



# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітньо-професійна програма Комп'ютерні науки

(назва)

## ЗАТВЕРДЖУЮ

Завідувач кафедри Інформаційних технологій, штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ

“ 28 ” квітня 2025 року

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Костіної Анастасії Володимирівни

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення вебпорталу аварійної служби ліфтової компанії "ОТІС"

керівник роботи Грама Михайло Петрович, доцент, кандидат технічних наук

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

затверджені наказом закладу вищої освіти від 28 квітня 2025 року № 254-кв

2. Строк подання здобувачем роботи 30 травня 2025 року

3. Вихідні дані до роботи Вихідні дані про компанію «ОТІС», Бізнес-процес прийому заявки, Дані про ліфти, Дані про працівників, Структура підрозділу технічного обслуговування, Навчальні матеріали

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Розділ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

Розділ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

Розділ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ВЕБ ДОДАТКУ

5. Перелік графічного матеріалу:

Організаційна структура підприємства, установи чи організації; Функціональні моделі; Моделі та схеми бази даних; Скриншоти інтерфейсу користувача;

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	PhD, ст. викладач Грама М. П.	28.04.2025	12.05.25
2	PhD, ст. викладач Грама М. П.	28.04.2025	16.05.25
3	PhD, ст. викладач Грама М. П.	28.04.2025	18.05.25

7. Дата видачі завдання \_\_\_\_\_ 28 квітня 2025 року \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз структури підприємства	01.05.2025	Виконано
2	Постановка задачі	03.05.2025	Виконано
3	Створення бізнес-моделі процесів	05.05.2025	Виконано
4	Огляд рішень-аналогів	06.05.2025	Виконано
5	Створення технічного завдання	08.05.2025	Виконано
6	Проектування бази даних	10.05.2025	Виконано
7	Розробка веб додатку	15.05.2025	Виконано
8	Створення інструкції користувача	16.05.2025	Виконано
9	Оформлення пояснювальної записки	28.05.2025	Виконано
10	Підготовка презентації	29.05.2025	Виконано

**Здобувач**

\_\_\_\_\_ (підпис)

**Анастасія КОСТИНА**

\_\_\_\_\_ (ім'я та прізвище)

**Керівник роботи**

\_\_\_\_\_ (підпис)

**Михайло ГРАМА**

\_\_\_\_\_ (ім'я та прізвище)

## АНОТАЦІЯ

Кваліфікаційна робота на тему «Розробка веб порталу аварійної служби ліфтової компанії "ОТІС"» присвячена розробленню сучасного веб рішення, що допоможе оперативно опрацьовувати заявки з аварійних питань. У ході роботи проаналізовано організаційну структуру департаменту сервісних операцій, наявність існуючих програмних рішень, їх недоліки та розроблено веб-портал з інтуїтивно зрозумілим інтерфейсом для користувачів. Програмне рішення реалізовує прямий взаємозв'язок між Клієнтом та Механіком.

Система створена за допомогою веб технологій таких як PHP, MySQL, JavaScript, HTML та CSS.

Робота містить 107 сторінок, 8 таблиць, 53 графічного матеріалу, 5 додатків, 37 літературних джерел.

**Ключові слова:** ВЕБ-ПОРТАЛ, АВАРІЙНА СЛУЖБА, ЛІФТ, PHP, MYSQL, БАЗА ДАНИХ, АВТОРИЗАЦІЯ, ЗАЯВКИ, ДИСПЕТЧЕР, КЛІЄНТ, ІНТЕРФЕЙС КОРИСТУВАЧА, ОТІС.

## SUMMARY

The qualification work on the topic "Development of a web portal for the emergency service of the elevator company "OTIS"" is dedicated to the development of a modern web solution that will help to promptly process applications for emergency issues. During the work, the organizational structure of the service operations department, the availability of existing software solutions, their shortcomings were analyzed and a web portal with an intuitive user interface was developed. The software solution implements a direct relationship between the Client and the Mechanic.

The system was created using web technologies such as PHP, MySQL, JavaScript, HTML and CSS.

The work contains 107 pages, 8 tables, 53 graphic material, 5 appendices, 37 literary sources.

**Keywords:** WEB PORTAL, EMERGENCY SERVICE, ELEVATOR, PHP, MYSQL, DATABASE, AUTHORIZATION, APPLICATIONS, DISPATCHER, CLIENT, USER INTERFACE, OTIS.

## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	11
1.1. Історія та базова характеристика Компанії Otis Elevator Company .....	11
1.2. Організаційна структура, роль і взаємодія підрозділів .....	12
1.3. Аналіз нинішнього стану комп'ютеризації на підприємстві ПрАТ «ОТІС» ...	17
1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів.....	29
1.5. Огляд існуючих рішень для розв'язання виявлених проблем в компанії "OTIS" .....	36
1.6. Обґрунтування доцільності проектування й розроблення вебпорталу аварійної служби ліфтової компанії ПрАТ «ОТІС» .....	41
1.7. Постановка задачі.....	43
1.8. Висновки до першого розділу.....	43
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ.....	45
2.1. Загальні положення.....	45
2.2. Призначення і цілі створення системи.....	45
2.3. Характеристика об'єкта автоматизації .....	46
2.4. Вимоги до системи.....	46
2.5. Склад і зміст робіт зі створення системи.....	60
2.6. Порядок контролю і приймання системи .....	60
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію .....	61
2.8. Вимоги до документації .....	61
2.9. Джерела розробки .....	62
РОЗДІЛ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	63
3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту .....	63
3.2. Проєктування та створення бази даних .....	66
3.3. Реалізація функцій системи.....	70

	7
3.4. Інструкція користувача.....	90
3.5. Тестування програмного продукту .....	99
ВИСНОВКИ.....	103
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	104
ДОДАТКИ.....	108
Додаток А. Організаційна структура департаменту сервісних операцій.....	108
Додаток Б. Функціональні моделі .....	109
Додаток В. Моделі та схеми бази даних .....	113
Додаток Г. Скриншоти інтерфейсу користувача .....	115
Додаток Д. Фрагменти коду програми.....	125

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	База даних
НР	Нормальна робота ліфтів
ОС	Операційна система
ПЗ	Програмне забезпечення
СУБД	Система управління базами даних
ТО	Технічне обслуговування
CSS	Мова стилів
HTML	Мова розмітки
JavaScript	Мова інтерактивності
MySQL	Система керування реляційними базами даних
PHP	Мова програмування для розробки веб-додатків

## ВСТУП

У сучасному світі, де кожна людина не може обійтись без смартфона, а документи вже оформлюються через додаток в телефоні, особливим стає впровадження інформаційних технологій в сферу інфраструктурних підприємств. Зокрема це стосується ліфтового обслуговування. Щодня в Україні експлуатуються тисячі ліфтів, в сотнях застрягають люди, через аварійні відключення світла, що вимагає швидкої реакції персоналу на усунення проблеми. Це неможливо досягти без злагодженої комунікації між заявником та виконавцем. Саме сучасні програмні рішення дозволяються автоматизувати обробку звернень та підвищити ефективність роботи ліфтових підприємств.

На підприємстві ПрАТ «ОТІС» вже використовуються програмні рішення для обліку заявок та ліфтів, але зростаюча хвиля аварійних відключень та застрягань створила необхідність в створенні додаткового способу комунікації, між клієнтами та ліфтовою компанією. Саме для цього був створений веб-портал, який буде розглянутий в цій кваліфікаційній роботі. Завдяки йому планується пришвидшити комунікацію, обробку аварійних заявок, покращити моніторинг виконаних робіт та зменшити навантаження на диспетчерів.

Тема кваліфікаційної роботи полягає в розробленні сучасного веб-порталу, який дозволить клієнту, користуючись зручним інтерфейсом, залишати заявки, слідкувати за їх статусами.

Науковою новизною в даній темі виступає використання новаторської системи передачі заявки напряму від клієнта до механіка, з мінімальним, або повністю відсутнім, залученням диспетчера, що дозволить збільшити пропускну спроможність заявок та збільшенню задоволення клієнтів.

Веб-додаток створено з використанням сучасних веб-технологій – PHP, JavaScript, HTML, CSS і MySQL.

Метою кваліфікаційної роботи є створення повноцінного веб-порталу для аварійної служби ліфтової компанії, який має забезпечити часткову або повну автоматизацію бізнес процесу створення, передачу та закриття заявок, що знизить навантаження на диспетчера. Для цього був проведений аналіз предметної області,

сформоване технічне завдання, щодо функціональності системи, побудована база даних, реалізований інтерфейс та логічна частина веб-додатку з урахуванням чотирьох рівнів доступу: клієнт, диспетчер, механік та адміністратор.

## РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1. Історія та базова характеристика Компанії Otis Elevator Company

Компанія Otis Elevator Company "OTIS" — це культовий бренд з багатою історією. Одна з найстаріших та найбільших компаній-виробників підйомного устаткування, до них входять: ескалатори, ліфти та рухомі доріжки.

Компанія "OTIS" була заснована у 1853 року в Нью-Йорку американським винахідником і підприємцем, який увійшов в історію як творець безпечного ліфта. Він розробив механізм, який блокував кабінку при обриві канату [33].

З 1900 року до сьогодні, компанія "OTIS" перетворилася у процвітаючу компанію, яка зосередилася на розширенні, якості та інноваціях.

За цей час, компанія відкрила біля 300 представництв по всьому світу. Вона охопила такі сфери діяльності:

- Виробництво підйомного обладнання;
- Обслуговування та модернізацію для загального і індивідуального використання;
- Інтелектуальні рішення та цифрові технології;
- Інфраструктурні проекти;
- Соціальна відповідальність та сталий розвиток.

В Україні компанія "OTIS" працює з 1992 року. Перше представництво було відкрито в місті Києві. Офіційно було взяте на облік в [26]:

Дата реєстрації 24.09.1992

Назва компанії: ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО "ОТІС" (ПрАТ «ОТІС»)

За 32 роки ПрАТ «ОТІС» в Україні дало потужний поштовх для заміни радянських рішень!

Зараз представництво ПрАТ «ОТІС» відкрито в Одесі, Івано-Франківську, Дніпрі, Чернігові, Краматорську. Працює понад 370 співробітників. Географія

поставки і сервісу в Україні охоплює більше 30 міст, обласних центрів, смт і розширює свої кордони в праці.

Метою Компанії "OTIS" у всіх країнах є прагнення бути лідером у сфері вертикального транспорту та забезпечення безпечного, ефективного комфорту пасажиром.

Місією Компанії "OTIS" у всіх країнах це створення сучасних, енергоефективних технологій, що дозволяють зменшити споживання ресурсів і викиду CO<sub>2</sub>, які допоможуть сприяти захисту екології для майбутнього людства.

## **1.2. Організаційна структура, роль і взаємодія підрозділів**

### **1.2.1. Організаційна структура OTIS**

Організаційна структура – це система взаємопов'язаних управлінських та виробничих ланок, та послідовність потоків ресурсів, функціональних зв'язків, у виробничій системі, все це забезпечує досягнення стратегічних завдань підприємства.

Структура має тісно пов'язану взаємодію з її елементами [5, с. 179; 7, с. 189]: рівнями управління, цілями, функціями, організацією управлінських процесів, чисельним і професійно-кваліфікаційним складом, а також зі ступенем централізації управлінських функцій та побудовою інформаційних зв'язків. В рамках цієї структури відбувається циркуляція інформаційних потоків та ухвалення управлінських рішень, до чого залучають менеджерів всіх рівнів та професійної спеціалізації.

Компанія "OTIS" в Америці та Європі застосовує матричну організаційну структуру, яка поєднує функціональні та регіональні підрозділи.

Матрична структура управління [8, с. 79, 19] – це структура, за якої, поряд із традиційними лінійними керівниками та функціональними підрозділами, створюються тимчасові проектні групи. Ці групи формуються зі спеціалістів різних функціональних підрозділів та залучаються до створення нових продуктів або послуг.

Основні елементи її структури:

#### 1. Функціональні підрозділи

Загалом Компанія "OTIS" має окремі департаменти, які відповідають за ключові напрямки діяльності:

- Дослідження та розробки (R&D);
- Виробництво;
- Продажі та маркетинг;
- Обслуговування та підтримка;
- Фінанси;
- Юридичний відділ;
- Управління персоналом;

#### 2. Регіональні підрозділи

Входять такі регіони: Північна Америка, Європа, Близький Схід та Африка (EMEA), Азійсько-Тихоокеанський регіон, Латинська Америка.

#### 3. Централізоване керівництво OTIS

Знаходиться у Фармінгтоні, це штат Коннектикут в США. До нього входять: Голова та президент Джуді Маркс; Рада директорів; Головні директора по країнам: CFO, CTO, CMO.

#### 4. Дочірні компанії та філії

Компанія "OTIS" до 1970 року мала 34 компанії за межами США, а сьогодні має понад 300 офісів по всьому світу.

Ця організаційна структура дозволяє Материнській компанії "OTIS" ефективно управляти глобальними процесами, адаптуватися до місцевих ринків і зберігати контроль над стратегічними напрямками.

### **1.2.2. Організаційна структура департаменту з сервісних операцій**

#### **ПрАТ «ОТІС»**

Більш детальноше, відокремимо і розглянемо структуру організації одного з департаментів сервісу в компанії ПрАТ «ОТІС» в Україні на рисунку 1.1.



Рисунок 1.1 - Організаційна структура в ПрАТ «ОТІС»

1) Директор департаменту сервісу це фахівець, який відповідає за організацію та управління обслуговуванням клієнтів, а також за підтримання високого рівня сервісу та задоволеності клієнтів [17].

Обов'язки директора департаменту сервісу:

- Забезпечення узгодженої роботи, яка пов'язана з виконанням основних завдань високого ступеня складності;
- Розподілення обов'язків між Начальником дільниці та Начальником аварійної служби;
- Забезпечення розробки проектів, нормативно-правових актів з питань, що стосуються діяльності департаменту.
- Перевірка та дотримання стандартів сервісного обслуговування;
- Контроль якості надання послуг.

2) Начальник дільниці відповідає за планове технічне обслуговування та виконання замовлень.

Обов'язки Начальника дільниці:

- Контроль роботи механіків;
- Контроль дотримання виконання планових обслуговувань ліфтів;
- Організація та розподіл робіт між механіками.

3.1) Лінійний механік виконує технічне обслуговування та ремонт ліфтового обладнання.

Обов'язки Лінійного механіка:

- Виконання планових та аварійних ремонтів;
- Ведення журналу з планових та аварійних ремонтів;
- Виконання модернізацій;
- Забезпечення безпечного користування ліфтом пасажирями;
- Випуск людей під час застрягання;
- Інформування про результати виконання заявки;
- Заміна зношених деталей.

3.2) Аварійний механік займається виїздом на заявки у разі надходження виклику в неробочий час лінійного механіка, або як підстрахування механіка, якщо в нього велике завантаження.

Обов'язки Аварійного механіка:

- Випуск людей під час застрягання;
- Усунення причин неполадок, якщо це можливо;
- Інформування про результати виконання виклику.

4) Начальник аварійної служби відповідає за реалізацію оперативного реагування на аварійні ситуації.

Обов'язки Начальника аварійної служби:

- Координація роботи диспетчерів з лінійними та аварійними механіками;
- Контроль своєчасного виконання заявок;
- Оформлення договорів на диспетчеризацію;
- Аналіз аварійних випадків та шукання рішення для їх запобігання.

5) Диспетчер несе відповідальність за прийом заявок від клієнтів та передачі лінійним або аварійним механікам.

Обов'язки диспетчера:

- Прийом викликів та їх реєстрація в базі даних;
- Передача інформації про поломку механікам;
- Перевірка виконання ремонту;
- Інформування пасажирів про причини зупинки.

### 1.2.3. Взаємодія структурних посадових осіб в департаменті з сервісних операцій ПрАТ «ОТІС»

Взаємодія всередині компанії є основою для побудови ефективної роботи на всьому ланцюжку для вирішення поставленої мети.

Вона дозволить:

- Вчасно розподіляти виклики між технічним обслуговуванням та аварійною;
- Дотримуватись термінів і контролю виконання робіт;
- Покращити звітність і прибутковість компанії.

В таблиці 1.1 надана взаємодія структурних посадових осіб в департаменті з сервісних операцій в ПрАТ «ОТІС».

*Таблиця 1.1. Взаємодія структурних посадових осіб в департаменті з сервісних операцій в ПрАТ «ОТІС»*

Отримувач	Відправник	Яка інформація надходить?	Яка інформація передається?
Директор сервісу	Начальник дільниці, начальник аварійної служби, диспетчер	Скарги від клієнтів, прецеденти на змінах, аналітика роботи, брак матеріалів	Рішення про дії, зміни, стратегічні плани, оновлення матеріалів
Начальник дільниці	Директор сервісу, механіки, диспетчер	План робіт, заявки на зупинки, розподіл обов'язків	Графіки ТО, технічні звіти
Начальник аварійної служби	Директор сервісу, аварійні механіки, диспетчер	Інформація про аварії, статуси викликів	Координація аварійних виїздів, звіти

Механік ТО	Начальник дільниці, диспетчер	Заявки на обслуговування, маршрути	Статус виконання робіт, запити на ремонт
Аварійний механік	Начальник аварійної служби, диспетчер	Аварійні виклики, місцезнаходження	Статус ліквідації аварії, потреба в підтримці
Диспетчер	Клієнти, автоматизована система	Заявки від користувачів, зупинки ліфтів, скарги	Направлення викликів механікам, інформування начальників

Підрозділи технічного обслуговування та аварійної служби є взаємопов'язані між собою багатьма бізнес процесами, які спрямовані на збільшення ефективності використання та безпеки ліфтового обладнання.

### 1.3. Аналіз нинішнього стану комп'ютеризації на підприємстві ПрАТ «ОТІС»

#### 1.3.1. Програми диспетчеризації

На підприємстві ПрАТ «ОТІС», в диспетчерському відділі, були виявленні та дослідженні 2 діючі системи диспетчеризації, які допомагають диспетчерам слідкувати за роботою ліфтів дистанційно, перезавантажувати їх та мати зв'язок з пасажирями.

Першою програмою є ОДС «ПРОМІНЬ», що являє собою комплекс для диспетчеризації всіх систем будинку (рис. 1.2). Програма для диспетчеризації це та сама кнопка в ліфті з дзвіночком. Якщо будинок оформлював договір на її підключення, то під час натискання відбувається зв'язок з цією програмою, або її аналогами, про один яких пізніше, але все залежить саме від того обладнання, яке було встановлене на кабіні.

Налаштування Керівництво користувача V3.5.13

**ПРОМІНЬ** Виробничо-конструкторська фірма (044) 405-55-11  
УКРАЇНА м. Київ www.promin.in.ua

Оператор: 1  
Змін операт: 1  
Заблокувати: 1  
11:35

Пошук будинку >>   Пошук № Тел. GSM:

буд. № 1	буд. № 5	"ДК" Вимкнено автоматично <input type="button" value="Зняти виклики ДК"/> <input type="button" value="Закрити"/>		буд. № 7/1	буд. № 7/3
буд. № 19/1	буд. № 25/1	буд. № 29/1	буд. № 33	буд. № 12 Г	буд. № 14 Е
буд. № 2	буд. № 6	буд. № 8-А	буд. № 4	буд. № 2	буд. № 24
буд. № 20	буд. № 45	буд. № 5	буд. № 7	буд. № 4 Секція 1	буд. № 4 Секція 2
буд. № 61/2	буд. № 2Б	буд. № 4Б	буд. № 1-Б	Міст № 1 (14.1 км)	Міст № 2 (14.2 км)
Міст № 4 (17 км)	Міст № 5 (28 км)	Міст № 6 (31 км)	буд. № 13 Секція № 1	буд. № 13 Секція № 2	буд. № 13 Секція № 3
буд. № 20	буд. № 47	буд. № 49	буд. № 36-С Секція № 1-3	буд. № 36-С Секція № 4-6	буд. № 46
буд. № 50	буд. № 1	буд. № 6	буд. № 128	буд. № 3 Секція № 1	буд. № 3 Секція № 2
буд. № 1	буд. № 2 Секція № 1	буд. № 2 Секція № 2	буд. № 3		

Рівень гуч. викликів

Останні дії оператора

Черга викликів [0]

Пульт № 1 | Карта | Телефони | вкф Промінь | Службова | БД Викликів

Рисунок 1.2 - Інтерфейс програми ОДС «ПРОМІНЬ»

Дана програма побудована на базі даних MSSQL Compact. Система дозволяє контролювати роботу передбачених точок в режимі реального часу: наявність живлення, чи в нормі фаза постачання, чи відкриті/закриті двері, наявність людей в кабіні. Завдяки цьому фіксуються несправності та інші події, які можуть вплинути на безпечну експлуатацію ліфтів.

Через програму є можливість дистанційно вимкнути та увімкнути ліфт (рис. 1.3). Якщо люди застрягли в ліфті, то через цю програму є доступ до голосового зв'язку.

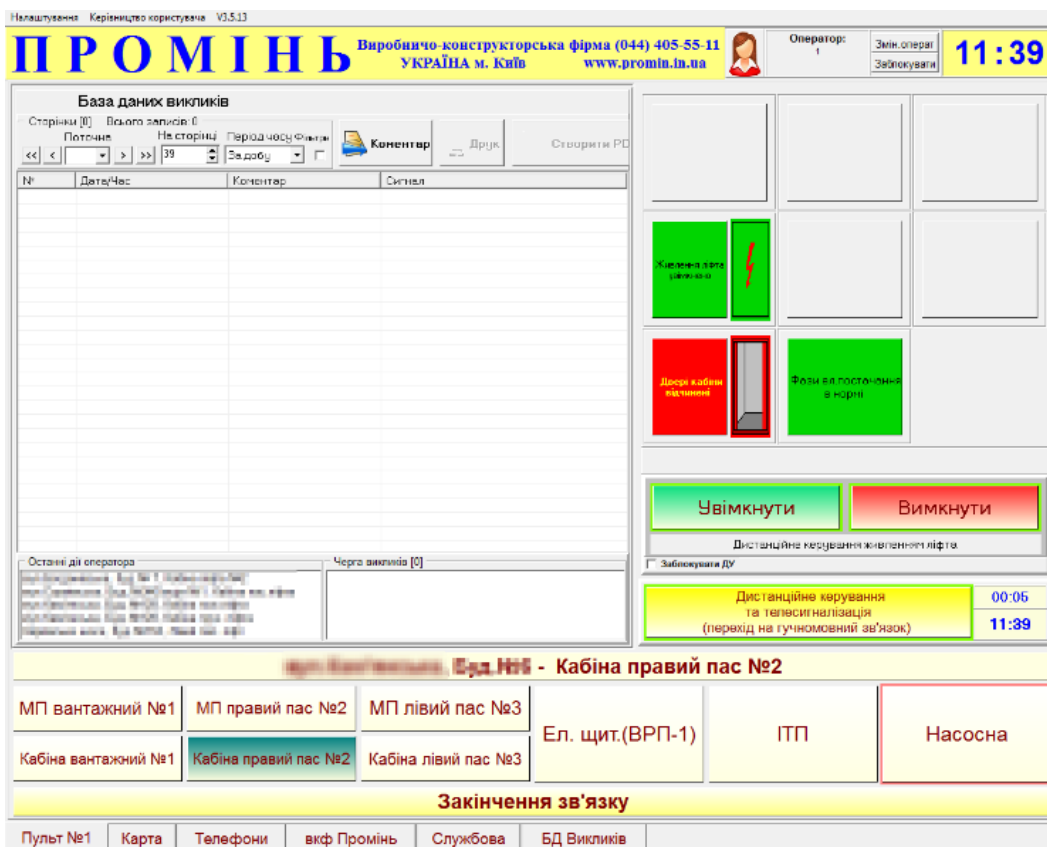


Рисунок 1.3 - Інтерфейс керування ліфтом

Система реєструє всі виклики та веде запис дзвінкі (рис. 1.4). Це дає сформувати звіти (рис. 1.5).

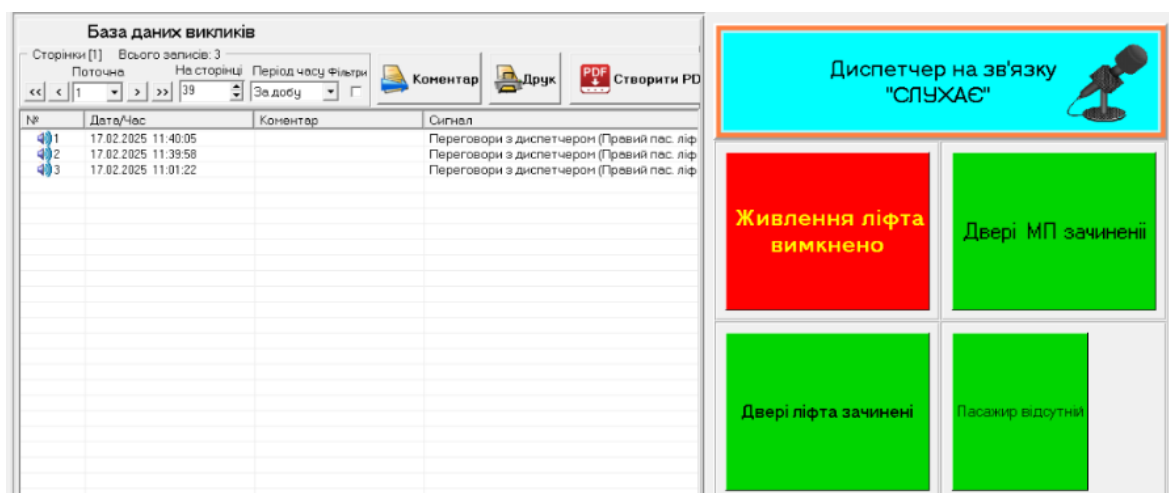


Рисунок 1.4 - Вигляд історії викликів та вид інтерфейсу під час зв'язку

Програма, використовуючи DTMF (Dual-Tone Multi-Frequency), може передавати команди на ліфтове обладнання через тональний набір по телефону.

Система контролює чи зачинені двері машинного приміщення, щоб уникнути несанкціонованого доступу.

База даних викликів					
Сторінки [1] Всього записів: 25		Поточна На сторінці		Період часу Фільтри	
<< < 1 > >>		39		За добу	
		Друк		Створити PDF	
Вулиця	№ будинку	Назва приміщення	Дата/Час	Коментар	Сигнал
	Буд. №1	МП пас.ліфта	17.02.2025 11:02:52		Переговори з диспетчером (Кабіна пас.ліфта)
	Буд. №1	МП груз.ліфта	17.02.2025 11:02:47		Переговори з диспетчером (Кабіна груз. ліфта)
	Буд. №1	МП пож.ліфта	17.02.2025 11:02:21		Переговори з диспетчером (Кабіна пож.ліфта)
	Буд. №1	МП груз.ліфта	17.02.2025 11:02:14		Переговори з диспетчером (Кабіна груз. ліфта)
	Буд. №1	МП пас.ліфта	17.02.2025 11:01:56		Переговори з диспетчером (Кабіна пас.ліфта)
	Буд. №1	Лівий пас. ліфт	17.02.2025 11:01:39		Переговори з диспетчером (Лівий пас. ліфт)
	Буд. №1	Вантажний ліфт	17.02.2025 11:01:31		Переговори з диспетчером (Вантажний ліфт)
	Буд. №1	Лівий пас. ліфт	17.02.2025 11:01:27		Переговори з диспетчером (Лівий пас. ліфт)
	Буд. №1	Правий пас. ліфт	17.02.2025 11:01:22		Переговори з диспетчером (Правий пас. ліфт)
	Буд. №1	Лівий пас. ліфт	17.02.2025 11:01:10		Переговори з диспетчером (Лівий пас. ліфт)
	Буд. №1	Насосна	17.02.2025 10:31:48		Переговори з диспетчером (Насосна)
	Буд. №1	ІТП	17.02.2025 10:31:46		Переговори з диспетчером (ІТП)
	Буд. №1	Ел.щит.№1	17.02.2025 10:31:40		Переговори з диспетчером (Ел.щит.№1)
	Буд. №1	Насосна	17.02.2025 10:31:38		Переговори з диспетчером (Насосна)
	Буд. №1	МП пож. ліфта	17.02.2025 10:15:26		Переговори з диспетчером (Кабіна пож.ліфта)
	Буд. №1	МП пож. ліфта	17.02.2025 10:15:19		Виклик на гучномовний зв'язок (Кабіна пож.ліфта)
	Буд. №1	МП лівий пас.№3	17.02.2025 08:32:58		Переговори з диспетчером (Кабіна лівий пас.№3)
	Буд. №1	МП лівий пас.№3	17.02.2025 08:32:49		Виклик на гучномовний зв'язок (Кабіна лівий пас.№3)
	Буд. №1	МП пож. ліфта	16.02.2025 19:23:13		Переговори з диспетчером (Кабіна пож.ліфта)
	Буд. №1	МП пож. ліфта	16.02.2025 19:22:59		Виклик на гучномовний зв'язок (Кабіна пож.ліфта)
	Буд. №1	Ел.щит.№2	16.02.2025 17:17:26		Освітлення увімкнено
	Буд. №1	МП вант. ліфта	16.02.2025 17:14:46		Переговори з диспетчером (Кабіна вант.ліфта)
	Буд. №1	МП вант. ліфта	16.02.2025 17:14:36		Виклик на гучномовний зв'язок (Кабіна вант.ліфта)
	Буд. №1	МП пож. ліфта	16.02.2025 14:53:03		Переговори з диспетчером (Кабіна пож.ліфта)
	Буд. №1	МП пож. ліфта	16.02.2025 14:52:55		Виклик на гучномовний зв'язок (Кабіна пож.ліфта)

Рисунок 1.5 - Звіти викликів

Підключення відбувається через 2 технології: Ethernet та/або GSM. Без GSM не може бути Ethernet, в той час як, без Ethernet може бути GSM, адже під час зникнення фази є можливість достукатись до диспетчера тільки через мобільний зв'язок. На рисунку 1.6 відображено зв'язок через GSM. В той час як на рисунку 1.7 – зв'язок через Ethernet.

Як і в будь-якій програмі, в ОДС «ПРОМІНЬ» є свої недоліки. Першою проблемою є те, що програма прив'язує диспетчера до диспетчерського відділу, через те, що зв'язок відбувається через спеціальний фізичний пристрій, який має бути підключений до комп'ютера, через що неможливо використовувати програму в хмарному середовищі.

Ще однією проблемою є те, що за розширенням та додаванням нових адрес треба звертатися напряму до виробника, за додаткову плату.

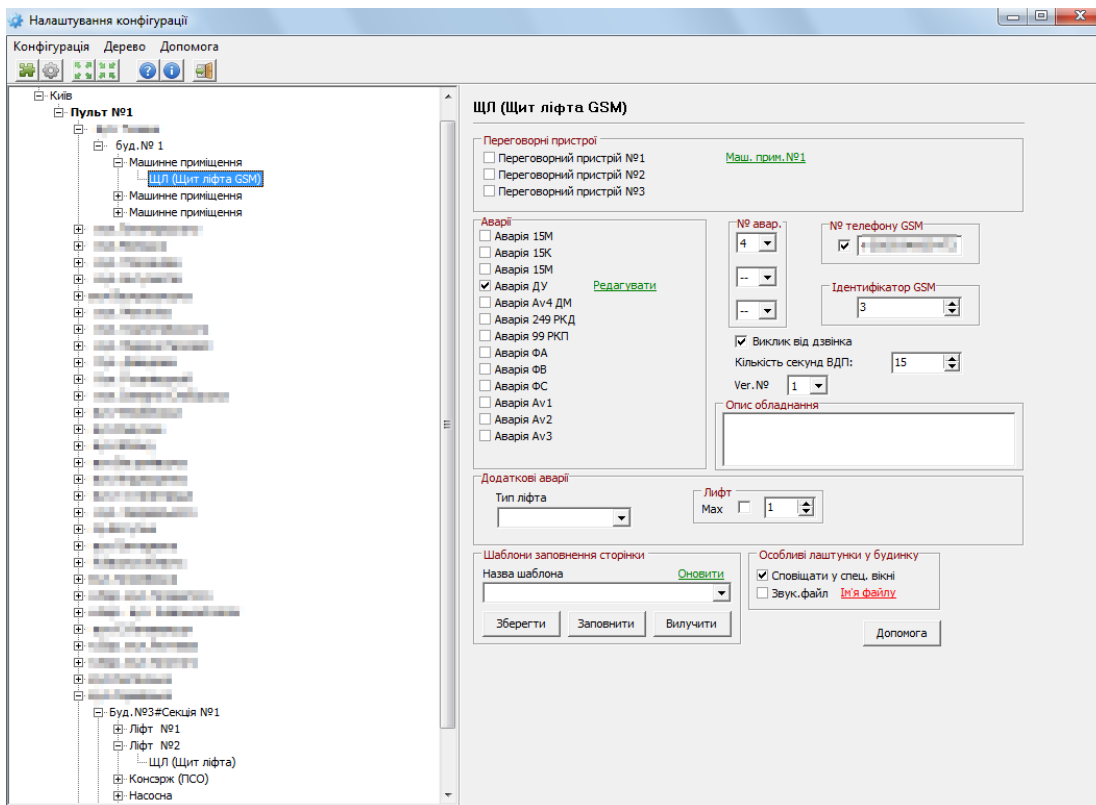


Рисунок 1.6 - Поле редагування існуючих ліфтів та адрес

Через використання GSM зв'язку має низьку якість звуку. Також дуже сильні спотворення спостерігаються під час використання модернізованого блоку "Еталон".

При роботі з додатком було знайдено помилку, яка полягала в тому, що під час знаходження на одній адресі, де багато ліфтів, програма не дає переключитись з одного на інший ліфт, поки оператор не закінчить зв'язок та не перепідключиться знову.

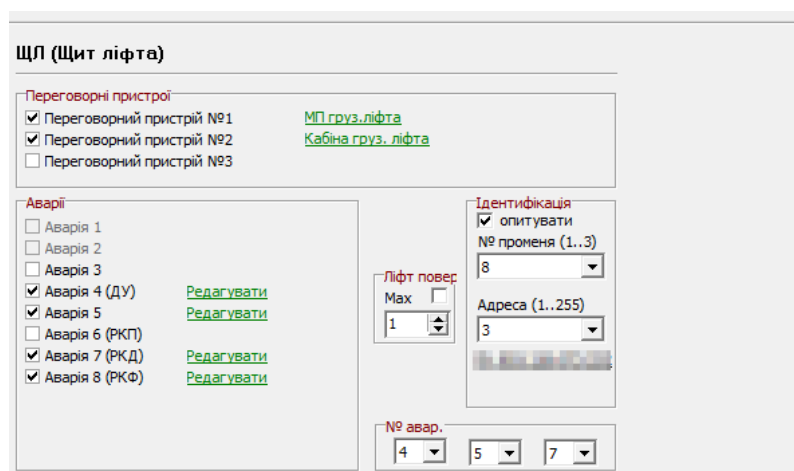


Рисунок 1.7 - Налаштування ліфта підключеного через Ethernet

Наступною програмою диспетчеризації є RMDedal – аналог «ПРОМІНЬ» (рис. 1.8). Вона забезпечує контроль стану ліфтового обладнання, дозволяє здійснювати дистанційне керування та забезпечує голосовий зв'язок між диспетчером і пасажирами. ОТІС використовує зазвичай модель DE-2060.

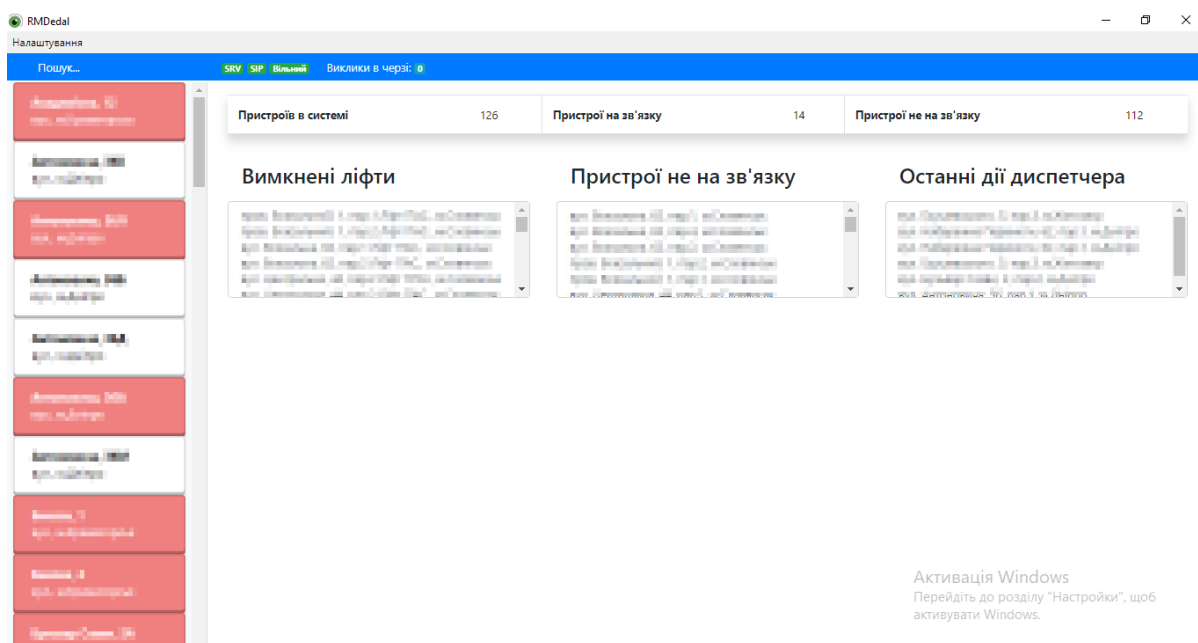


Рисунок 1.8 - Інтерфейс програми RMDedal

Як і «ПРОМІНЬ» вміє використовувати DTMF коди для перезапуску ліфту, що допомагає зменшити навантаження на механіків, адже в 90% випадків дистанційне перезавантаження допомагає у ситуаціях застрягань, але тільки при встановленому обладнанні.

Відстежує стан контрольних точок ліфтів (рис. 1.9). При виявленні неполадок надсилає СМС: немає фази; немає 220В; відкриті двері; розряджені акумулятори системи.

Система підтримує GSM та Ethernet. Існує контроль доступу в машинне приміщення.

З мінусів можна виділити те, що якщо ведеться активний виклик інші виклики в черзі і не висвічуються на екран, що може створити проблеми, якщо диспетчер забуде закінчити зв'язок.

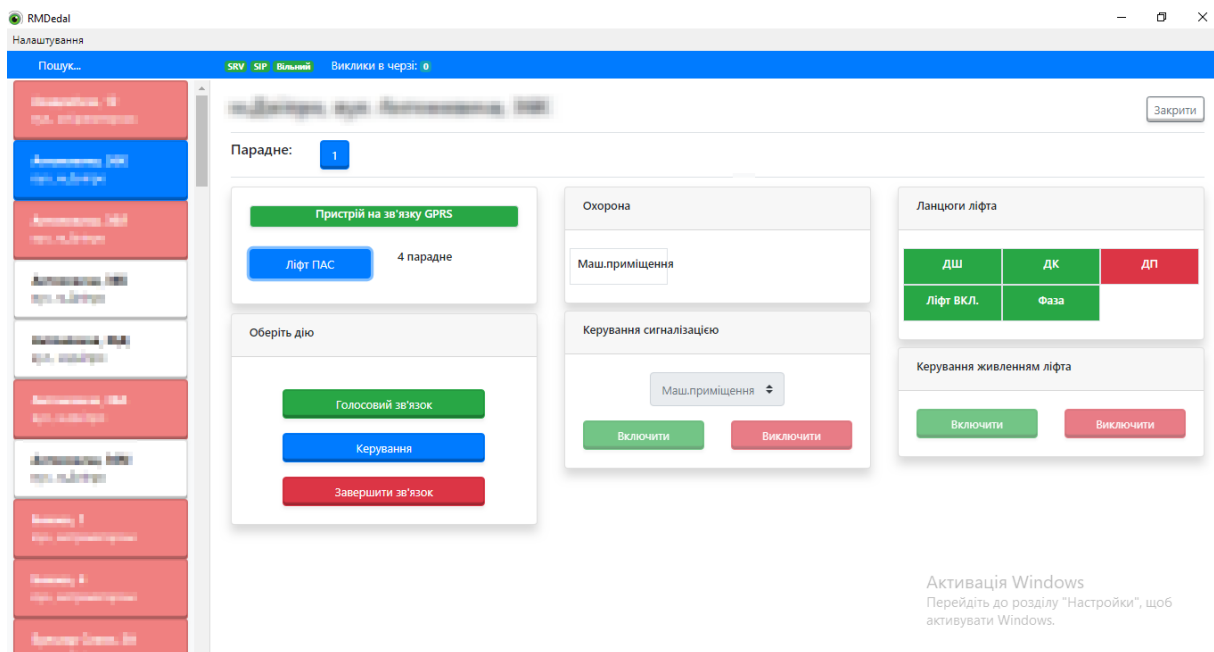


Рисунок 1.9 - Інтерфейс програми під час керування та зв'язку з ліфтом

### 1.3.2. Програма для обліку заявок ЛАС

Найважливішою програмою у відділі з обслуговування є ЛАС (Ліфтова аварійна система), завдяки якій ведеться облік заявок (рис. 1.10). Історія програми тягнеться ще з 90х років, коли ОТІС увійшла в Україну. Тоді ця програма була написана на DOS, а керування пунктами меню здійснювалось тільки за допомогою клавіатури. Пізніше в ОТІС з'явилися перші комп'ютери, на операційній системі Windows, і постало питання в перенесенні програми на графічний інтерфейс. За образом та подобою була написана програма на Delphi, яка використовується і досі.

Характерною особливістю цієї програми є те, що в керуванні присутній гібридний інтерфейс – функції продубльовані на графічний інтерфейс та на F-ки, що дає змогу швидше відкривати необхідні пункти меню, але також нестандартне та незвичне для нових робітників. Всі дані зберігаються в базі даних MSSQL.

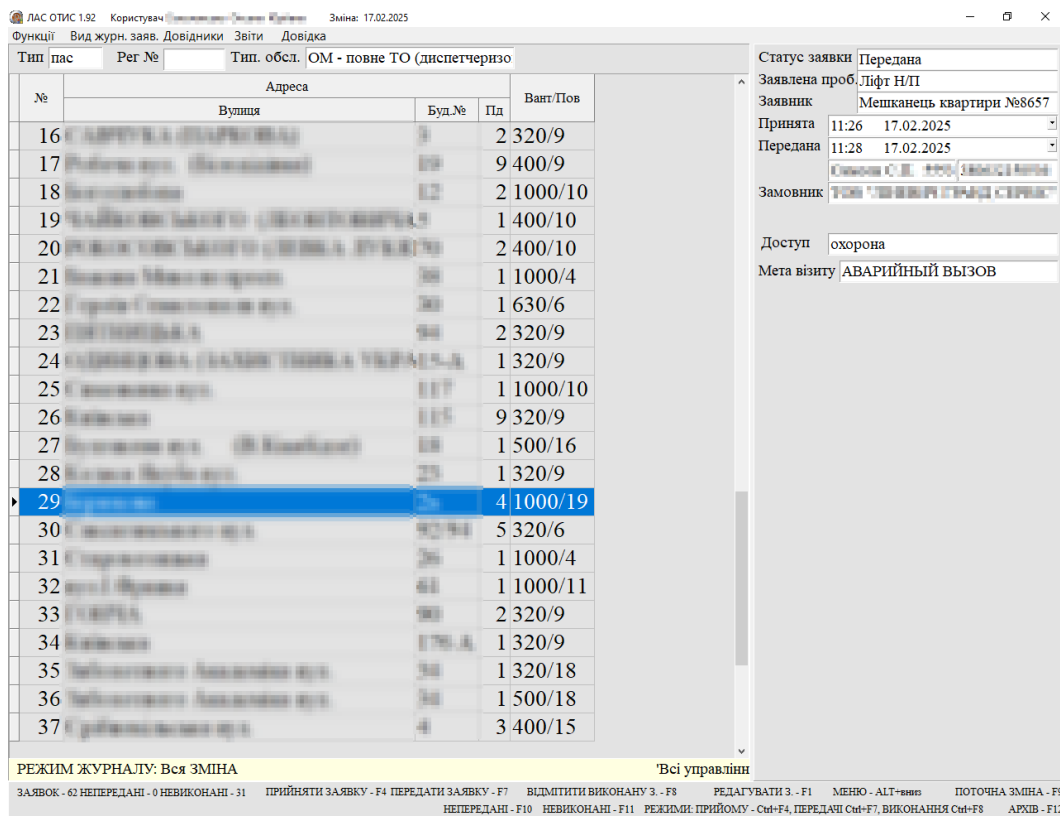


Рисунок 1.10 - Інтерфейс ЛАС

Головними функціями програми є взаємодія з заявками. Найбільш використовувані з них є:

- 1) Прийняття заявки (F4, Enter);
- 2) Передача заявки (F7);
- 3) Відмітка виконаної заявки (F8);
- 4) Редагування (F1);
- 5) Відміна заявки як помилка вводу;
- 6) Всі заявки, які поступили за поточну зміну, та не закриті за минулі (F9);
- 7) Заявки які не були передані (F10);
- 8) Всі невиконані заявки (F11).

Життєвий цикл заявки починається з її прийому. Клієнт може зателефонувати на гарячу лінію або через програми диспетчеризації зв'язатися з оператором. Вказується вулиця, номер будинку та під'їзд. Оператор дивиться чи є така заявка в роботі та повідомляє про її наявність/відсутність. При відсутності

починається прийом заявки (рис. 1.11). Вказується який тип проблеми змусив про неї повідомити.

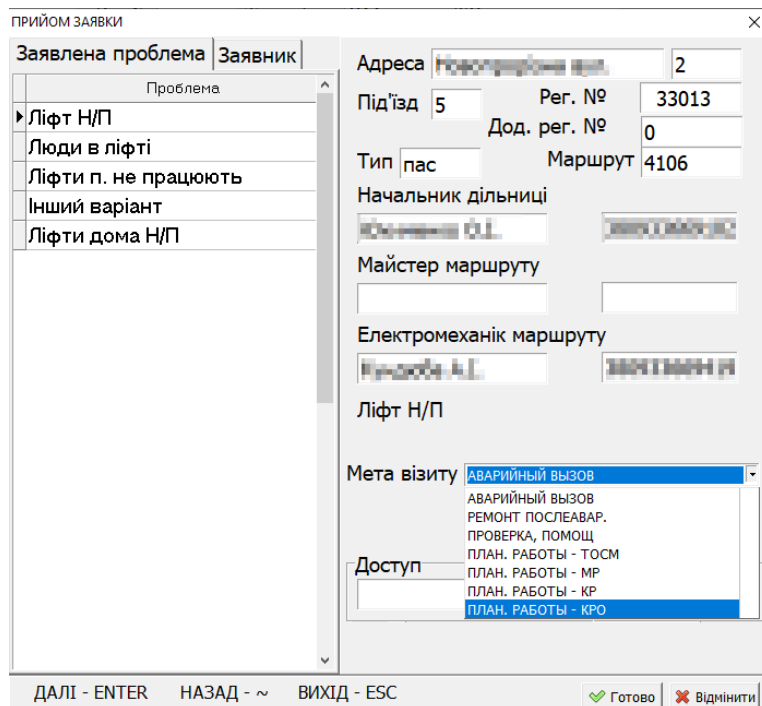


Рисунок 1.11 - Прийом заявки

Далі диспетчер реєструє особу, яка повідомила про заявку.

Якщо це був житель будинку, то вказується номер квартири (рис. 1.12).

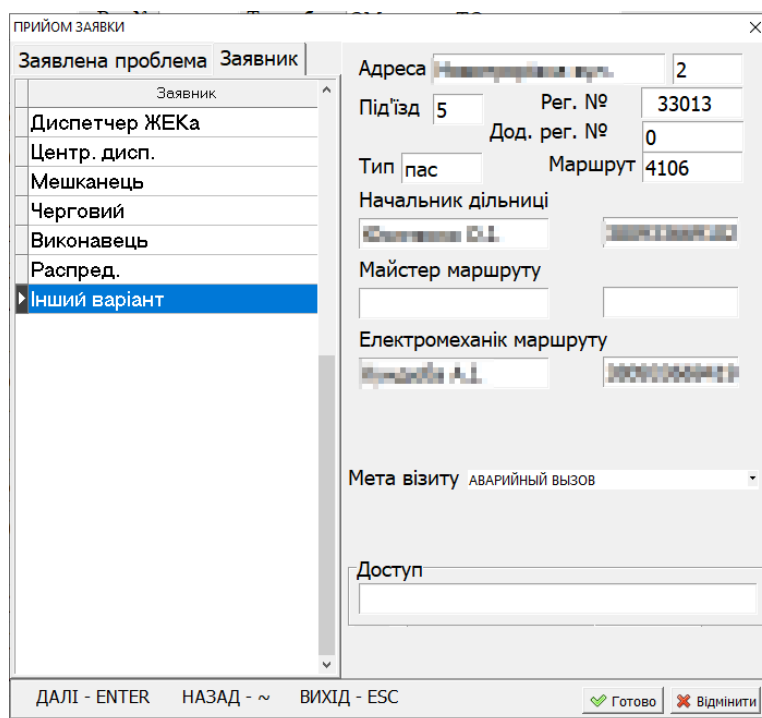


Рисунок 1.12 - Вибір заявника

Після цього заявка передається механіку. В програмі вказується кому саме вона була передана. Передача відбувається через телефон. Фіксується час передачі.

Коли заявка виконана механік має повідомити про це диспетчера надавши інформацію про причини зупинки, час виконання заявки та стан ліфта. Всю цю інформацію диспетчер заносить у вікні «Відмітити виконану заявку» (рис. 1.13).

Рисунок 1.13 - Відмітка виконаної заявки

Якщо диспетчер помилково ввів щось в заявку, то він має можливість редагувати її пункти, які можна побачити на рисунку 1.14.

Рисунок 1.14 - Меню вибору редагування

Якщо ж диспетчер помилково вже створив заявку, то в програмі немає пункту видалення. Щоб вирішити це питання оператор має можливість закрити заявку як помилку вводу (рис. 1.15.)

Статус заявки	Помилка вводу
Заявлена проб.	Ліфт Н/П
Заявник	Мешканець квартири №3588
Принята	12:23 17.02.2025

Рисунок 1.15 - Вигляд помилки вводу в закритій заявці

Існують ситуації, коли ліфт запускається сам і люди повідомляють про це. Тоді диспетчер закриває заявку статусом: Знята заявником (рис. 1.16).

Статус заявки	Знята заявником
Заявлена проб.	Люди в ліфті
Заявник	Мешканець квартири №6282
Принята	08:35 17.02.2025
Передана	08:36 17.02.2025

Рисунок 1.16 - Вигляд заявки, яка знята заявником

Важливим моментом є реєстрація ліфтів, які не вийшло полагодити – непрацюючі ліфти або ліфти в зупинці. Під час закриття заявки існує поле Стан ліфту. Якщо обрано, що ліфт не запущено, то він автоматично відправляється в список зупинених ліфтів але необхідно ввести причину зупинки (рис. 1.17).

Непрацюючі ліфти									
Додати Дата пуску ліфта Причина простою Історія ліфту Запланована дата пуску Компетенція замовника Дата зупинки ліфта									
Дільниця <b>Дальницька дільниця</b>									
Рег №	53574	Тип обсл.	ОМ - повне ТО	Хто дав інф.		Виконано А.І.			
Адреса		ПД	Тип	Вант/Пов	Причина Н/З	Дата зупинки	План. дата пуску	Комп. замов	
Вулиця	Буд№								
Вулиця Київська	2/10/1	1	пас	320/8	Треб зам канат.	29.06.2022 13:51:00		Ні	
Вулиця Миколаївська	5	2	вант	500/16	зломане водило	29.01.2025 19:10:00		Ні	
Вулиця Миколаївська	19-Д	1	пас	400/16	Потрібна заміна трансформатора	09.02.2025 22:05:00		Ні	
Вулиця Миколаївська	19-Д	5	вант	1000/13	Потрібн.перечалка троса ОС	12.02.2025 17:45:00		Ні	
Суданська вулиця	1	3	вант	630/15	Потр.заміна ВА-1	13.02.2025 14:30:00		Ні	
Вулиця Зіньківська	10-10	1	пас	400/16	Потрібна заміна троса	13.02.2025 15:04:00		Ні	
Миколаївська вулиця (Роздільний І-А)		2	пас	320/10	згорів дверний привод	14.02.2025 18:31:00		Ні	
Дальницька вулиця (Дальницька вулиця 10)		2	вант	320/12	Вибиває ВА-1	14.02.2025 19:45:00		Ні	
Вулиця Київська	10-10	1	пас	400/10	Потр.заміна датчика точної зупинки	15.02.2025 14:20:00		Ні	
Вулиця Миколаївська	19-Д	1	вант	500/16	Зламаний глебан ДК	15.02.2025 20:45:00		Ні	
Удільницька вулиця	10	9	пас	400/17	Потр.наладчик	16.02.2025 14:45:00		Ні	

Рисунок 1.17 - Список непрацюючих ліфтів

В самій програмі ведеться відслідковування активності сеансів користувачів, тобто коли він зайшов та вийшов. Є 3 рівні доступу: диспетчер, керівник та адміністратор.

Цікаво те, що в програмі є вбудований редактор звітів, який реалізований через Fast Report (рис. 1.18).

Історія Ліфту за адресою: [адреса]											
за період 17.02.2024 8:30:00 - 17.02.2025 12:47:48											
Мета візиту	Дата час ЗУПИНКИ	ЗАЯВЛЕНА ПРОБЛЕМА	ЗАЯВНИК	ВИКОНАВ Ел. МЕХАНІК	ЧАС ПРИБ. ЗАВ.	ПОВ ПАС	КОД/ШИФР НЕСПР.	КОМЕНТАР	ЛІФТ ПУЩЕНО ?	ПРОСТ. днів: год:хв	ДОДАТКОВА ІНФ. ДАТА ПУСКУ
Під.№4" Рег.№ пас 1000/19 Дата встановлення 01.01.2019 Замовник: [адреса]											
Завод виробник ПрАТ "ОПС" Майстер: Електромеханік маршруту: [адреса]											
A	12.00 16.05.2024	Ліфт Н/П	Мешканець квартири №9040	[адреса]	12.03 12.25	1	900/6	збій електрон.	Пущено	0:00:25	
A	17.14 17.05.2024	Ліфт Н/П	Черговий	[адреса]	17.30 18.00	1	900/6	збій частотника	Пущено	0:00:46	Кандидат
A	10.11 21.05.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	12.00 12.40	1	900/6	збій частотника	Пущено	0:02:29	Критичний
A	11:52 22.05.2024	Ліфт Н/П	Мешканець квартири №9040	[адреса]	12.10 12.40	1	900/6	збій частотного перетворювача	Пущено	0:00:48	Критичний
A	12.04 25.05.2024	Ліфт Н/П	Черговий	[адреса]	13.00 13.30	1	900/6	перезапуск	Пущено	0:01:26	Критичний
A	10.23 05.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	11.00 11.15	1	1703/3	Потр. заміна контактів ДК	Ліфт Н/П	1:06:22	16.45 06.06.2024
A	14.29 08.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	14.40 15.05	1	900/6	перезапуск	Пущено	0:00:36	Критичний
A	09:35 09.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	09.38 10.00	1	909/6	збій електрон.	Пущено	0:00:25	Критичний
A	11:26 11.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	11.35 12.05	1	900/6	перезапуск	Пущено	0:00:39	Критичний
A	08:38 14.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	08.45 09.30	1	900/6	РНР	Пущено	0:00:52	Критичний
A	08:35 17.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	08.55 09.40	1	900/6	перезапуск	Пущено	0:01:05	Критичний
A	10:58 19.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	12.00 12.40	1	900/6	перезапуск	Пущено	0:01:42	Критичний
A	12:55 20.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	13.10 13.30	1	900/6	відсутнє світло	Пущено	0:00:35	Критичний
A	12:55 22.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	14.00 14.20	1	900/6	перезапуск	Пущено	0:01:25	Критичний
A	11:47 23.06.2024	Ліфт Н/П	Мешканець квартири №0474	[адреса]	12.00 12.25	1	900/6	перезапуск	Пущено	0:00:38	Критичний
A	09:28 25.06.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	10.45 11.00	1	908/3	Потр. заміна плати керування	Ліфт Н/П	2:01:05	10:33 27.06.2024
A	12:29 08.07.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	12.45 13.30	1	900/6	РНР	Пущено	0:01:01	Критичний
A	10:04 09.07.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	08.30 08.45	1	900/6	перезапуск	Пущено	0:22:41	Критичний
A	07:42 12.07.2024	Ліфт Н/П	Мешканець квартири № [адреса]	[адреса]	08.01 08.25	1	900/6	перезапуск	Пущено	0:00:43	Критичний
A	13:26 14.07.2024	Ліфт Н/П	Мешканець квартири №0474	[адреса]	14.00 14.25	1	900/6	перезапуск	Пущено	0:00:59	Критичний
A	12:14 19.07.2024	Ліфт Н/П	Диспетчер ЖЕКa	[адреса]	12.17 12.35	1	1402/6	Отс. напр. (фази на ВУ)	Ліфт Н/П	0:01:50	14:04 19.07.2024
---	10:00 21.07.2024	ЗУПИНКА	Самсон	[адреса]	00.00 00.00	0	0/0	Причину не виявлено	Пущено	1:01:00	11:00 22.07.2024
A	20:06 06.10.2024	Люди в ліфті	Черговий	[адреса]	20.08 20.25	1	4444/6	ЛОЖНЫЙ ВЫЗОВ	Пущено	0:00:19	
A	10:44 29.10.2024	Ліфт Н/П	Мешканець квартири №5645	[адреса]	14.35 15.00	1	900/6	перезапуск	Пущено	0:04:16	Кандидат

Рисунок 1.18 - Звіт по всім проблемам за рік для певного ліфту

На початку кожної зміни диспетчер вносить чергових аварійних механіків, які працюють в цю зміну. Під час оформлення заявки у кожного ліфта має бути внесено як потрапити в машинне приміщення.

Які є види мети візиту наведено в рисунку 1.19.

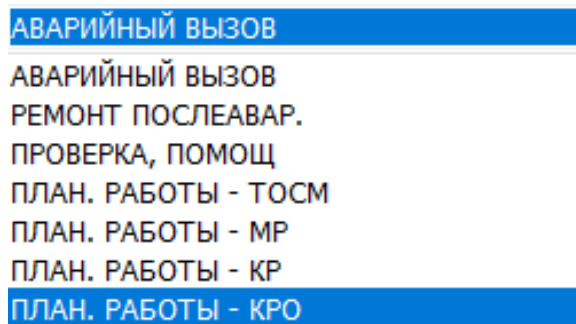


Рисунок 1.19 - Види причин візиту

## 1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів

### 1.4.1. Функціональна модель в компанії "OTIS"

Під час проходження практики були зібрані дані, на основі яких можна побудувати контекстні діаграми. Ці діаграми відображають взаємозв'язок між суб'єктами та інформаційною системою. Завдяки ним, може здійснюватися декомпозиція системи до двох рівнів.

Кожна діяльність, яка відображена на основі батьківської діаграми, може бути деталізована за допомогою дочірньої діаграми, що виконує роль її деталізації. На діаграмі декомпозиції відображають декілька блоків діяльності, що являє собою функції, які є складовими діяльності головної діаграми під час процесу аналізу, де шукають детальні результати. Під час створення дочірньої моделі всі стрілки та зв'язки автоматично дублюються з батьківської схеми. Для встановлення з'єднання батьківської та дочірніх діаграм використовують С-номери, що дають можливість виключити чіткі зв'язки між діаграмами [6, с. 14]. Декомпозиція дозволяє аналізувати внутрішні процеси та взаємозв'язок між контекстними діаграмами в структурі.

Декомпозиція була проведена в програмному засобі ALLFUSION PROCESS MODELER.

Дана програма підтримує три методології побудов діаграм - IDEF0, IDEF3 і DFD, а також змішаних моделей [15]. Були використані діаграми IDF0 та DFD.

DFD (Data Flow Diagram) або діаграма потоків даних – це засіб графічного структурного аналізу, який дозволяє описати зовнішні стосовно системи джерела та споживачів інформації, логічні процеси обробки, напрями руху потоків даних та сховища, з якими здійснюється взаємодія [12].

IDEF0 – це підхід до функціонального моделювання, який використовується для побудови функціональної моделі, що демонструє взаємозв'язок між функціями системи, її структурою, а також потоками інформації і матеріальними ресурсами, які зв'язують ці функції [13].

В Erwin був проаналізований та змодельований бізнес-процес діяльності аварійної служби, з проведеною декомпозицією першого рівня (рис. 1.20).

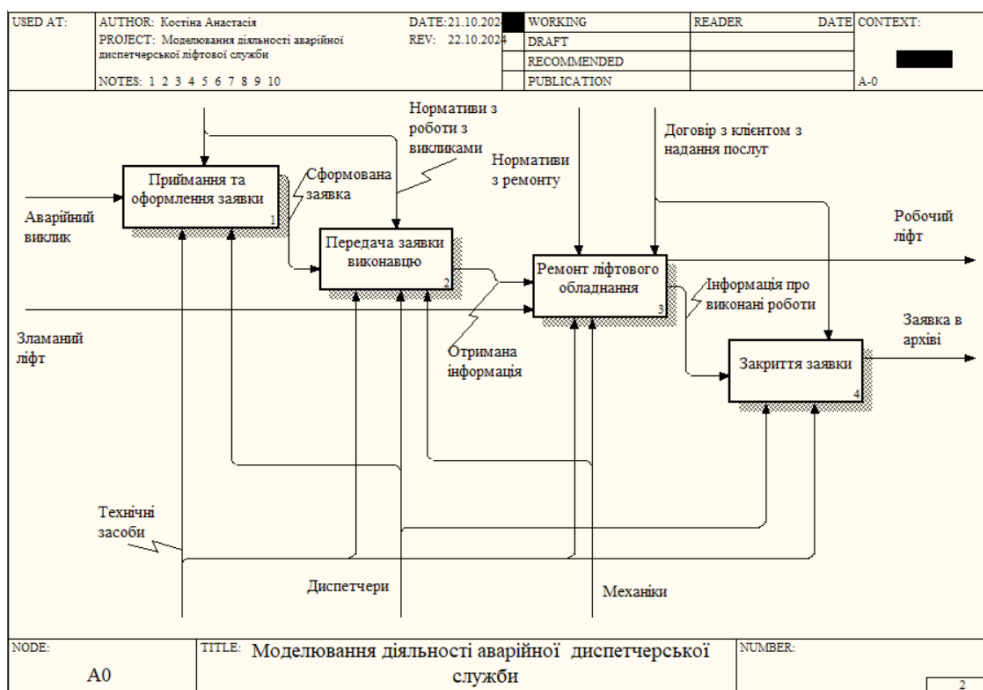


Рисунок 1.20 – Моделювання діяльності аварійної диспетчерської служби в нотації IDF0

Проведено декомпозицію другого рівня в нотації DFD Приймання та оформлення заявки (рис. 1.21).

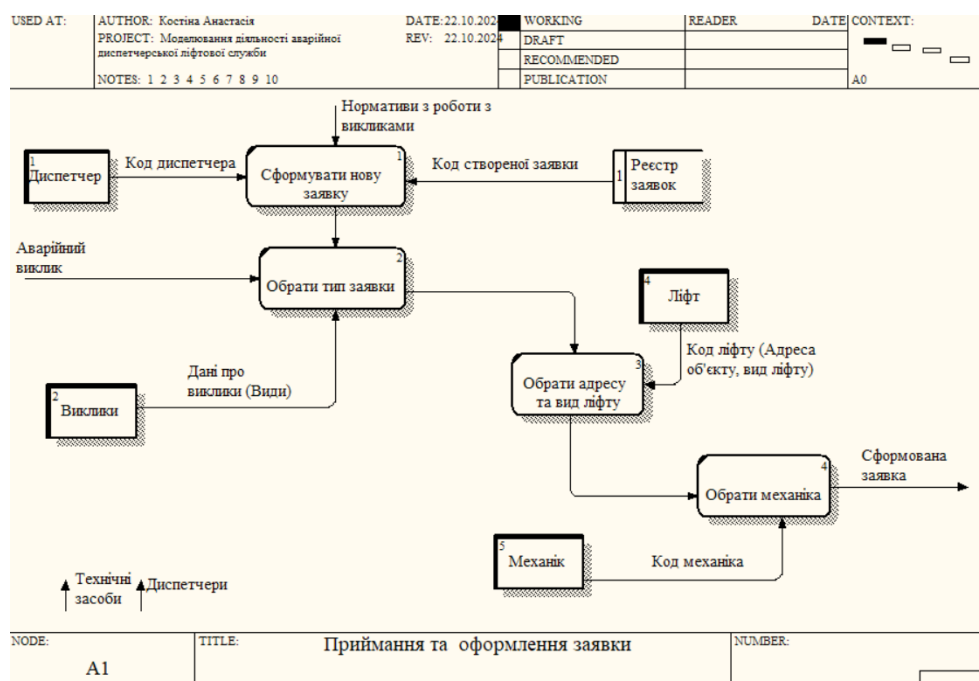


Рисунок 1.21 – Приймання та оформлення заявки в нотації DFD

Проведено декомпозицію другого рівня в нотації IDFO Передача заявки виконавцю (рис. 1.22).

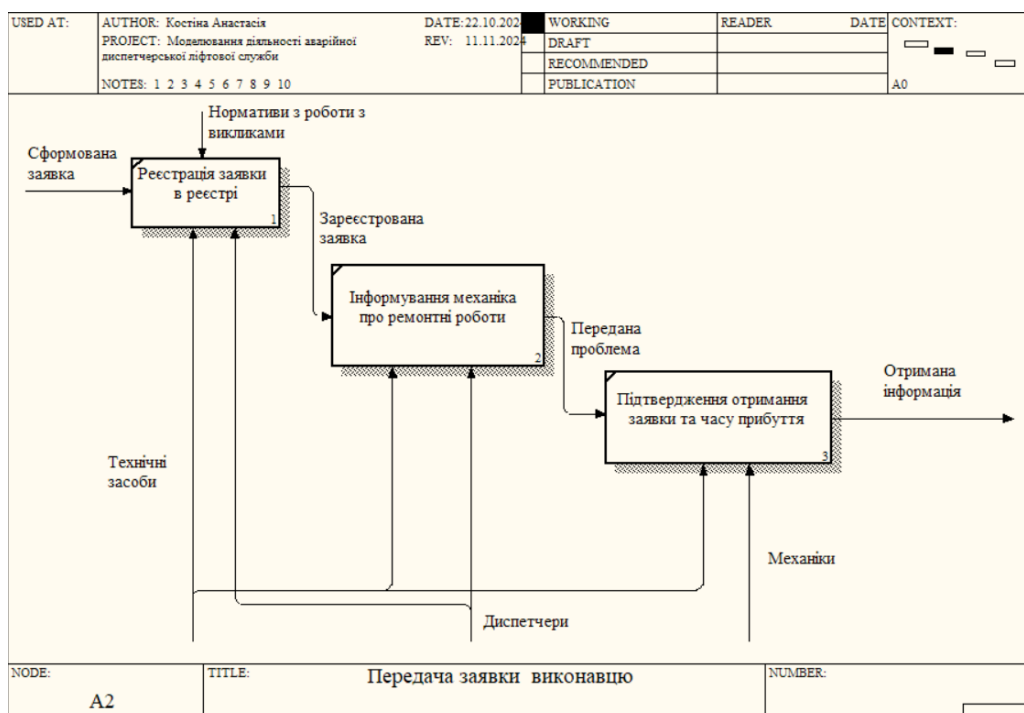


Рисунок 1.22 – Передача заявки виконавцю в нотації IDFO

Проведено декомпозицію другого рівня в нотації IDFO Ремонт ліфтового обладнання (рис. 1.23).

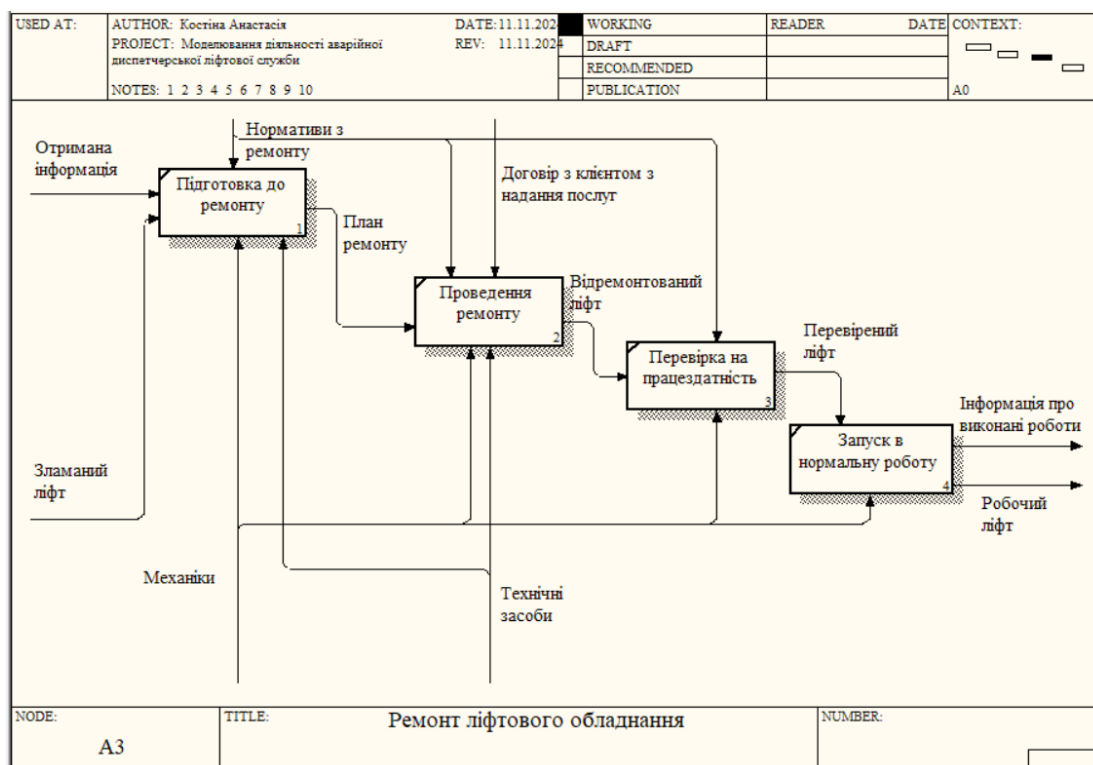


Рисунок 1.23 - Ремонт ліфтового обладнання в нотації IDFO

Проведено декомпозицію другого рівня в нотації IDFO Закриття заявки (рис. 1.24).

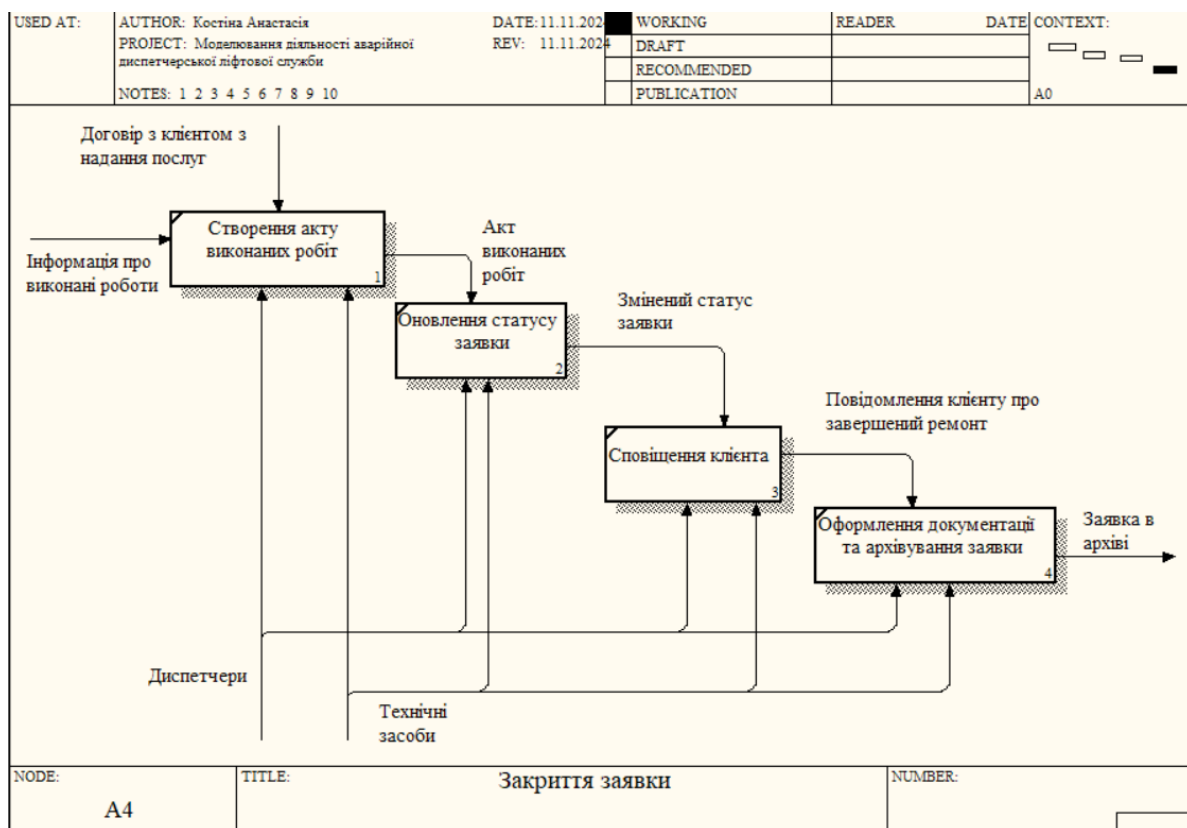


Рисунок 1.24 – Закриття заявки в нотації IDFO

Термін «бізнес-процес» у формалізованому вигляді можна вважати за сукупність взаємопов'язаних дій, що мають чітко визначені складові: вхідні дані, механізм обробки, результат, засоби управління, а також суб'єкти, які ініціюють процес (постачальники) та отримують його результати (споживачі).

В широкому розумінні бізнес-процес – це структурована послідовність дій, що охоплюють всі стадії життєвого циклу предмета діяльності: від створення початкової ідеї, проектування та реалізації до кінцевого результату, наприклад введення об'єкта в експлуатацію, постачання товарів, надання послуг, тощо. Це певний системно-замкнений процес [25].

Також бізнес процес прийому заявок був досліджений в Bizagi Process Modeler – безкоштовному програмному забезпеченні для створення діаграм процесів і документації в нотації стандарту BPMN; завдяки даному програмному

забезпеченню можна не тільки створити, а й опублікувати результати роботи в форматах від MS Word до інтерактивного HTML [37].

Нотація BPMN (Business Process Model and Notation) – це стандартизована графічна система, що призначена для моделювання бізнес-процесів і відображення взаємодії між учасниками в них. Діаграма дозволяє створювати зрозумілі та стандартизовані блок-схеми, що полегшує їх аналіз, оптимізацію та автоматизацію [20].

На рисунку 1.25 показаний бізнес процес в нотації BPMN по прийняттю та обробці заявки.

В даному бізнес процесі містяться точки входу, а саме дзвінок клієнта до диспетчера. Диспетчером створюється заявка, яка заноситься в ЛАС (Ліфтова аварійна система). Заявка передається механіку, який уточнює інформацію та, підготовлюючись до ремонту, виїжджає на об'єкт, де проводить ремонт ліфтового обладнання.

Якщо ремонт був вдалий, то механік повідомляє про це диспетчера, після чого, вносяться причини поломки, час передачі, час прибуття та час завершення ремонту. Заявка закривається.

Якщо ремонт не вдалось провести механіком, причини можуть бути різні, наприклад: через його складність чи необхідність заміни деталей, механік передає цю інформацію диспетчеру та просить занести ліфт у зупинку. Диспетчер вказує причину та закриває заявку як непрацюючий ліфт і програма заносить його до списку зупинених ліфтів.

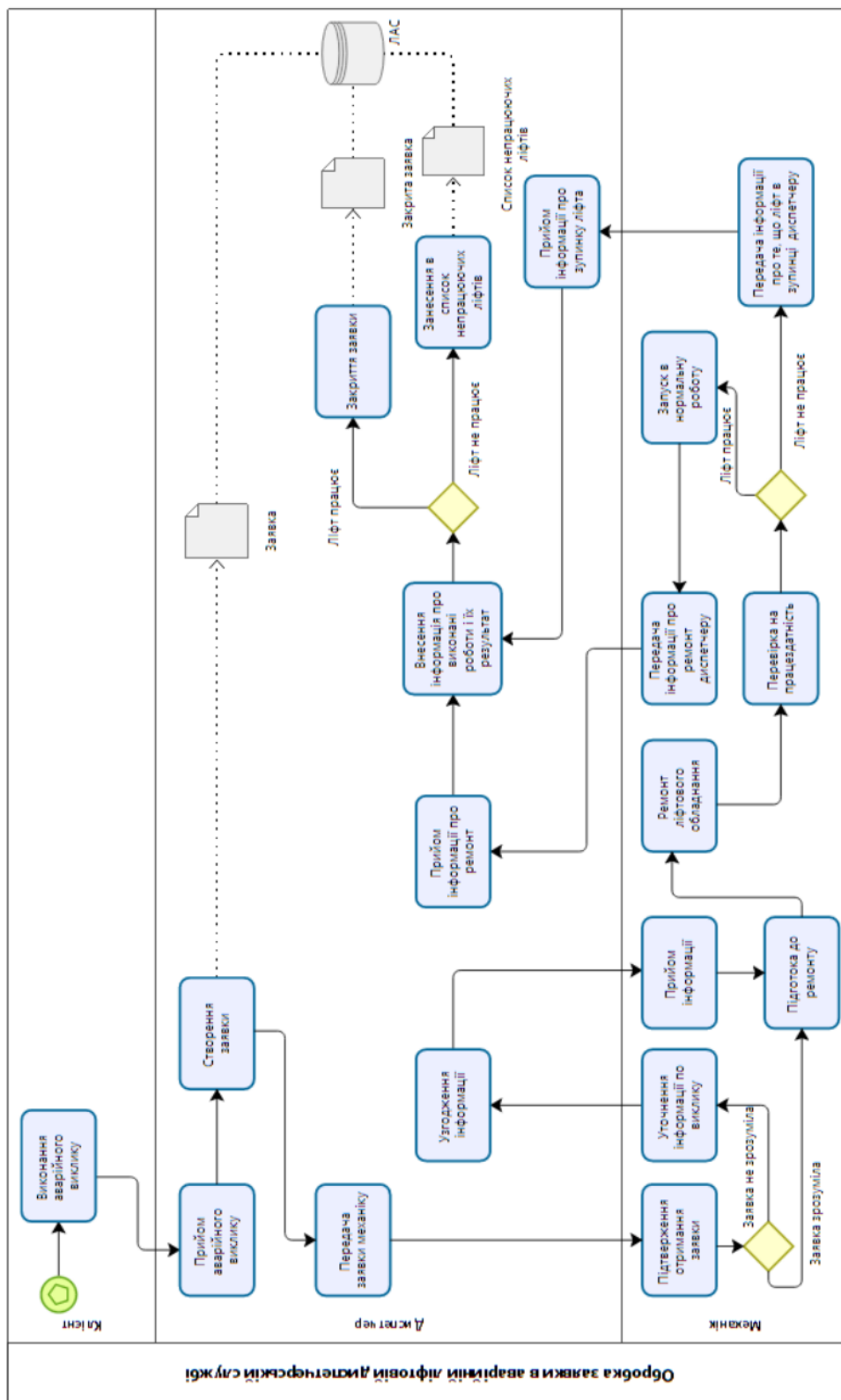


Рисунок 1.25 - Бізнес процес прийняття та обробки заявки в нотатії BPMN

#### 1.4.2. Виявлені проблеми в програмному забезпеченні в компанії "OTIS"

Ліфти відіграють важливу роль в нашому суспільстві, забезпечуючи комфортні умови для житла. Їхня несправність спричиняє незручності, особливо

для людей з інвалідністю, літніх осіб, а також для мешканців, яким необхідно перевозити громіздкі або важкі речі, такі як дитячі коляски чи великі покупки.

Оперативне реагування на несправності ліфтів є важливим завданням, проте існуючі системи диспетчеризації мають низку суттєвих недоліків. Зокрема, передача заявки про несправність часто здійснюється лише телефоном, що може бути незручним або навіть неможливим у певних ситуаціях. Наприклад, під час відключень електроенергії, спричинених російською агресією, мешканці можуть залишитися без зв'язку через відсутність мобільного покриття. Водночас дедалі більше людей використовують безперебійні Wi-Fi-мережі, що створює передумови для переходу на веб-застосунки як альтернативний спосіб комунікації з аварійною службою.

Окрім технічних обмежень, існує ще й людський фактор. Диспетчери щоденно опрацьовують сотні дзвінків, що значно навантажує їх і впливає на ефективність роботи. Деякі пасажери можуть поводитися агресивно, що призводить до стресових ситуацій для операторів. Також часто трапляються випадки, коли люди неправильно або нечітко називають адресу, що ускладнює швидке реагування на виклик.

З недоліків ЛАС можна відмітити те, що програма має застарілий та незручний, неінтуїтивний інтерфейс в якому можна загубитись. Причиною є вихідний код, який вже не оновлюється. Також недоліком є те, що якщо на ліфт вже створена заявка, до прикладу – скрипить, то при ситуації, що в цьому ліфті застрягли, диспетчер не може додати нову заявку на цей ліфт, адже ще є попередня незакрита. Дописати теж не варіант, адже всі заявки реєструються за фактичним часом отримання. Також незручним є те, що в списку заявок не показується поле «Механік», що робить неможливим відсортувати по механікам, адже, якщо майстер не надав сам інформацію, то диспетчер має зателефонувати механіку та перевірити статус виконання заявки, а їх, в одній особі, може бути купа.

## 1.5. Огляд існуючих рішень для розв'язання виявлених проблем в компанії "OTIS"

OTRS (Open Ticket Request System) – це сучасна програма для ведення заявок, яка переважно використовується в сфері обслуговування ІТ клієнтів, технічної підтримки та управління внутрішніми процесами [34] (рис. 1.26).

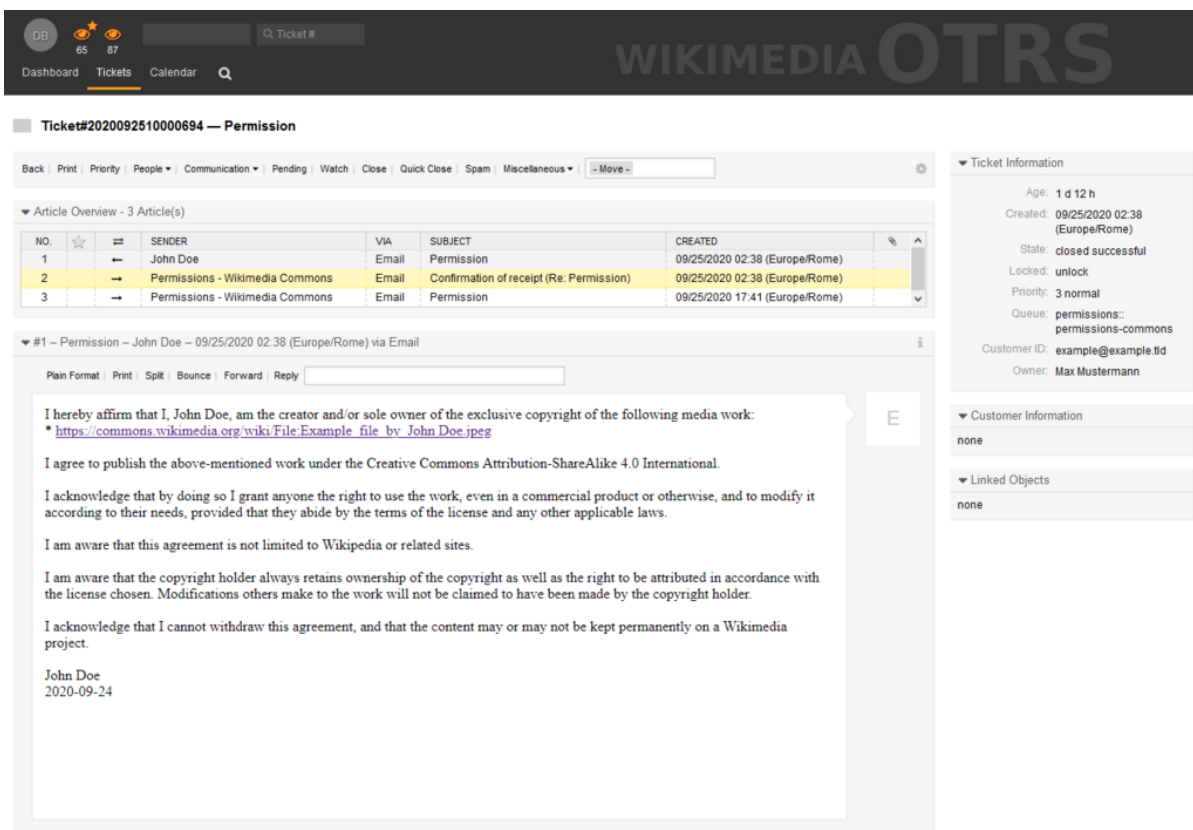


Рисунок 1.26 - Інтерфейс OTRS

Основними можливостями OTRS є управління заявками та оптимізація процесів [34]. Є можливість автоматично створювати, розподіляти та відстежувати заявки в реальному часі. Доступна інтеграція з електронною поштою, що дозволяє вести обмін інформацією без додаткових дій. Автоматичні сповіщення для механіків. Можливість перегляду детального журналу історії змін по кожній заявці.

В програмі передбачені ролі та права доступу, які можна задати програмно [34]. Наприклад диспетчери можуть створювати, редагувати та передавати заявки, механіки отримують доступ лише до тих заявок, які їм призначені, а керівники мають доступ до звітності, аналітики та управління ресурсами, завдяки вбудованим

інструментам для аналізу продуктивності співробітника, статистики частоти поломок, тощо.

OTRS доступний у веб інтерфейсі, що дозволяє користувачу заходити через мобільні пристрої. Є підтримка API для інтеграції інших систем [34].

В програми є дві версії: безкоштовна та платна. Безкоштовна версія (Community edition) доступна для самостійного розгортання та вимагає власного адміністрування. Платні версії (OTRS Group Managed) мають постійну підтримку від розробника, що включає оновлення та безпеку, але ціни можуть варіюватись від 5000 до 10000 в рік [34]. Проведемо порівняння між ЛАС та OTRS в таблиці 1.2.

*Таблиця 1.2. Порівняння OTRS та ЛАС*

<b>Характеристика</b>	<b>OTRS</b>	<b>ЛАС</b>
Інтерфейс	Сучасний, веб-орієнтований, адаптивний	Застарілий, DOS-подібний
Передача заявок	Автоматична, через систему	Вручну через телефон
Інформація про механіків	Диспетчер бачить зайнятість механіків	Потрібно телефонувати
Автоматизація	Є бізнес-процеси, сценарії та правила	Відсутня
Аналітика	Детальна статистика та звітність	Мінімальні можливості
Мобільна версія	Є	Немає
Інтеграція	API для підключення зовнішніх сервісів	Відсутня

У порівнянні з ЛАС OTRS має також і недоліки, які перешкоджають її впровадженню. Складність впровадження пов'язана з тим, що необхідні фахівці для налаштування та інтеграції, вартість стандартних та розширених версій є

великою, деякі функції програми можуть бути надлишковими для ліфтової служби, що може ускладнити користування. Система є адаптованою більше для ІТ-підтримки, що робить складності у налаштуванні, оптимізації та додаткової конфігурації. Також іноземний інтерфейс може сповільнити процес навчання.

GLPI (Gestionnaire Libre de Parc Informatique) – це система управління ІТ-заявками, яка широко використовується для контролю різних процесів, включаючи технічне обслуговування [30] (рис. 1.27). Ця система може бути адаптована під потреби ліфтової служби, дозволяючи вести облік заявок та стежити за історією обслуговування ліфту.

Основними можливостями є прийом, обробка та закриття заявок з можливістю автоматичного призначення виконавців, створення їх за кваліфікаціями (аварійні, технічні, профілактичні тощо), наявність автоматичних сповіщень [30].

Можливість введення бази даних усіх ліфтів з їх технічними характеристиками та історією обслуговування.

Програма надає доступ до формування детальних звітів, відстеження продуктивності механіків та до аналітики найчастіших поломок, що може допомогти в покращенні технічного обслуговування [30].

Можливість підключення інтеграцій для слідкування за станом ліфтів та API до існуючих баз даних компанії.

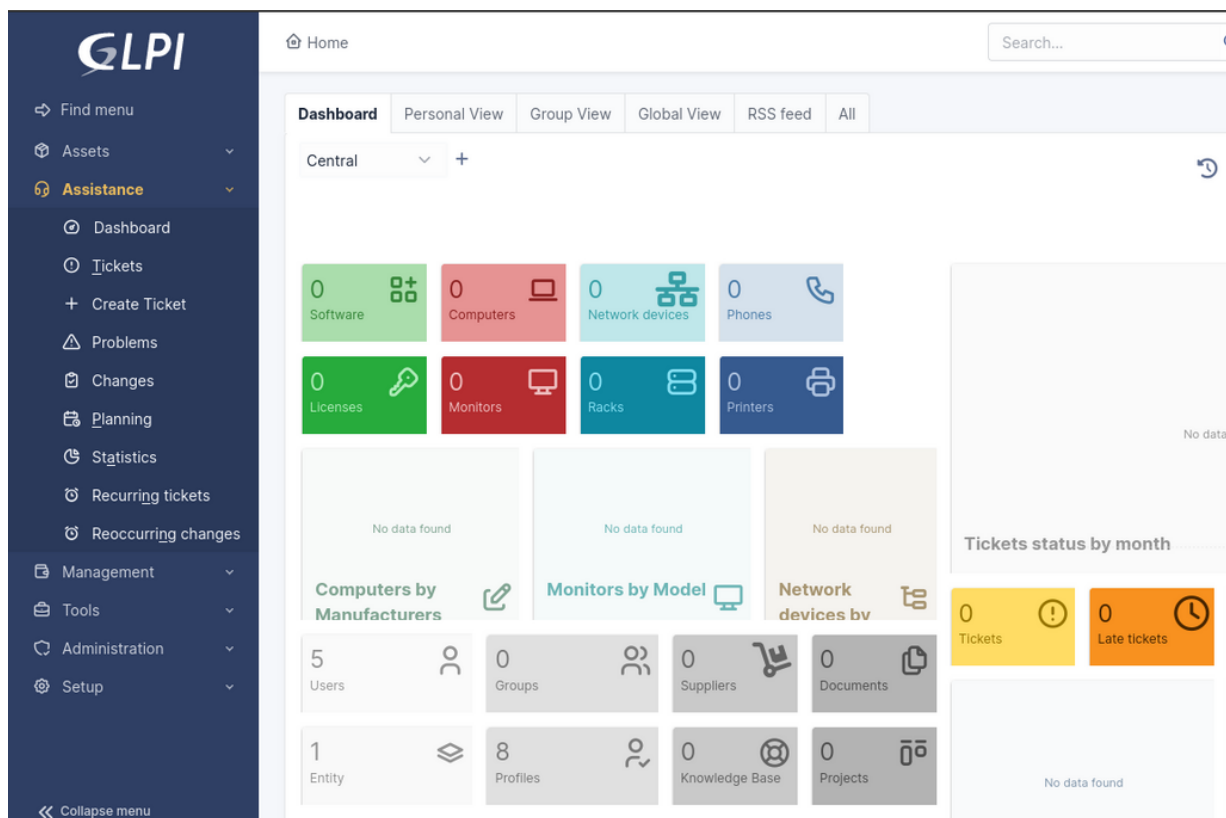


Рисунок 1.27 - Інтерфейс GLPI

Проведемо порівняння між ЛАС та GLPI в таблиці 1.3.

Таблиця 1.3. Порівняння GLPI та ЛАС

Характеристика	GLPI	ЛАС
Інтерфейс	Сучасний, веб-орієнтований, адаптивний	Застарілий, DOS-подібний
Автоматизація	Є автоматичний розподіл заявок	Відсутня
Аналітика	Детальна статистика та звітність	Мінімальні можливості
Інтеграція	API для підключення зовнішніх сервісів	Відсутня

До недоліків у порівнянні з ЛАС можна відмітити те, що налаштування програми може зайняти багато часу перед запуском. Необхідно перенавчати персонал до нового інтерфейсу також іноземна може сповільнити цей процес. LPI орієнтований більше на управління IT-активами, тому деякі функції можуть бути зайвими або потребувати модифікації під потреби ліфтової служби. Система не містить готових рішень для обліку ліфтів та специфічних технічних параметрів, що може вимагати додаткових розширень або кастомізації.

Проведемо порівняння двох програм, їх переваг, недоліків та можливостей застосування у ліфтовій аварійній службі в таблиці 1.4.

*Таблиця 1.4. Порівняння GLPI та OTRS*

<b>Характеристика</b>	<b>OTRS</b>	<b>GLPI</b>
Основне призначення	Система управління заявками та техпідтримкою	Управління заявками + облік IT-активів
Інтерфейс	Веб-орієнтований, сучасний, але складний	Веб-інтерфейс, інтуїтивно зрозумілий
Простота налаштування	Потребує складної конфігурації	Легший у налаштуванні, але потребує доопрацювання
Автоматизація	Розвинені бізнес-процеси, гнучкі сценарії	Менш розвинена, але достатня для заявок
Ролі та доступ	Гнучка система ролей та прав	Гнучка система ролей та прав
Інтеграція з іншими системами	Підтримка API, інтеграція з e-mail, SMS тощо	API для інтеграції, можливість підключення плагінів

Звітність та аналітика	Розширена статистика, звітність у реальному часі	Вбудована аналітика, але слабша за OTRS
Підтримка мобільних пристроїв	Веб-доступ через браузер	Веб-доступ через браузер, є мобільний застосунок
Вартість	Безкоштовна версія + платні пакети підтримки (від 5000 євро/рік)	Безкоштовна версія + платна хмарна версія (від 3000 євро/рік)
Гнучкість налаштувань	Висока, але потребує додаткової конфігурації	Висока, легко розширюється через плагіни
Придатність для ліфтової аварійної служби	Потребує серйозного налаштування під процеси	Менш складний у налаштуванні, але може потребувати кастомізації

Обидві системи мають свої переваги, і вибір залежить від потреб та ресурсів компанії. OTRS більше підходить для великих організацій із складними бізнес-процесами. Його переваги – потужна автоматизація, аналітика та масштабованість. Проте складне впровадження та вища вартість можуть стати перешкодами. GLPI більш гнучкий у налаштуванні, особливо якщо потрібно не тільки облік заявок, але й керування активами (наприклад, станом ліфтів та обладнання). Проте в нього слабша аналітика і може знадобитися додаткове налаштування для адаптації під диспетчерську службу.

### **1.6. Обґрунтування доцільності проектування й розроблення вебпорталу аварійної служби ліфтової компанії ПрАТ «ОТІС»**

З огляду на існуюче програмне забезпечення для обліку заявок та провівши аналіз аналогічних рішень, було зроблено висновок про необхідність розробки та впровадження нового програмного продукту. Основною причиною вибору власної

розробки є відсутність на ринку готових програмних рішень, які б включали веб-інтерфейс не лише для працівників компанії, а й для клієнтів.

Головна ідея кваліфікаційної роботи полягає у зменшенні навантаження на гарячу лінію шляхом впровадження системи «заповни сам». Такий підхід дозволить користувачам самостійно створювати та вносити заявки до бази даних без необхідності телефонного звернення до диспетчера. Водночас диспетчери продовжуватимуть виконувати функцію приймання та оформлення заявок, але очікується суттєве зниження кількості телефонних дзвінків, а також зменшення кількості помилок у введенні даних через неправильне або нечітке зазначення адреси.

Основними перевагами власного розроблення програмного забезпечення є:

- Повна кастомізація – можливість адаптації веб-порталу під конкретні потреби компанії, включаючи дизайн, функціонал та інтеграцію з іншими системами;
- Відсутність постійних витрат на ліцензування та підтримку сторонніх продуктів – компанія матиме повний контроль над програмним забезпеченням і не залежатиме від зовнішніх постачальників послуг;
- Безпека даних – уникнення ризиків, пов'язаних із передачею конфіденційної інформації третім сторонам, що мінімізує можливість витоку даних;
- Зменшення ризиків під час виконання заявок – у процесі роботи механіки не завжди мають можливість відповісти на телефонний дзвінок. Це може призвести до того, що заявка не дійде до виконавця вчасно або механік відволічеться на розмову/рінгтон, що, у свою чергу, може спричинити аварійну ситуацію. Впровадження веб-додатку дозволить уникнути таких ризиків, адже інформація буде доступною в електронному вигляді без необхідності голосового спілкування.

## 1.7. Постановка задачі

Було запропоновано створити зручний веб-додаток зі зрозумілим інтерфейсом та швидким реагуванням, який буде адаптуватися під різні пристрої. На програмне рішення буде покладена відповідальність за передачу заявок за схемою «Клієнт-Оператор-Механік».

Ця система призначена для збереження всієї інформації про ліфти, їх проблеми. Вона має забезпечити зручний та швидкий доступ до даних, дозволити легко знаходити, редагувати та додавати нову інформацію, а також оперативно формувати необхідну звітну документацію.

Основні функції системи:

- Реєстрація клієнта;
- Авторизація користувача з рівнями доступу: адміністратор, клієнт, механік, оператор;
- Створення заявки під ролями оператор, клієнт;
- Надання доступу механіку до всіх заявок, що належать йому;
- Закриття заявок диспетчером або механіком у зручний для них час;
- Створення звітів;
- Повідомлення клієнтів через пошту про статус виконання заявки.

Планується, що механіку буде приходити повідомлення про заявку на пошту. Веб-портал буде розроблено з використанням php та JavaScript, базою даних буде служити MySQL задля того, щоб відокремити зовнішній додаток від внутрішніх даних. В базі даних будуть занесені дані про користувачів системи, адреси, заявки, ліфти.

## 1.8. Висновки до першого розділу

У першому розділі було подано історію створення компанії Otis Elevator Company "OTIS" та базову характеристику її діяльності. Також було розглянуто структуру та взаємодію посадових осіб у департаменті сервісних операцій.

На основі зібраних даних було проведено аналіз стану комп'ютеризації та організації робочих процесів у ПрАТ «ОТІС». У результаті, було виявлено низку проблем, після чого керівництву компанії були надані відповідні пропозиції.

Було запропоновано впровадження веб-порталу для аварійної ліфтової служби ПрАТ «ОТІС» з метою підвищення ефективності роботи. Нова система покращить взаємодію між клієнтами та сервісною службою, зменшить навантаження на персонал і підвищить ефективність обробки заявок.

Розробка цього рішення дозволить автоматизувати та впорядкувати робочі процеси, що, у свою чергу, підвищить рівень безпеки та комфорту для клієнтів. Гнучкість і масштабованість системи забезпечать можливість подальшого вдосконалення, інтеграції з іншими сервісами та розширення функціоналу відповідно до потреб компанії.

Отже, створення сучасного веб-порталу для передачі заявок і моніторингу їх виконання є актуальним та необхідним кроком. Такий інструмент не лише спростить взаємодію мешканців із аварійною службою, а й зменшить навантаження на диспетчерів, підвищить точність переданої інформації та дозволить ефективніше реагувати на аварійні ситуації.

## **РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ**

### **2.1. Загальні положення**

2.1.1. Найменування системи: «Веб-портал аварійної служби ліфтової компанії “ОТІС”».

2.1.2. Результати робіт зі створення даної системи повинні оформлюватися згідно з вимогами державних стандартів на відповідні етапи розроблення. Порядок оформлення та передачі результатів визначається змістом і календарним планом виконання проєкту.

2.1.3. За необхідності на наступних стадіях розробки окремі положення цього технічного завдання можуть уточнюватися чи доповнюватися.

### **2.2. Призначення і цілі створення системи**

#### **2.2.1. Призначення системи.**

Система призначена для автоматизації роботи аварійно-диспетчерської служби ліфтової компанії “ОТІС”. Вона забезпечує єдиний веб-портал для реєстрації та обробки аварійних викликів, пов’язаних з несправностями ліфтів, координації дій персоналу з їх усунення, моніторингу статусів ремонтних робіт та ведення відповідної звітності. Система дозволяє диспетчерам оперативно приймати заявки про несправності ліфтів, призначати на них виїзних механіків, відстежувати хід виконання робіт, а клієнтам – повідомляти про проблеми та отримувати інформацію про хід і результати їхнього вирішення. Таким чином, впровадження порталу забезпечить своєчасну та скоординовану реакцію на аварійні ситуації з ліфтами.

#### **2.2.2. Цілі створення системи.**

Основною метою створення системи є підвищення оперативності та ефективності роботи аварійної служби за рахунок автоматизації процесів обліку аварійних заявок і ремонту ліфтів. Система забезпечить оперативне отримання повної та достовірної інформації щодо кожного випадку несправності, мінімізує людський фактор при передачі даних та дозволить накопичувати статистичні дані для подальшого аналізу. Це створить умови для оптимізації діяльності аварійно-

диспетчерської служби: скорочення часу простою ліфтів, покращення контролю за виконанням ремонтів та підвищення рівня обслуговування клієнтів.

Завдання оперативного управління аварійно-диспетчерською службою вимагають координації між багатьма учасниками процесу. В умовах функціонування автоматизованої системи з'являється можливість виконувати всі поставлені завдання на сучасному рівні, забезпечуючи необхідні точність, оперативність, зменшення навантаження на операторів та достовірність обробки інформації.

### **2.3. Характеристика об'єкта автоматизації**

#### **2.3.1. Короткі відомості про об'єкт автоматизації.**

Об'єктом автоматизації є діяльність аварійно-диспетчерської служби ПрАТ «ОТІС» у місті Києві. Базове підприємство впровадження системи – ПрАТ «ОТІС» (03062, Україна, м. Київ, вул. Чистяківська, буд. 32). Аварійна служба підприємства займається прийомом повідомлень про несправності ліфтів від мешканців і обслуговуючих організацій, диспетчеризацією викликів та координацією роботи виїзних механіків. На даний час аварійна служба розподілена по містах на дільниці, за якими закріплені відповідні адреси обслуговування і чергові аварійні механіки. Система впроваджується з метою автоматизації зазначених процесів на цьому підприємстві.

### **2.4. Вимоги до системи**

#### **2.4.1. Вимоги до системи в цілому.**

##### **2.4.1.1. Вимоги до структури і функціонування системи.**

2.4.1.1.1. Система повинна мати багатокористувацьку клієнт-серверну архітектуру, що використовує єдину БД. Веб-портал реалізується як клієнтська частина (інтерфейс у веб-браузері) та серверна частина (сервер додатків і БД). Передбачається підтримка чотирирівневої системи доступу: Адміністратор, Диспетчер, Механік, Клієнт – кожна категорія користувачів матиме визначені права доступу до функцій та даних системи.

Згідно з функціональною структурою роботи аварійної служби, система повинна забезпечувати мережеву взаємодію між автоматизованими робочими місцями:

- адміністратора;
- диспетчера аварійної служби;
- лінійного та аварійного механіка;
- клієнта (мешканця будинку або представника організації, що подає заявку).

2.4.1.1.2. У процесі функціонування системи в мережевому середовищі має бути передбачений механізм виявлення відхилень від нормального процесу обробки заявок, а також збоїв у роботі апаратного забезпечення чи програмних помилок. Користувачі мають бути забезпечені своєчасними відповідними діагностичними повідомленнями про помилки. Взаємодія між підсистемами (модулями) має бути реалізована на основі єдиної бази даних із використанням засобів локальних мереж або Інтернет. Система повинна реалізовувати логіку автоматичного спрямування (маршрутизації) заявок до відповідальних осіб згідно з типом проблеми та належністю адреси до певної ділянки, що підвищить швидкість реагування на аварійні виклики.

2.4.1.2. Вимоги до чисельності і кваліфікації персоналу.

2.4.1.2.1. Персонал, який використовує автоматизовану систему, повинен відповідати наступним вимогам:

- пройти необхідне навчання і мати навички роботи на персональному комп'ютері;
- дотримуватись технологічних інструкцій при роботі з системою в діалоговому режимі;
- виконувати експлуатаційні правила роботи на ПК відповідно до інструкцій з технічної експлуатації;
- дотримуватися встановлених правил зберігання інформації (регулярно виконувати резервне копіювання бази даних) та забезпечувати цілісність даних;

- дотримуватися правил техніки безпеки при роботі з комп'ютером (вимоги електробезпеки, ергономіки робочого місця тощо).

2.4.1.2.2. Користувачами системи є штатні співробітники компанії та клієнти, зокрема: адміністратор, диспетчери, лінійні та аварійні механіки, представники клієнтів. Впровадження системи не потребує розширення штату – додаткове наймання працівників не передбачається, оскільки функції з обслуговування системи можуть виконуватися наявним ІТ-спеціалістом, або суміщатись з чинними обов'язками диспетчерів або керівників. Кількість користувачів визначається існуючою структурою підприємства та числом клієнтів, що обслуговуються: орієнтовно, 1 адміністратор, 3–5 диспетчерів (залежно від кількості змін та навантаження), від 10 механіків, а кількість клієнтів, які матимуть доступ до порталу має може доходити до 100.

2.4.1.2.3. Вхід до системи здійснюється лише авторизованими користувачами за допомогою логіна та пароля, які визначають рівень доступу користувача. Залежно від привілеій, користувач отримує повний чи обмежений доступ до функцій системи (з правом редагування даних або лише перегляду) . Таким чином, система підтримує розмежування доступу до інформації між різними категоріями персоналу, що забезпечує як безпеку даних, так і зручність роботи кожного користувача в межах його компетенції.

2.4.1.2.4. Персонал підприємства повинен бути ознайомлений з інструкцією користувача системи і мати чітко визначені посадові обов'язки щодо роботи з новою системою. За необхідності проводиться додаткове навчання диспетчерів і механіків особливостям роботи у веб-порталі (навігація по інтерфейсу, введення даних, формування звітів). Рівень кваліфікації користувачів (диспетчерів, механіків) – середній (базова комп'ютерна грамотність, уміння працювати з веб-браузером). Адміністратор системи повинен мати поглиблені знання в області ІТ (володіти навичками роботи з СУБД, адміністрування веб-серверів, основами кібербезпеки).

#### 2.4.1.3. Вимоги до надійності.

2.4.1.3.1. Система є багатофункціональною і призначена для постійної роботи (у режимі 24/7 або принаймні протягом усього робочого дня диспетчерської служби). Всі функції системи повинні виконуватися точно і коректно, без критичних збоїв. Надійність програмно-технічного комплексу визначається надійністю програмного забезпечення (в першу чергу СУБД) та апаратних засобів, на яких функціонує система. Оцінка надійності може проводитися по кожній функції окремо, відповідно до чинних стандартів. Основними показниками надійності системи є:

$L_i$  — ймовірність безвідмовного виконання задачі в заданий термін (імовірність того, що  $i$ -тий запит буде виконаний);

$K_r$  — коефіцієнт готовності ПТК (програмно-технічного комплексу);

$T_v$  — середній час відновлення ПТК;

$T_e$  — мінімальний час між двома відмовами за календарний місяць.

2.4.1.3.2. Комплекс технічних засобів системи повинен дозволяти виконання всіх функціональних задач з різних робочих місць (диспетчерів, механіків, адміністратора) навіть у випадку виходу з ладу окремих станцій. У разі відмови сервера повинна бути можливість оперативного відновлення роботи системи з резервної копії на резервному обладнанні (або переключення на дублюючий сервер, якщо передбачено). Для захисту даних від втрати при аварійних ситуаціях (збій обладнання, відключення електроживлення тощо) необхідно застосовувати регулярне резервне копіювання бази даних та зберігання копій на надійному носії.

2.4.1.3.3. Очікуваний рівень безвідмовності системи – не менше 99% часу доступності протягом місяця. Середній час відновлення після відмов (при наявності актуальної резервної копії) не повинен перевищувати 2 годин. Відмова окремих компонентів (клієнтського місця або мережевого з'єднання одного з користувачів) не повинна призводити до порушення роботи системи в цілому – інші користувачі мають продовжувати роботу без перешкод.

#### 2.4.1.4. Вимоги до безпеки.

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів системи слід дотримуватися вимог наступних стандартів з охорони праці та пожежної безпеки:

- ДСТУ 2293-99 “Охорона праці. Терміни та визначення”;
- ДСТУ ISO 6309:2007 “Протипожежний захист. Вимоги до знаків безпеки і пожежного обладнання”;
- ДСТУ 12.0.230:2008 “Система стандартів безпеки праці. Пристрої захисного відключення. Загальні вимоги”;
- ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011 – вимоги щодо пожежної безпеки ліфтового господарства.

По допустимих рівнях освітленості, вібраційних і шумових навантажень на робочих місцях слід дотримуватися вимог відповідно до ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

Сама програмна система повинна забезпечувати безпечне виконання операцій, не створюючи ризиків для життя і здоров'я людей. Оскільки система не втручається безпосередньо в керування ліфтом, а лише інформує персонал, вона не може безпосередньо спричинити аварійну ситуацію з обладнанням. Проте система має сприяти загальній безпеці пасажирів ліфтів за рахунок прискорення реагування на несправності та зменшення часу простою ліфтів. У разі виникнення аварійних ситуацій в роботі самої системи (відмова серверу, збій БД) повинен бути визначений відповідальний адміністратор, який оперативно приймає рішення щодо відновлення системи (перехід на резервне обладнання, відновлення з бекапу).

#### 2.4.1.5. Вимоги до ергономіки та технічної естетики.

Загальні ергономічні та естетичні вимоги до системи повинні відповідати чинним стандартам (ДСТУ 8604:2015, ДСТУ 7298:2013). Освітленість робочого місця користувача системи повинна відповідати нормам ДСТУ EN 12464-1:2016 та ДБН В.2.5-28:2006 (нормативи штучного освітлення).

Засоби відображення інформації (монітори) слід розміщувати таким чином, щоб кут спостереження екрану становив не більше  $45^\circ$ , мінімальна відстань від очей до екрану – 0,3 м, рекомендована – не менше 0,5 м.

При розробленні інтерфейсу програмного забезпечення слід створити зручний та інтуїтивно зрозумілий інтерфейс користувача (меню, діалоги, підказки) для запобігання зайвому навантаженню і втомлюваності користувача при роботі з системою.

2.4.1.6. Вимоги по експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи.

2.4.1.6.1. Види технічного обслуговування системи визначаються відповідно до ДСТУ EN 13306:2019. Загальні вимоги щодо експлуатації, технічного обслуговування і ремонту повинні відповідати ДСТУ 3576-97.

2.4.1.6.2. Для розміщення технічних засобів системи необхідні площі, визначені в ДБН В.2.2-9-2009 (нормативи площ для обчислювальної техніки). При цьому слід дотримуватися вимог, зазначених в експлуатаційній документації на обладнання. Напруга живлення технічних засобів системи – 220/380 В змінного струму частотою  $(50 \pm 1)$  Гц. Допустиме відхилення напруги – від +10% до –15%. Тривалість перерв у електроживленні не повинна перевищувати 0,001 с.

2.4.1.6.3. Кількість, кваліфікація і режими роботи обслуговуючого персоналу повинні відповідати рекомендаціям, зазначеним в технічних умовах та інструкціях з експлуатації обладнання системи.

2.4.1.6.4. Склад, розміщення і умови зберігання компонентів технічних засобів системи визначаються вимогами, зазначеними в експлуатаційній документації на ці компоненти.

2.4.1.6.5. Регламент (періодичність) обслуговування системи має відповідати рівню складності та умовам роботи обладнання, щоб у разі відмови системи була забезпечена можливість роботи в аварійному режимі.

2.4.1.7. Вимоги до захисту інформації від несанкціонованого доступу.

2.4.1.7.1. Для надійного збереження інформації і контролю доступу в системі необхідно використовувати такі засоби захисту:

- засоби керування доступом, що надаються серверними операційними системами (наприклад, сімейства Windows Server);
- засоби захисту локальної мережі та мережеві екрани (Firewall) для запобігання несанкціонованому доступу ззовні;
- механізми захисту клієнт-серверної СУБД, зокрема:
  - тригери, подання (для контролю цілісності даних);
  - внутрішні процедури та функції (для перевірки та обробки даних);
  - розмежування прав доступу шляхом налаштування груп користувачів і ролей.

2.4.1.7.2. Кожен сеанс роботи користувача в системі повинен починатися з введення індивідуального пароля. Система парольного захисту повинна мати засоби періодичної примусової зміни паролів або використовувати стандартні засоби середовища розроблення. Для надійного захисту від несанкціонованого доступу кожен співробітник матиме власний обліковий запис та пароль.

2.4.1.8. Вимоги щодо збереження інформації при аваріях.

2.4.1.8.1. Необхідно передбачити засоби резервного збереження даних системи. Після кожного оновлення або коригування інформації у базі даних повинні створюватися резервні копії (архівні дампи БД). Має бути забезпечена можливість відновлення бази даних із архівної копії у випадку її пошкодження або втрати даних.

2.4.1.8.2. Резервний архів та основна база даних мають зберігатися на різних фізичних носіях чи пристроях, що унеможливило одночасну втрату даних при виході з ладу обладнання.

2.4.1.9. Вимоги по захисту від впливу зовнішніх факторів.

2.4.1.9.1. Електрична складова електромагнітного поля завад у приміщеннях, де розміщені технічні засоби системи, не повинна перевищувати  $0,3 \text{ В/м}^2$  в діапазоні частот від 0,15 до 300 МГц. Для захисту від впливу електромагнітних полів та промислових завад слід передбачити встановлення відповідних екранів та фільтрів.

2.4.1.9.2. Засоби, які усувають вплив шкідливих факторів на функціонування комплексу технічних засобів, повинні бути спроектовані відповідно до ДБН В.2.2-9-2009. Стійкість обчислювальних засобів до зовнішніх впливів повинна відповідати ДСТУ 2506-94.

#### 2.4.1.10. Вимоги до патентної частини.

Патентні дослідження при створенні даної системи не проводяться. Розробка програмного забезпечення здійснюється з дотриманням ліцензійних умов використання інструментальних засобів та компонентів, що не порушує чинних патентів і авторських прав.

#### 2.4.1.11. Вимоги по стандартизації та уніфікації.

Система повинна використовувати загальноприйняті класифікатори і стандарти кодування інформації. Формати даних, що передаються і зберігаються, мають відповідати міжнародним стандартам (наприклад, кодування символів UTF-8, стандартизовані формати дати і часу тощо) з метою забезпечення сумісності та уніфікації.

#### 2.4.2. Вимоги до функцій.

Перелік функцій системи із зазначенням вхідної та вихідної інформації наведено в таблиці 2.1. Функції системи мають забезпечити раціональну організацію роботи користувачів на основі безперервної технології обробки інформації: ведення бази даних (довідників адрес, ліфтів, користувачів тощо), реєстрацію і підтримку життєвого циклу заявок, формування різнорідних звітів, а також виконання інших функцій, визначених цим технічним завданням. При цьому пріоритетом є зручність введення та використання інформації користувачами за рахунок розвинутого інтерфейсу з меню та підказками на екрані.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

Найменування функції	Вхідна інформація	Вихідна інформація
Реєстрація нового клієнта в системі	Дані, введені користувачем (ПІБ, адреса, тощо)	Створений запис облікового запису; підтвердження реєстрації (повідомлення на екрані)
Створення аварійної заявки на ремонт ліфта	Дані заявки від користувача (адреса, ліфт, тип проблеми); довідники адрес і ліфтів з БД	Нова заявка у базі
Обробка та маршрутизація заявки диспетчером	Нові заявки з категорією "інші" (непідтверджені автоматично); довідники механіків та діляниць	Оновлена заявка (підтверджена та призначена механіку або відхилена); сповіщення механіка про нову заявку
Моніторинг стану ліфтів	Дані з датчиків або журналу зупинок ліфтів; БД ліфтів (статуси); інформація від механіків	Список ліфтів із зазначенням стану (працює/в зупинці); перелік ліфтів, що перебувають в аварійній зупинці
Управління станом ліфта (зупинка/ запуск)	Вибір ліфта та команда від користувача (диспетчера/ механіка); поточний статус ліфта	Оновлений статус ліфта (відображається як "в зупинці" або "працює"); повідомлення про успішне виконання операції

Закриття (завершення) заявки механіком або диспетчером	Обрана активна заявка; звіт від механіка або диспетчера (причина та усунення несправності), поточний статус ліфта	Оновлений запис заявки (статус “закрито”, причина несправності, коментар); оновлений статус ліфта (якщо занесено в статус “в зупинці”)
Формування звітів про роботу аварійної служби	Дані БД про заявки та ліфти (історія заявок, стан обладнання)	Звіти (вибірки) за заданий період у вигляді таблиць на екрані або друкованих форм (кількість заявок, час реагування, кількість несправних ліфтів тощо)
Адміністрування системи (управління користувачами)	Дані про нових співробітників	Нові співробітники компанії в БД

#### 2.4.3. Вимоги до видів забезпечення.

##### 2.4.3.1. Вимоги до математичного забезпечення (МЗ).

Система не потребує розроблення спеціальних методів чи алгоритмів обчислювального характеру для реалізації покладених на неї функцій. Існуючих засобів обраної системи управління базами даних достатньо для виконання всіх операцій, тому спеціальні вимоги до математичного забезпечення не висуваються.

##### 2.4.3.2. Вимоги до інформаційного забезпечення (ІЗ).

2.4.3.2.1. Інформаційне забезпечення системи повинно містити дані, достатні для виконання всіх покладених на систему функцій. ІЗ має гарантувати раціональну організацію зберігання інформації та доступу до неї. Необхідно передбачити створення і ведення довідникової інформації (бази даних адрес обслуговування, переліку ліфтів з їх характеристиками, списку користувачів та їх ролей, типових причин несправностей тощо).

2.4.3.2.2. Поповнювання бази даних необхідною інформацією (довідниками адрес, обладнання тощо) покладається на сторону замовника згідно методик та шаблонів, які були розроблені розробниками системи. Структура, наповнення та принципи організації інформації будуть визначені у проєктній документації, зокрема в логічній моделі бази даних, і можуть уточнюватися на етапі технічного проєктування.

2.4.3.2.3. Слід передбачити захист даних від втрати при аваріях і збоях електроживлення – зокрема, використання резервного копіювання бази даних та механізмів журналу транзакцій для відновлення незбережених змін.

2.4.3.3. Вимоги до лінгвістичного забезпечення (ЛЗ).

2.4.3.3.1. Для розроблення програмних засобів, які реалізують функції системи і забезпечують роботу користувацького інтерфейсу, повинні використовуватися сучасні мови програмування високого рівня, що дозволяють створювати структуровані програмні компоненти. Зокрема, серверна частина веб-порталу розробляється мовою PHP із застосуванням об'єктно-орієнтованого підходу, клієнтська частина – з використанням мови JavaScript, розмітки HTML/CSS. Для доступу до даних та маніпулювання ними використовується мова запитів SQL обраної СУБД (MySQL).

2.4.3.3.2. Організація діалогу користувача з системою будується на основі наборів меню і контекстних підказок, орієнтованих на виконання користувачем визначених функцій. Запити користувача до системи (пошук інформації, введення заявок тощо) повинні формулюватися переважно у формі, наближеній до природної мови, з використанням зрозумілої термінології предметної області.

2.4.3.4. Вимоги до програмного забезпечення (ПЗ).

2.4.3.4.1. Загальносистемне програмне забезпечення повинно забезпечувати надійне і якісне виконання функціональних завдань системи. До загальносистемного ПЗ належать:

- операційна система сервера – Windows Server 2019 або CloudLinux 8.9; операційна система клієнтських робочих станцій – Windows 10 (або сумісна ОС для кінцевих користувачів);

- система керування базою даних (СУБД) – MySQL 8.x;
- рекомендоване використання веб-сервера Apache;
- на стороні клієнта загальносистемним програмним забезпеченням є сучасний веб-браузер (Google Chrome, Mozilla Firefox, Microsoft Edge або аналогічний) останньої версії, що підтримує HTML5, CSS3 та JavaScript ES6. Операційна система клієнтських машин може бути Windows 10/11 або інша, сумісна з вимогами браузера (наприклад, Linux дистрибутив). Наявність офісного ПЗ або спеціальних драйверів на клієнтських ПК не є критичною – достатньо стандартного мережевого підключення та браузера.

#### 2.4.3.4.2. Загальні вимоги до системного програмного забезпечення:

- мінімальні вимоги до апаратних ресурсів (оптимальне використання пам'яті та дискового простору);
- максимальна швидкодія при виконанні операцій вводу/виводу та обробки даних;
- повне задоволення потреб функціональних завдань системи.

#### 2.4.3.4.3. Вимоги до операційної системи сервера:

- мінімальне споживання ресурсів комп'ютера для власних потреб ОС (особливо оперативної пам'яті та дискового простору);
- забезпечення максимальної швидкодії при керуванні зовнішніми пристроями і мережевими запитам;
- висока стабільність та засоби відмово стійкості (автоматичне відновлення сервісів після збоїв, журнали подій тощо).

#### 2.4.3.4.4. Вимоги до СУБД:

- повне задоволення потреб функціональних задач за обсягами та структурою даних;
- надійність і забезпечення цілісності даних;
- ефективне управління великими обсягами даних;
- висока швидкість виконання запитів користувачів;

- низькі вимоги до технічних ресурсів (ефективне використання пам'яті, CPU).

2.4.3.4.5. Вимоги до прикладного програмного забезпечення (спеціального ПЗ) для введення, обробки:

- забезпечувати виведення необхідних даних на екран у вигляді зручних форм і повідомлень;
- супроводжувати введення даних контролем коректності та сигналізацією про помилки з можливістю виправлення під час введення;
- підтримувати керований комп'ютером діалог при введенні даних (навігація по формах, підказки);
- надавати можливість виведення даних у друкованому вигляді (звітні форми) на запит користувача.

2.4.3.5. Вимоги до технічного забезпечення.

2.4.3.5.1. Технічні засоби системи (Таблиця 2.2) повинні забезпечувати виконання функцій, перелічених у таблиці 2.1.

2.4.3.5.2. Обчислювальна техніка має підтримувати обмін інформацією в необхідних обсягах та з необхідною швидкістю для гарантування своєчасної обробки заявок і взаємодії користувачів у реальному часі.

*Таблиця 2.2. Орієнтовані вимоги до технічного забезпечення системи*

Основні характеристики	Сервер	Клієнт
Процесор (CPU)	4-ядерний, 64-розрядний, 2.5 ГГц (або потужніше)	2-ядерний, 2.0 ГГц (або аналогічний)
Оперативна пам'ять (RAM)	8 Гб і більше	8 Гб (мінімум 4 Гб)
Накопичувач даних (HDD/SSD)	500 Гб HDD (рекомендовано RAID1/5 для надійності)	250 Гб HDD/SSD

Мережеве підключення	1 Gbit/s Ethernet (локальна мережа або Інтернет)	Широкопasmовий доступ до Інтернет/ЛАН
Периферійні пристрої	- (не потрібні безпосередньо на сервері)	Монітор 17", клавіатура, миша; принтер (за потреби)

2.4.3.5.3. Примітка: Наведені характеристики є мінімальними для комфортної роботи системи. За наявності більш продуктивного обладнання система зможе обслуговувати більшу кількість одночасних користувачів та швидше обробляти дані. Вибір конкретної конфігурації сервера залежить від фінансових можливостей Замовника та планованого навантаження (масштабу впровадження).

#### 2.4.3.6. Вимоги до метрологічного забезпечення.

Система не виконує функцій вимірювання фізичних величин і не використовує вимірювальних пристроїв або приладів. Всі дані, що обробляються, мають довідковий або обліковий характер (текст, числа, час, дати), тому метрологічне забезпечення для даного програмного продукту не є застосовним. Вимоги до цього виду забезпечення не висуваються.

#### 2.4.3.7. Вимоги до організаційного забезпечення.

2.4.3.7.1. Організаційне забезпечення функціонування системи повинно розроблятися у відповідності до вимог державних стандартів з автоматизованих систем управління (серія стандартів ЄСАС).

2.4.3.7.2. Впровадження системи не потребує збільшення штатної чисельності підприємства. Територіальне розміщення робочих місць не передбачається, адже система загальнодоступна і доступ можна отримати навіть з телефону.

2.4.3.7.3. До функціонування системи встановлюються такі організаційні вимоги:

- Наказом керівника підприємства визначається перелік співробітників, які мають право доступу до системи (адміністратори, диспетчери, аварійні механіки та лінійні механіки);
- Контроль за роботою системи та прийняття рішень при нестандартних або аварійних ситуаціях у процесі експлуатації здійснює відповідальний за систему, призначений наказом по підприємству.

## 2.5. Склад і зміст робіт зі створення системи

2.5.1. Стадії створення системи і терміни виконання робіт наведено в таблиці 2.3.

*Таблиця 2.3. – Орієнтовані вимоги до технічного забезпечення системи*

Найменування етапу	Строк виконання (до дати)
Передпроектне дослідження об'єкта автоматизації	01.05.2025
Технічне завдання (розробка і погодження)	06.05.2025
Технічний проєкт (розробка системи)	13.05.2025
Оформлення документації (дипломного проєкту)	18.05.2025

## 2.6. Порядок контролю і приймання системи

2.6.1. Впровадження розробленої системи планується на діючому підприємстві (в аварійно-диспетчерській службі компанії “ОТІС”). Перед введенням системи в експлуатацію необхідно провести приймальні випробування згідно з вимогами стандарту ДСТУ 3974-2000. Приймальні випробування – це заключний етап перевірки працездатності системи, метою якого є визначення відповідності розробленої системи встановленим в технічному завданні вимогам та вирішення питання про допуск системи до експлуатації. Перевірка працездатності системи та ухвалення рішення, щодо можливості її провадження в режим дослідної експлуатації здійснюється розробниками разом із Замовником. Програму і

методику випробувань розробники готують заздалегідь, і підлягають погодженню та затвердженню Замовником до початку тестування.

2.6.2. Передача системи в режим дослідної експлуатації здійснюється на підставі затвердженого технічного завдання та розробленої заздалегідь інструкції користувача. Після завершення дослідної експлуатації формується список рекомендованих або необхідних доробок із зазначеними термінами їх реалізації. Введення системи в промислову експлуатацію підтверджується підписанням акту приймання-передачі між розробником і замовником.

## **2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію**

Для введення системи в експлуатацію замовник має здійснити комплекс підготовчих робіт на об'єкті автоматизації:

- проводить докупку або повна закупка необхідних технічних засобів (закупівля та встановлення серверного обладнання, робочих станцій, мережевого оснащення тощо);
- забезпечує проведення навчання користувачів щодо роботи з системою (основи роботи на ПК для персоналу, інструктаж з правил роботи у веб-порталі, вивчення керівництва користувача);
- здійснює дослідну експлуатацію системи, за результатами якої приймає рішення чи вводити систему в режим постійної експлуатації.

## **2.8. Вимоги до документації**

2.8.1. Для системи формується повний комплект документації, що несе за собою: технічне завдання (даний документ) та технічний проєкт (документ, що містить деталізований дизайн системи, структуру бази даних, схеми алгоритмів, інструкції тощо). За підсумками дипломного проєкту може додатково розроблятися користувацька документація (керівництво оператора/адміністратора системи).

2.8.2. Документація на систему створюється у відповідності до вимог чинних державних стандартів серії ДСТУ 19 «Єдина система програмної документації» та серії ДСТУ 24 «Єдина система стандартів автоматизованих систем управління».

Структура і оформлення документів відповідають ДСТУ 3008-2015 (звітна документація для сфери науки і техніки) та іншій чинній нормативній базі.

## 2.9. Джерела розробки

2.9.1. Під час підготовки цього технічного завдання та проєктної документації для системи були використані такі нормативні документи та літературні джерела:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. (Використаний як стандарт щодо структури та оформлення тексту технічного завдання та пояснювальної записки дипломного проєкту);
- ДСТУ 3973–2000. Система розроблення та поставлення продукції на виробництво. (Використаний як базовий стандарт, що визначає стадії створення автоматизованих систем та вимоги до змісту технічного завдання);
- ДСТУ 3974-2000. Система розроблення та поставлення продукції на виробництво. Правила виконання дослідно-конструкторських робіт. Загальні положення . (Регламентує порядок проведення приймальних випробувань та дослідної експлуатації дослідних зразків програмних засобів; застосовувався при визначенні порядку приймання системи);
- ДСТУ Б В.2.5–82:2016. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. (Нормативний документ, врахований при формулюванні вимог з електробезпеки обладнання, на якому працює система).

## РОЗДІЛ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту

Веб-портал ліфтової компанії ОТІС було розроблено безпосередньо на хостинговому сервері Ukrline, що дозволило від початку проєктування перевіряти роботу системи в умовах наближених, до реальних робочих. Завдяки цьому підходу була усунута потреба в перенесенні проєкту з локального середовища, було спрощено тестування на різних пристроях та браузерях, була забезпечена стабільна інтеграція з реальними сервісами.

Сервер працює під ОС CloudLinux 8.9, яка базується на Red Hat Enterprise Linux (RHEL).

CloudLinux – це ОС, створена американською компанією CloudLinux Inc (США), яка відповідає за її підтримку. CloudLinux здобула значне поширення у сфері хостингу завдяки своїй унікальності та спеціальним можливостям. CloudLinux побудована на основі таких дистрибутивів як ОС CentOS і Red Hat Enterprise Linux, що дає повну сумісність з ПЗ, яке було написано для даних ОС. Це окремий комерційний продукт, який орієнтований на потреби хостинг-компаній і користувачів, чії розробки розміщені на даному рішенні [21].

Характеристики хостингової платформи:

- Apache: 2.4.37;
- MySQL: 8.0.32;
- PHP: 8.2.19.

Веб-сервер Apache використовується в сфері веб розробки для обробки запитів користувачів та наданні доступу до сторінок порталу.

Apache – це кросплатформлене ПЗ для серверів з відкритими кодами, яке відіграє важливу роль в наповненні Інтернету. Він призначений для обслуговування веб-вмісту, як HTML сторінки, зображення, мультимедія, тощо. Був розроблений

та підтримується спільнотою Apache Software Foundation. Має перевагу у виді гнучкості, потужності та широкої сумісності з різними ОС [18].

Для зберігання інформації про заявки, користувачів та ліфтів використовується реляційна СУБД MySQL, що забезпечує надійність, підтримку транзакцій, зовнішніх ключів та високу швидкодію. База MySQL була обрана з огляду на її технічні переваги, безкоштовну ліцензію, а також, завдяки створенню окремої бази для відкритого програмного продукту, була надана ізоляція від корпоративної бази, що забезпечує безпеку та гнучкість у розробці.

MySQL — це одна з найпопулярніших систем управління реляційними базами даних (СУРБД), яка призначена для збереження, структурування та обробки інформації. Популярність була здобута завдяки своїй простоті використання, надійності, високій продуктивності, стабільності роботи та ефективності [32].

Серверна частина була реалізована з використанням PHP, що є однією з найпоширеніших мов для веб-розробки, що в майбутньому, якщо будуть необхідні оновлення, дозволить швидко знайти спеціаліста. Обрана версія є стабільною та сумісною з бібліотеками та фреймворками.

PHP (англ. PHP: Hypertext Preprocessor - "PHP: препроцесор гіпертексту") – скриптова мова програмування, що застосовується для розробки сайтів та веб-додатків. На даних час вона є підтримувана майже всіма хостинг-компаніями, такими як UkrLine, та є лідером серед мов програмування, для динамічних веб-сторінок. PHP проста для навчання та опанування [24].

Для створення інтерфейсу було використано HTML5, CSS3 та JavaScript, що дозволило створити динамічний, адаптивний та зручний інтерфейс, що буде сумісним з сучасними браузерами.

HTML (мова розмітки гіпертексту) – це основа створення веб-сторінок, яка працює через систему тегів для структурування та компоновати елементів веб-сторінок. Вона визначає які елементи мають бути присутні на сторінці та де саме будуть розташовані, до прикладу Заголовки, Текстові блоки, Зображення або гіперпосиланням. Це як каркас будинку, на основі якого формують просторову структуру [31].

CSS (аббревіатура від Cascading Style Sheets) - це спеціальна мова стилів, за допомогою якої розробляють вигляд сторінок (як і де відобразити елементи веб-сторінки), написаних мовами розмітки даних [22].

JavaScript – це скриптова мова програмування високого рівня, яка активно застосовується створення динамічних та інтерактивних застосунків і сайтів. Завдяки JavaScript веб-сторінки набувають інтерактивності: додається анімація, перевірка форм, спливаючі повідомлення, оновлення контенту в реальному часі без перезавантаження сторінки, реакція інтерфейсу на наші дії. Мова JavaScript забезпечує функціональність сторінки [23].

Для додавання адміна в систему було використано фреймворк Laravel, який був доданий через Shell-клієнт на сервері. Як альтернативу можна було б використати HTML-форму, але використання даного фреймворку дало змогу пришвидшити розгортання користувача з мінімальним ризиком помилок та високим рівнем безпеки.

Laravel — це міжплатформовий PHP-фреймворк, що використовується для створення веб-додатків. Він має багато переваг у використанні, адже дозволяє розробнику користуватися можливостями великої бібліотеки попередньо запрограмованих функцій [10].

Для надсилання листів підтвердження чи зміни паролю на пошту користувача, було використано бібліотеку PHPMailer, що забезпечує підтримку SMTP, кодувань, вкладень та налаштувань відправника. Альтернативою могло б послужити використання стандартної функції mail() у PHP, але вона блокується багатьма поштовими сервісами.

PHPMailer - це зручний інструмент, який допомагає відправляти електронні листи з ваших веб-додатків, створених за допомогою мови програмування PHP. Він надає простий та надійний спосіб відправки пошти через веб-сайти [36].

Для формування звітів у форматі PDF було використано бібліотеку mPDF. Можливим аналогом могла б виступити бібліотека TCPDF, але mPDF має простішу інтеграцію та кращу підтримку верстання HTML сторінок у PDF без додаткових змін.

mPDF — це бібліотека PHP із відкритим вихідним кодом, яка дозволяє користувачам створювати, змінювати та маніпулювати PDF-документами у власних програмах PHP. Бібліотека підтримує створення PDF-файлів із HTML у кодуванні UTF-8 і приймає HTML у кодуванні UTF-8 [35].

Фреймворки та зовнішні бібліотеки були встановлені за допомогою Composer, що являє собою офіційним менеджером залежностей для PHP. Завдяки ньому в файли сайту були завантажені необхідні компоненти.

Редагування коду відбувалось через вбудований файловий менеджер хостингу та завдяки локальним рішенням: Visual Studio Code та Note Pad++.

Для публікації був створений піддомен evis, який був зареєстрований через сервіс NIC.UA, що був обраний через простоту інтерфейсу, можливості керування DNS-записами та повній сумісності з хостингом UkrLine.

Всі обрані технології в переважній більшості є безкоштовними або передбачають повний доступ за одну оплату. Широко підтримуються розробниками, що несе за собою актуальність, стабільність, масштабованість та перспективи подальшого розвитку та оновлення проєкту.

### **3.2. Проєктування та створення бази даних**

Для зберігання та обробки інформації веб-порталу аварійної служби ліфтової компанії «ОТІС» було спроектовано базу даних, яка має назву lift. Структура бази побудована з урахуванням особливостей дослідженої предметної області, а саме – обробка заявок, облік користувачів з ролями, адреси, технічного стану та довідників.

На першому етапі було створено логічну та фізичну моделі бази даних в середовищі Erwin Data Modeler, яка включила в себе ключові сутності, атрибути та зв'язки. Модель дала можливість візуально відобразити структуру, виявити проблеми, перевірити нормалізацію та забезпечити логічну узгодженість таблиць.

Програмне забезпечення Erwin Data Modeler це інструментальний засіб, який забезпечує розробника усіма необхідними функціями для створення, візуалізації і

документування моделей бази даних, а також дозволяє виконувати подальше перетворення логічної чи фізичної моделі в базу даних [27].

Логічну модель бази даних показано на рисунку 3.1.

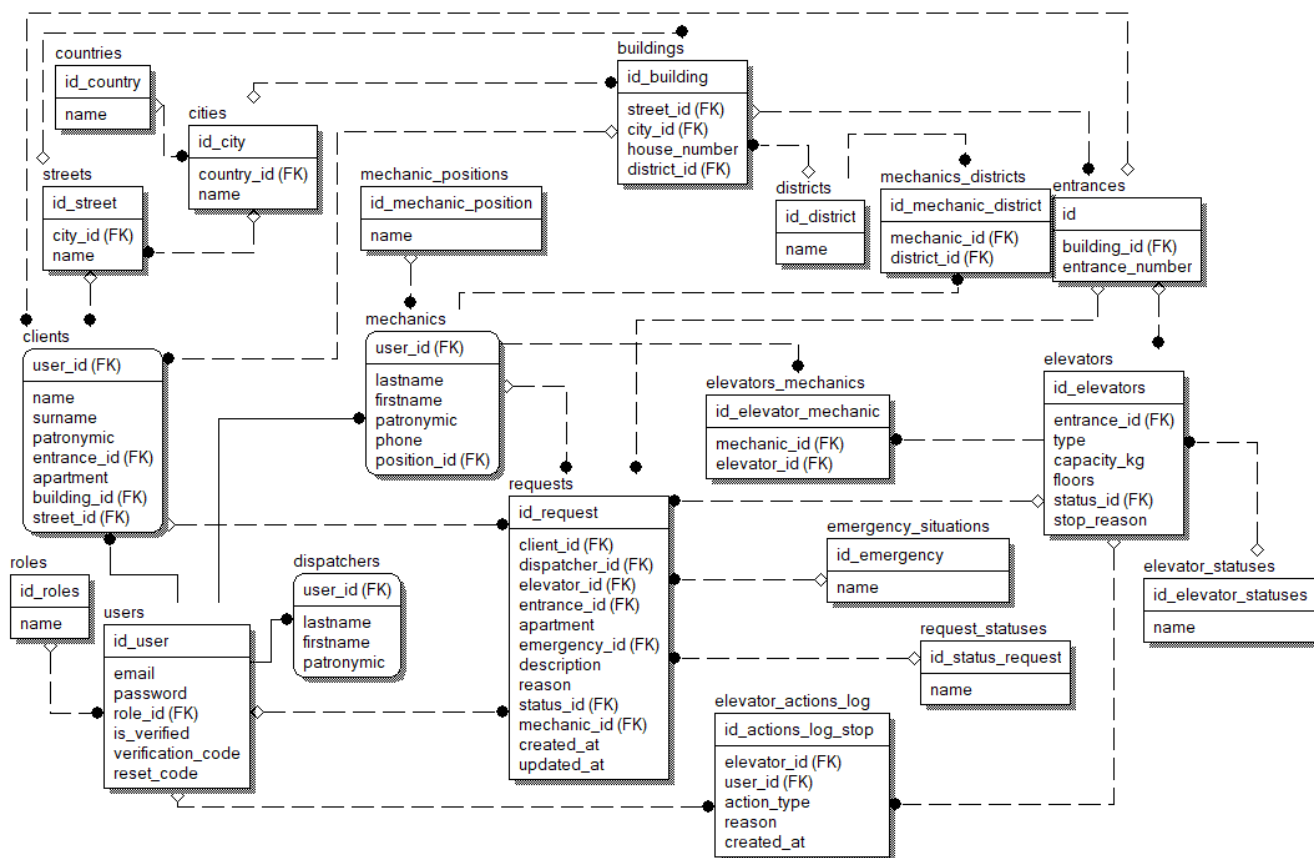


Рисунок 3.1 – Логічна модель бази даних розроблена в Erwin Data Modeler

Після доведення логічної моделі до потрібного вигляду було створено базу даних через виконання SQL-запитів в інтерфейсі phpMyAdmin, до якого був доступ через хостинг UkrLine. В цей процес увійшло створення таблиць, задання типів даних, первинних та вторинних ключів, а також наповнення довідкових таблиць (roles, emergency\_situations, requests\_statuses, elevator\_statuses).

phpMyAdmin - це безкоштовний веб-інструмент, розроблений на мові PHP, який використовується для адміністрування серверами баз даних MySQL. За його допомогою можна виконувати основні адміністративні операції, такі як створення бази, виконання SQL-запитів, керування користувачами та правами доступу [11].

Основні таблиці є: users, clients, mechanics, dispatchers, requests, elevators, entrances, buildings, streets, cities, countries, roles, emergency\_situations,

request\_statuses, elevator\_actions\_log, mechanics\_districts та інші. Всі вони зв'язані через зовнішні ключі, що несе за собою забезпечення даних та логічну зв'язаність інформації.

Кожен користувач системи зберігається в таблицю users, яка включає поля: email, password, role, email, is\_verified. Інформація про користувачів, залежно від їх ролі, зберігається в таблицях clients, dispatchers, mechanics, які зв'язані з users через зовнішній ключ user\_id.

Усі заявки фіксуються в таблиці requests, де знаходиться інформація про клієнта, диспетчера, ліфт, під'їзд, тип аварійної ситуації, статус, призначеного механіка та час обробки.

Реалізована таблиця elevator\_actions\_log, яка дозволяє вести історію зупинок та запусків ліфтів, де фіксується дата, причина та того, хто виконав цю дію.

Є таблиця elevators\_mechanics завдяки якій відбувається закріплення лінійного механіка до конкретного ліфту. В mechanics\_districts є призначення конкретного аварійного механіка до дільниці, яка закріплена за адресою в таблиці buildings завдяки полю district\_id. Таблиця mechanic\_positions має довідник посад механіка: лінійний або аварійний механік.

Адреса реалізована через ієрархічну структуру таблиць countries → cities → streets → buildings → entrances. Це дає змогу формалізовано зберігати адреси об'єктів, ефективно реалізовувати пошук та фільтрацію.

Ліфти з таблиці elevators закріплені за конкретним під'їздом, який, в свою чергу, закріплений за будинком, який розташований на певній вулиці, в конкретному місті та країні.

Таблиця districts є довідником для зберігання назв дільниць, що використовуються для маршрутизації заявок та прив'язки механіків.

Схему бази даних можна побачити на рисунку 3.2.

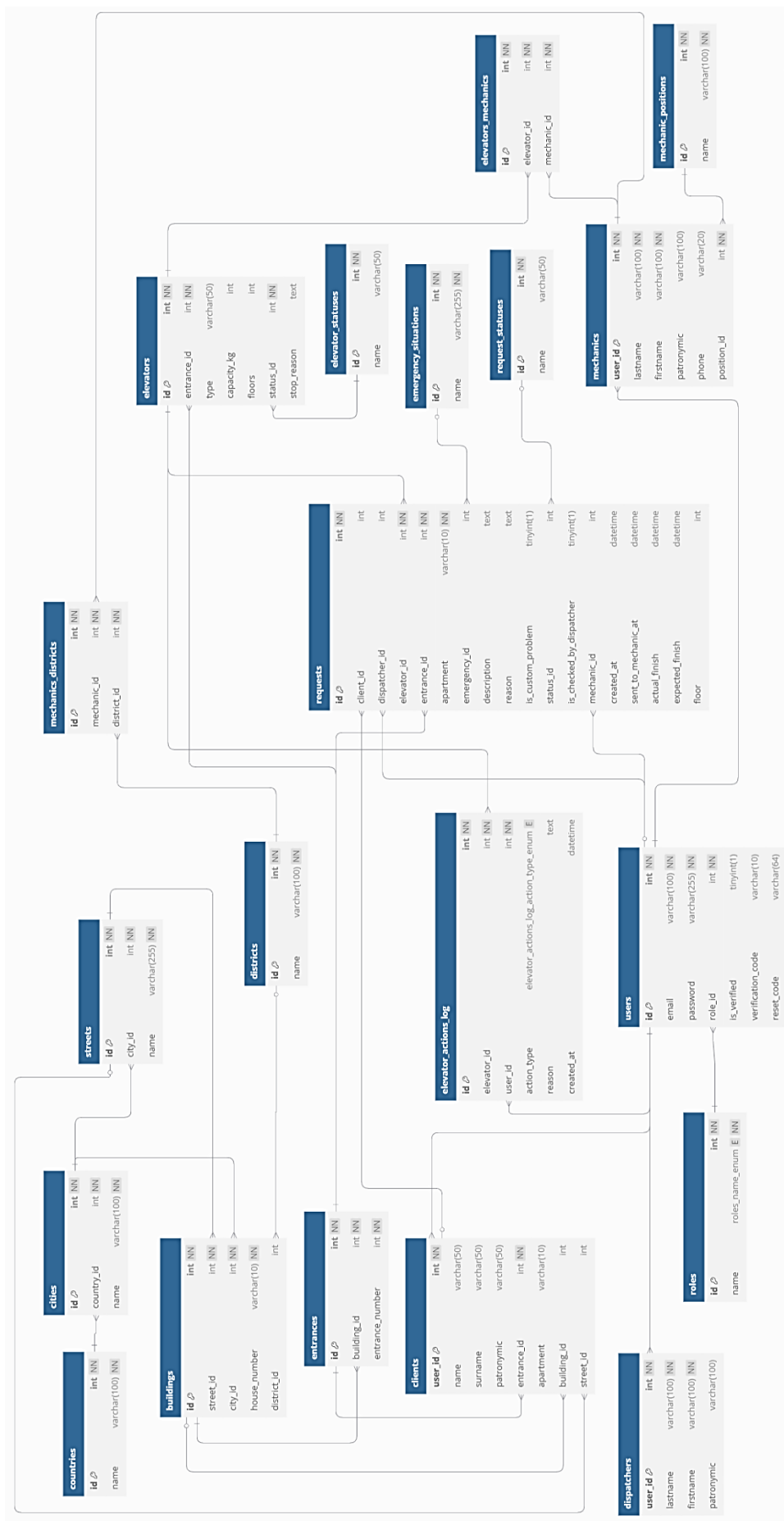


Рисунок 3.2 – Схема базы данных

Завдяки структурованій базі вдалось досягти високої узгодженості та надійності. Структура легко масштабується та може бути розширена в майбутньому.

### 3.3. Реалізація функцій системи

#### 3.3.1. Протокол шифрування

Веб-портал був реалізований з використанням хостингу та домену. Для безпечного підключення був отриманий сертифікат від Let`s Encrypt для протоколу шифрування з'єднання https, що захищає дані, які передаються між користувачами та сервером, що унеможлиблює перехоплення.

#### 3.3.2. Реалізація сторінки авторизації

Поле авторизація, як і інші сторінки користувачів, в плані верстки було реалізовано завдяки створеному стилю сайту `evis.lift.org.ua`, який було занесено в загальну папку `styles.css`. Вигляд авторизації наведено на рисунку 3.3.

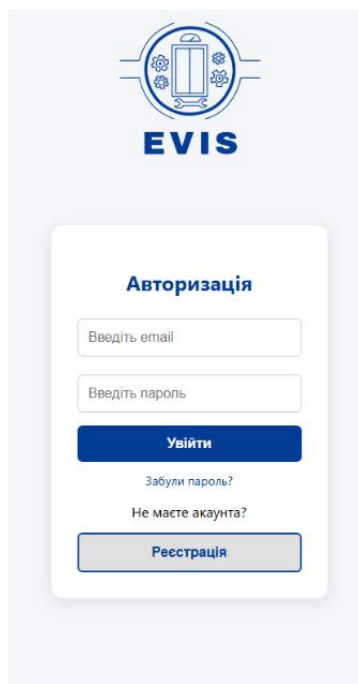


Рисунок 3.3 – Авторизація EVIS

Це, можна сказати, головна сторінка системи. Сторінка реалізована з використанням кнопок та полів для введення. Вміє адаптуватися до різних екранів.

Адаптація в коді:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Кожна кнопка веде до своєї сторінки та реалізовує свій функціонал.

Реалізація переходу через кнопку в коді:

```
<div class="forgot-password">
    <a href="forgot-password.php">Забули пароль?</a>
</div>
    <p class="no-account">Не маєте акаунта?</p>
    <a href="registration.php">
        <button type="button"
class="secondary">Реєстрація</button>
    </a>
</div>
```

Підключення до БД відбувається через файл db.php – загальний файл для підключення по всьому сайту:

```
<?php
$servername = "195.64.155.154";
$username = "*****";
$password = "*****";
$dbname = "lift";

$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Помилка підключення: " . mysqli_connect_error());
}
?>
```

При переході на сторінку реєстрації реалізовані поля для введення (рис. 3.4)

The image shows a registration form titled "Реєстрація" (Registration). It contains the following input fields from top to bottom: "Ім'я" (Name), "Прізвище" (Surname), "По батькові" (Patronymic), "Email", "Пароль" (Password), "Країна" (Country), "Місто" (City), "Вулиця" (Street), "Номер будинку" (Building number), "Під'їзд" (Entrance), and "Квартира" (Apartment). Below the fields is a blue button labeled "Зареєструватися" (Register) and a link labeled "← Повернутись до входу" (← Return to login).

Рисунок 3.4 – Сторінка реєстрації

Йде захист від розголошення адрес обслуговування третім особам, які є в системі. Це одна з перевірок чи користується клієнт послугами ОТІС. Підказка до полів Країна, Місто, Вулиця з'являється тільки після введення 3х символів, а будинку після першого (рис. 3.5).

The image shows the registration form with the following data entered: "Анастасія" in the Name field, "Костіна" in the Surname field, "Володимирівна" in the Patronymic field, "nastia@gmail.com" in the Email field, and "....." in the Password field. The Country field is set to "Укр|" (Ukrainian). Below the Country field, a dropdown menu is open, showing a single option: "Україна" (Ukraine).

Рисунок 3.5 – Демонстрація підказки

Реалізовано це через SQL запит та JavaScript.

Приклад коду `get_countries.php`:

```
ini_set('display_errors', 1);

error_reporting(E_ALL);
require_once "db.php"; // Підключення до БД

header('Content-Type: application/json');

$query = trim($_GET['q'] ?? '');

if (strlen($query) < 3) {
    echo json_encode([]);
    exit;
}

$stmt = mysqli_prepare($conn, "SELECT id, name FROM countries WHERE name LIKE
CONCAT(?, '%') LIMIT 10");
mysqli_stmt_bind_param($stmt, "s", $query);
mysqli_stmt_execute($stmt);
$result = mysqli_stmt_get_result($stmt);

$countries = [];
while ($row = mysqli_fetch_assoc($result)) {
    $countries[] = ['id' => $row['id'], 'name' => $row['name']];
}

echo json_encode($countries);
```

Якщо не ввести всі поля, то буде помилка:

```
<?php if ($error === 'invalid_address'): ?>
    <p class="form-error">Оберіть з підказки наступні поля</p>
<?php endif; ?>
```

Додавання нового користувача реалізовано за допомогою окремого коду `register_client.php`, що перевіряє чи заповнені всі поля та існування користувача з таким email в системі. При успішній перевірці всіх заповнених полей хешується пароль, створюється код верифікації та виконується SQL запити на додавання користувача та клієнта.

```
$password = password_hash($raw_password, PASSWORD_DEFAULT);
```

```

$verification_code = rand(100000, 999999);
$role_id = 4; // клієнт

$insert_user = mysqli_prepare($conn,
    "INSERT INTO users (email, password, role_id, is_verified, verification_code)
    VALUES (?, ?, ?, 0, ?)");
mysqli_stmt_bind_param($insert_user, "ssis", $email, $password, $role_id,
$verification_code);
$password = password_hash($raw_password, PASSWORD_DEFAULT);
$verification_code = rand(100000, 999999);
$role_id = 4; // клієнт

$insert_user = mysqli_prepare($conn,
    "INSERT INTO users (email, password, role_id, is_verified, verification_code)
    VALUES (?, ?, ?, 0, ?)");
mysqli_stmt_bind_param($insert_user, "ssis", $email, $password, $role_id,
$verification_code);

```

Після успішного додавання користувача в БД йому надсилається код верифікації на пошту завдяки технології PHPMailer (рис. 3.6).

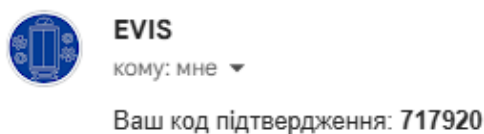


Рисунок 3.6 – Лист з кодом підтвердження акаунту

Код реалізації надсилання email:

```

<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'PHPMailer/src/PHPMailer.php';
require 'PHPMailer/src/SMTP.php';
require 'PHPMailer/src/Exception.php';

function sendVerificationEmail($recipientEmail, $verificationCode) {
    $mail = new PHPMailer(true);

    try {
        // SMTP конфіг

```

```

$mail->isSMTP();
$mail->Host      = 'smtp.gmail.com';
$mail->SMTPAuth  = true;
$mail->Username  = 'eviselevator@gmail.com';
$mail->Password  = '*****';
$mail->SMTPSecure = 'tls';
$mail->Port      = 587;

// Від кого і кому
$mail->setFrom('eviselevator@gmail.com', 'EVIS');
$mail->addAddress($recipientEmail);

// Контент
$mail->isHTML(true);
$mail->CharSet = 'UTF-8';
$mail->Subject = 'EVIS: код підтвердження реєстрації';
$mail->Body    = "Ваш код підтвердження: <b>$verificationCode</b>";

$mail->send();
return true;
} catch (Exception $e) {
    error_log("PHPMailer Error: {$mail->ErrorInfo}");
    return false;
}
}

```

Після цього поле в БД `is_verified` буде 1 та клієнт отримає доступ.

Забули пароль реалізовано схожим методом з надсиланням посилання для відновлення на email. Система передбачає ситуацію якщо користувач забув ввести код підтвердження та надає йому шанс, надіслати новий код та ввести повторно:

```

$_SESSION['resend_email'] = $email;
$error = "Пошта не підтверджена. <a href='resend_code.php'>Надіслати код повторно</a>";

```

Після введення паролю та логіну існуючого перевіреного користувача йде перевірка на роль:

```

$_SESSION['user_id'] = $user_id;
$_SESSION['role_id'] = $role_id;
$_SESSION['name'] = $name;

```

```

switch ($row['role_id']) {
    case 1:
        header("Location: Admin/admin_home.php");
        break;
    case 2:
        header("Location: Mechanic/mechanic_home.php");
        break;
    case 3:
        header("Location: Dispatcher/dispatcher_home.php");
        break;
    case 4:
    default:
        header("Location: Client/client_home.php");
        break;
}
exit;

```

### 3.3.3. Реалізація сторінки клієнта

Успішний вхід переносить нас на сторінку клієнта, де в нього реалізовані функції: Додати нову заявку, Відслідковування заявок, Персональні дані, де є зміна адреси та паролю. Головна сторінка продемонстрована на рисунку 3.7, де реалізовані відображення актуальних заявок та, якщо наявні, ліфти в зупинці.

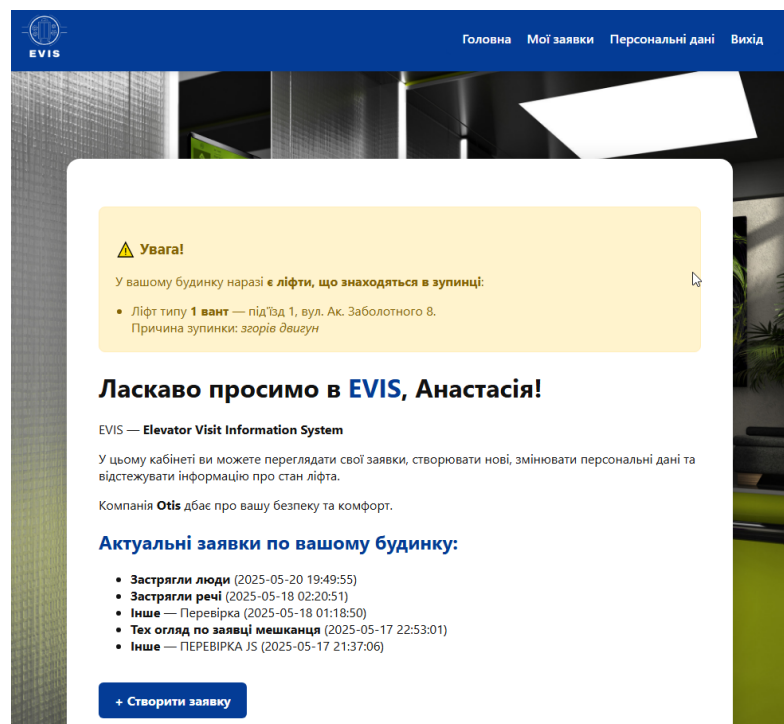


Рисунок 3.7 – Головна сторінка клієнта

Функція створити заявку передбачає обирання існуючих або кастомних проблем (Інше), для надсилання заявки, продемонстрованих на рисунку 3.8.

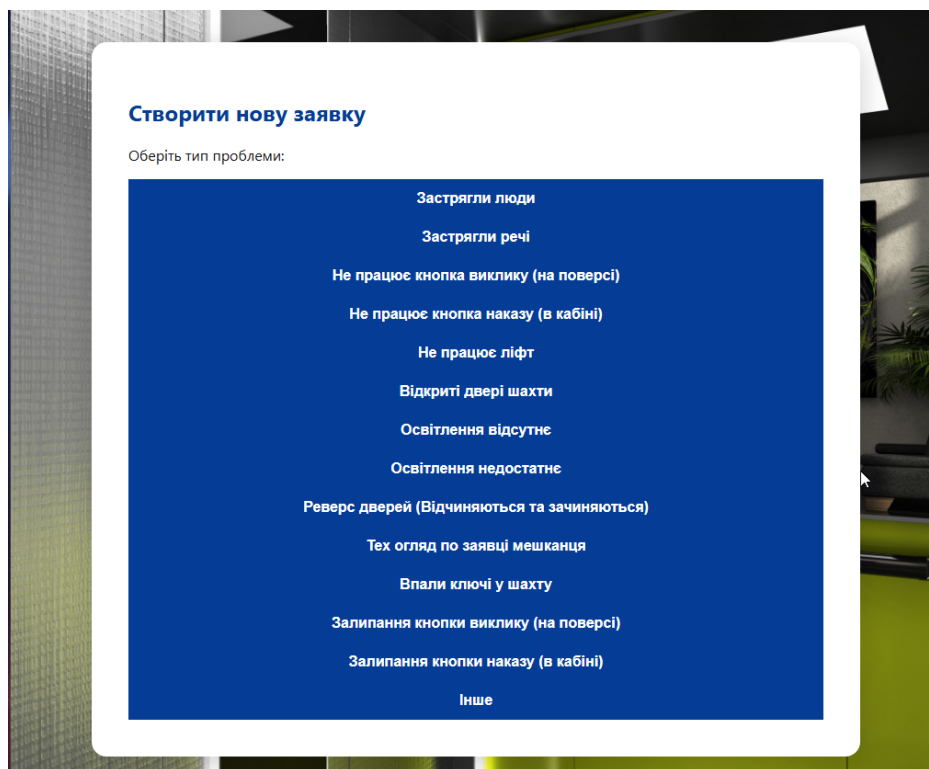


Рисунок 3.8 – Створення нової заявки

Реалізація створення нової заявки відбувається за допомогою посилань та полів. При виборі ліфтів та поверхів відбуваються запити в БД за наявними ліфтами за цією адресою. Після натиснення кнопки створити йде перевірка чи існує ліфт з такою заявкою вже відкритою, або, якщо він в зупинці. Тоді заявка не створюється. При успішному створенні заявки відправляються заповнені дані зі статусом Надіслано механіку, та з повідомленням на пошту для закріпленого механіка. Також при виборі Інше, в базу відправляється додатково опис, але статус стає Очікує перевірку диспетчером.

На сторінці персональні дані реалізована Зміна паролю та Зміна адреси, що наведено на рисунку 3.9. На рисунку 3.10 представлено поле зміни паролю.

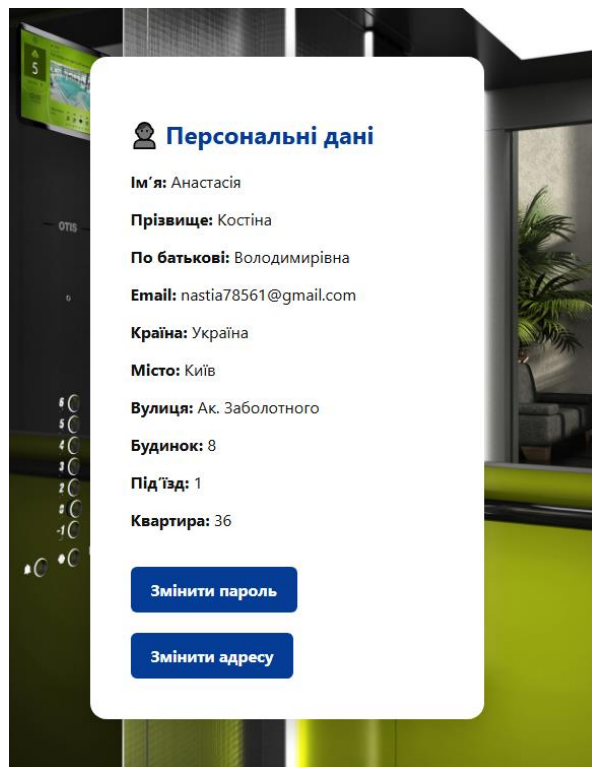


Рисунок 3.9 – Сторінка персональні дані

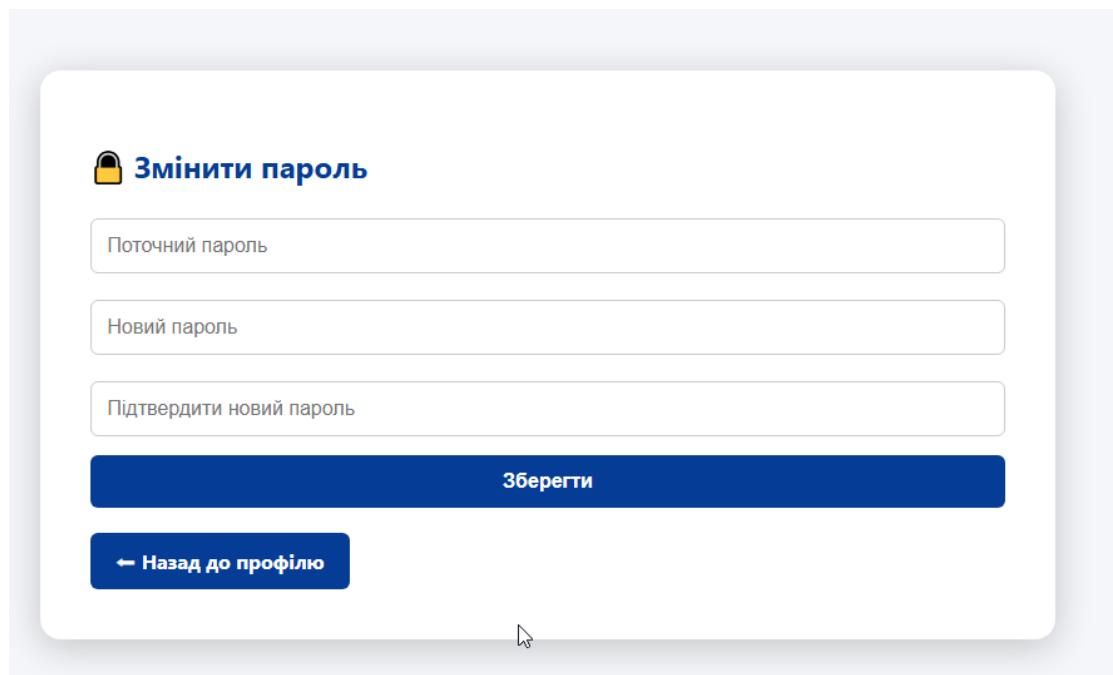


Рисунок 3.10 – Зміна паролю

Код для зміни паролю:

```
<?php
session_start();
require_once "../db.php";
```

```

        if (!isset($_SESSION['user_id']) || !isset($_SESSION['role_id']) ||
$_SESSION['role_id'] != 4) {
            header("Location: ../index.php");
            exit;
        }

$user_id = $_SESSION['user_id'];
$message = '';

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $current = trim($_POST['current_password']);
    $new = trim($_POST['new_password']);
    $confirm = trim($_POST['confirm_password']);

    if ($new !== $confirm) {
        $message = "Паролі не співпадають.";
    } else {
        $stmt = $conn->prepare("SELECT password FROM users WHERE id = ?");
        $stmt->bind_param("i", $user_id);
        $stmt->execute();
        $res = $stmt->get_result();
        $user = $res->fetch_assoc();

        if (!password_verify($current, $user['password'])) {
            $message = "Неправильний поточний пароль.";
        } else {
            $hashed = password_hash($new, PASSWORD_DEFAULT);
            $update = $conn->prepare("UPDATE users SET password = ? WHERE id = ?");
            $update->bind_param("si", $hashed, $user_id);
            if ($update->execute()) {
                header("Location: profile.php?password_updated=1");
                exit;
            } else {
                $message = "Помилка при зміні пароля.";
            }
        }
    }
}
}
?>

```

На рисунку 3.11 представлено сторінку зміни адреси.

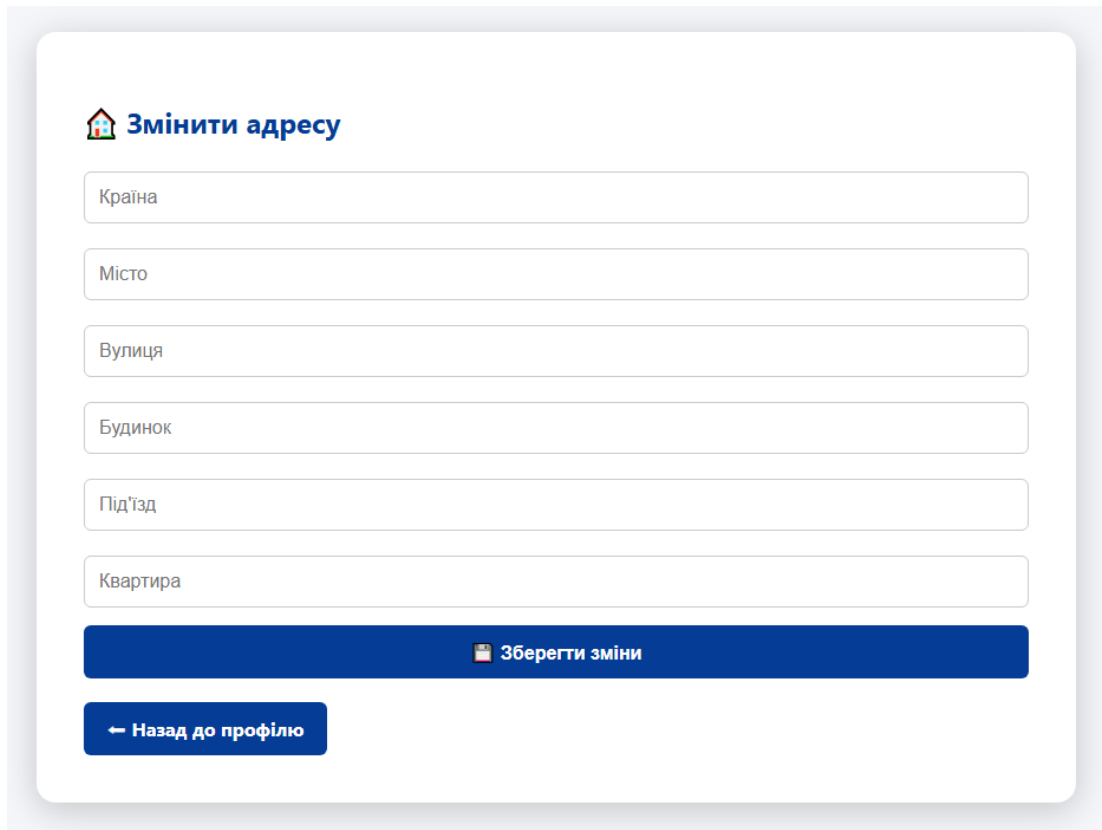


Рисунок 3.11 – Зміна адреси

При зміні адреси минулі заявки залишаються.

Код зміни адреси:

```
<?php
session_start();
require_once "../db.php";

if (!isset($_SESSION['user_id']) || !isset($_SESSION['role_id']) ||
$_SESSION['role_id'] != 4) {
    header("Location: ../index.php");
    exit;
}

$user_id = $_SESSION['user_id'];

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $country_id = $_POST['country_id'] ?? null;
    $city_id = $_POST['city_id'] ?? null;
    $street_id = $_POST['street_id'] ?? null;
```

```

$building_id = $_POST['building_id'] ?? null;
$entrance_id = $_POST['entrance_id'] ?? null;
$apartment = trim($_POST['apartment'] ?? '');

if (!$country_id || !$city_id || !$street_id || !$building_id || !$entrance_id
|| empty($apartment)) {
    header("Location: edit_address.php?error=invalid_address");
    exit;
}

// Типізація
$building_id = (int)$building_id;
$entrance_id = (int)$entrance_id;
$street_id = (int)$street_id;
$user_id = (int)$user_id;

// Оновлюємо адресу в таблиці clients
$stmt = $conn->prepare("UPDATE clients SET building_id = ?, entrance_id = ?,
street_id = ?, apartment = ? WHERE user_id = ?");
if (!$stmt) {
    header("Location: edit_address.php?error=db_prepare");
    exit;
}

$stmt->bind_param("iisii", $building_id, $entrance_id, $street_id, $apartment,
$user_id);

if ($stmt->execute()) {
    $_SESSION['address_updated'] = true;
    header("Location: profile.php?address_updated=1");
    exit;
} else {
    header("Location: edit_address.php?error=update_failed");
    exit;
}
} else {
    header("Location: edit_address.php");
    exit;
}
}

```

### 3.3.4. Реалізація сторінки диспетчера

На сторінці диспетчера реалізовано функції закриття заявки, редагування механіка, створенні нової, перевірка заявок Інше, запуск зупинених ліфтів (рис.3.12). При виборі Заявки в таблиці будуть показані всі заявки (рис. 3.13). Поле змінення та закриття заявки реалізовані завдяки модальним сторінкам через JavaScript (рис. 3.14).

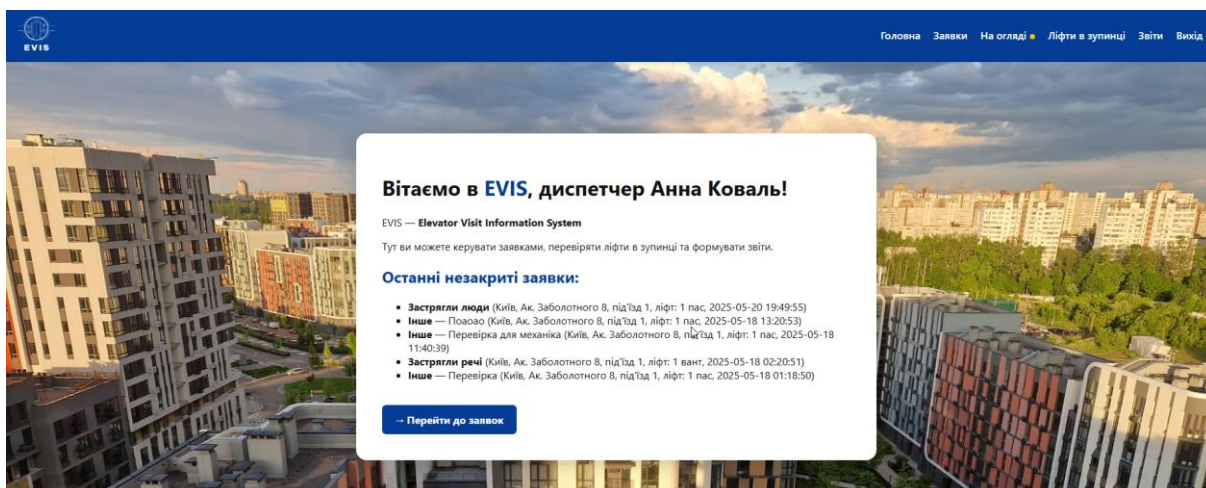


Рисунок 3.12 – Головна диспетчер

**Заявки**

Всі   Закриті за сьогодні   Незакриті за сьогодні   Незакриті   [+ Нова заявка](#)

Фільтр за механіком: — Усі —

Дата	Проблема	Адреса	Квартира	Поверх	Ліфт	Механік	Статус	Дії
2025-05-20 19:49:55	Застрагли люди	Київ, Ак. Заболотного 8, під'їзд 1	2	—	1 пас	Андреев Д.В.	Передана механіку	
2025-05-18 13:20:53	Повазо	Київ, Ак. Заболотного 8, під'їзд 1	36	—	1 пас	Андреев Д.В.	Очікує перевірки диспетчером	
2025-05-18 11:40:39	Перевірка для механіка	Київ, Ак. Заболотного 8, під'їзд 1	36	—	1 пас	Андреев Д.В.	Очікує перевірки диспетчером	
2025-05-18 02:20:51	Застрагли речі	Київ, Ак. Заболотного 8, під'їзд 1	36	—	1 вант	Андреев Д.В.	Передана механіку	
2025-05-18 01:18:50	Перевірка	Київ, Ак. Заболотного 8, під'їзд 1	36	—	1 пас	Андреев Д.В.	Очікує перевірки диспетчером	
2025-05-17 22:53:01	Тех огляд по заявці мешканця	Київ, Ак. Заболотного 8, під'їзд 1	36	—	1 вант	Андреев Д.В.	Передана механіку	
2025-05-17 21:37:06	ПЕРЕВІРКА JS	Київ, Ак. Заболотного 8, під'їзд 1	36	—	1 пас	Андреев Д.В.	Очікує перевірки диспетчером	

Рисунок 3.13 – Вигляд таблиці заявок

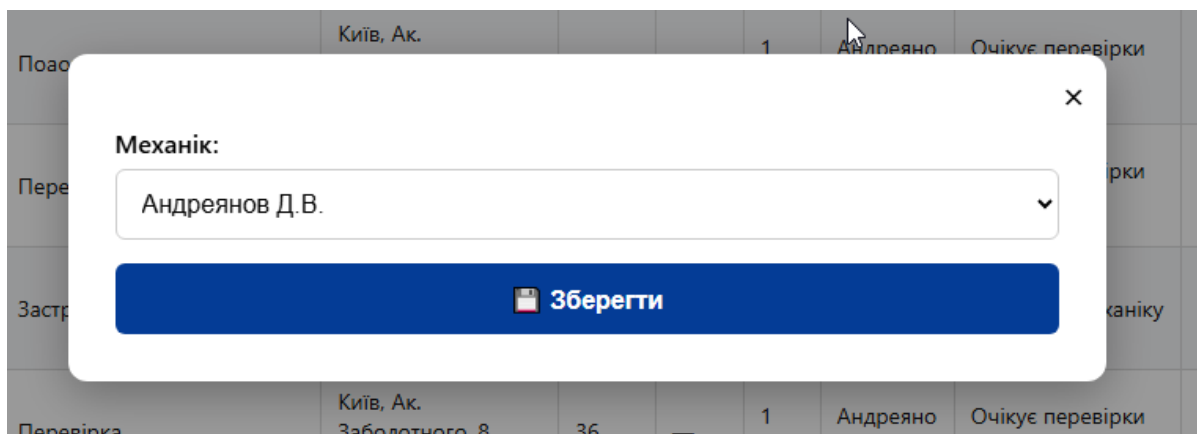


Рисунок 3.14 – Зміна механіка

При виборі іншого механіка дані перезаписуються в базу і заявка з'являється в інтерфейсі іншого механіка. Модальне вікно закриття реалізовано за допомогою модального вікна та JS для перевірки виборів в ComboBox (рис. 3.15).

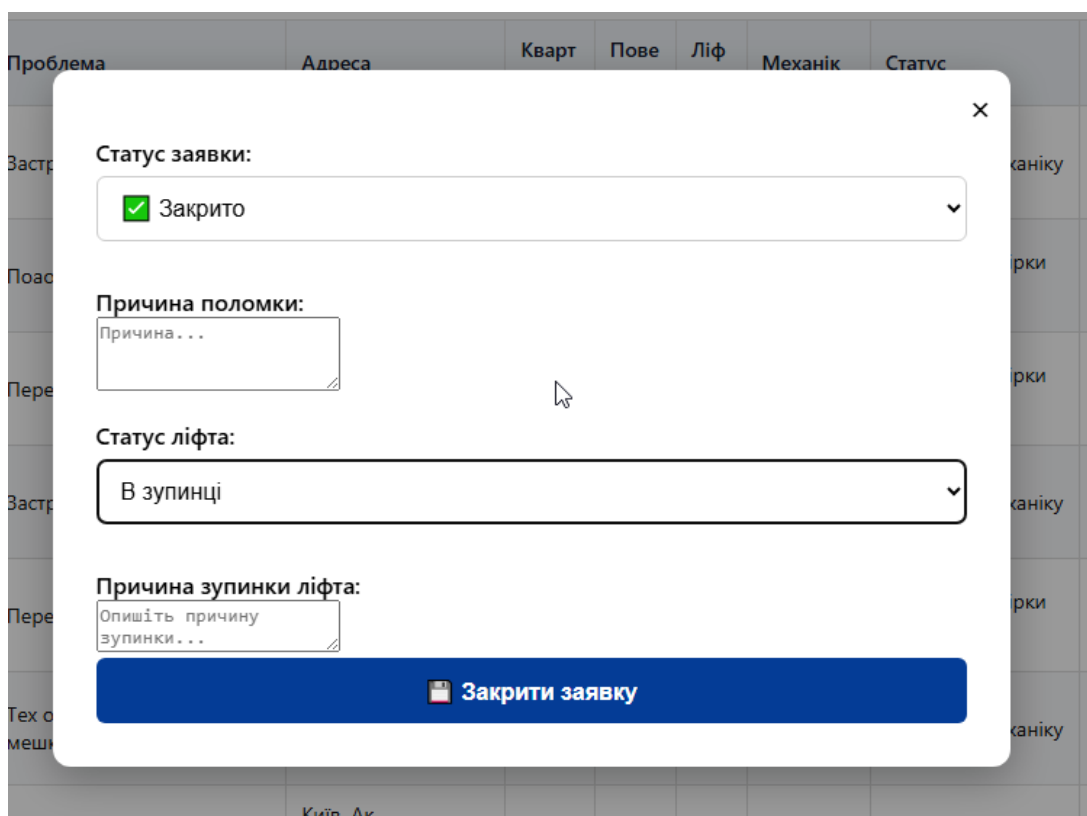


Рисунок 3.15 – Вигляд модального вікна закриття при виборі таких наборів даних

Після заповнення всіх полів дані будуть перезаписані в базі. Додасться час закриття заявки та оновиться статус. Якщо ліфт було зупинено, то причина зупинки буде внесена в таблицю ліфтів та `elevator_action_log`.

Створити заявку вручну несе за собою внесення всієї інформації за допомогою select2 та JavaScript, який приховує поля (рис. 1.16). У випадках, які передбачають уточнення інформації про поверх або Інше вилізають додаткові поля. Вибір механіків реалізований завдяки SQL запиту:

```
$stmt = $conn->prepare("
    SELECT m.user_id, m.firstname, m.lastname, m.patronymic
    FROM mechanics m
    WHERE m.user_id IN (
        SELECT em.mechanic_id FROM elevators_mechanics em
        JOIN elevators e ON em.elevator_id = e.id
        WHERE e.entrance_id = ?
        UNION
        SELECT md.mechanic_id FROM mechanics_districts md
        WHERE md.district_id = ?
    )
");
```

**Створити заявку вручну**

Оберіть місто + вулицю:  
Київ, Ак. Заболотного

Будинок:  
8, під'їзд 1

Ліфт:  
1 вант

Тип аварії:  
Застрягли речі

Квартира:  
67

Механік:  
Сидорчук О.І.

**Створити заявку**

Рисунок 3.16 – Заповнене вікно створення заявки

Якщо клієнт обрав інше, то диспетчер може або закрити заявку як помилкову, або підтвердити (рис. 3.17). При виборі Підтвердження статус зміниться на Надіслано механіку та відправить повідомлення.

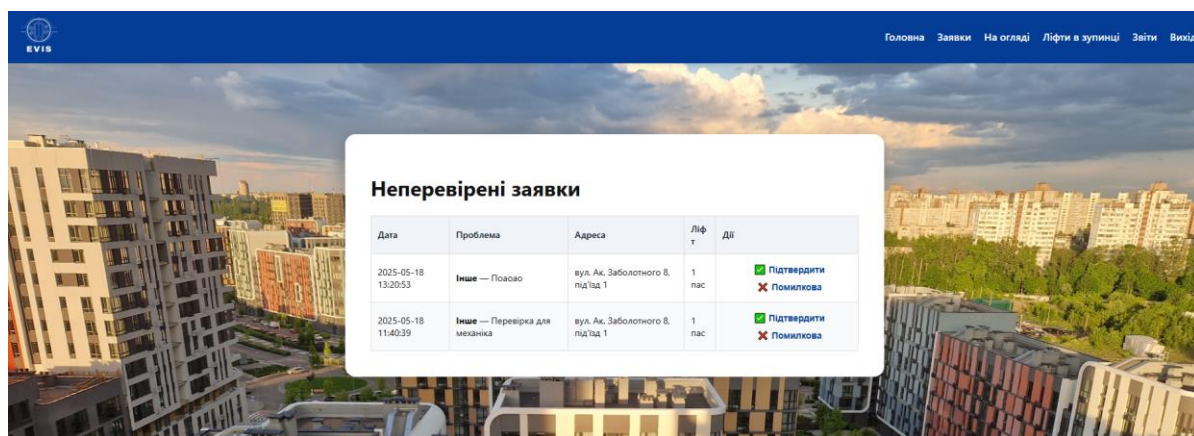


Рисунок 3.17 – Вигляд неперевіраних заявок

Код перевірки який статус був обраний:

```

if ($action === 'approve') {
    // Підтверджено – статус 3 (Надіслано механіку)
    $stmt = $conn->prepare("
        UPDATE requests
        SET is_checked_by_dispatcher = 1, status_id = 3, sent_to_mechanic_at = ?
        WHERE id = ?
    ");
    $stmt->bind_param("si", $now, $request_id);
} else {
    // Помилкова – статус 5 (закрита)
    $stmt = $conn->prepare("
        UPDATE requests
        SET is_checked_by_dispatcher = 1, status_id = 5, actual_finish = ?
        WHERE id = ?
    ");
    $stmt->bind_param("si", $now, $request_id);
}

```

Також на всіх сторінках реалізований індикатор нової заявки на перевірку, який розташований біля пункту меню На огляді та інформування про наявність нової заявки при знаходженні клієнта на сторінці Заявка або На огляді. Кожні 20 секунд скрипт перевіряє наявність нових заявок.

```
function checkNewRequests() {
```

```
$.getJSON('check_new_requests.php', { last: lastCheckTime }, function (data) {
    if (data.count > 0) {
        $('#new-count').text(data.count);
        $('#new-requests-alert').fadeIn();
    }
});
}
```

```
// Кожні 20 секунд перевірка нових
setInterval(checkNewRequests, 20000);
```

Системою передбачено зняття ліфта з зупинки, що робить статус ліфта Працює та вводить лог коли був запущений ліфт (рис.3. 18).

```
$update_stmt = $conn->prepare("UPDATE elevators SET status_id = 1, stop_reason =
NULL WHERE id = ?");
```

```
$update_stmt->bind_param("i", $elevator_id);
$update_stmt->execute();
```

```
// 2. Записати у лог
```

```
$log_stmt = $conn->prepare("
    INSERT INTO elevator_actions_log (elevator_id, user_id, action_type, reason,
created_at)
    VALUES (?, ?, 'запуск', NULL, ?)
");
$log_stmt->bind_param("iis", $elevator_id, $dispatcher_id, $now);
$log_stmt->execute();
```

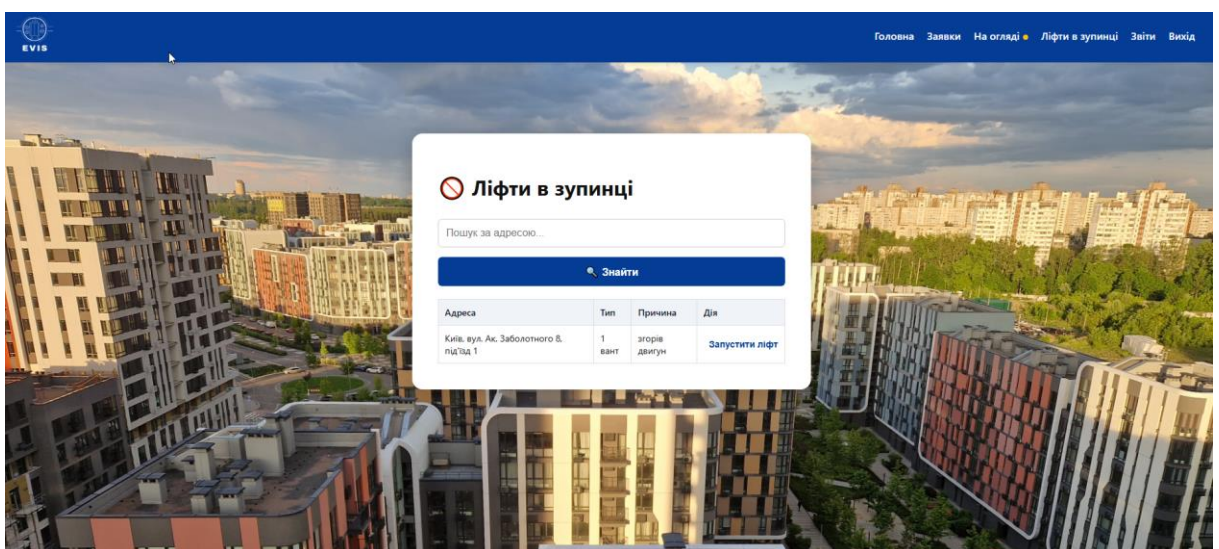


Рисунок 3.18 – Сторінка запуску ліфтів

Диспетчер може переглядати звіти та формувати їх в PDF файл завдяки mPDF (рис. 3.19).

```
echo "</tbody></table>";
$html = ob_get_clean();

$mpdf = new Mpdf(['default_font' => 'dejavusans']);
$mpdf->WriteHTML($html);
$mpdf->Output("EVIS-report-{$report_type}.pdf",
\Mpdf\Output\Destination::INLINE);
```

### Кількість заявок по кожному ліфту

Адреса	Тип	Заявок	К-сть зупинок	Помилкових
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 пас	23	2	10
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 вант	6	1	0
Київ, вул. Хрещатик 1, під'їзд 1	1 пас	1	1	0
Київ, вул. Хрещатик 1, під'їзд 2	1 пас	1	0	0
Київ, вул. Хрещатик 1, під'їзд 1	1 вант	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	2 пас	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	1 пас	0	0	0

Рисунок 3.19 – Вигляд таблиці звіту у вигляді PDF

### 3.3.5. Реалізація сторінки механіка

Сторінка механіка схожа за своєю структурою до диспетчера, але цей користувач відрізняється тим, що він отримує повідомлення на пошту про нові заявки (рис. 3.20).

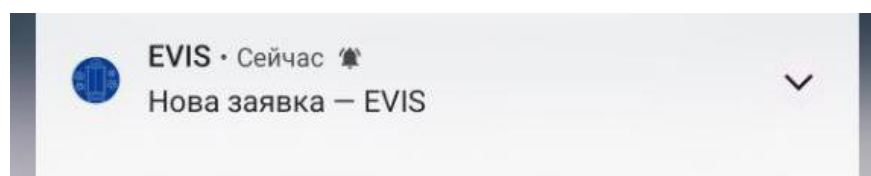


Рисунок 3.20 – Вигляд сповіщення про нову заявку для механіка

В іншому все було ідентичне до логіки диспетчера, окрім двох моментів: механік не може редагувати поля та створювати нові заявки. Механіку виводяться тільки його заявки та зі статусами передані механіку (рис. 3.21).

```
$req_stmt = $conn->prepare("SELECT r.description, r.created_at, es.name AS emergency,
    s.name AS street, b.house_number, en.entrance_number, ci.name AS city,
    e.type AS elevator_type
FROM requests r
JOIN emergency_situations es ON r.emergency_id = es.id
JOIN entrances en ON r.entrance_id = en.id
JOIN buildings b ON en.building_id = b.id
JOIN streets s ON b.street_id = s.id
JOIN cities ci ON s.city_id = ci.id
JOIN elevators e ON r.elevator_id = e.id
WHERE r.status_id NOT IN (2, 4, 5) AND r.mechanic_id = ?
ORDER BY r.created_at DESC
LIMIT 5");
```

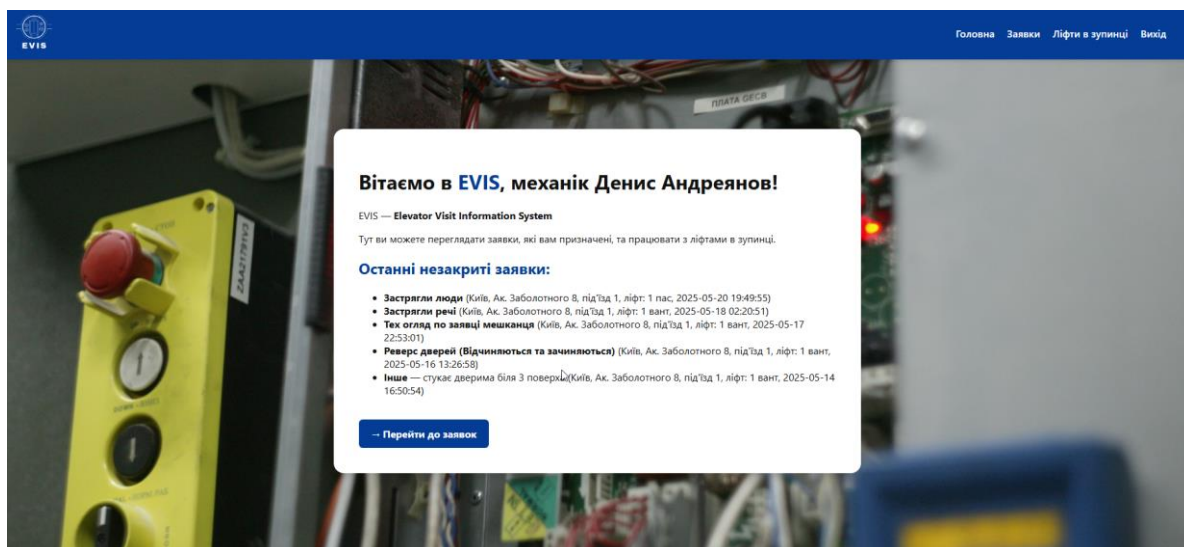


Рисунок 3.21 – Головна механіка

### 3.3.6. Реалізація сторінки адміна

Перед створенням сторінки адміна був створений сам адмін, через Shell-клієнт серверу, за допомогою Laravel.

```
use Illuminate\Support\Facades\Hash;
use App\Models\User;
```

```
User::create([
    'email' => admin@evis.lift.org.ua',
```

```
'password' => Hash::make('*****'),  
'role_id' => 1,  
'is_verified' => true,  
]);
```

В моїй системі адмін може створювати нових користувачів, які є робітниками компанії та переглядати звіти. На рисунку 3.22 наведена головна адміна. Поля введення нового користувача наведені на рисунку 3.23, а звіти на рисунку 3.24.

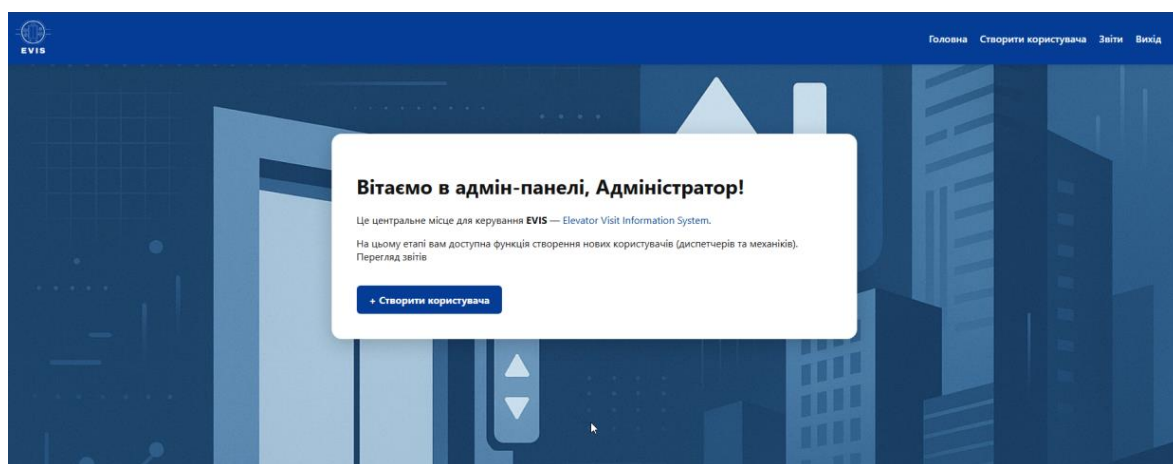


Рисунок 3.22 – Головна адміна

Рисунок 3.23 – Створення нового користувача

Адреса	Тип	Заявок	К-сть зупинок	Помилки
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 пас	23	2	10
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 вант	6	1	0
Київ, вул. Хрещатик 1, під'їзд 2	1 пас	1	0	0
Київ, вул. Хрещатик 1, під'їзд 1	1 пас	1	1	0
Київ, вул. Хрещатик 1, під'їзд 1	1 вант	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	1 пас	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	2 пас	0	0	0

Рисунок 3.24 – Сторінка звітів

## 3.4. Інструкція користувача

### 3.4.1. Інструкція для клієнта

#### 3.4.1.1. Загальні положення.

В даному підрозділі викладена інструкція стосовно роботи клієнта з веб-порталом аварійної служби ліфтової компанії ПрАТ «ОТІС» - <https://evis.lift.org.ua>. Система створена для подавання заявок на обслуговування ліфтів, перегляд їх статусу та керування персональними даними.

#### 3.4.1.2. Вимоги до середовища.

Для коректної роботи порталу рекомендовано використовувати пристрої з сучасними браузерами, такими як: Google Chrome, Mozilla Firefox, Microsoft Edge. Необхідна наявність інтернет підключення.

#### 3.4.1.3. Реєстрація користувача.

##### 3.4.1.3.1. Перейдіть за посиланням <https://evis.lift.org.ua>.

##### 3.4.1.3.2. У разі відсутності акаунту натисніть кнопку «Реєстрація».

3.4.1.3.3. Заповніть обов'язкові поля, такі як Ім'я, Прізвище, По-батькові, Адресу електронної пошти, Пароль.

##### 3.4.1.3.4. Для введення адреси:

- У полях, такі як Країна, Місто та Вулиця введіть 3 символи, після чого система запропонує відповідні значення;
- Оберіть відповідне значення з запропонованих варіантів;
- У полі Будинок та Під'їзд введіть цифру та оберіть її з підказки;
- Введіть номер квартири.

3.4.1.3.5. Натисніть Зареєструватись.

3.4.1.3.6. Перевірте електронну пошту та введіть у відповідне поле код підтвердження. У разі втрати поля коду – авторизуйтеся на сторінці авторизація та скористайтесь функцією «Повторно надіслати код».

3.4.1.4. Авторизація.

3.4.1.4.1. Введіть в відповідних полях електронну пошту та пароль.

3.4.1.4.2. В разі втрати паролю натисніть Забули пароль?. Введіть електронну адресу та перейдіть за посиланням в листі, для встановлення нового паролю.

3.4.1.5. Робота з головною сторінкою.

Після авторизації клієнту доступна інформація по активним заявкам та непрацюючим ліфтам, якщо вони наявні. Доступні такі пункти меню як Головна, Мої заявки, Персональні дані, Вихід.

3.4.1.6. Створення нової заявки.

3.4.1.6.1. На головній сторінці оберіть Створити заявку

3.4.1.6.2. Вкажіть:

- Тип проблеми (з наданого переліку, або Інше);
- Поверх (якщо необхідно);
- Опис проблеми, якщо обрано Інше.

3.4.1.6.3. Якщо відкрита заявка з таким типом проблеми система повідомить про дублювання.

3.4.1.6.4. Після заповнення інформації про заявку з'явиться вікно для підтвердження зазначеного ліфта та типу проблеми. Перевірте дані та підтвердіть створення.

Примітка: ліфт, який перебуває в статусі «В зупинці», не приймає заявку «Не працює ліфт».

### 3.4.1.7. Перегляд створених заявок.

#### 3.4.1.7.1. У меню оберіть «Мої заявки».

3.4.1.7.2. Доступна історія про створенні заявки: статус, дата створення, тип проблеми, опис (за наявності) (рис.3.25).

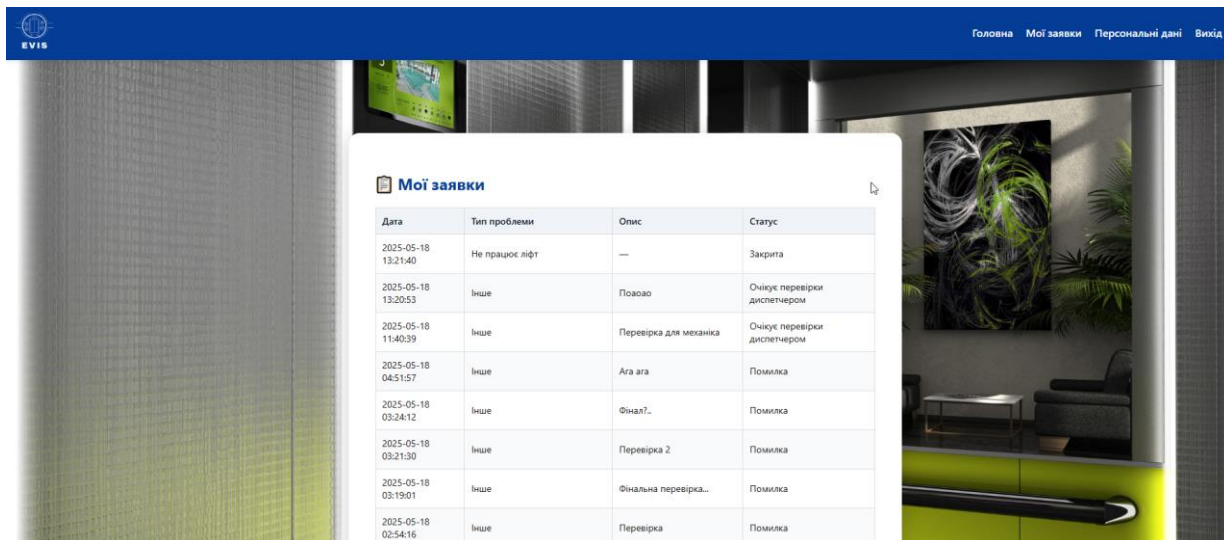


Рисунок 3.25 – Сторінка заявок

### 3.4.1.8. Персональні дані.

3.4.1.8.1. Перейдіть у пункт меню «Персональні дані» для перегляду чи редагування інформації про себе.

#### 3.4.1.8.2. Доступні дії:

- Змінити пароль – введіть поточний пароль, новий та підтвердіть;
- Змінити адресу – оберіть нову адресу, аналогічно до реєстрації. Архів заявок залишиться прив'язаним до попередньої адреси.

### 3.4.1.9. Завершення сеансу.

3.4.1.9.1. Для виходу з особистого кабінету натисніть на Вихід в меню.

## 3.4.2. Інструкція для диспетчера

### 3.4.2.1. Загальні положення.

В цьому підпункті надані інструкції для працівників, які виконують функції диспетчеризації в веб-порталі аварійної служби «ОТІС». Диспетчер відповідає за обробку заявок, перевірку заявок, наданих клієнтом з типом проблеми Інше, призначення механіків, зупинка та запуск ліфтів, перегляд звітів.

#### 3.4.2.2. Вимоги до середовища.

Для коректної роботи порталу рекомендовано використовувати пристрої з сучасними браузером, такими як: Google Chrome, Mozilla Firefox, Microsoft Edge. Необхідна наявність інтернет підключення.

#### 3.4.2.3. Авторизація.

3.4.2.3.1. Перейдіть за посиланням <https://evis.lift.org.ua>.

3.4.2.3.2. Увійдіть до системи, використовуючи корпоративну пошту та пароль, який був наданий адміністратором.

3.4.2.3.3. У разі втрати паролю зверніться до адміністратора. Не використовуйте функцію Забули пароль?.

#### 3.4.2.4. Робота з головною сторінкою.

Після авторизації диспетчеру доступна інформація по останнім активним заявкам. Доступні такі пункти меню як Головна, Заявки, На огляді, Ліфти в зупинці, Звіти, Вихід.

#### 3.4.2.5. Робота з заявками.

##### 3.4.2.5.1. Перегляд поточних заявок.

3.4.2.5.1.1. Для повного доступу до всіх заявок перейдіть до пункту меню Заявки.

##### 3.4.2.5.1.2. Доступні такі фільтри:

- Всі – незакриті заявки за сьогодні та інші дні, закриті заявки за сьогодні;
- Незакриті – незакриті заявки за сьогодні та інші дні;
- Незакриті за сьогодні – заявки, що були створені сьогодні та ще не закриті;
- Закриті за сьогодні – заявки, завершені за сьогодні.

3.4.2.5.1.3. Для фільтрації за механіком використовуйте випадаючий список з прізвищами механіками.

##### 3.4.2.5.2. Створення нової заявки.

3.4.2.5.2.1. Натисніть кнопку «Нова заявка».

3.4.2.5.2.2. Заповніть наступні поля:

- Вулиця, будинок, під'їзд – з автопідказкою після введення символів;
- Ліфт – оберіть зі списку;

- Тип аварії – виберіть з переліку;
- Поверх – внесіть, якщо вимагається типом аварії;
- Опис проблеми – внесіть інформацію, якщо обраний тип «Інше»;
- Квартира заявника;
- Механік – оберіть лінійного або аварійного в списку.

3.4.2.5.2.3. Натисніть «Створіть заявку».

3.4.2.5.2.4. Перевірте заявку в списку.

3.4.2.5.3. Редагування заявок.

3.4.2.5.3.1. Для зміни механіка натисніть піктограму олівця, оберіть іншого механіка та натисніть Зберегти.

3.4.2.5.4. Закриття заявки.

3.4.2.5.4.1. Натисніть піктограму галочки.

3.4.2.5.4.2. Вкажіть тип завершення:

- Закрита – вкажіть причину поломки та метод усунення, поточний статус ліфта (працює або в зупинці). Якщо обрано статус в зупинці – введіть причину.
- Помилка – не вимагає додаткового уточнення.

3.4.2.5.4.3. Після зупинки ліфта закриття його заявок блокується до запуску.

3.4.2.5.4.4. Натисніть Зберегти.

3.4.2.6. Робота з заявками типу «Інше».

3.4.2.6.1. Перейдіть до пункту меню «На огляді».

3.4.2.6.2. Якщо біля цього пункту відображається жовтий індикатор – є неперевірені заявки.

3.4.2.6.3. Перегляньте опис проблеми та перевірте його правильність.

3.4.2.6.4. Доступні дії:

- Підтвердити – заявка буде додана до загального списку та надіслана механіку;
- Закрити як помилку – заявка буде закрита, як помилка, без подальшої обробки.

3.4.2.7. Ліфти в зупинці.

3.4.2.7.1. Перейдіть до пункту «Ліфті в зупинці»;

3.4.2.7.2. Знайдіть потрібну адресу та ліфт;

3.4.2.7.3. Натисніть «Запустити ліфт»;

3.4.2.7.4. Підтвердіть запуск ліфт в діалоговому вікні. Усі дії фіксуються в базі даних.

3.4.2.8. Повідомлення про нові заявки.

3.4.2.8.1. Слідкуйте за новими заявками у верхній частині таблиці заявок;

3.4.2.8.2. У разі відображення повідомлення про надходження нової заявки оновіть сторінку.

3.4.2.8.3. Працює в пунктах Заявки та На огляді.

3.4.2.9. Формування звітів.

3.4.2.9.1. У розділі Звіти доступна така аналітика:

- Загальна кількість заявок по ліфтам;
- Заявки по механікам;
- ТОП-10 найпроблемніших ліфтів;
- Щоденні звіти.

3.4.2.9.2. Для збереження у форматі PDF натисніть Завантажити PDF. Доступне збереження звітів та друкування.

3.4.2.10. Завершення сеансу.

3.4.2.10.1. Для виходу з особистого кабінету натисніть на Вихід в меню.

### **3.4.3. Інструкція для механіка**

3.4.3.1. Загальні положення.

В цьому підпункті надані інструкції для механіка, які працюють з веб-порталом аварійної служби ПрАТ «ОТІС». Система надає доступ до заявок, які пов'язані з ліфтом, який закріплений за механіком, дозволяє закривати виконані заявки, а також керувати статусом ліфта.

3.4.3.2. Вимоги до середовища.

Для коректної роботи порталу рекомендовано використовувати пристрої з сучасними браузером, такими як: Google Chrome, Mozilla Firefox, Microsoft Edge. Необхідна наявність інтернет підключення.

#### 3.4.3.3. Авторизація.

3.4.3.3.1. Перейдіть за посиланням <https://evis.lift.org.ua>.

3.4.3.3.2. Увійдіть до системи, використовуючи корпоративну пошту та пароль, який був наданий адміністратором.

3.4.3.3.3. У разі втрати паролю зверніться до адміністратора. Не використовуйте функцію Забули пароль?.

#### 3.4.3.4. Робота з заявками.

3.4.3.4.1. Перегляд поточних заявок.

3.4.3.4.1.1. Інформація стосовно нової заявки надходить на пошту.

3.4.3.4.1.2. На головній сторінці після входу відображаються останні незакриті заявки.

3.4.3.4.1.3. Для перегляду повного списку заявок перейдіть до пункту Заявки.

3.4.3.4.1.4. Доступні такі фільтри:

- Всі – незакриті заявки за сьогодні та інші дні, закриті заявки за сьогодні;
- Незакриті – незакриті заявки за сьогодні та інші дні;
- Незакриті за сьогодні – заявки, що були створені сьогодні та ще не закриті;
- Закриті за сьогодні – заявки, завершені за сьогодні.

3.4.3.4.2. Закриття заявки.

3.4.3.4.2.1. Натисніть піктограму галочки.

3.4.3.4.2.2. Вкажіть тип завершення:

- Закрита – вкажіть причину поломки та метод усунення, поточний статус ліфта (працює або в зупинці). Якщо обрано статус в зупинці – введіть причину.
- Помилка – не вимагає додаткового уточнення.

3.4.3.4.2.3. Після зупинки ліфта закриття його заявок блокується до запуску.

3.4.3.4.2.4. Натисніть Зберегти.

3.4.3.5. Ліфти в зупинці.

3.4.3.5.1. Перейдіть до пункту «Ліфті в зупинці»;

3.4.3.5.2. Знайдіть потрібну адресу та ліфт;

3.4.3.5.3. Натисніть «Запустити ліфт»;

3.4.3.5.4. Підтвердіть запуск ліфт в діалоговому вікні. Усі дії фіксуються в базі даних.

3.4.3.6. Повідомлення про нові заявки.

3.4.3.6.1. Слідкуйте за новими заявками у верхній частині таблиці заявок;

3.4.3.6.2. У разі відображення повідомлення про надходження нової заявки оновіть сторінку.

3.4.3.6.3. Працює в пункті Заявки.

3.4.3.7. Завершення сеансу.

3.4.3.7.1. Для виходу з особистого кабінету натисніть на Вихід в меню.

### **3.4.4. Інструкція для адміна**

3.4.4.1. Загальні положення.

Адміністратор даної системи має повноваження в додаванні нових користувачів компанії «ОТІС» з ролями Диспетчер та Механік, формувати звіти та базове технічне обслуговування веб-порталу. Доступ надається авторизованим працівникам.

3.4.4.2. Вимоги до середовища.

Для коректної роботи порталу рекомендовано використовувати пристрої з сучасними браузером, такими як: Google Chrome, Mozilla Firefox, Microsoft Edge. Необхідна наявність інтернет підключення. Для оновлення даних необхідно необхіден доступ до shell-клієнта сервера.

3.4.4.3. Авторизація.

3.4.4.3.1. Перейдіть за посиланням <https://evis.lift.org.ua>.

3.4.4.3.2. Увійдіть до системи, використовуючи корпоративну пошту та пароль.

3.4.4.3.3. У разі втрати паролю оновлення можливе виключно через термінал з використанням Laravel shell-команд.

3.4.4.3.4. Бажано не використовувати «Забули пароль?», але якщо не вийшло оновити доступ через shell-команди, то допускається використання.

3.4.4.4. Створення нового облікового запису.

3.4.4.4.1. Загальні положення.

Форма створення доступна після натиснення Створити користувача на головній сторінці.

3.4.4.4.2. Створення диспетчера.

3.4.4.4.2.1. Укажіть:

- Адресу електронної пошти;
- Пароль;
- Роль: Диспетчер;
- Прізвище, ім'я, по-батькові.

3.4.4.4.3. Створення механіка.

3.4.4.4.3.1. Укажіть:

- Адресу електронної пошти;
- Пароль;
- Роль: Механік;
- Прізвище, ім'я, по-батькові;
- Номер телефону;
- Посаду: Лінійний або Аварійний.

3.4.4.5. Формування звітів.

3.4.4.5.1. У розділі Звіти доступна така аналітика:

- Загальна кількість заявок по ліфтам;
- Заявки по механікам;
- ТОП-10 найпроблемніших ліфтів;
- Щоденні звіти.

3.4.4.5.2. Для збереження у форматі PDF натисніть Завантажити PDF. Доступне збереження звітів та друкування.

3.4.4.6. Завершення сеансу.

3.4.4.6.1. Для виходу з особистого кабінету натисніть на Вихід в меню.

### 3.5. Тестування програмного продукту

Тестування програмного забезпечення проводилось з метою перевірки правильності реалізації логіки та всіх функцій веб-порталу. Перевірялась відповідність до ТЗ та забезпеченню безпечної, надійної та стабільної роботи. В процесі використовувались такі типи перевірки: модульне, інтеграційне, системне та приймальне.

Checklist (контрольний список) – це структурований перелік перевірок або дій, які слід виконати в процесі тестування програмного продукту згідно визначеному порядку та сценарію [28].

#### 3.5.1. План тестування

В таблиці 3.1 наведені основні функціональні блоки веб-порталу, цілі, очікуваний результат та фактичний результат.

Таблиця 3.1. Чек-лист

Модуль	Функціональність	Очікуваний результат	Результат тестування
Реєстрація клієнта	Автопідказка вулиць після 3 символів	Список вулиць з БД	Успішно
Реєстрація клієнта	Вибір будинку після вибору вулиці	Вивід лише будинків на обраній вулиці	Успішно
Створення заявки	Повідомлення про існуючу незакриту заявку	Якщо вже є — виводиться попередження	Успішно
Клієнт: Перегляд заявок	Виводяться заявки клієнта	Дані правильні	Успішно
Диспетчер: перевірка заявок	Кнопка “Оновити” при новій заявці	Повідомлення працює	Успішно

Механік: закриття заявки	Зміна статусу заявки, оновлення в БД	Заявка оновлена	Успішно
Адмін: створення користувача	Додавання диспетчера/механіка	Новий запис у users	Успішно
Авторизація	Логін із підтвердженням пошти	Доступ дозволений лише після підтвердження	Успішно
Email підтвердження	Код надсилається, повторна відправка	Працює коректно	Успішно
Захист даних	Хешування паролів	Паролі у базі в хеші	Успішно

### 3.5.2. Рівні тестування

Модульне тестування проводилось для окремих скриптів: register\_client.php, create\_request.php, on\_review.php та інших. Кожен блок введення/обробки, SQL-запити та валідація були перевірені.

Інтеграційне тестування було проведено при перевірці взаємодії між модулями: від реєстрації клієнта до закриття заявки механіком. Особливу увагу було приділено логіці, де одна дія активувувала декілька залежних процесів, як внесення ліфту в зупинку.

Системне тестування проводилось на сервері Ukrline. Повний життєвий цикл був протестований в найпопулярніших браузерях таких як Chrome, Firefox і Edge. Також було перевірено відображення сторінок в мобільній версії. Виявлена проблема з часовим поясом була успішно усунена.

Приймальне тестування було проведено з декількома людьми для імітації реальної ситуації прийому та обробки заявки. Система виправдала очікування.

### 3.5.3. Тест-кейси

Було розроблено 3 тест-кейси для функцій

Тест-кейс 1: Створення заявки з проблемою «Інше»

- Клієнт обирає ліфт
- Обирає тип інше
- Вводить опис
- Надсилає заявку
- Очікування: Заявка з'являється у диспетчера в загальному списку та у вкладці «На огляді»
- Результат: Працює

Тест-кейс 2: Автоматичне виведення повідомлення про оновлення списку заявок

- Відкрити вкладку «На огляді»
- Створити нову заявку в ролі клієнта з типом Інше
- Перевірка появи точки біля пункту меню або поява повідомлення про оновлення, якщо диспетчер на цій сторінці
- Очікування: Виводиться повідомлення або з'являється точка
- Результат: Працює

Тест-кейс 3: Заборона створення нової заявки типу Ліфт не працює на ліфт у зупинці

- Встановити статус ліфта "В зупинці"
- Спробувати створити заявку
- Очікування: Система видає заборону
- Результат: Працює

### 3.5.4. Оцінка покриття тестами

На кожен функціональний модуль системи припадало в середньому 2-4 тестів. Найбільше тестів забрали модулі реєстрації, створення заявки та обробки

заявки диспетчером/механіком. Найменше змін потребувала частина з виводом звітів, адже вона базується на готовій бібліотеці mPDF і вже має стабільну логіку.

Візуалізація результатів тестування представлена у вигляді діаграми, яка свідчить про рівномірне покриття всіх ключових компонентів системи (рис. 3.26).

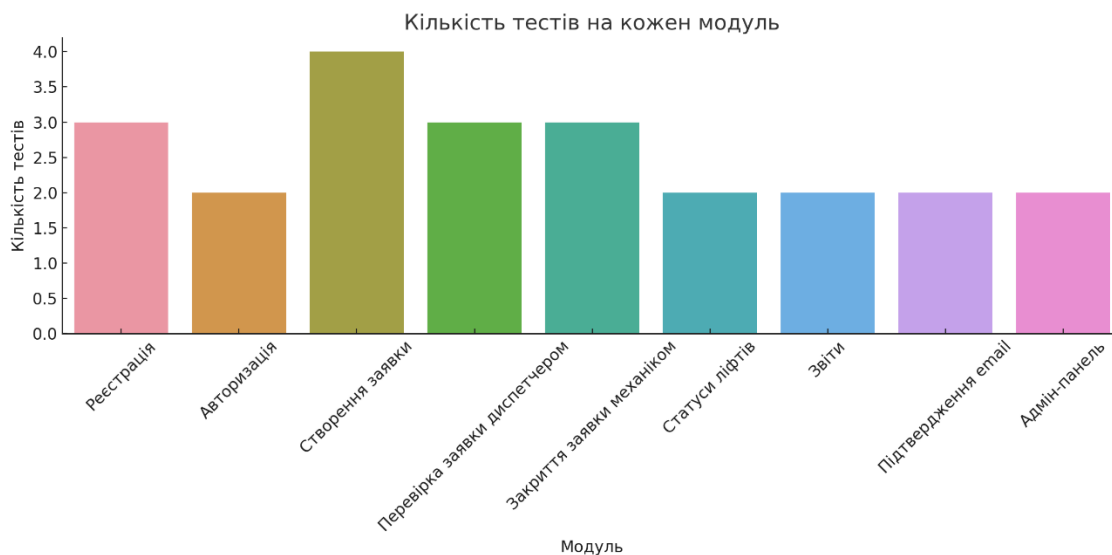


Рисунок 3.26 – Діаграма кількості тестів на кожен модуль

## ВИСНОВКИ

В рамках кваліфікаційної роботи було автоматизовано процес обробки ліфтових заявок. Основною метою проєкту було створення веб-порталу для взаємодії клієнтів, диспетчерів та механіків. Мета була виконана. Всі заплановані завдання, в тому ж числі вивчення реальної робочої ситуації на підприємстві, створення бази даних, розробка веб-порталу та впровадження основного функціоналу, були виконані.

Запропоноване програмне рішення надає змогу фіксувати проблеми ліфтового обладнання з повним або частковим вилученням телефонних дзвінків, що зменшує час на обробку звернення та мінімізує навантаження на диспетчерів.

Користувачі Клієнти мають доступ до функцій подання заявок, перегляду їх статусу та оновлення персональних даних. Для працівників доступні функції моніторингу технічного стану ліфтового обладнання, обробка заявок, формування звітів.

Система реалізує багаторівневу авторизацію та перевірку через електронну пошту, що підвищує надійність та безпечність. Також домен сайту має діючий SSL-сертифікат, що підвищує безпеку та зменшує ризики витоку інформації.

Важливим було і використання інтуїтивно зрозумілого інтерфейсу, що спрощує адаптацію клієнта до нового ПЗ. Модулі сайту були перевірені в ході тестування, що підтвердило стабільність роботи ресурсу та його відповідність до технічних вимог.

До подальшого розвитку можна віднести впровадження мобільного додатку, покращення інтерфейсу в деяких модулях, розширення API для інтеграції з іншими сервісами компанії, автоматизація аналітики, збільшення функцій на адмін панелі, створення англійської версії та прогнозування типових несправностей.

Очікуваним ефектом від впровадження виступає зменшення навантаження на диспетчерів, підвищення швидкості реагування на виклики, покращення координації дій технічного персоналу, підвищення задоволеності клієнтів та прозорості сервісного обслуговування.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ Б В.2.5–82:2016. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. – [Чинний від 2018-01-01]. – Київ : Мінрегіон України, 2016. – 32 с.
2. ДСТУ 3008:2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – [Чинний від 2016-07-01]. – Київ : ДП «УкрНДНЦ», 2015. – 18 с.
3. ДСТУ 3973:2000. Система розроблення та поставлення продукції на виробництво. Правила виконання науково-дослідних робіт. Загальні положення. – [Чинний від 2001-01-01]. – Київ : Держстандарт України, 2000. – 18 с.
4. ДСТУ 3974:2000. Система розроблення та поставлення продукції на виробництво. Правила виконання дослідно-конструкторських робіт. Загальні положення. – [Чинний від 2001-01-01]. – Київ : Держстандарт України, 2000. – 20 с.
5. Кузьмин О. Є. Теоретичні і практичні засади менеджменту : навчальний посібник / О. Є. Кузьмин, О. Г. Мельник. – 2-е вид. доп. і перероб. – Львів : Національний університет «Львівська політехніка» (Інформаційно-видавничий центр «ІНТЕЛЕКТ+» Інститут післядипломної освіти), «Інтелект-Захід», 2003. – 352 с.
6. М'якшило О.М, Харкянен О.В. Проектування Інформаційних Систем. Конспект Лекцій: Київ НУХТ 2018. – 46с.
7. Пушкар Р. М. Менеджмент : теорія і практика: підручник / Р. М. Пушкар, Н. П. Тарнавська. – 3-тє вид., перероб. і доп. – Тернопіль : Карт-бланш, 2005. – 486 с.
8. Ситник Й. С. Менеджмент організацій : навчальний посібник /Ситник Й. С. – Львів : «Тріада плюс», 2008. – 456 с.
9. Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки» денної та заочної

- форми здобуття освіти [Електрон. ресурс] / уклад. С. В. Грибков, Н. В. Ліманська, М. П. Костіков. – К.: НУХТ, 2025. – 43 с.
10. Анатолій Проботиук | Що таке Laravel? *probotiuk.com*. – Режим доступу: <https://probotiuk.com/blog/sho-take-laravel> (дата звернення: 18.05.2025).
  11. Вступ — phpMyAdmin 6.0.0-dev documentation.*phpMYAdmin*. – Режим доступу: <https://docs.phpmyadmin.net/uk/latest/intro.html> (дата звернення: 18.05.2025).
  12. Головна | *Elib LNTU*. – Режим доступу: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/Кондіус%20%20Оготовва/page18.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/Кондіус%20%20Оготовва/page18.html) (дата звернення: 18.05.2025).
  13. Головна | *Elib LNTU*. – Режим доступу: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/Кондіус%20%20Оготовва/page9.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/Кондіус%20%20Оготовва/page9.html) (дата звернення: 18.05.2025).
  14. Етапи створення веб-додатку. Основи PHP та MySQL. *Створення інтернет магазину, створити інтернет магазин ціна, розробка*. – Режим доступу: <https://promoter.net.ua/articles/etapi-stvorenniya-veb-dodatku-osnovi-php-ta-mysql.html> (дата звернення: 19.05.2025).
  15. Лабораторна робота Вивчення вікна програми All Fusion Process Modeler, інструментів моделювання та основних фігур, які використовуються при моделюванні, робота зі стрілками та їх різновидами. Побудова контекстної діаграми бізнес-процесу. *KURSOVIKS*. – Режим доступу: [https://kuroviks.com.ua/bd\\_kompyuternyye/article\\_post/403-laboratorna-robota-vivchennya-vikna-programi-all-fusion-process-modeler-instrumentiv-modelyuvannya-ta-osnovnikh-figur-yaki-vikoristovuyutsya-pri-modelyuvanni-robota-zi-strilkam](https://kuroviks.com.ua/bd_kompyuternyye/article_post/403-laboratorna-robota-vivchennya-vikna-programi-all-fusion-process-modeler-instrumentiv-modelyuvannya-ta-osnovnikh-figur-yaki-vikoristovuyutsya-pri-modelyuvanni-robota-zi-strilkam) (дата звернення: 16.05.2025).
  16. Надаємо Web Хостинг для сайтів | VDS/VPS - UkrLine. *UkrLine*. URL: <https://ukrline.com.ua/ua/> (дата звернення: 16.05.2025).
  17. Посадова інструкція - Директор департаменту. *Посадові інструкції*. – Режим доступу: [https://www.borovik.com/index\\_instruction.php?Gins=mymtc&lang\\_i=1](https://www.borovik.com/index_instruction.php?Gins=mymtc&lang_i=1) (дата звернення: 17.05.2025).

18. Сервер *apache. Ranktracker.* – Режим доступу: <https://www.ranktracker.com/uk/seo/glossary/apache-server/> (дата звернення: 17.05.2025).
19. Типи організаційних структур управління підприємством - бібліотека *buklib.net. Головна - Бібліотека BukLib.net.* – Режим доступу: <https://buklib.net/books/32066/> (дата звернення: 18.05.2025).
20. Що таке BPMN-діаграма і навіщо вона потрібна з прикладами | IAMPМ. *IAMPМ.* – Режим доступу: <https://iampm.club/ua/blog/shho-take-bpmn-diagrama-i-navishho-vona-potribna-z-prikladami-2/> (дата звернення: 18.05.2025).
21. Що таке *CloudLinux. УН.ua.* – Режим доступу: <https://uh.ua/ua/kb/vps/software-installation/cloud-linux.html> (дата звернення: 16.05.2025).
22. Що таке html? *CSS.IN.UA.* – Режим доступу: [https://css.in.ua/article/shcho-take-html\\_10](https://css.in.ua/article/shcho-take-html_10) (дата звернення: 17.05.2025).
23. Що таке JavaScript і для чого він потрібен – GoIT Global. *GoIT.* – Режим доступу: <https://goit.global/ua/articles/shcho-take-javascript-i-dlia-choho-vin-potriben/> (дата звернення: 17.05.2025)..
24. Що таке PHP? - *hyperhost wiki. HyperHost.* – Режим доступу: <https://hyperhost.ua/uk/wiki/chto-takoe-php> (дата звернення: 17.05.2025).
25. 10.4. Бізнес-процеси: сутність поняття, учасники, види бізнес- процесів :: *megalib.com.ua. Електронна бібліотека онлайн MegaLib.com.ua.* – Режим доступу: [http://megalib.com.ua/content/10202\\_Biznes\\_procesi\\_sytnist\\_ponyatty\\_a\\_uchasniki\\_vidi\\_biznes\\_procesiv.html](http://megalib.com.ua/content/10202_Biznes_procesi_sytnist_ponyatty_a_uchasniki_vidi_biznes_procesiv.html) (дата звернення: 16.05.2025).
26. 14357579 - ПРАТ "ОТІС" - Основна інформація - *Clarity Project. Clarity Project.* – Режим доступу: <https://clarity-project.info/edr/14357579> (дата звернення: 15.05.2025).
27. *CA ERWin Data Modeler (ERWin) - Бази даних.: Підручники для студентів онлайн.* – Режим доступу:

- [https://stud.com.ua/77235/informatika/erwin\\_data\\_modeler\\_erwin](https://stud.com.ua/77235/informatika/erwin_data_modeler_erwin) (дата звернення: 16.05.2025).
28. Checklist – чудовий помічник для QA. *qagroup.com.ua* – Режим доступу: <https://qagroup.com.ua/publications/checklist-chudovyj-pomichnyk-dlia-qa> (дата звернення: 16.05.2025).
29. Domain registrar NIC.UA. *Domain registrar | NIC.UA.* – Режим доступу: <https://nic.ua/> (дата звернення: 17.05.2025).
30. Home - GLPI Project. *GLPI Project.* URL: <https://glpi-project.org>.
31. HTML і CSS: що це, кому та для чого потрібно. GoIT. – Режим доступу: <https://goit.global/ua/articles/html-i-css-shcho-tse-komu-ta-dlia-choho-potribno/> (дата звернення: 18.05.2025).
32. MySQL: визначення, основні характеристики та компоненти, історія – Simplen Trec. *Simplen Trec – платформа для розвитку та самовдосконалення.* – Режим доступу: <https://simplentrec.com/mysql-introduction/> (дата звернення: 18.05.2025).
33. Otis History | Elevator history timeline. *OTIS.* – Режим доступу: <https://www.otis.com/uk/ua/our-company/history> (дата звернення: 18.05.2025).
34. OTRS software solutions for customer service management, ITSM, ISMS, SOAR and more. *OTRS.* – Режим доступу: <https://otrs.com/?lang=en> (дата звернення: 16.05.2025).
35. PHP API – створюйте та редагуйте PDF-файл, захищений... *FILEFORMAT.* – Режим доступу: <https://products.fileformat.com/uk/pdf/php/mpdf/> (дата звернення: 18.05.2025).
36. PHPMailer для Чайників | Навіщо він потрібен? *Ukrline.* – Режим доступу: <https://ukrline.com.ua/> (дата звернення: 18.05.2025).
37. Yuliia. 9 програм для моделювання бізнес процесів - ZOHO Partner. *Впровадження Zoho CRM для вашого бізнесу від офіційних партнерів.* – Режим доступу: <https://uk.crmoz.com/blogs/post/business-process-modeling> (дата звернення: 16.05.2025).

## ДОДАТКИ

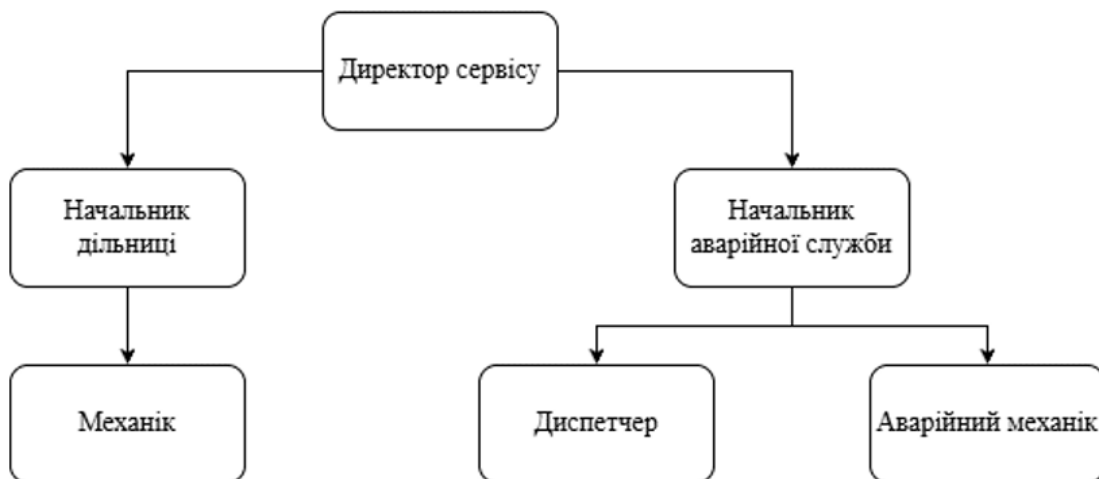
**Додаток А. Організаційна структура департаменту сервісних операцій**

Рисунок А.1 - Організаційна структура в ПрАТ «ОТІС»

## Додаток Б. Функціональні моделі

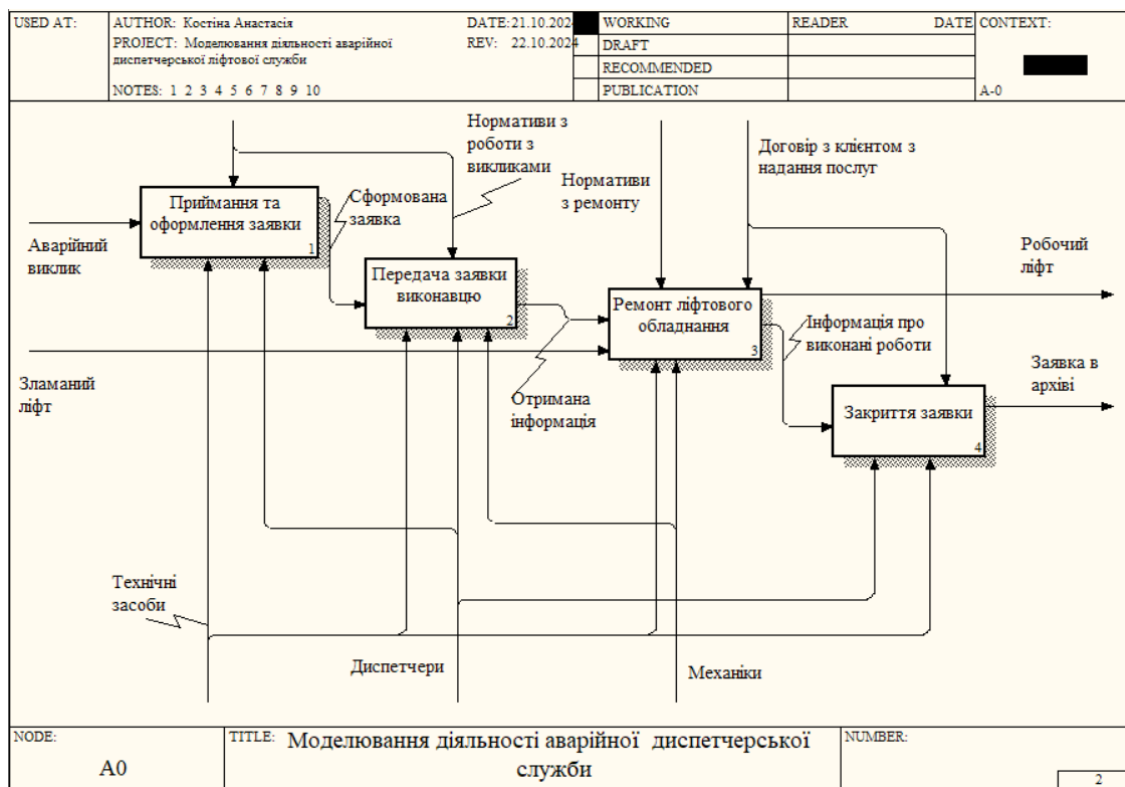


Рисунок Б.1 – Моделювання діяльності аварійної диспетчерської служби в нотатції IDF0

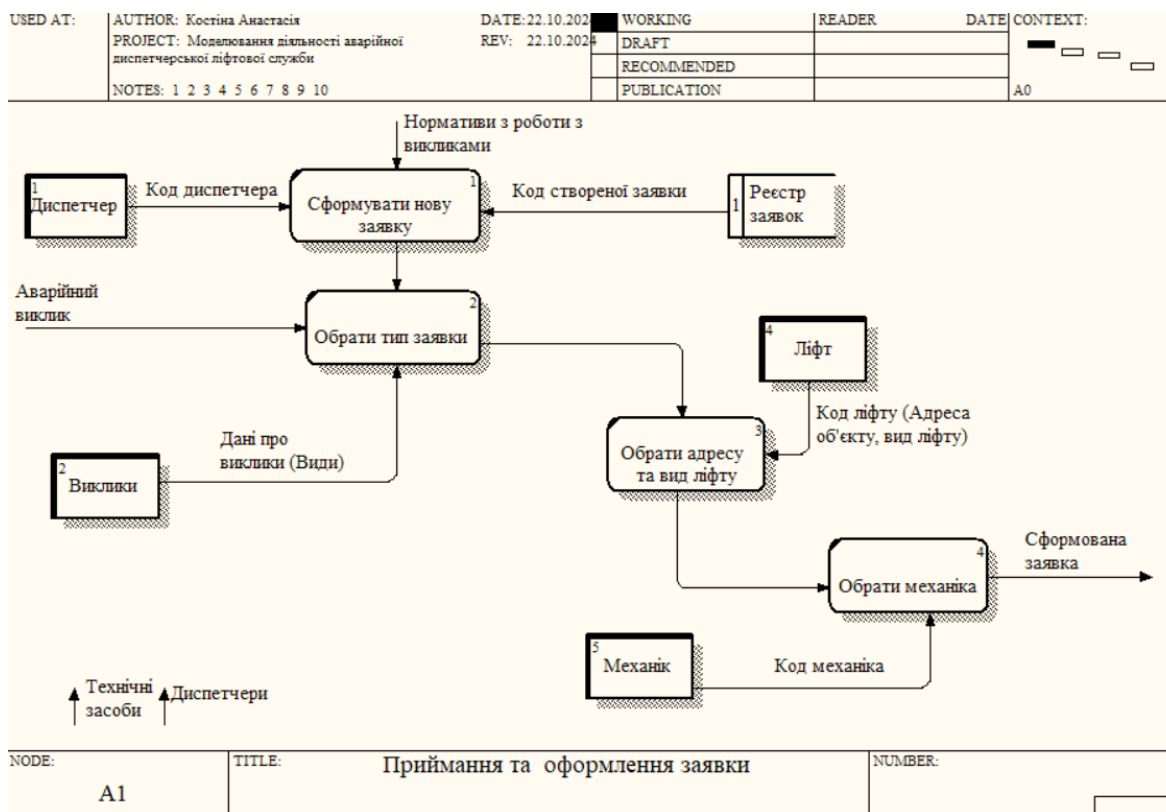


Рисунок Б.2 – Приймання та оформлення заявки в нотатції DFD

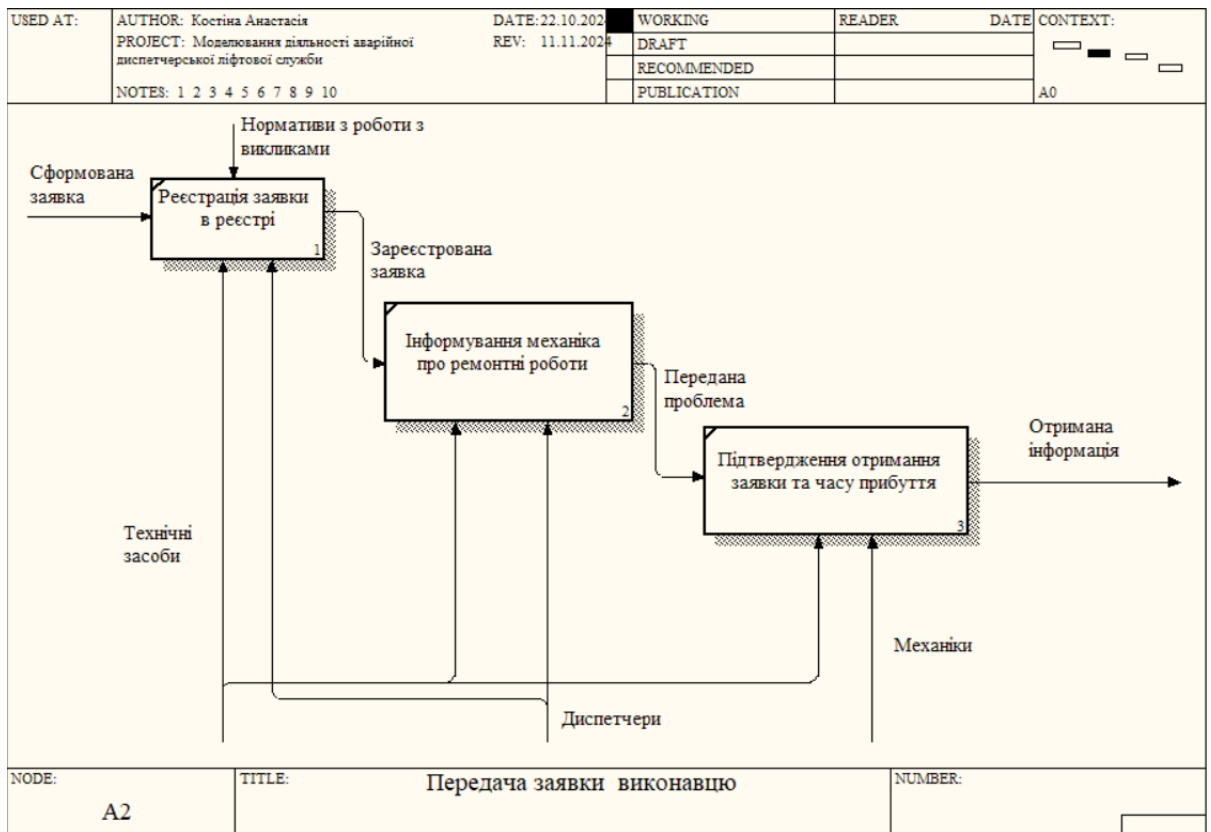


Рисунок Б.3 – Передача заявки виконавцю в нотації IDF0

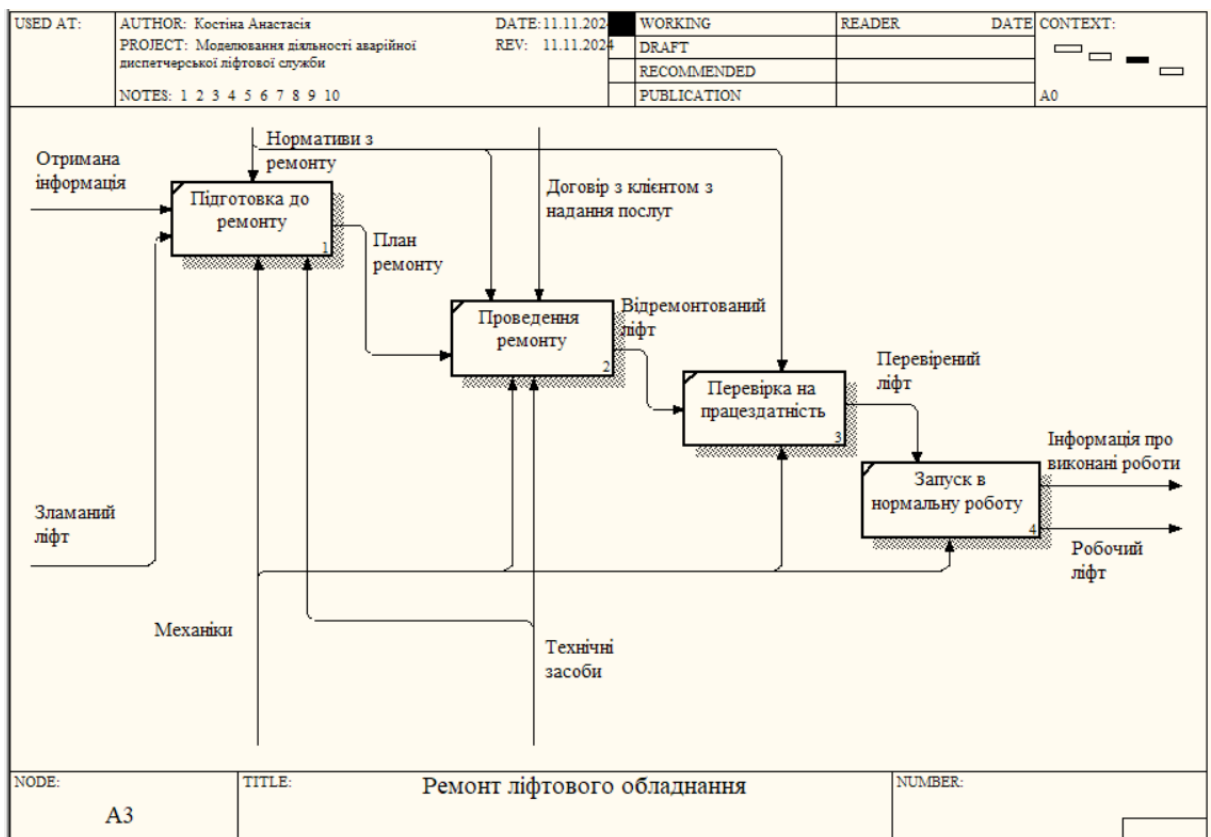


Рисунок Б.4 - Ремонт ліфтового обладнання в нотації IDF0

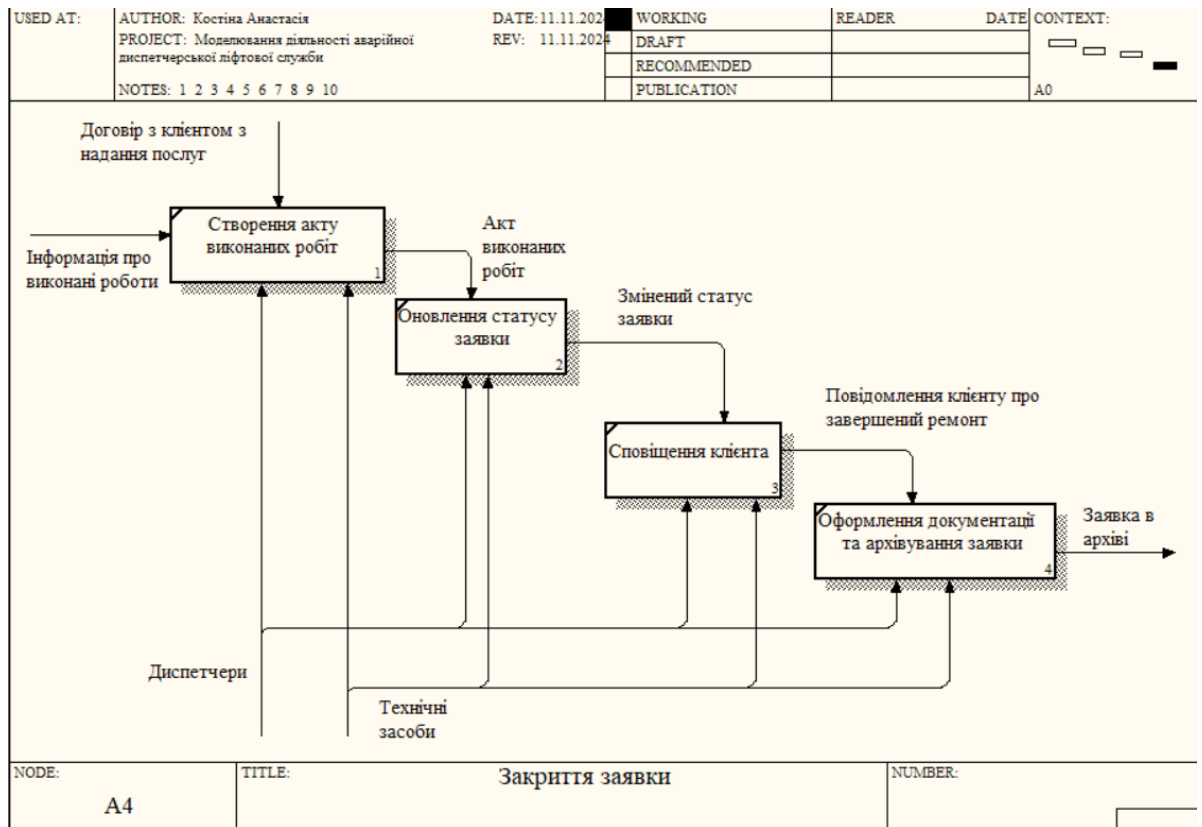


Рисунок Б.5 – Закриття заявки в нотатції IDF0

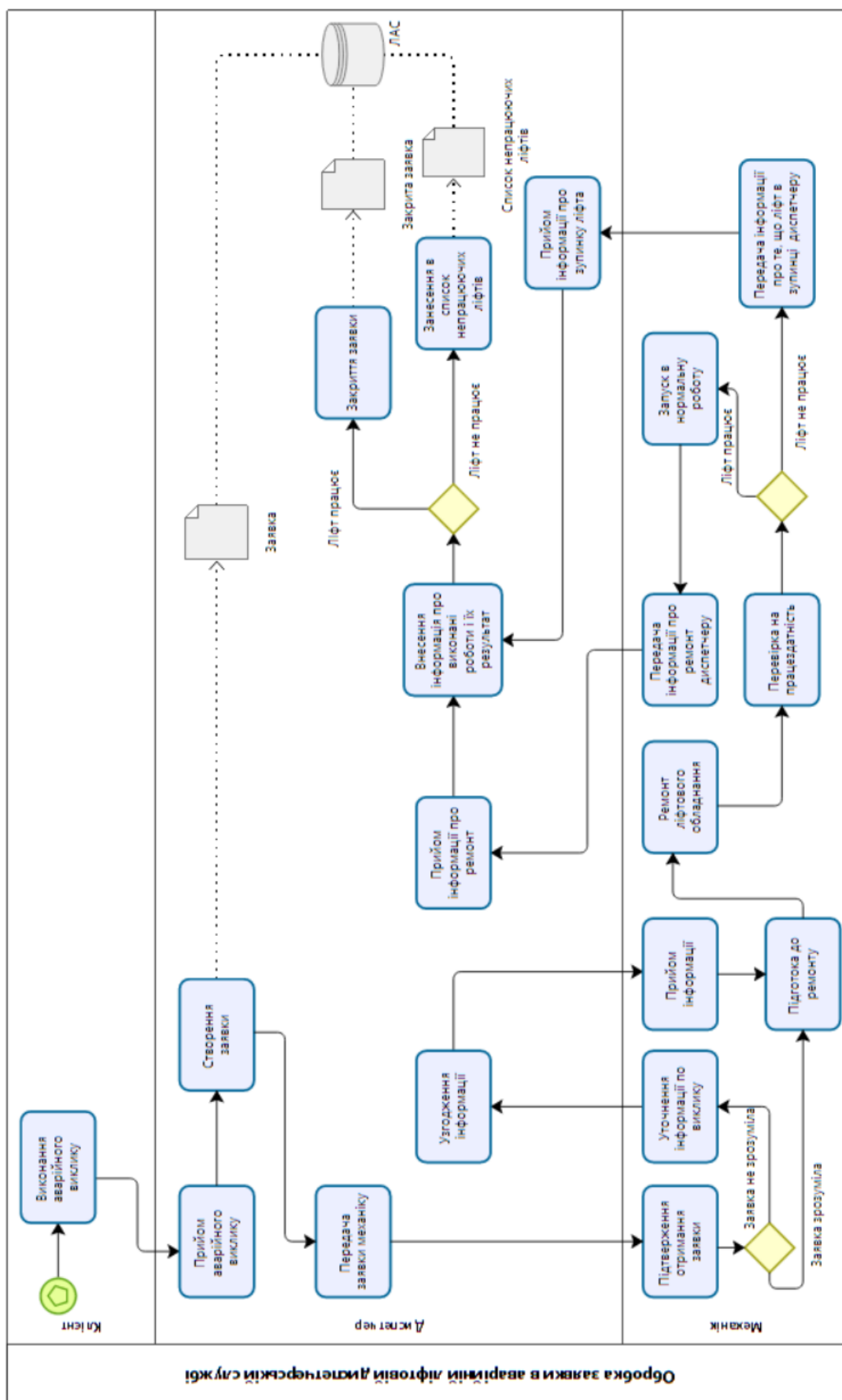


Рисунок Б.6 - Бізнес процес прийняття та обробки заявки в нотації BPMN

## Додаток В. Моделі та схеми бази даних

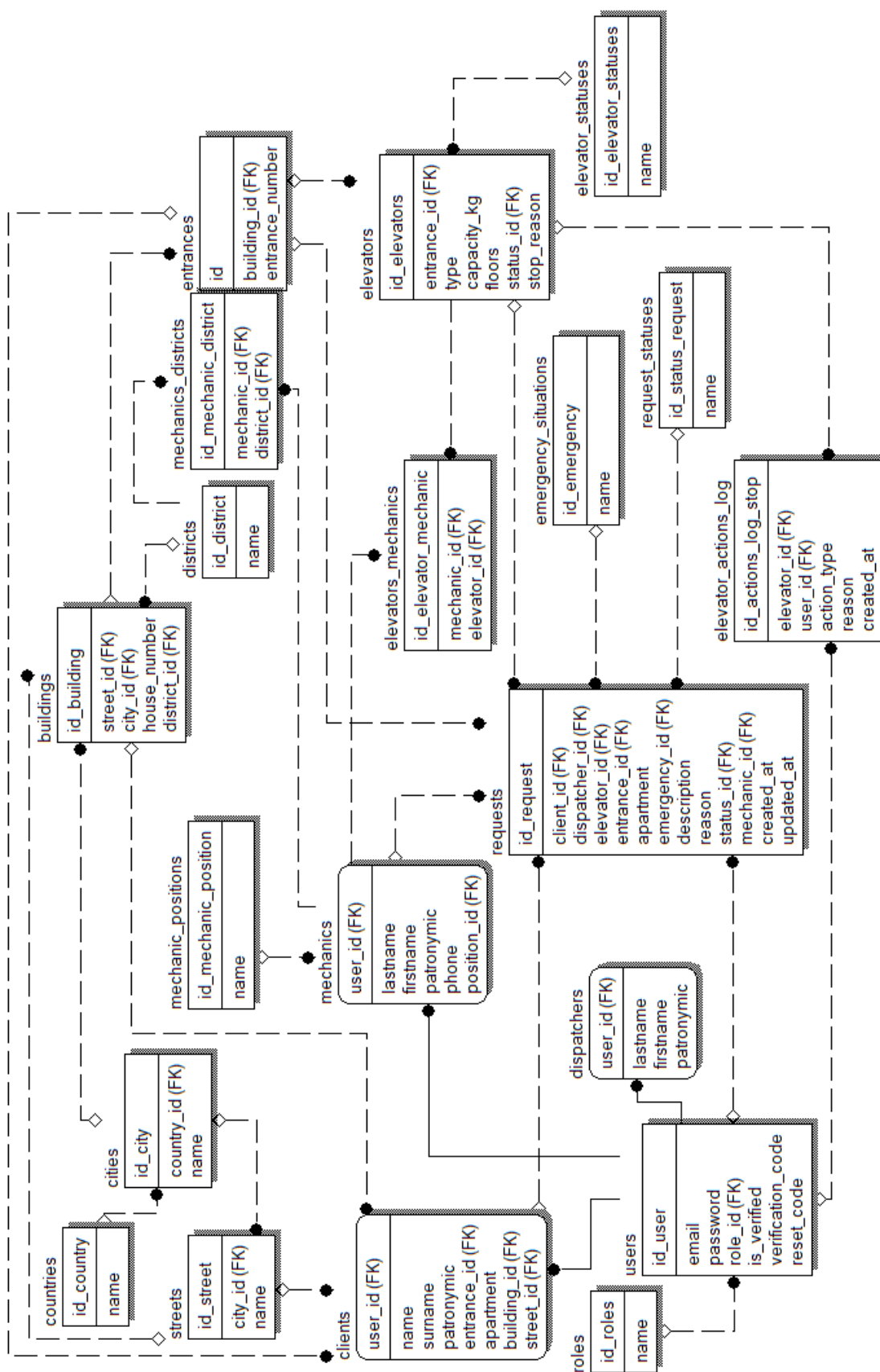


Рисунок В.1 – Логічна модель бази даних

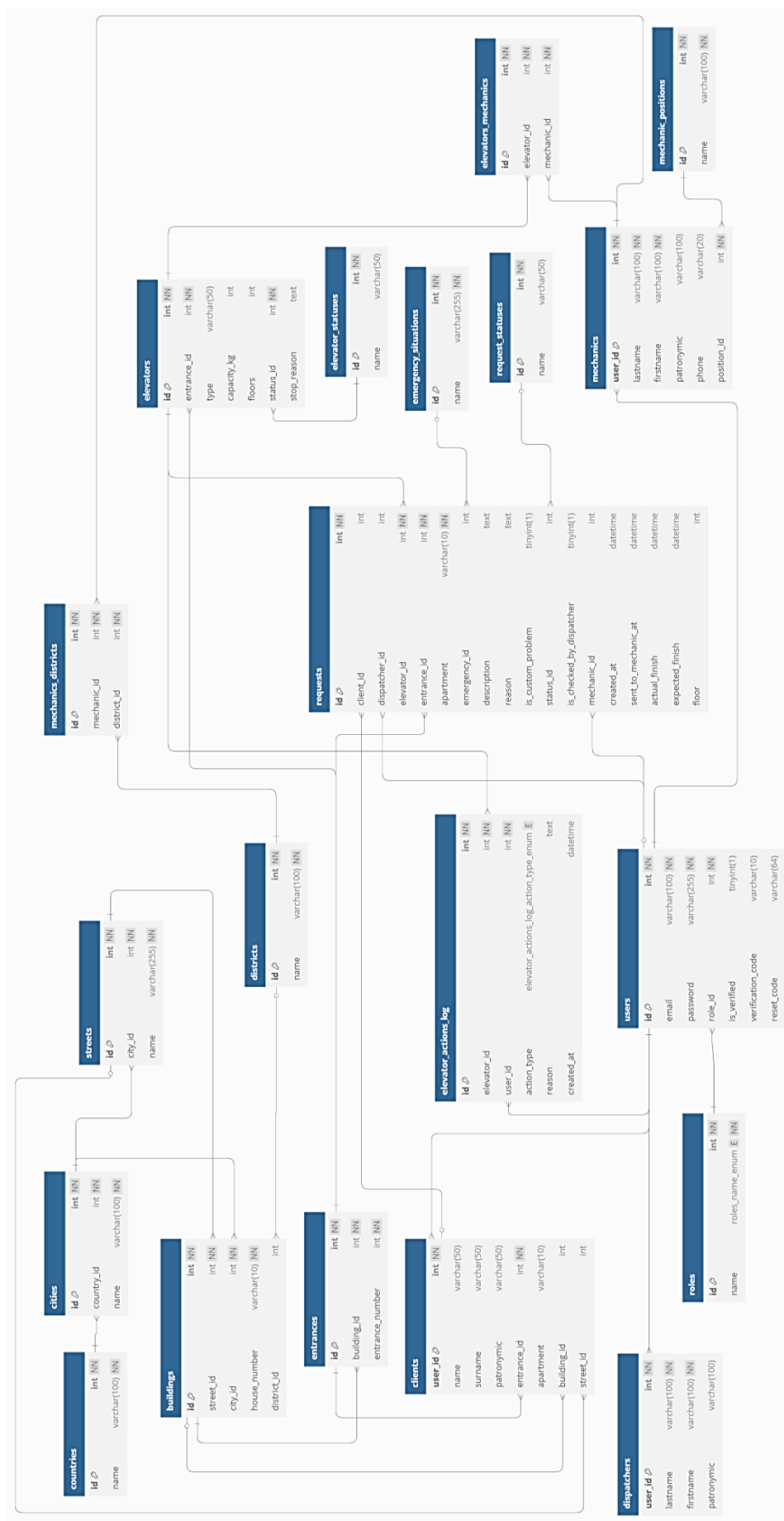
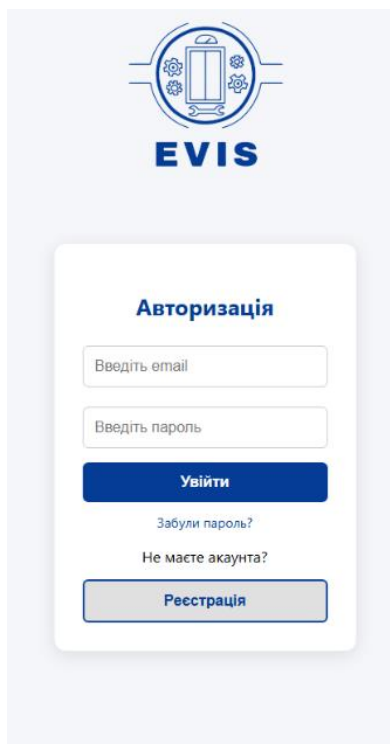


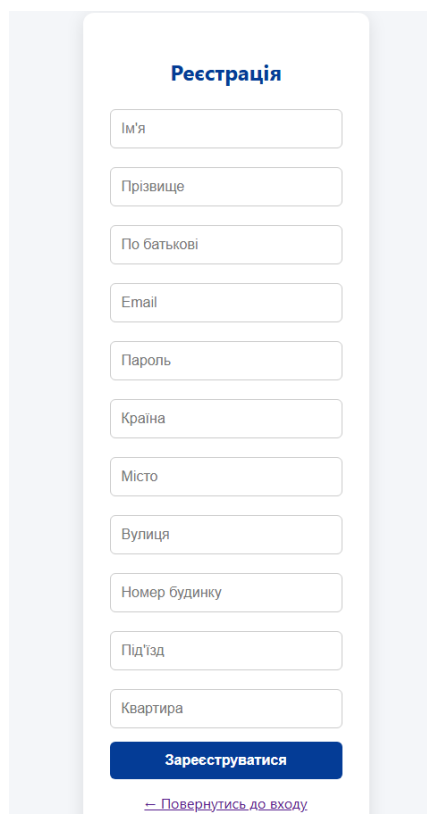
Рисунок В.2 – Схема бази даних

## Додаток Г. Скриншоти інтерфейсу користувача



The screenshot shows the EVIS login interface. At the top is the EVIS logo, which consists of a circular emblem with a book and gears, and the text "EVIS" below it. The main heading is "Авторизація". Below this are two input fields: "Введіть email" and "Введіть пароль". A blue button labeled "Увійти" is positioned below the password field. Underneath the button are two links: "Забули пароль?" and "Не маєте акаунта?". At the bottom of the form is a grey button labeled "Реєстрація".

Рисунок Г.1 – Авторизація EVIS



The screenshot shows the EVIS registration form. The heading is "Реєстрація". The form contains several input fields: "Ім'я", "Прізвище", "По батькові", "Email", "Пароль", "Країна", "Місто", "Вулиця", "Номер будинку", "Під'їзд", and "Квартира". A blue button labeled "Зареєструватися" is located at the bottom of the form. Below the button is a link: "— Повернутись до входу".

Рисунок Г.2 – Сторінка реєстрації

## Реєстрація

Анастасія

Костіна

Володимирівна

nastia@gmail.com

.....

Укр]

• Україна

Рисунок Г.3 – Демонстрація підказки



EVIS

кому: мене ▾

Ваш код підтвердження: 717920

Рисунок Г.4 – Лист з кодом підтвердження акаунту

EVIS

Головна Мої заявки Персональні дані Вихід

**⚠ Увага!**

У вашому будинку наразі є ліфти, що знаходяться в зупинці:

- Ліфт типу **1 вант** — під'їзд 1, вул. Ак. Заболотного 8.  
Причина зупинки: згорів двигун

**Ласкаво просимо в EVIS, Анастасія!**

EVIS — Elevator Visit Information System

У цьому кабінеті ви можете переглядати свої заявки, створювати нові, змінювати персональні дані та відстежувати інформацію про стан ліфта.

Компанія **Otis** дбає про вашу безпеку та комфорт.

**Актуальні заявки по вашому будинку:**

- Застрягли люди (2025-05-20 19:49:55)
- Застрягли речі (2025-05-18 02:20:51)
- Інше — Перевірка (2025-05-18 01:18:50)
- Тех огляд по заявці мешканця (2025-05-17 22:53:01)
- Інше — ПЕРЕВІРКА JS (2025-05-17 21:37:06)

+ Створити заявку

Рисунок Г.5 – Головна сторінка клієнта

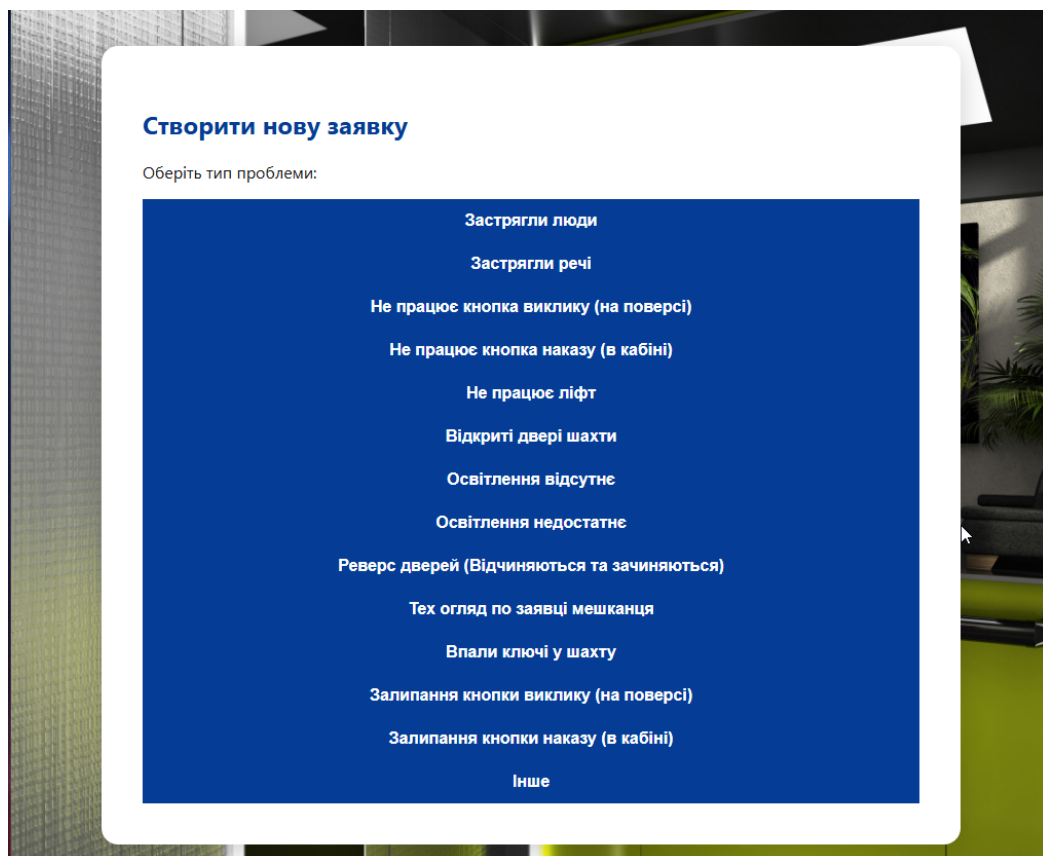


Рисунок Г.6 – Створення нової заявки

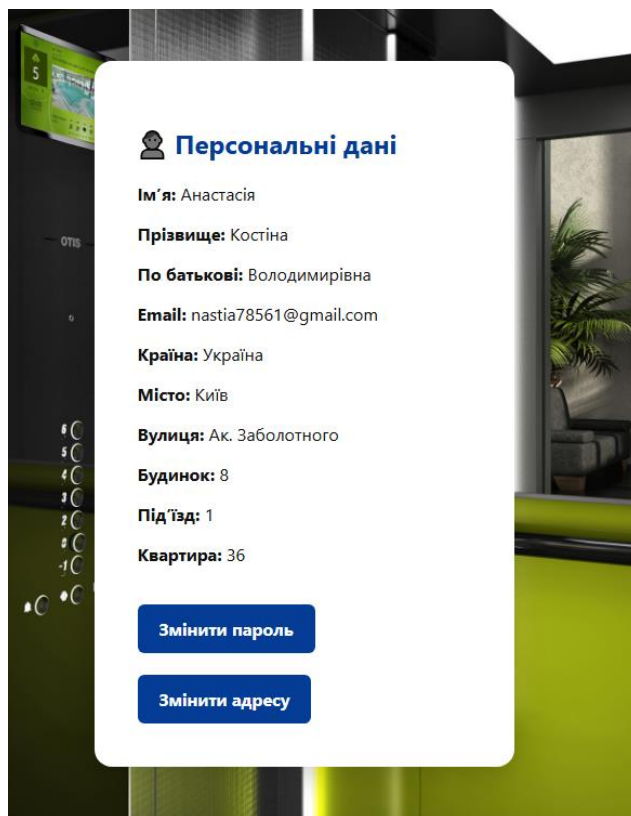
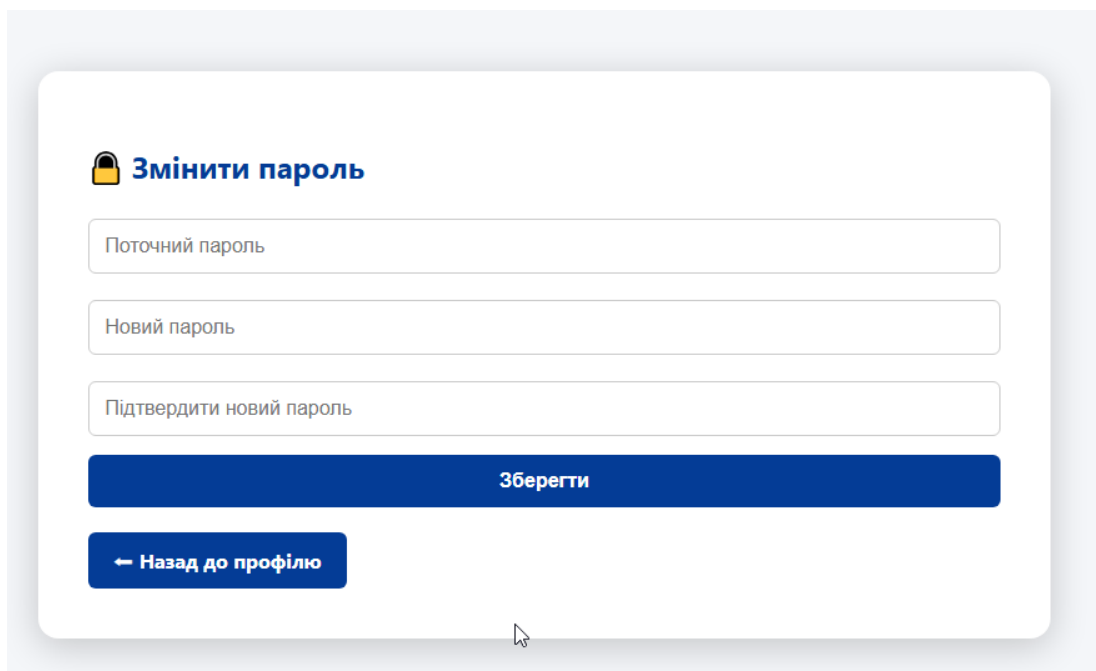


Рисунок Г.7 – Сторінка персональні дані



**🔒 Змінити пароль**

Поточний пароль

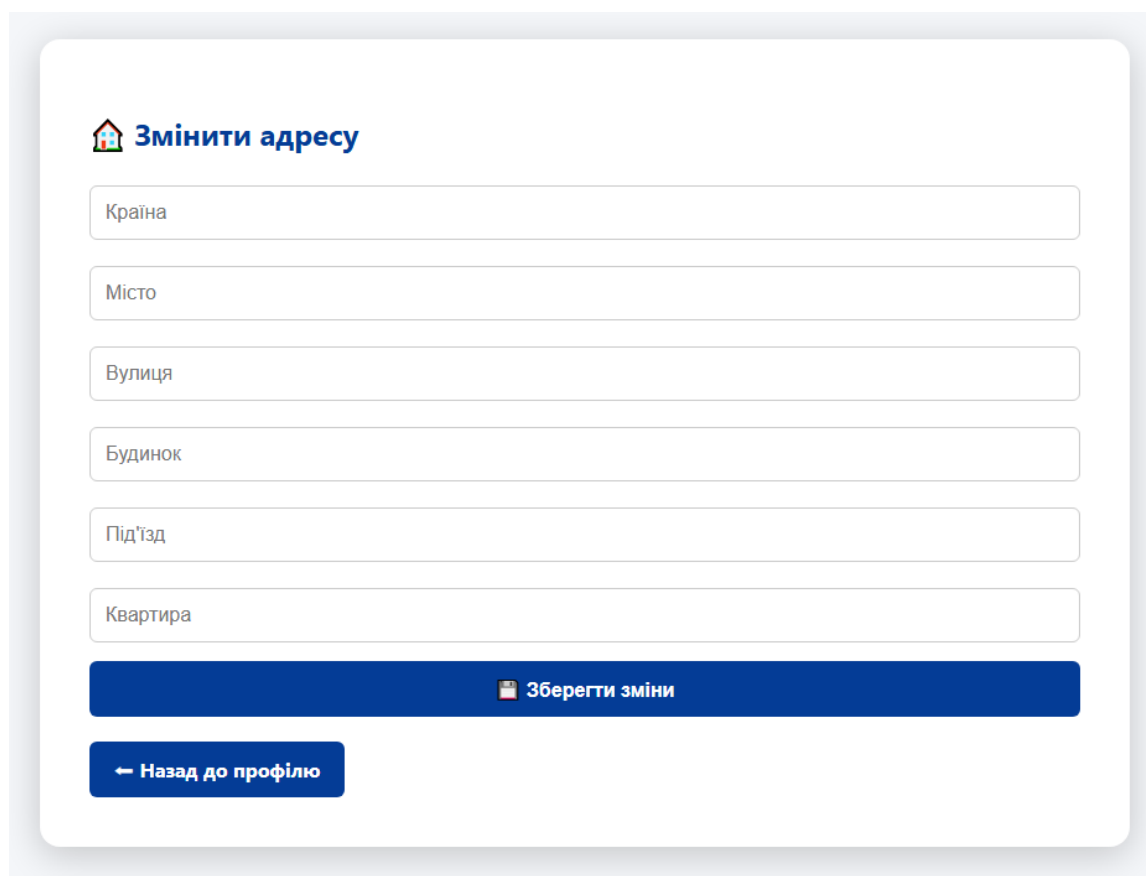
Новий пароль

Підтвердити новий пароль

**Зберегти**

**← Назад до профілю**

Рисунок Г.8 – Зміна паролю



**🏠 Змінити адресу**

Країна

Місто

Вулиця

Будинок

Під'їзд

Квартира

**📄 Зберегти зміни**

**← Назад до профілю**

Рисунок Г.9 – Зміна адреси

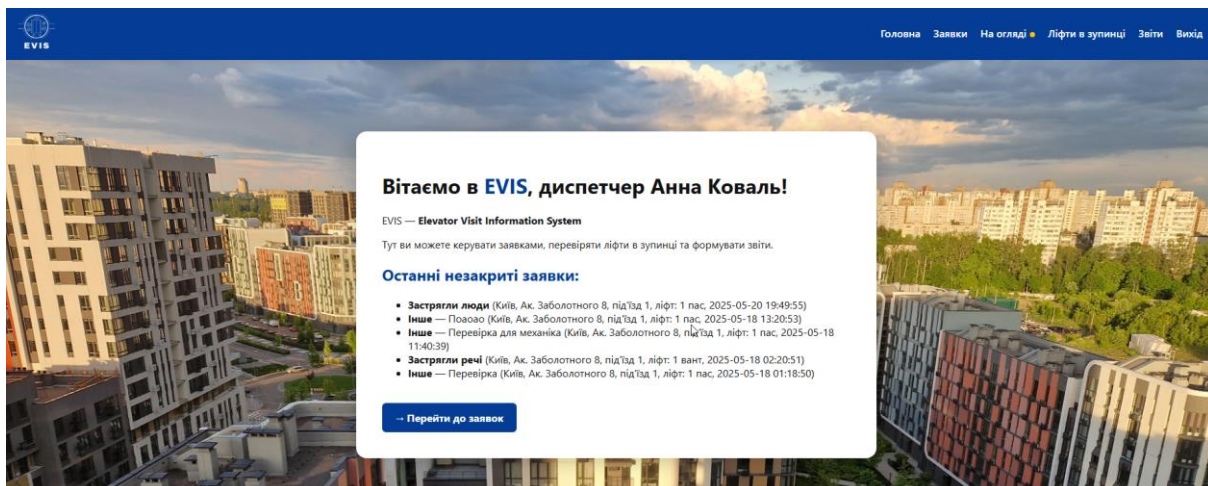


Рисунок Г.10 – Головна диспетчер

Дата	Проблема	Адреса	Квартира	Поверх	Ліфт	Механік	Статус	Дії
2025-05-20 19:49:55	Застрягли люди	Київ, Ак. Заболотного, 8, під'їзд 1	2	—	1 пас	Андреянов Д.В.	Передана механіку	
2025-05-18 13:20:53	Поазао	Київ, Ак. Заболотного, 8, під'їзд 1	36	—	1 пас	Андреянов Д.В.	Очікує перевірки диспетчером	
2025-05-18 11:40:39	Перевірка для механіка	Київ, Ак. Заболотного, 8, під'їзд 1	36	—	1 пас	Андреянов Д.В.	Очікує перевірки диспетчером	
2025-05-18 02:20:51	Застрягли речі	Київ, Ак. Заболотного, 8, під'їзд 1	36	—	1 вант	Андреянов Д.В.	Передана механіку	
2025-05-18 01:18:50	Перевірка	Київ, Ак. Заболотного, 8, під'їзд 1	36	—	1 пас	Андреянов Д.В.	Очікує перевірки диспетчером	
2025-05-17 22:53:01	Тех огляд по заявці мешканця	Київ, Ак. Заболотного, 8, під'їзд 1	36	—	1 вант	Андреянов Д.В.	Передана механіку	
2025-05-17 21:37:06	ПЕРЕВІРКА JS	Київ, Ак. Заболотного, 8, під'їзд 1	36	—	1 пас	Андреянов Д.В.	Очікує перевірки диспетчером	

Рисунок Г.11 – Вигляд таблиці заявок

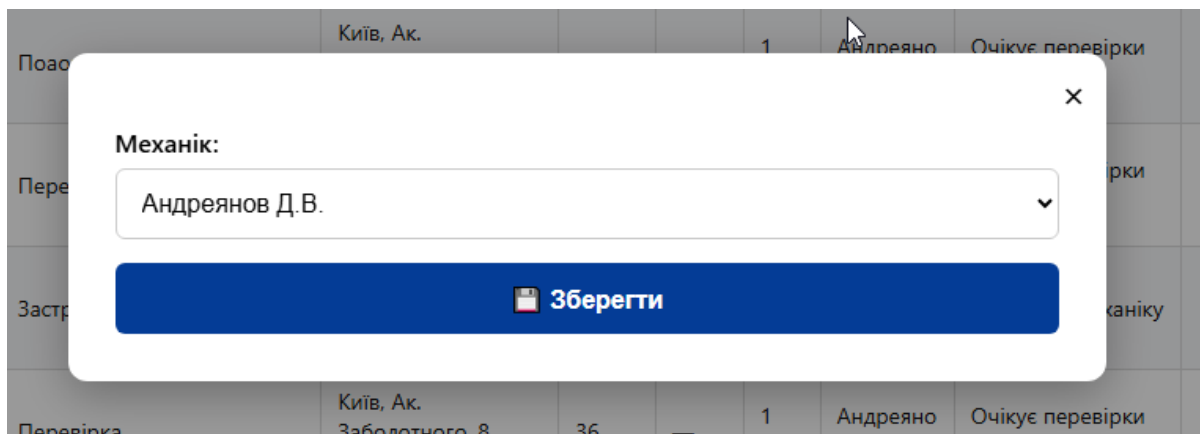


Рисунок Г.12 – Зміна механіка

Статус заявки:

✓ Закрито

Причина поломки:

Причина...

Статус ліфта:

В зупинці

Причина зупинки ліфта:

Опишіть причину зупинки...

Закрити заявку

Рисунок Г.13 – Вигляд модального вікна закриття при виборі таких наборів даних

**Створити заявку вручну**

Оберіть місто + вулицю:

Київ, Ак. Заболотного

Будинок:

8, під'їзд 1

Ліфт:

1 вант

Тип аварії:

Застрягли речі

Квартира:

67

Механік:

Сидорчук О.І.

Створити заявку

Рисунок Г.14 – Заповнене вікно створення заявки

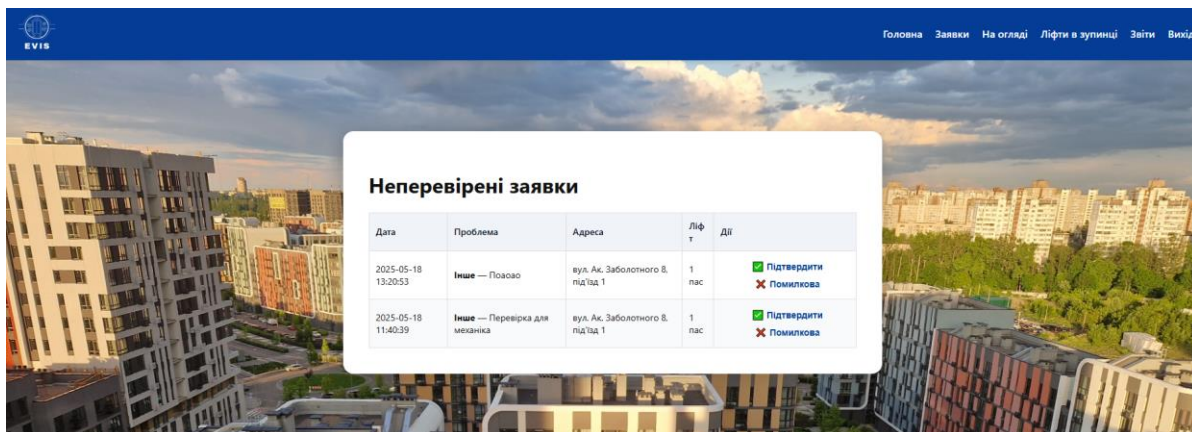


Рисунок Г.15 – Вигляд неперевірених заявок

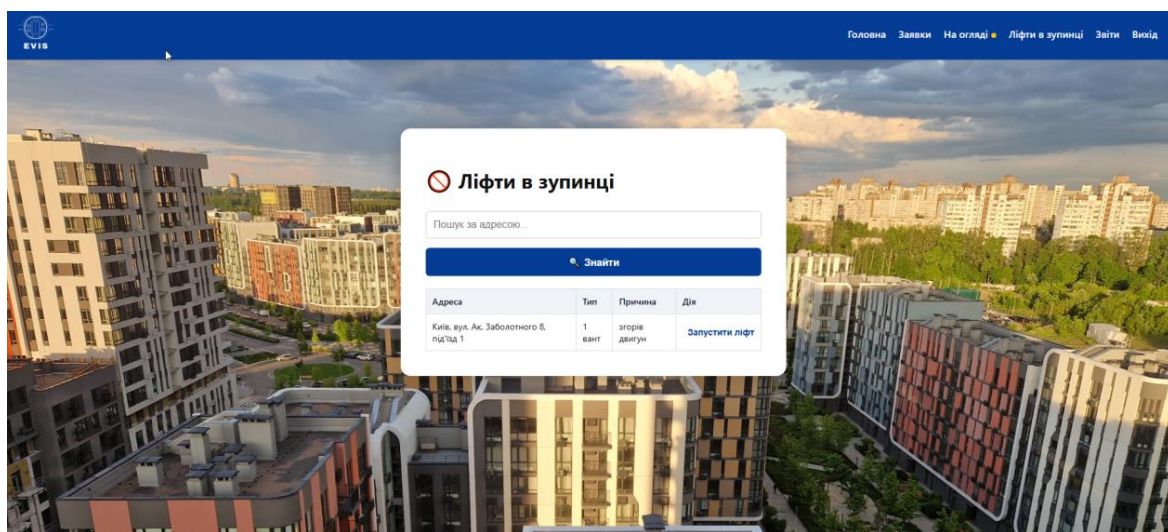


Рисунок Г.16 – Сторінка запуску ліфтів

### Кількість заявок по кожному ліфту

Адреса	Тип	Заявок	К-сть зупинок	Помилкових
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 пас	23	2	10
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 вант	6	1	0
Київ, вул. Хрещатик 1, під'їзд 1	1 пас	1	1	0
Київ, вул. Хрещатик 1, під'їзд 2	1 пас	1	0	0
Київ, вул. Хрещатик 1, під'їзд 1	1 вант	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	2 пас	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	1 пас	0	0	0

Рисунок Г.17 – Вигляд таблиці звіту у вигляді PDF

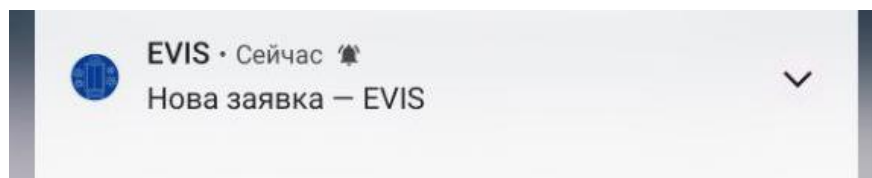


Рисунок Г.18 – Вигляд сповіщення про нову заявку для механіка

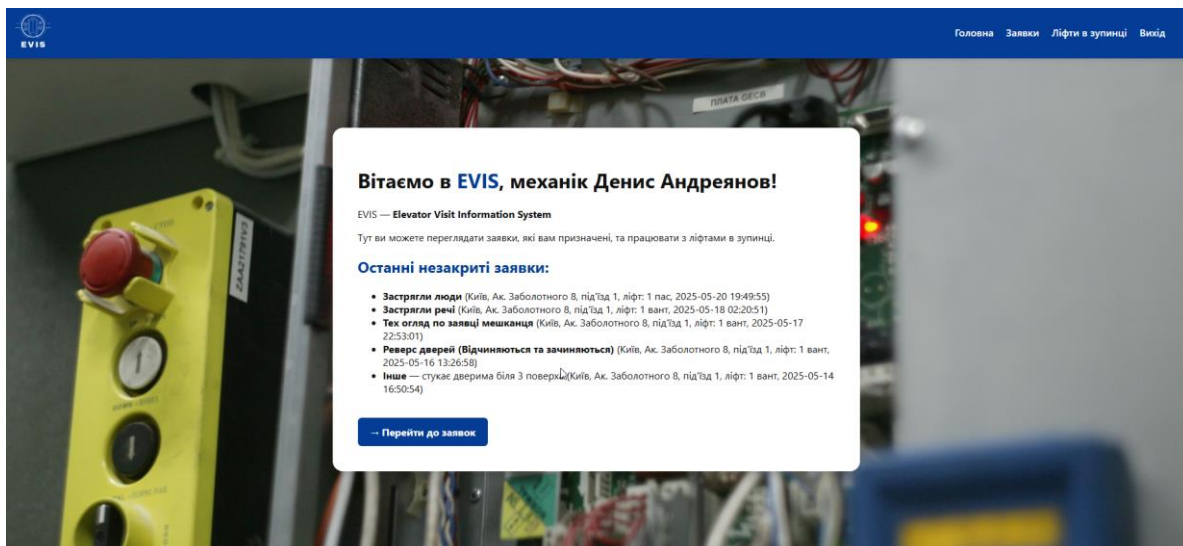


Рисунок Г.19 – Головна механіка

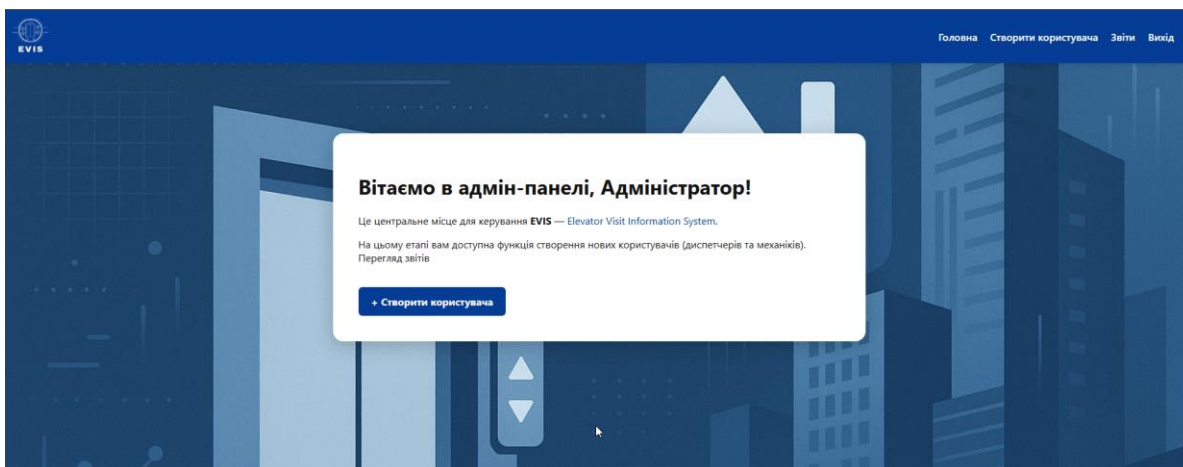


Рисунок Г.20 – Головна адміна

**Створити нового користувача**

Email:

Пароль:

Роль:

Прізвище:

Ім'я:

По-батькові:

Рисунок Г.21 – Створення нового користувача

**Звіти**

Кількість заявок по кожному ліфту

Адреса	Тип	Заво к	К-сть зупинок	Помилкови х
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 пас	23	2	10
Київ, вул. Ак. Заболотного 8, під'їзд 1	1 вант	6	1	0
Київ, вул. Хрещатик 1, під'їзд 2	1 пас	1	0	0
Київ, вул. Хрещатик 1, під'їзд 1	1 пас	1	1	0
Київ, вул. Хрещатик 1, під'їзд 1	1 вант	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	1 пас	0	0	0
Київ, вул. Богдана Хмельницького 10А, під'їзд 1	2 пас	0	0	0

Рисунок Г.22 – Сторінка звітів

**Мої заявки**

Дата	Тип проблеми	Опис	Статус
2025-05-18 13:21:40	Не працює ліфт	—	Закрита
2025-05-18 13:20:53	Інше	Поаао	Очікує перевірки диспетчером
2025-05-18 11:40:39	Інше	Перевірка для механіка	Очікує перевірки диспетчером
2025-05-18 04:51:57	Інше	Ага ага	Помилка
2025-05-18 09:24:12	Інше	Фінал?..	Помилка
2025-05-18 03:21:30	Інше	Перевірка 2	Помилка
2025-05-18 03:19:01	Інше	Фінальна перевірка...	Помилка
2025-05-18 02:54:16	Інше	Перевірка	Помилка

Рисунок Г.23 – Сторінка заявок клієнта

## Додаток Д. Фрагменти коду програми

### Д.1. index.php

```

<!DOCTYPE HTML>

<?php
session_start();
require_once "db.php";

$error = "";
$success = $_GET['verified'] ?? "";
$logged_out = isset($_GET['logout']) ? true : false;

// Обробка входу
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $email = trim($_POST['email']);
    $password = trim($_POST['password']);

    $stmt = mysqli_prepare($conn, "SELECT id, password, is_verified, role_id
FROM users WHERE email = ?");
    mysqli_stmt_bind_param($stmt, "s", $email);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

    if ($row = mysqli_fetch_assoc($result)) {
        if (password_verify($password, $row['password'])) {
            if ($row['is_verified']) {
                $user_id = $row['id'];
                $role_id = $row['role_id'];
                $name = 'Користувач'; // За замовчуванням

                // Витягуємо ім'я залежно від ролі
                switch ($role_id) {
                    case 1: // Адмін (можна не витягувати, ім'я — "Адміністратор")
                        $name = 'Адміністратор';
                        break;

                    case 2: // Механік
                        $stmt2 = $conn->prepare("SELECT firstname, lastname FROM
mechanics WHERE user_id = ?");
                        $stmt2->bind_param("i", $user_id);
                        $stmt2->execute();
                        $res2 = $stmt2->get_result();
                        if ($row2 = $res2->fetch_assoc()) {
                            $name = $row2['firstname'] . ' ' . $row2['lastname'];

```

```

    }
    break;

    case 3: // Диспетчер
        $stmt3 = $conn->prepare("SELECT firstname, lastname FROM
dispatchers WHERE user_id = ?");
        $stmt3->bind_param("i", $user_id);
        $stmt3->execute();
        $res3 = $stmt3->get_result();
        if ($row3 = $res3->fetch_assoc()) {
            $name = $row3['firstname'] . ' ' . $row3['lastname'];
        }
        break;

    case 4: // Клієнт
    default:
        $stmt4 = $conn->prepare("SELECT name FROM clients WHERE
user_id = ?");

        $stmt4->bind_param("i", $user_id);
        $stmt4->execute();
        $res4 = $stmt4->get_result();
        if ($row4 = $res4->fetch_assoc()) {
            $name = $row4['name'];
        }
        break;
    }

    // Cecia
    $_SESSION['user_id'] = $user_id;
    $_SESSION['role_id'] = $role_id;
    $_SESSION['name'] = $name;

    // Перехід до кабінету
    switch ($row['role_id']) {
        case 1:
            header("Location: Admin/admin_home.php");
            break;
        case 2:
            header("Location: Mechanic/mechanic_home.php");
            break;
        case 3:
            header("Location: Dispatcher/dispatcher_home.php");
            break;
        case 4:

```

```

        default:
            header("Location: Client/client_home.php");
            break;
        }
        exit;

    } else {
        $_SESSION['resend_email'] = $email;
        $error = "Пошта не підтверджена. <a
href='resend_code.php'>Надіслати код повторно</a>";
    }
    } else {
        $error = "Неправильний пароль.";
    }
    } else {
        $error = "Користувача з таким email не знайдено.";
    }
}
?>

<html>
<head>
    <link rel="icon" type="image/x-icon" href="favicon.png">
    <title>Авторизація</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="page-container">
        <div class="logo">
            
        </div>

        <div class="login-container">

            <?php if (!empty($error)): ?>
                <div class="form-error"><?= $error ?></div>
            <?php endif; ?>

            <?php if ($success == '1'): ?>
                <div class="success-message">Вашу пошту підтверджено! Тепер
можете увійти 🐼 </div>
            <?php endif; ?>

```

```

        <?php if ($logged_out): ?>
            <div class="success-message">Ви вийшли з системи. До зустрічі!
    ✎ </div>
        <?php endif; ?>

        <h2>Авторизація</h2>
        <form action="index.php" method="POST">
            <input type="email" name="email" placeholder="Введіть email"
required>
            <input type="password" name="password" placeholder="Введіть
пароль" required>
            <button type="submit">Увійти</button>
        </form>

        <div class="forgot-password">
            <a href="forgot-password.php">Забули пароль?</a>
        </div>

        <p class="no-account">Не маєте акаунта?</p>
        <a href="registration.php">
            <button type="button" class="secondary">Реєстрація</button>
        </a>
    </div>
</div>
</body>

</html>

```

## Д.2. forgot-password.php

```

<?php
session_start();
require_once "db.php";
require_once "send_reset_email.php"; // нова функція надсилання

$message = "";

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $email = trim($_POST['email']);

    $stmt = mysqli_prepare($conn, "SELECT id FROM users WHERE email =
?");
    mysqli_stmt_bind_param($stmt, "s", $email);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

```

```

if ($user = mysqli_fetch_assoc($result)) {
    $code = bin2hex(random_bytes(16)); // 32-символьний токен
    $update = mysqli_prepare($conn, "UPDATE users SET reset_code = ?
WHERE id = ?");
    mysqli_stmt_bind_param($update, "si", $code, $user['id']);
    mysqli_stmt_execute($update);

    $sent = sendResetEmail($email, $code); //

    $message = $sent
        ? "Інструкції надіслано на пошту."
        : "Сталася помилка при відправці листа.";
    } else {
        // навмисно однакове повідомлення з міркувань безпеки
        $message = "Інструкції надіслано на пошту.";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Відновлення паролю</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
<div class="page-container">
    <div class="login-container">
        <h2>Відновлення паролю</h2>
        <?php if ($message): ?>
            <div class="success-message"><?= $message ?></div>
        <?php else: ?>
            <form method="POST">
                <input type="email" name="email" placeholder="Ваш email"
required>
                <button type="submit">Надіслати посилання</button>
            </form>
        <?php endif; ?>
        <a href="index.php">← Назад до авторизації</a>
    </div>
</div>
</body>
</html>

```