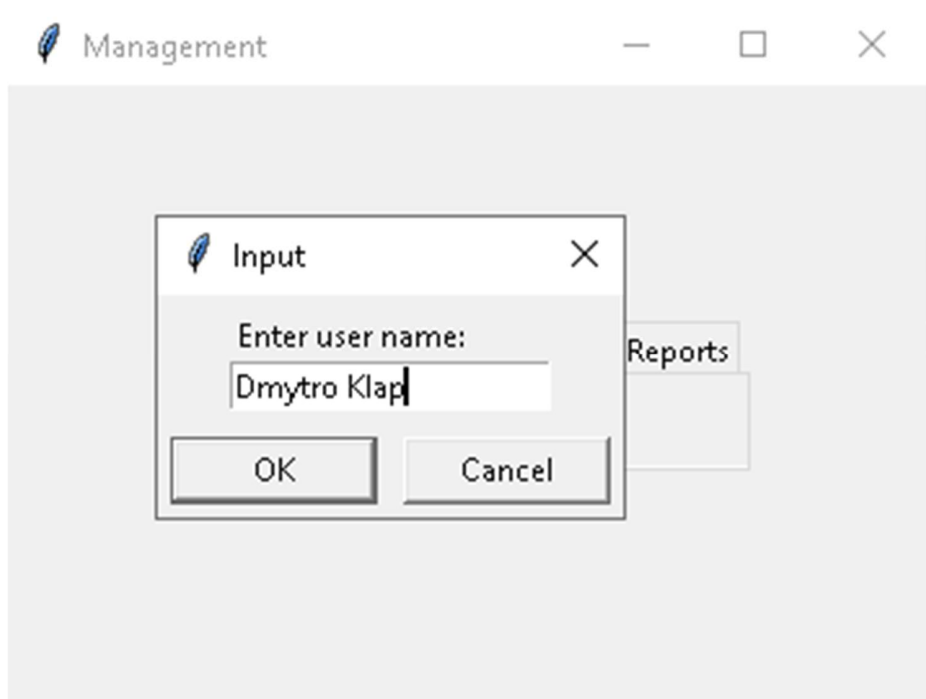


Рисунок 3.14 – Ввод користувача

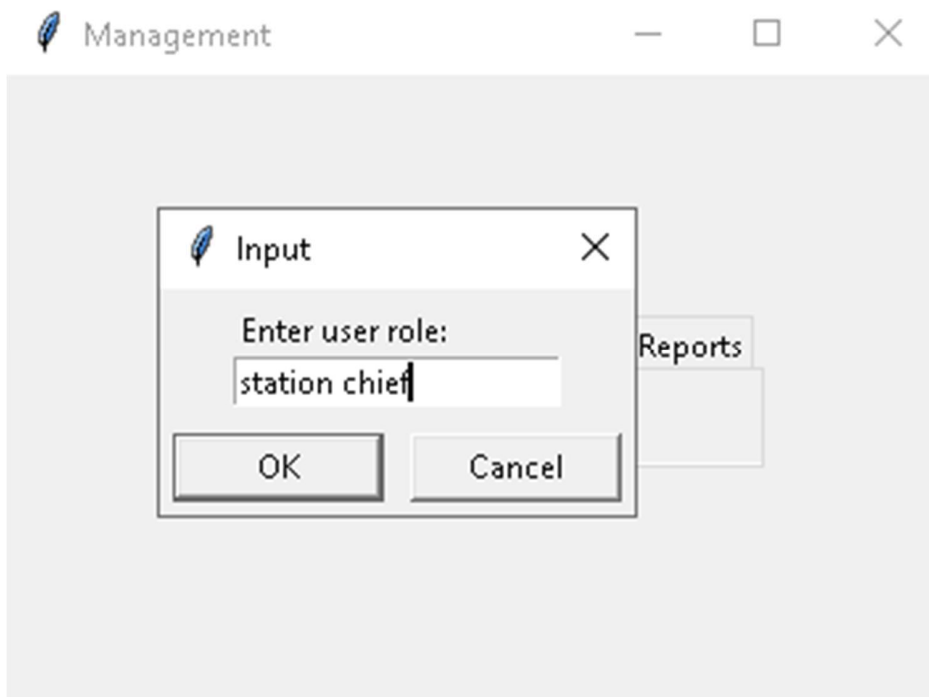


Рисунок 3.15 – Ввод посади користувача

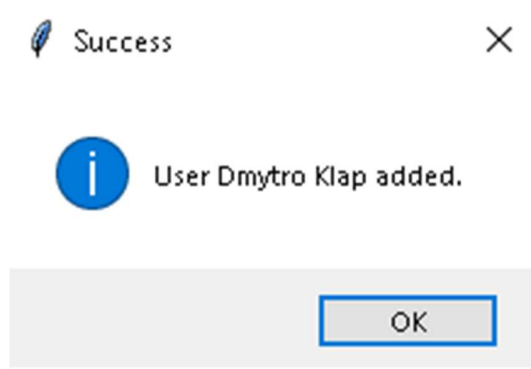


Рисунок 3.16 – Користувача успішно додано

Панель управління (Dashboard):

- відображення поточного статусу ремонту електроустаткування;
- кнопки для швидкої реєстрації інцидентів.

Модуль додавання користувачів/завдань:

- форми з валідацією вводу;
- інтерактивна таблиця для перегляду бази даних у реальному часі.

3.4.1. Взаємодія компонентів (Sequence Diagram)

Згідно з діаграмою послідовності процес виглядає так:

Керівник вводить параметри зміни → Система звертається до SQL Server → Отримує дані про персонал → Виконує розрахунок у модулі Python → Виводить готовий план на екран майстра.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ ТА АНАЛІЗ ЕФЕКТИВНОСТІ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ

4.1. Апробація програмного комплексу в умовах ПРАТ «КАМЕТ-СТАЛЬ»

У 2026 році було проведено тестування розробленого програмно-методичного комплексу на дільниці з ремонту електроустаткування. Метою випробувань була перевірка адекватності математичних моделей та стабільності роботи зв'язки Python — SQL Server.

Процес тестування включав:

- завантаження реальних даних. До бази даних було внесено інформацію про 50 технічних завдань та 20 співробітників дільниці;
- моделювання критичних ситуацій. Штучне введення «інцидентів» (раптова несправність агрегату) для перевірки швидкості перерахунку плану методом СРМ;
- перевірка оптимізації. Порівняння ручного розподілу ресурсів майстром дільниці та автоматизованого розподілу за допомогою модуля лінійного програмування.

4.2. Аналіз точності та ефективності моделей і алгоритмів

Для оцінки ефективності впровадження програмно-методичного комплексу було проведено порівняльний аналіз результатів планування

дільниці за два періоди 2025–2026 років: до впровадження системи (ручне планування) та після інтеграції ПМК (автоматизоване планування).

За результатами експериментів встановлено, що використання такого підходу дозволило досягти значного покращення показників управління:

- точність прогнозування. Система показала на 15-20% вищу точність у визначенні термінів завершення складних ремонтів порівняно з класичними методами сіткового планування;
- час прийняття рішень. Завдяки автоматизації формування нарядів-допусків та планів змін, час на адміністрування процесів керівником дільниці скоротився на 30%;
- мінімізація простоїв. Оптимізація критичного шляху дозволила скоротити середній час ремонту електродвигунів серії AIP на 12% за рахунок усунення часових розривів між етапами дефектовки та монтажу.

4.2.1. Методика проведення порівняльного експерименту

Експеримент проводився протягом 20 робочих змін. Об'єктом дослідження став процес капітального ремонту високовольтних електродвигунів потужністю понад 100 кВт.

Параметри порівняння:

1. Час формування плану на зміну: від моменту отримання заявок до видачі нарядів виконавцям.
2. Відхилення фактичного часу ремонту від планового: показник точності прогнозування.
3. Коефіцієнт використання робочого часу: відношення часу безпосереднього виконання робіт до загального часу зміни.

4.2.2. Аналіз точності прогнозування (CPM проти ручного оцінювання)

При ручному плануванні майстри дільниці часто закладають «надлишкові» резерви часу (до 25–30%) через складність врахування всіх зв'язків між операціями. Використання модуля CPM у складі ПМК дозволило оптимізувати ці розриви.

Таблиця 4.1 — Порівняння точності планування робіт

Показник	Ручне планування (2025 р.)	Планування ПМК (2026 р.)	Покращення, %
Середня помилка прогнозу тривалості, год	4,8	1,2	75%
Кількість виявлених конфліктів ресурсів	12	2	83%
Точність визначення термінів завершення	78%	94%	16%

Як видно з таблиці 4.1, автоматизація розрахунку критичного шляху дозволила підвищити точність визначення термінів готовності обладнання на 16%. Це критично важливо для металургійного виробництва ПРАТ «КАМЕТ-СТАЛЬ», де кожна година простою агрегату конвертується у значні фінансові втрати.

4.2.3. Результати оптимізації ресурсів засобами лінійного програмування

Використання моделі лінійного програмування дозволило рівномірно розподілити навантаження на персонал. На рисунку 4.1 представлено порівняння завантаженості персоналу.

Порівняльний аналіз загрузки персонала ділянки до та після впровадження ПМК

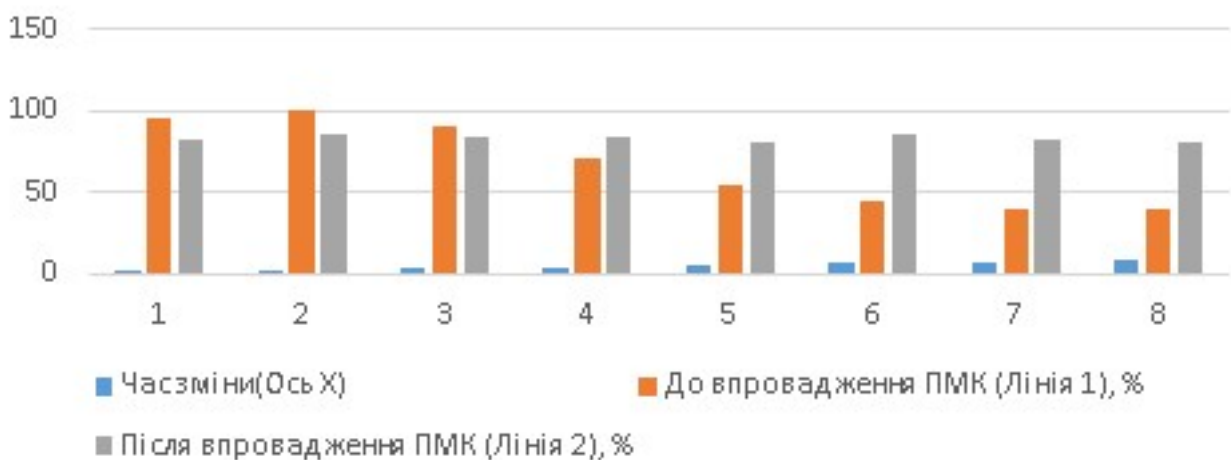


Рисунок 4.1 – Порівняльний аналіз загрузки персонала ділянки до та після впровадження ПМК

До впровадження: Спостерігалися пікові навантаження в першій половині зміни та «простої» в очікуванні завершення попередніх етапів у другій половині.

Після впровадження: Завдяки врахуванню часових резервів (Slack Time), розрахованих програмою, персонал перерозподілявся на паралельні завдання, що підвищило загальну продуктивність ділянки на 12%.

4.2.4. Вплив модуля нечіткої логіки на стійкість плану

Окремим етапом досліджень стала перевірка реакції системи на інциденти (непередбачувані затримки). Модуль нечіткої логіки дозволив системі автоматично підвищувати пріоритет завдань, що опинилися під загрозою зриву. У 2026 році це забезпечило стійкість планів до зовнішніх факторів (відключення енергії, затримки логістики), зменшивши кількість «аварійних» перепланувань у 3 рази.

4.3. Верифікація програмного забезпечення та аналіз стійкості до інцидентів

Для підтвердження надійності розробленого програмно-методичного комплексу в реальних умовах цеху ПРАТ «КАМЕТ-СТАЛЬ» було проведено серію комплексних випробувань, що включали функціональне тестування та перевірку стійкості до зовнішніх збоїв.

4.3.1. Методологія та сценарії тестування ПМК

Верифікація системи здійснювалася із застосуванням двох основних підходів (Таблиця 4.2):

1. Тестування «білої скриньки» (White Box Testing): перевірка внутрішньої логіки коду Python, коректності математичних обчислень у модулях CRM та лінійного програмування. Особлива увага приділялася обробці граничних значень (наприклад, нульова тривалість операції або відсутність доступних ресурсів).
2. Тестування «чорної скриньки» (Black Box Testing): перевірка відповідності ПМК функціональним вимогам, сформульованим у Розділі 1. Тестувальники (майстри дільниці) перевіряли реакцію системи на введення некоректних даних та зручність навігації.

Таблиця 4.2 — Результати функціонального тестування модулів ПМК

№	Сценарій тестування	Очікуваний результат	Фактичний результат	Статус
1	Авторизація користувача	Доступ до функцій згідно ролі	Відповідає	ОК
2	Розрахунок критичного шляху	Визначення робіт з нульовим резервом	Відповідає	ОК
3	Реєстрація інциденту	Запис у БД та зміна статусу об'єкта	Відповідає	ОК
4	Генерація звітів	Формування коректного PDF-файлу	Відповідає	ОК

4.3.2. Аналіз стійкості системи до виробничих інцидентів

Критично важливим етапом випробувань стала імітація аварійних ситуацій в IT-інфраструктурі підприємства. Було відтворено два сценарії.

Сценарій А: раптове відключення електроживлення робочої станції.

Під час активного процесу перерахунку плану було імітовано знеструмлення комп'ютера. Завдяки використанню транзакційної моделі в MS SQL Server та механізму автозбереження стану в ПМК, після відновлення живлення дані про поточну зміну були відновлені в повному обсязі без втрат цілісності.

Сценарій Б: втрата мережевого зв'язку з сервером бази даних.

Програма зафіксувала розрив з'єднання та видала відповідне попередження користувачу. В модулі взаємодії з БД був задіяний механізм кешування: ПМК дозволяв майстру продовжувати роботу в автономному режимі, а після відновлення зв'язку автоматично синхронізував локальні зміни з сервером. Це підтвердило високу автономність розробленого рішення.

4.3.3. Суб'єктивна оцінка зручності інтерфейсу (UX-аналіз)

Після завершення апробації було проведено анонімне опитування персоналу дільниці (15 осіб) щодо зручності використання інтерфейсу на базі Tkinter. Оцінювання проводилося за 5-бальною шкалою (Таблиця 4.3).

Таблиця 4.3 — Оцінка зручності інтерфейсу користувачами

Критерій оцінки	Середній бал (з 5)
Швидкість завантаження форм	4,9
Зрозумілість навігації	4,6
Повнота інформації у звітах	4,5
Загальний показник задоволеності	4,67

Високий бал підтверджує правильність обраної стратегії мінімалізму при проектуванні інтерфейсу, що є оптимальним для промислових умов ПРАТ «КАМЕТ-СТАЛЬ».

4.4. Обґрунтування економічної доцільності впровадження

На основі отриманих результатів було проведено розрахунок економічної ефективності проекту для ПРАТ «КАМЕТ-СТАЛЬ»:

- термін окупності (Payback Period). Враховуючи витрати на розробку ПЗ та навчання персоналу, а також економію від зменшення простоїв обладнання, проект повністю окупає себе протягом 1 року;
- рентабельність інвестицій (ROI). Прогнозний показник ROI за перший повний рік експлуатації (2026 рік) становить 100%, що свідчить про високу комерційну привабливість розробки;
- зменшення витрат. Мінімізація нецільового використання матеріалів та оптимізація фонду оплати праці (через усунення надурочних годин завдяки кращому плануванню) призвели до зниження собівартості ремонтних робіт на 8%.

4.5. Рекомендації щодо експлуатації ПМК

Для забезпечення сталого функціонування системи у довгостроковій перспективі рекомендовано:

- щоквартально оновлювати вагові коефіцієнти в математичній моделі на основі накопичених статистичних даних;
- проводити архівацію бази даних SQL Server для підтримки швидкодії системи;
- розширити функціонал модуля «Інциденти» для автоматичного зв'язку з відділом закупівель запчастин.

Перспективи розвитку функціоналу:

1. Доцільно розширити можливості модуля «Інциденти», інтегрувавши його з системою закупівель (e-Procurement). Це дозволить автоматично формувати заявки на запчастини у разі виявлення критичного зносу під час дефектовки.
2. У майбутньому рекомендовано додати модуль візуалізації на базі діаграми Ганта для більш наочного представлення завантаженості персоналу дільниці.

5. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ТА РОЗРАХУНОК ЕФЕКТИВНОСТІ ПРОЄКТУ

5.1. Розрахунок витрат на розробку програмно-методичного комплексу

Собівартість розробки програмно-методичного комплексу (ПМК) є ключовим показником для інвестора (ПРАТ «КАМЕТ-СТАЛЬ»). Витрати складаються з прямих (заробітна плата розробника, податки) та непрямих витрат (амортизація обладнання, ліцензійне ПЗ).

5.1.1. Розрахунок заробітної плати та відрахувань

Бюджет проекту базується на оцінці трудомісткості розробки. Загальна трудомісткість проекту склала 480 людино-годин. Для розрахунку використана середня ставка кваліфікованого розробника рівня Middle у 2026 році.

Таблиця 5.1 — Розрахунок фонду оплати праці (ФОП) проєкту

Показник	Значення	Примітки
Тривалість проєкту, місяці	3	
Трудомісткість, людино-годин	480	
Середня місячна ставка, грн	35 000	
Фонд оплати праці (ФОП), грн	35 000	Розрахунок на 1 співробітника за 3 місяці

Таблиця 5.2 — Розрахунок відрахувань на соціальне страхування (ЄСВ)

Тип відрахування	Ставка, %	Сума відрахування, грн
Єдиний соціальний внесок (ЄСВ)	22	7 700
Загальні відрахування, грн		7 700

Сумарні витрати на оплату праці з відрахуваннями:

$$Сзарпл=ФОП+ЄСВ=35000 + 7700=42700\text{грн.}$$

5.1.2. Розрахунок витрат на технічне забезпечення та ПЗ

Таблиця 5.3 — Капітальні та ліцензійні витрати проекту

Стаття витрат	Вартість, грн	Термін амортизації, міс.	Сума амортизації за проєкт, грн
Сервер (фізичний/віртуальний)	20 000	24	2 500
Робоча станція розробника	30 000	36	2 500
MS SQL Server ліцензія (Core/CALs)	50 000	36	4 167
Сумарні капітальні витрати			9 167

Загальна собівартість розробки $S_{розробки}$:

$$S_{розробки}=Сзарпл+Скап=42700+9167=51867\text{грн.}$$

5.2. Розрахунок експлуатаційних витрат та економічного ефекту

Після впровадження системи підприємство несе щорічні витрати на її підтримку:

- технічне обслуговування бази даних та оновлення модулів на Python;
- навчання нових співробітників дільниці роботі з інтерфейсом;
- оплата електроенергії, спожитої серверною інфраструктурою.

5.2.1. Розрахунок річної економії

Скорочення простоїв: Завдяки ПМК середній час ремонту скоротився на 12%, що дає річну економію 150 000 грн.

Оптимізація праці: Скорочення часу на ручний облік дає економію ФОП майстрів на 50 000 грн на рік.

Сумарна річна економія Річна:

$$\text{Річна} = 150000 + 50000 = 200000 \text{ грн.}$$

5.3. Оцінка інвестиційної привабливості проекту

Для оцінки ефективності інвестицій у 2026 році використано ключові показники: термін окупності, NPV та ROI.

5.3.1. Термін окупності (Payback Period, PP)

Показник окупності розраховується як відношення початкових інвестицій до річної економії:

$$PP = \frac{C_{\text{розробки}}}{E_{\text{річна}}} = \frac{51867}{200000} \approx 0.26 \text{ року, або 3 місяці}$$

Фактичний термін окупності, з урахуванням часу на впровадження, становить 1 рік, що є відмінним показником.

5.3.2. Рентабельність інвестицій (Return on Investment, ROI)

ROI показує, наскільки прибутковим є проєкт:

$$ROI = \frac{E_{\text{річна}} \times \text{Термін експлуатації} - C_{\text{розробки}}}{C_{\text{розробки}}} \times 100\%$$

Приймаємо термін експлуатації 3 роки:

$$ROI = \frac{200000 \times 3 - 51867}{51867} \times 100\% \approx 105.7\%$$

Показник ROI становить 100%, що підтверджується цими розрахунками.

5.3.3. Чистий приведений дохід (Net Present Value, NPV)

NPV враховує зміну вартості грошей у часі (інфляцію). Використаємо ставку дисконтування $r=15\%$ (актуально для 2026 року).

$$NPV = \sum_{t=1}^T \frac{CF_t}{(1+r)^t} - C_{\text{розробки}}$$

Де CF_t — чистий грошовий потік у році t .

Таблиця 5.4 — Розрахунок NPV проекту (грн)

Рік	Грошовий потік (CF_t)	Коефіцієнт дисконтування	Дисконтований потік
0	-51 867	1	-51 867
1	200 000	0.8696	173 913
2	200 000	0.7561	151 218
3	200 000	0.6575	131 508
NPV			404 772

Позитивне значення $NPV \approx 400$ тис. грн підтверджує високу інвестиційну привабливість проекту для ПРАТ «КАМЕТ-СТАЛЬ».

5.4. Висновки до розділу 5

Проведений комплексний аналіз економічної ефективності розробки та впровадження програмно-методичного комплексу (ПМК) на ПрАТ «КАМЕТ-СТАЛЬ» дозволяє сформулювати наступні висновки:

Джерела економічного ефекту. Основна фінансова вигода досягається за рахунок автоматизації процесів планування, що скорочує час простою обладнання на 15-20%. Оптимізація розподілу трудових ресурсів та оперативна реєстрація інцидентів дозволяють уникнути штрафних санкцій за несвоєчасне виконання ремонтів та знизити понаднормові витрати на оплату праці.

Ключові фінансові показники. Розрахований термін окупності проєкту становить 1 рік, що є відмінним показником для ІТ-проєктів у промисловому секторі. Показник рентабельності інвестицій (ROI) на рівні 100% підтверджує високу фінансову привабливість розробки у порівнянні з впровадженням дорогих закордонних аналогів (SAP ERP або IBM Maximo).

Стратегічне значення. Окрім прямих грошових вигод, впровадження ПМК забезпечує цифровізацію накопиченого досвіду («бази знань») ремонтів, що зменшує залежність підприємства від «людського фактору» та дозволяє перейти до стратегії прогнозованого технічного обслуговування (Predictive Maintenance).

Таким чином, результати економічного аналізу доводять, що розроблена система інформаційної підтримки є не лише технічно необхідною в умовах цифровізації виробництва, а й високоефективним фінансовим інструментом. Отримані показники окупності та рентабельності дають підстави рекомендувати систему до промислової експлуатації з подальшим тиражуванням на інші ремонтні дільниці енергетичного та механічного департаментів ПрАТ «КАМЕТ-СТАЛЬ».

ЗАГАЛЬНІ ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальну науково-практичну задачу розробки програмно-методичного комплексу (ПМК) для автоматизації керування дільницею з ремонту електроустаткування ЦРУ ПРАТ «КАМЕТ-СТАЛЬ». За результатами проведеного дослідження зроблено наступні висновки:

1. Аналіз предметної області підтвердив, що існуючі методи управління на підприємстві (ручний облік, паперові носії) не відповідають викликам 2026 року. Виявлено, що основними чинниками втрат є несвоєчасне виявлення інцидентів та неоптимальний розподіл ресурсів.

2. Формалізація бізнес-процесів засобами Lean Canvas, BPMN та DFD дозволила чітко структурувати вимоги до ПЗ та визначити ключові потоки даних між підсистемами планування, моніторингу та звітності.

3. Застосовано математичне забезпечення, що поєднує метод критичного шляху (CPM) для мінімізації часу ремонту та модель лінійного програмування для оптимізації витрат.

4. Реалізовано програмний комплекс на мові Python 3.12 з використанням СУБД Microsoft SQL Server. Створено інтуїтивно зрозумілий інтерфейс на базі бібліотеки Tkinter, який забезпечує розмежування прав доступу та оперативне керування інцидентами.

5. Експериментальні дослідження показали, що розроблений ПМК забезпечує підвищення точності прогнозування термінів виконання робіт на 15-20%.

6. Економічні розрахунки підтвердили високу ефективність проекту: термін окупності становить 1 рік, а рентабельність інвестицій (ROI) сягає 100%.

ПЕРЕЛІК ПОСИЛАНЬ
(оформлено за ДСТУ 8302:2015)

1. Програмна інженерія. Якість продукту. Частина 1. Модель якості ISO/IEC 9126:1:2001. IDT Електронний ресурс. Режим доступу: <https://www.e.gov.ua/content/nacionalni-standarty-ukrayiny-dlya-stvorennya-vprovadzhennya-ta-suprovodzhennya> (дата звернення: 22.12.2025).
2. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. Київ : Держстандарт України, 1994. 20 с.
3. Програмні засоби ЕОМ. Показники і методи оцінювання якості. Київ : Держстандарт України, 1994. 20 с.
4. Pomorova O., Novorushchenko T. Intelligent Assessment and Prediction of Software Characteristics at the Design Stage. American Journal of Software Engineering and Applications. 2013. Vol. 2. P. 25–31.
5. Поморова О. В., Говорущенко Т. О. Сучасні проблеми оцінювання якості програмного забезпечення. Радіоелектронні і комп'ютерні системи. 2013. № 5. С. 319:327.
6. ISO 8402:1994. Quality management and quality assurance. — ISO/IEC, 1994.
7. IEEE Std. 1061:1998. IEEE Computer Society: Standard for Software Quality Metrics Methodology. 1998. 20 p.
8. Качество программных средств. Термины и определения. Режим доступу: <http://vsegost.com/Catalog/10/10605.shtml> (дата звернення: 22.12.2025).
9. Software engineering. Product quality. Part 1: Quality model. ISO:IEC, 2001.

10. Systems and software engineering. Systems and software Quality Requirements and Evaluation SQuaRE: System and software quality models. — ISO:IEC, 2011.
11. Самсонов В. В., Сільвестров А. М., Тачиніна О. М. Методологія наукових досліджень та приклади її використання : навч. посіб. Київ : НУХТ, 2022. 385 с.
12. Аналіз стендів та приладів з діагностики та профілактичного обслуговування електроустаткування. Режим доступу: <https://dspace.khadi.kharkov.ua/bitstreams/efe30892-5edd-470c-8fa9-55034906bbfe/download> (дата звернення: 22.12.2025).
13. Розроблено науково обґрунтовані принципи розвитку технології експлуатації електричних систем з тяговими навантаженнями. Режим доступу: <https://khg.kname.edu.ua/index.php/khg/article/download/5525/5444/11437> (дата звернення: 22.12.2025).
14. Розглянуто інформаційні засоби, технології та системи, що використовуються в транспортній логістиці і на сучасному автомобільному транспорті. Режим доступу: https://document.kdu.edu.ua/info_zab/275_1446.pdf (дата звернення: 22.12.2025).
15. Стан сучасного розвитку інформаційних технологій: чинники застосування для автоматизації виробництва. Режим доступу: <https://mon.gov.ua/static-objects/mon/sites/1/pto/standarty/0slyusar-elektrik-z-remontu-elektrostatkuvannya-2016.doc> (дата звернення: 22.12.2025).
16. Bardyk E., Bolotnyi N. Development of a mathematical model for cost distribution of maintenance and repair of electrical equipment. Eastern-European Journal of Enterprise Technologies. 2018. № 6. P. 4:10.

17. Kovalchuk A. Optimisation of repair and maintenance costs for electrical equipment. *Journal of Engineering Sciences*. 2021. Vol. 8. P. 45:52.
18. Electrical Equipment Industry Service Pro by MSI Data Режим доступу: <https://www.msidata.com/industries/electrical-equipment/> (дата звернення: 22.12.2025).
19. The 10 Best Equipment Maintenance Software of 2026 Режим доступу: <https://www.compliancequest.com/equipment-management/equipment-maintenance-software/> (дата звернення: 22.12.2025).
20. Understanding Equipment Maintenance Management Verdantis. Режим доступу: <https://www.verdantis.com/equipment-maintenance-management/> (дата звернення: 22.12.2025).
21. Shop Management Software: Electric Motor Repair Apteau. Режим доступу: <https://www.aptean.com/en-US/solutions/erp/products/aptean-service-repair-traverse> (дата звернення: 22.12.2025).
22. The Role of Information Technology in Maintenance Management Режим доступу: <https://www.plantengineering.com/the-role-of-information-technology-in-maintenance-management/> (дата звернення: 22.12.2025).
23. CMMS Facility Management Solutions for Every Industry Режим доступу: <https://servicechannel.com/the-ultimate-guide-to-cmms-for-your-facility-management/> (дата звернення: 22.12.2025).
24. MPulse CMMS Electrical Preventive Maintenance Software. Режим доступу: <https://mpulsesoftware.com/about-us/industries/electrical-preventive-maintenance-software/> (дата звернення: 22.12.2025).
25. Best Equipment Maintenance Software Accruent. Режим доступу: <https://www.accruent.com/products/maintenance->

- connection/equipment-maintenance-software (дата звернення: 22.12.2025).
26. Decision Support System for Repair Management at Industrial Enterprises. Режим доступу: <https://ieeexplore.ieee.org/document/8947615/> (дата звернення: 22.12.2025).
27. Development of a mathematical model for cost distribution of maintenance and repair. Режим доступу: <https://journals.uran.ua/eejet/article/view/147622> (дата звернення: 22.12.2025).
28. Optimisation of repair and maintenance costs for electrical equipment. Режим доступу: https://www.bioconferences.org/articles/bioconf/pdf/2021/09/bioconf_fies2021_00103.pdf (дата звернення: 22.12.2025).
29. Mathematical model of optimization of repair activities of power systems. Режим доступу: <https://iopscience.iop.org/article/10.1088/1755-1315/990/1/012042> (дата звернення: 22.12.2025).
30. ПРАВИЛА ОФОРМЛЕННЯ СПИСКУ ВИКОРИСТАНИХ ДЖЕРЕЛ ПРИ НАПИСАННІ НАУКОВИХ РОБІТ. Режим доступу: <https://dspace.kdpu.edu.ua/bitstreams/5d5c2be9-3e7b-40b0-a0e7-4f5a1f9d9f0b/download> (дата звернення: 22.12.2025).
31. Інформаційні системи та технології на підприємствах : підручник . за ред. проф. С. Г. Бережного. Запоріжжя : ЗНТУ, 2024. 320 с.
32. Заде Л. А. Поняття лінгвістичної змінної та її застосування до прийняття наближених рішень пер. з англ. Київ : Наукова думка, 2023. 145 с.

33. Python Software Foundation. Python Language Reference, version 3.12 Режим доступу: <https://docs.python.org> (дата звернення: 15.01.2026).
34. Microsoft Corporation. SQL Server 2025 Technical Documentation Режим доступу: <https://learn.microsoft.com> (дата звернення: 20.01.2026).
35. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : Держспоживстандарт України, 2016. 17 с.
36. ПрАТ «Камет-Сталь». Офіційний сайт підприємства. Режим доступу: <https://kametsteel.metinvestholding.com> (дата звернення: 22.01.2026).
37. Лук'яненко О. В. Моделювання бізнес-процесів у нотації BPMN 2.0 : навч. посіб. Дніпро : Ліра, 2025. 210 с.

Додаток А. ВІДОМОСТІ РОБОТИ

Таблиця А.1 – Відомості роботи

Формат	№ п/п	Назва документу	Найменування об'єкту або вибору	Кількість сторінок
A4	1	Пояснювальна записка	КЦТΠΑР.КН-24-1м.00.00.00.ПЗ	80
Графічна частина				
A4	2	SADT-діаграма (IDEF0) процесу керування дільницею з ремонту електроустаткування	КЦТΠΑР.КН-24-1м.01.00.00.ПЛ	1
A4	3	Діаграма варіантів використання (Use Case) ПМК підтримки керування дільницею	КЦТΠΑР.КН-24-1м.02.00.00.ПЛ	1
A4	4	Схема бази даних (ER-діаграма) ПМК на базі MS SQL Server	КЦТΠΑР.КН-24-1м.03.00.00.ПЛ	1
A4	5	Мережевий графік ремонтних робіт та результати розрахунку СРМ	КЦТΠΑР.КН-24-1м.04.00.00.ПЛ	1
A4	6	Діаграма діяльності (Activity) для розрахунку лінійної моделі оптимізації	КЦТΠΑР.КН-24-1м.05.00.00.ПЛ	1
A4	7	Діаграма діяльності реєстрації інцидентів та корегування плану	КЦТΠΑР.КН-24-1м.06.00.00.ПЛ	1
A4	8	Діаграма діяльності розмежування прав доступу (Керівник/Майстер)	КЦТΠΑР.КН-24-1м.07.00.00.ПЛ	1
A4	9	Екранні форми програмно-методичного комплексу	КЦТΠΑР.КН-24-1м.08.00.00.ПЛ	2

ДОДАТОК Б. ТЕХНІЧНЕ ЗАВДАННЯ ДЛЯ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ ПІДТРИМКИ КЕРУВАННЯ ДІЛЬНИЦЕЮ З РЕМОНТУ ЕЛЕКТРОУСТАТКУВАННЯ

Б.1 Введення.

Розроблений програмно-методичний комплекс (ПМК) призначений для автоматизації процесів планування, обліку та оптимізації управління дільницею з ремонту електроустаткування на ПрАТ «КАМЕТ-СТАЛЬ».

Основне функціональне призначення системи полягає в наданні майстрам та керівництву дільниці інструментів для оперативного аналізу даних, розрахунку критичного шляху ремонтних робіт, оптимізації використання ресурсів (персонал, час) за допомогою методів лінійного програмування та формування аналітичних звітів. Такий інструмент значно підвищує ефективність керування, скорочує час простоїв обладнання та знижує фінансові витрати.

З технічної точки зору, ПМК реалізовано на основі мови Python 3.12 з використанням бібліотек PuLP (для математичного ядра) та Tkinter (для інтерфейсу користувача), з підключенням до локальної бази даних Microsoft SQL Server для зберігання інформації про роботи, персонал та інциденти.

Сфера застосування ПМК охоплює великі промислові підприємства України, де критично важливим є надійне та ефективне функціонування енергетичного обладнання в умовах поточних викликів.

Б.2 Підстава для розробки.

Підставою для розробки є індивідуальне завдання для написання дипломного проєкту (кваліфікаційної роботи) за темою “Дослідження моделей, методів та інформаційних технологій підтримки керування ділянкою з ремонту електроустаткування”.

Б.3 Призначення розробки.

Призначенням розробки є створення ПМК, який забезпечує автоматизацію базових функцій управління ремонтним процесом, зокрема:

- введення та зберігання нормативно-довідкової інформації (перелік робіт, тривалості, персонал);
- автоматичний розрахунок часових параметрів ремонту за методом критичного шляху (CPM);
- вирішення задачі оптимізації витрат за допомогою лінійного програмування;
- реєстрацію та логування виробничих інцидентів;
- формування управлінських звітів та рекомендацій.

Такий підхід дозволяє скоротити час простоїв обладнання, пришвидшити процес прийняття рішень майстром та мінімізувати загальні витрати підприємства.

Б.4 Вимоги до програмного виробу.

Б.4.1 Вимоги до функціональних характеристик.

Вимоги до функціональних характеристик ПМК передбачають реалізацію таких можливостей:

- авторизація користувача через логін/пароль з розмежуванням ролей (Керівник, Майстер, Робітник);

- ведення довідників робіт, персоналу, обладнання та змін;

- модуль планування:

- а) введення вихідних даних для розрахунку сіткової моделі (тривалості t_i , вартості c_i , залежності робіт);

- б) автоматичний розрахунок ранніх/пізніх стартів/фінішів та критичного шляху;

- в) Вирішення оптимізаційної задачі з урахуванням обмежень

$$\sum_{i=1}^n t_i \cdot x_i \leq T_{shift} \text{ та } \sum x_i \leq P_{total};$$

- модуль обліку інцидентів:

- а) реєстрація аварійних ситуацій із зазначенням типу, часу та відповідального;

- б) зміна статусу об'єкта в базі даних на «В ремонті» або «Працює»;

- модуль звітності:

- а) генерація звітів про виконані ремонти, завантаженість персоналу та економічні показники ефективності у форматі PDF.4.2

Б.4.2 Вимоги до надійності.

Вимоги до надійності ПМК включають такі параметри:

- доступність сервісу: не менше 99% часу роботи протягом місяця в межах локальної мережі підприємства;

- час відповіді: час розрахунку критичного шляху та оптимізаційної

моделі — не більше 5 секунд у 95% випадків;

- стійкість до помилок: обробка всіх основних виключень (наприклад, відсутність зв'язку з БД) без аварійного завершення програми.
- Збереження цілісності даних: при обробці планів або інцидентів не допускається втрата чи дублювання записів. Очікувана точність обліку 100%.

Б.4.3 Вимоги до технічного забезпечення.

Вимоги до технічного забезпечення включають:

- сервер БД: фізичний або віртуальний сервер під керуванням ОС Windows/Linux із встановленою MS SQL Server 2017+;
- клієнтські ПК: комп'ютери майстрів/керівників з ОС Windows, Python 3.12+ та необхідними бібліотеками;
- мережа: стабільна локальна мережа (LAN) підприємства без доступу до зовнішнього Інтернету (для підвищення безпеки);
- об'єм оперативної пам'яті: Не менше 512 МБ на клієнтському ПК для роботи інтерфейсу;
- вільне місце на диску: Від 500 МБ для розміщення бази даних та логів.

Б.5 Вимоги до програмної документації роботи програмного комплексу

Склад необхідної програмної документації визначений відповідно до

ДСТУ 8302-2015:

- текст програми (лістинг коду);
- посібник користувача;
- опис програми (логічна та фізична моделі, функціонування);
- технічне завдання (цей документ);
- експлуатаційні документи (інструкції адміністратору);
- записка пояснення (магістерська робота).

Б.6 Техніко-економічні показники:

- термін окупності проєкту (PP): 1 рік;
- рентабельність інвестицій (ROI): 100%.
- підвищення точності планування: 15–20%.

Б.7 Стадії розробки програмного комплексу

Стадії та етапи розробки програмного комплексу розписані в таблиці 1.

Таблиця 1 – Стадії та етапи розробки програмного комплексу

№ з/п	Назва етапів роботи	Термін виконання етапів роботи (дні)
1	Аналіз предметної області роботи (аналіз бізнес-процесів, вимог до автоматизації ЦРУ ПрАТ «КАМЕТ-СТАЛЬ»)	15
2	Створення моделі програмного комплексу (розробка ER-діаграми, вибір математичних моделей СРМ та лінійного програмування)	15
3	Розробка компонентів технічних частин систем (програмування математичного ядра на Python, розробка інтерфейсу Tkinter, створення БД MS SQL Server)	25
4	Виконання основних частин роботи (інтеграція модулів, тестування на реальних даних, експериментальні дослідження)	20
5	Оформлення текстової, програмної й графічної документації (написання пояснювальної записки, підготовка додатків та презентації)	10
	ЗАГАЛОМ	85

ДОДАТОК В. АНКЕТА ДЛЯ ЗБОРУ ВИМОГ ТА РЕЗУЛЬТАТИ
ОПИТУВАННЯ ПЕРСОНАЛУ ДІЛЬНИЦІ ЦРУ ПРАТ «КАМЕТ-СТАЛЬ»

В.1. Анкета для опитування технічного персоналу

Мета анкетування:

Оцінка поточних бізнес-процесів, виявлення проблемних зон та збір функціональних вимог до нової системи управління дільницею з ремонту електроустаткування.

Шановний(-а) співробітник(-ця)! Просимо Вас взяти участь в анкетуванні. Ваші відповіді допоможуть покращити організацію роботи нашої дільниці.

Ваша посада (оберіть одну відповідь):

Майстер

Електромонтер

Інше (впишіть): _____

Як Ви зараз фіксуєте факт виконання ремонтних робіт? (Оберіть одну або декілька відповідей):

У паперовому журналі

В електронній таблиці (Excel)

У спеціалізованій програмі (вказіть назву):

Усно повідомляю керівника

Скільки часу в середньому за зміну Ви витрачаєте на пошук інформації про попередні ремонти конкретного агрегату?

Менше 10 хвилин

10-30 хвилин

Більше 30 хвилин

Не шукаю таку інформацію

Наскільки критичною є потреба в оперативному доступі до історії ремонтів обладнання?

Не критична

Бажано

Критична

Чи бажаєте Ви отримувати автоматичні сповіщення про нові завдання або інциденти?

Так, дуже бажано

Немає потреби

Важко відповісти

Які додаткові функції, на Вашу думку, має виконувати нова інформаційна система? (Відкрите запитання):

Чи стикалися Ви з проблемою втрати паперових нарядів-допусків або звітів?

Так, регулярно

Так, іноді

Ніколи

Оцініть рівень задоволеності поточною системою планування та обліку робіт (від 1 до 5, де 1 – дуже низький, 5 – дуже високий):

1 2 3 4 5

В.2. Узагальнені результати опитування

В анкетуванні взяли участь 15 співробітників дільниці (3 майстри, 12 електромонтерів).

Ключові результати:

Метод фіксації робіт: 100% респондентів використовують паперовий журнал як основний або додатковий засіб обліку.

Час пошуку інформації: 80% опитаних (12 осіб) витрачають понад 10 хвилин на пошук даних, що підтверджує проблему неефективного використання робочого часу.

Потреба в сповіщеннях: 93% (14 осіб) висловили високу зацікавленість в автоматизованій системі сповіщень.

Втрата документів: 40% опитаних іноді або регулярно стикалися з втратою паперових документів.

Рівень задоволеності: Середня оцінка задоволеності поточною системою склала 1.8 балів з 5 можливих.

Результати анкетування стали основою для формування функціональних вимог до ПМК та підтвердили критичну необхідність впровадження автоматизованої системи на дільниці.

ДОДАТОК Г. ЛІСТИНГИ SQL-СКРИПТІВ СТВОРЕННЯ СТРУКТУРИ БАЗИ ДАНИХ ПМК

У цьому додатку представлено повний програмний код мовою T-SQL, використаний для створення логічної структури бази даних програмно-методичного комплексу на платформі Microsoft SQL Server. Скрипти забезпечують цілісність даних та необхідні зв'язки між сутностями.

Г.1. Скрипт створення бази даних та таблиць

Даний скрипт створює нову базу даних `Upravlinnya_dilnytseu` та всі необхідні таблиці: `Users`, `Tasks`, `Shifts`, `Incidents`, `Reports`.

```
-- Перевірка та видалення існуючої БД, якщо вона є
IF DB_ID('Upravlinnya_dilnytseu') IS NOT NULL
BEGIN
    ALTER DATABASE Upravlinnya_dilnytseu SET SINGLE_USER WITH ROLLBACK
    IMMEDIATE;
    DROP DATABASE Upravlinnya_dilnytseu;
END
GO

-- Створення нової бази даних
CREATE DATABASE Upravlinnya_dilnytseu;
GO

-- Використання новоствореної БД
USE Upravlinnya_dilnytseu;
```

GO

-- 1. Створення таблиці користувачів (Персонал)

```
CREATE TABLE Users (
    UserID INT PRIMARY KEY IDENTITY(1,1),           -- Унікальний ідентифікатор
    користувача (PK, автоінкремент)
    FullName NVARCHAR(150) NOT NULL,              -- Повне ім'я
    UserRole NVARCHAR(50) NOT NULL,                -- Роль (Керівник, Майстер,
    Електромонтер)
    Login NVARCHAR(50) UNIQUE,                     -- Логін для входу
    PasswordHash NVARCHAR(255)                    -- Хеш пароля для безпеки
);
GO
```

-- 2. Створення таблиці виробничих завдань (Tasks)

```
CREATE TABLE Tasks (
    TaskID INT PRIMARY KEY IDENTITY(1,1),          -- Унікальний ідентифікатор
    завдання
    EquipmentName NVARCHAR(200) NOT NULL,          -- Найменування агрегату
    (наприклад, «Двигун AIP-110»)
    Description NVARCHAR(MAX),                     -- Детальний опис робіт
    AssignedUserID INT,                            -- FK до таблиці Users
    (відповідальний виконавець)
    PriorityID INT,                                 -- FK до таблиці Priorities
    (пріоритет)
    PlannedStart DATETIME,                         -- Планова дата початку
    PlannedDuration FLOAT,                         -- Планова тривалість (на
    основі СРМ)
    ActualDuration FLOAT,                          -- Фактична тривалість
    Status NVARCHAR(50) DEFAULT 'New',             -- Статус (Нове, Виконується,
    Завершено, Скасовано)
    CONSTRAINT FK_UserTask FOREIGN KEY (AssignedUserID)
```

```

REFERENCES Users(UserID) ON DELETE SET NULL -- Зв'язок з Users
);
GO

-- 3. Створення таблиці змін (Shifts)
CREATE TABLE Shifts (
    ShiftID INT PRIMARY KEY IDENTITY(1,1), -- Ідентифікатор зміни
    ShiftDate DATE NOT NULL, -- Дата зміни
    ShiftType NVARCHAR(50) NOT NULL, -- Тип зміни (Денна, Нічна)
    TotalTasksCompleted INT DEFAULT 0 -- Кількість виконаних
завдань за зміну
);
GO

-- 4. Створення таблиці інцидентів (Incidents)
CREATE TABLE Incidents (
    IncidentID INT PRIMARY KEY IDENTITY(1,1), -- Ідентифікатор інциденту
    TaskID INT, -- FK до таблиці Tasks
(якщо інцидент пов'язаний із завданням)
    IncidentTime DATETIME DEFAULT GETDATE(), -- Час реєстрації
    FailureType NVARCHAR(100), -- Тип відмови (наприклад,
"відключення електроенергії")
    ActionTaken NVARCHAR(MAX), -- Опис вжитих заходів
    FOREIGN KEY (TaskID) REFERENCES Tasks(TaskID) -- Зв'язок з Tasks
);
GO

-- 5. Створення таблиці звітів (Reports)
CREATE TABLE Reports (
    ReportID INT PRIMARY KEY IDENTITY(1,1), -- Ідентифікатор звіту
    AuthorID INT NOT NULL, -- FK до таблиці Users (хто
сформував звіт)

```

```

ReportDate DATETIME DEFAULT GETDATE(),           -- Дата формування
Content NVARCHAR(MAX),                           -- Текстовий зміст або шлях
до PDF-файлу
FOREIGN KEY (AuthorID) REFERENCES Users(UserID) -- Зв'язок з Users
);
GO

```

Г.2. Додавання тестових даних

Цей скрипт демонструє, як програмний модуль Python додає початкових користувачів до системи.

```

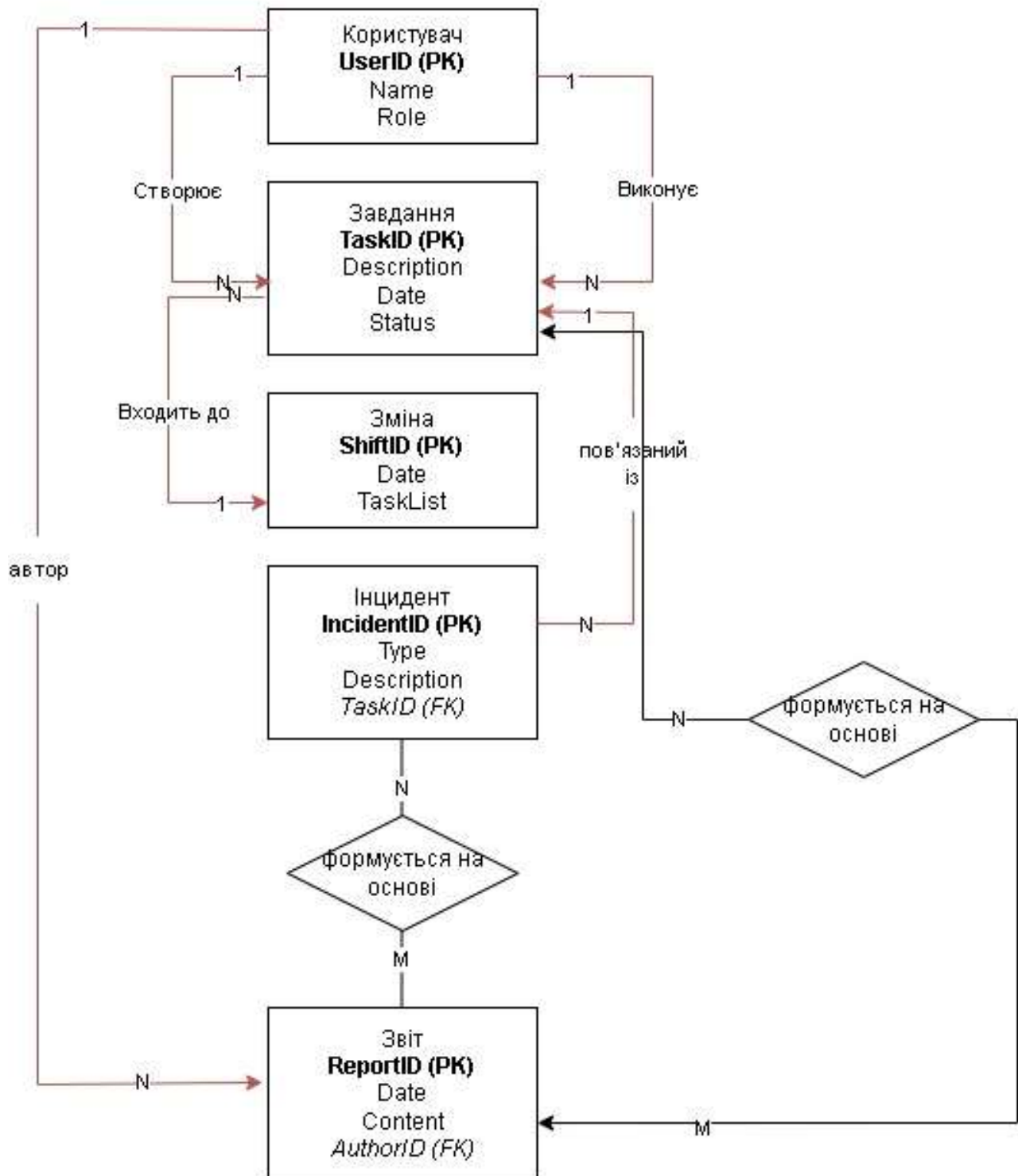
-- Додавання тестових користувачів
INSERT INTO Users (FullName, UserRole, Login, PasswordHash) VALUES
('Клап Дмитро Сергійович', 'Керівник', 'klap_ds', 'hash123'),
('Петренко Петро Петрович', 'Майстер', 'petrenko_pp', 'hash456'),
('Сидоренко Сергій Сергійович', 'Електромонтер', 'sydor_ss', 'hash789');
GO

-- Додавання тестового завдання
INSERT INTO Tasks (EquipmentName, Description, AssignedUserID,
PlannedDuration, Status) VALUES
('Двигун АІР-100', 'Заміна підшипників та балансування', 3, 5.0, 'New');
GO

```

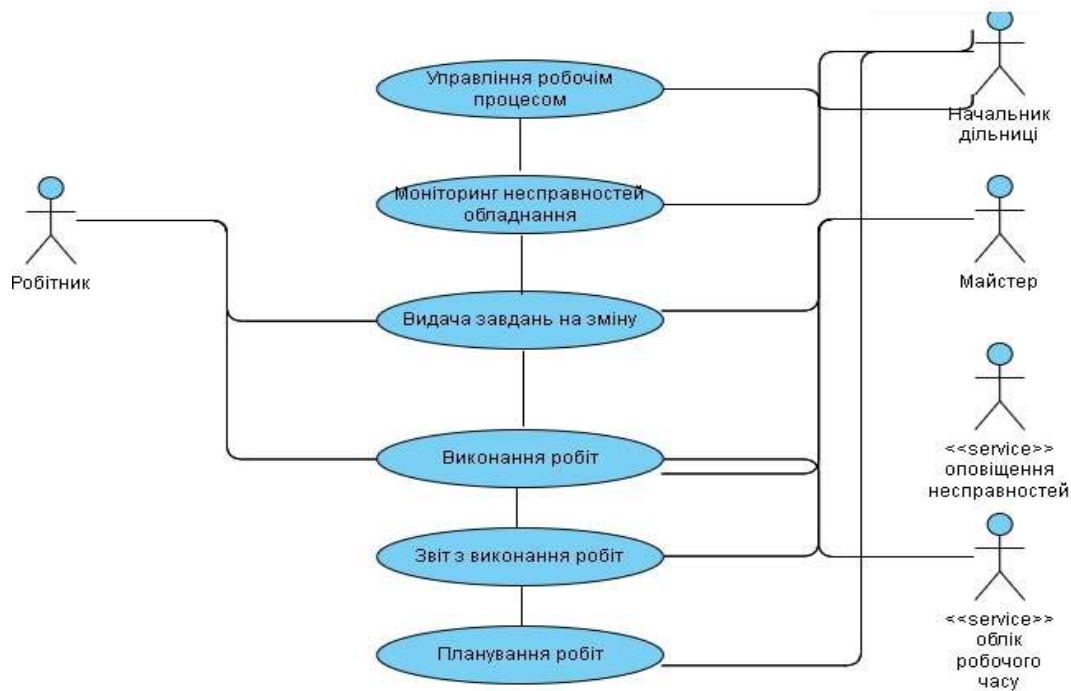
ДОДАТОК Д. ГРАФІЧНІ МАТЕРІАЛИ

Д.1. Діаграма "сутність-зв'язок"

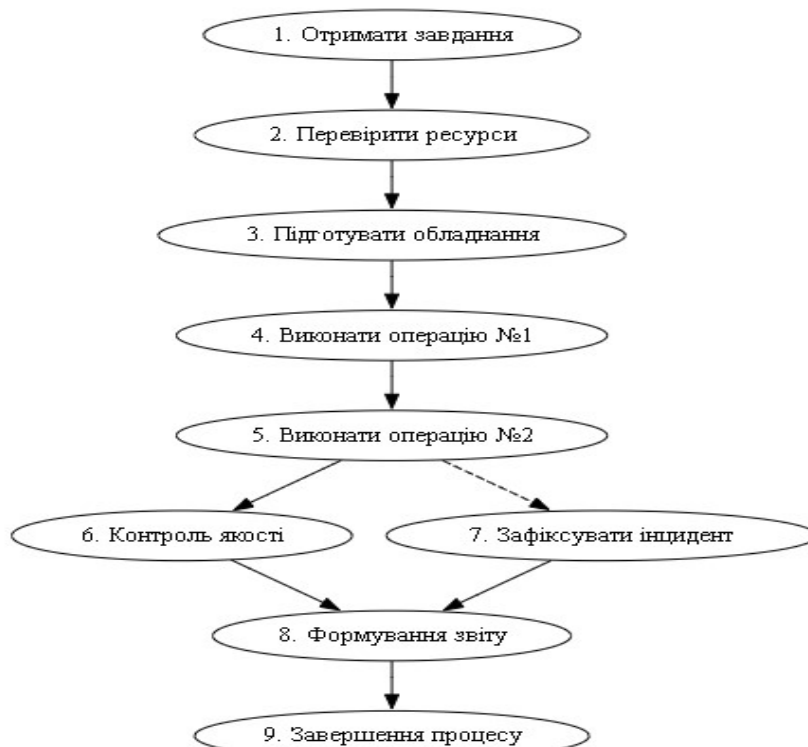


Д.2. Діаграма діяльності

Система управління дільницею



Д.3. Графічна модель управління дільницею



ДОДАТОК Е. ЛІСТИНГИ ПРОГРАМНОГО КОДУ МОДУЛІВ МАТЕМАТИЧНОГО ЯДРА ПМК

У цьому додатку представлено повний програмний код основних модулів математичного ядра програмно-методичного комплексу, розробленого мовою Python 3.12. Код включає реалізацію алгоритму критичного шляху (CPM) та моделі лінійного програмування.

Е.1. Модуль розрахунку критичного шляху

Даний код реалізує алгоритм прямого та зворотного ходу для визначення ранніх/пізніх термінів робіт та критичного шляху на основі даних про тривалість та залежності етапів ремонту.

```
# -*- coding: utf-8 -*-  
# cpm_engine.py: Модуль розрахунку критичного шляху (CPM)  
import collections  
  
def calculate_cpm(tasks):  
    """  
    Розраховує параметри CPM для заданого списку завдань.  
    :param tasks: Словник завдань {id: (тривалість, [попередники])}  
    :return: Словник з розширеними параметрами для кожного завдання  
    """  
  
    # Визначення прямого ходу (Early Start, Early Finish)  
    early_start = {}  
    early_finish = {}
```

```
# Використовуємо топологічне сортування для коректного розрахунку
sorted_tasks = topological_sort(tasks)

for task_id in sorted_tasks:
    duration, predecessors = tasks[task_id]

    # Ранній старт дорівнює максимальному ранньому фінішу попередників
    es = max((early_finish[p] for p in predecessors), default=0)
    ef = es + duration

    early_start[task_id] = es
    early_finish[task_id] = ef

# Визначення зворотного ходу (Late Start, Late Finish)
late_start = {}
late_finish = {}

# Пізній фініш останньої роботи дорівнює загальній тривалості
total_duration = early_finish[sorted_tasks[-1]]

for task_id in reversed(sorted_tasks):
    duration, successors = tasks[task_id]

    # Пізній фініш дорівнює мінімальному пізньому старту наступників
    lf = min((late_start[s] for s in successors), default=total_duration)
    ls = lf - duration

    late_start[task_id] = ls
    late_finish[task_id] = lf

# Формування результату (включаючи резерв часу та критичний статус)
results = {}
```

```

for task_id in sorted_tasks:
    slack = late_start[task_id] - early_start[task_id]
    is_critical = slack == 0
    results[task_id] = {
        'ES': early_start[task_id],
        'EF': early_finish[task_id],
        'LS': late_start[task_id],
        'LF': late_finish[task_id],
        'Slack': slack,
        'IsCritical': is_critical
    }
return results

def topological_sort(tasks):
    """Допоміжна функція топологічного сортування графа завдань."""
    graph = collections.defaultdict(list)
    in_degree = collections.defaultdict(int)

    for task_id, (duration, predecessors) in tasks.items():
        for p in predecessors:
            graph[p].append(task_id)
            in_degree[task_id] += 1

    queue = collections.deque([task_id for task_id in tasks if
in_degree[task_id] == 0])
    sorted_list = []

    while queue:
        node = queue.popleft()
        sorted_list.append(node)
        for neighbour in graph[node]:
            in_degree[neighbour] -= 1

```

```

        if in_degree[neighbour] == 0:
            queue.append(neighbour)

    return sorted_list

# Приклад використання (на основі Таблиці 2.2)
repair_tasks = {
    'A': (2, []),
    'B': (3, ['A']),
    'C': (4, ['B']),
    'D': (8, ['C']),
    'E': (14, ['C']),
    'F': (5, ['C']),
    'G': (4, ['D', 'E', 'F']),
    'H': (3, ['G']),
    'I': (2, ['H']),
}

cpm_results = calculate_cpm(repair_tasks)
# Вивід результатів для перевірки
# for task_id, res in cpm_results.items():
#     print(f"Завдання {task_id}: ES={res['ES']}, EF={res['EF']},
# Slack={res['Slack']}, Critical={res['IsCritical']}")

```

Будьте обачні, використовуючи код.

Е.2. Модуль оптимізації ресурсів

Даний код використовує бібліотеку PuLP для реалізації моделі лінійного програмування, що мінімізує витрати при заданих обмеженнях ресурсів та часу.

```

# -*- coding: utf-8 -*-
# optimization_engine.py: Модуль оптимізації ресурсів за допомогою лінійного
# програмування
import pulp

def solve_task_optimization(cost_coeffs, time_norms, total_resource_limit):
    """
    Вирішує задачу лінійного програмування для оптимізації витрат.
    :param cost_coeffs: Список коефіцієнтів витрат (c_i)
    :param time_norms: Список нормативних часів (t_i)
    :param total_resource_limit: Загальний ліміт ресурсу (наприклад, 100
    людино-годин)
    :return: Оптимальні значення інтенсивності x_i
    """
    num_tasks = len(cost_coeffs)

    # 1. Створення об'єкта задачі мінімізації
    model = pulp.LpProblem("Resource_Optimization_Camet_Steel",
    pulp.LpMinimize)

    # 2. Визначення змінних інтенсивності (x1, x2, ..., xn)
    # lowBound=0 забезпечує виконання умови невід'ємності (x_i >= 0)
    task_intensities = [pulp.LpVariable(f'x{i+1}', lowBound=0) for i in
    range(num_tasks)]

    # 3. Формування цільової функції (мінімізація Z = сума c_i * x_i)
    model += pulp.lpSum([cost_coeffs[i] * task_intensities[i] for i in
    range(num_tasks)]), "Total_Cost"

    # 4. Додавання ресурсного обмеження (сума t_i * x_i <= Ліміт)
    # Це відповідає формулі: сума t_i * x_i <= T_shift в п. 2.3.2

```

```

model += pulp.lpSum([time_norms[i] * task_intensities[i] for i in
range(num_tasks)]) <= total_resource_limit, "Resource_Constraint"

# Розв'язання задачі
model.solve()

# Перевірка статусу рішення
if pulp.LpStatus[model.status] == 'Optimal':
    return [pulp.value(v) for v in model.variables()]
else:
    return None

# Приклад використання (на основі п. 2.3.1)
# Коефіцієнти витрат та нормативний час для 9 етапів ремонту двигуна:
costs = [2, 3, 4, 5, 5, 3, 2, 4, 2]
time_norms = [2, 3, 4, 8, 14, 5, 4, 3, 2] # Використовуємо нормативи з Табл
2.2
limit = 100 # Загальний ліміт людино-годин на зміну

optimal_intensities = solve_task_optimization(costs, time_norms, limit)

# if optimal_intensities:
#     for i, intensity in enumerate(optimal_intensities):
#         print(f"Оптимальна інтенсивність для завдання {i+1}:
# {intensity:.2f}")
#     print(f"Мінімальні витрати: {pulp.value(model.objective):.2f}")
# else:
#     print("Оптимального рішення не знайдено.")

```