

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Розробка системи опрацювання даних з використанням технологій
React та удосконаленням РБД технологією Firebase на замовлення ТОВ
Газмережі

Виконав(ла): студент(ка) VI курсу, групи СПМ-61
спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

	(підпис)	Соловій Т.Б. (прізвище та ініціали)
Керівник	(підпис)	Цуприк Г.Б. (прізвище та ініціали)
Нормоконтроль	(підпис)	Стоянов Ю.М. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Петрик М.Р. (прізвище та ініціали)
Рецензент	(підпис)	Стадник Н.Б. (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

« »

20__ р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту Соловію Тарасу Богдановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи опрацювання даних з використанням технологій React та удосконаленням РБД технологією Firebase на замовлення ТОВ Газмережі

Керівник роботи Цуприк Галина Богданівна к.т.н.доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » 11 2025 року № 4/7-1045

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Мова програмування JavaScript, компонентний фреймворк React.js, хмарна платформа Firebase, середовище розробки Visual Studio Code, бібліотеки React Router, SweetAlert2, Lucide Icons для інтерфейсних елементів, система контролю версій Github.

4. Зміст роботи (перелік питань, які потрібно розробити)
проаналізувати предметну область та визначити основні бізнес-процеси ТОВ «Газмережі», що потребують автоматизації; сформулювати цілі, задачі та вимоги до інформаційної системи замовника; визначити користувачів системи та описати основні варіанти використання; спроектувати архітектуру системи та структуру бази даних; побудувати UML-діаграми для відображення логіки роботи системи; розробити інтерфейс та програмні модулі застосунку; реалізувати ключові функції системи, включаючи авторизацію, роботу з даними та ролями.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Презентація на 15 слайдів з розкриттям теми і поясненням.

6. Консультанти розділів роботи

[illegible]

7. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

(прізвище та ініціали)

Керівник роботи

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Розробка системи опрацювання даних з використанням технологій React та удосконаленням РБД технологією Firebase на замовлення ТОВ Газмережі. Кваліфікаційна робота освітнього ступеня «магістр». Тернопільський національний технічний університет, кафедра програмної інженерії, група СПМ-61, спеціальність 121 «Інженерія програмного забезпечення». Тернопіль, 2025. Сторінок 73, рисунків 29, таблиць 12, додатків 2, бібліографічних посилань 32.

Метою роботи є створення веб-системи для збору, зберігання та аналітичної обробки даних газопостачального підприємства з використанням сучасних вебтехнологій.

Об'єктом дослідження – є процес створення веб-системи основною функцією якої є робота з великими об'ємами даних з використанням сучасних інформаційних технологій.

Предмет дослідження – методи та інструменти побудови веб-систем управління даними на основі React, Firebase та модулів штучного інтелекту.

Методи дослідження включають системний аналіз, моделювання бізнес-процесів, об'єктноорієнтоване проєктування та Agile-підхід.

У роботі виконано аналіз предметної області, розроблено архітектуру системи, структуру бази даних, реалізовано клієнтську та серверну частини, інтегровано модуль ШІ для автоматизованої аналітики.

Наукова новизна полягає у створенні адаптивної веб-системи з інтегрованим ШІ-модулем, що забезпечує автоматичну обробку даних та інтелектуальну підтримку користувачів.

Практична значущість полягає у можливості впровадження системи на підприємствах газового сектору для підвищення ефективності обліку, аналізу та візуалізації даних.

Ключові слова: вебсистема, газопостачання, управління даними, React, Firebase, штучний інтелект, аналітика..

ABSTRACT

Development of a data processing system using React technologies and Firebase technology to improve RDB, commissioned by Gazmeri LLC. Master's degree thesis. Ternopil National Technical University, Department of Software Engineering, group SPm-61, specialty 121 “Software Engineering.” Ternopil, 2025. 73 pages, 29 figures, 12 tables, 2 appendices, 32 bibliographical references.

The purpose of the thesis is to develop a web-based system for collecting, storing, and analyzing operational data of a gas supply company using modern web technologies.

The object of research is the process of creating a web system whose main function is to work with large volumes of data using modern information technologies.

The subject of research is the methods and tools for building data management web systems based on React, Firebase, and integrated artificial intelligence modules.

The applied research methods include system analysis, business process modeling, object-oriented design, and the Agile software development approach.

The work presents the analysis of the domain, system architecture design, database structure development, implementation of client-side and server-side functionality, and integration of an AI module for automated analytics.

Scientific novelty lies in creating an adaptive web-based data management system with an AI module capable of automated data processing and intelligent user support.

The practical value is the possibility of implementing the system in gas sector enterprises to improve data collection, analysis, visualization, and operational efficiency.

Keywords: web system, gas supply, data management, React, Firebase, artificial intelligence, analytics.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ...	8
1.1 Аналіз предметної області	8
1.2 Постановка завдання та цілей	11
1.3 Пошук акторів та варіантів використання	13
1.4 Опис ключових вимог замовника ТОВ «Газмережі»	15
РОЗДІЛ 2. ПРОЄКТУВАННЯ І РОЗРОБКА	19
2.1. Вибір процесу розробки та обґрунтування вибору технологій	19
2.2. Архітектурна модель системи	21
2.3. Проєктування структури бази даних і взаємозв'язків компонентів	23
2.4. Побудова UML-діаграм.....	26
2.5. Проєктування інтерфейсу користувача та логіки взаємодії.....	32
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ СИСТЕМИ	33
3.1. Структура проєкту та реалізація основних модулів	33
3.2. Програмна реалізація ключових функцій системи.....	34
3.2.1. Інтеграція модуля штучного інтелекту	44
3.4 Методика тестування системи.....	46
3.5. Демонстрація отриманих результатів	52
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	65
4.1. Охорона праці.....	65
4.2. Забезпечення безпеки життєдіяльності при роботі з ПК	67
ВИСНОВКИ	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	73
ДОДАТКИ	77

ВСТУП

В сучасних умовах цифровізації енергетичного сектору питання ефективного управління даними стає одним із ключових чинників стабільності та конкурентоспроможності підприємств. Галузь газопостачання, що забезпечує критично важливі потреби населення та промисловості, потребує системного підходу до збору, аналізу та оброблення інформації про постачання, споживання, облік та технічний стан мереж. Відсутність автоматизованих систем управління даними призводить до затримок у прийнятті рішень, дублювання інформації, помилок у звітності та зниження рівня контролю за технологічними процесами. Тому розробка сучасної вебсистеми для збору, оброблення та аналізу даних у газопостачальній компанії є актуальним завданням, яке сприятиме підвищенню ефективності роботи підприємства, забезпеченню прозорості бізнес-процесів і оперативності реагування на зміни в інфраструктурі.

Метою даної роботи є розробка та впровадження вебсистеми для збору, оброблення та аналізу даних діяльності ТОВ «Газмережі», яка забезпечить автоматизацію основних інформаційних процесів і підвищить якість управлінських рішень. Система повинна враховувати динамічні зміни вимог замовника, а також забезпечувати масштабованість і надійність завдяки використанню сучасних вебтехнологій.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Провести аналіз предметної області та дослідити існуючі інформаційні системи в галузі газопостачання;
- Визначити вимоги замовника та сформулювати технічне завдання з урахуванням можливих змін у процесі розробки;
- Спроекувати архітектуру системи та структуру бази даних, що забезпечить ефективне зберігання і доступ до інформації;
- Реалізувати клієнтську та серверну частини системи з використанням технологій React і Firebase;

- Інтегрувати модуль штучного інтелекту для автоматизованого консультування користувачів та підтримки аналітичних запитів;
- Провести тестування системи, оцінити її продуктивність, надійність та відповідність вимогам замовника.

Об'єктом дослідження є процес управління даними в газопостачальній компанії, що охоплює збір, оброблення, зберігання та аналіз інформації про об'єкти інфраструктури й клієнтів.

Предметом дослідження є методи, технології та інструменти створення веб-системи збору та оброблення даних із використанням React і Firebase, а також впровадження модулів штучного інтелекту для аналітики й інтелектуальної взаємодії з користувачами.

У процесі дослідження застосовано методи системного аналізу, моделювання бізнес-процесів, а також об'єктно-орієнтованого проєктування. Для реалізації системи використано підхід Agile, що передбачає ітераційний процес розробки з урахуванням змін вимог замовника. У програмній реалізації застосовано стек технологій React (Frontend), Firebase (Backend, Realtime Database, Auth).

Наукова новизна роботи полягає у створенні адаптивної вебсистеми управління даними з інтегрованим модулем штучного інтелекту, який динамічно реагує на зміни вимог замовника, автоматизує аналіз даних і забезпечує інтелектуальне консультування користувачів. Запропонована система реалізує новий підхід до автоматизації бізнес-процесів газопостачальних компаній через поєднання хмарних технологій, компонентної архітектури, оброблення даних у реальному часі та інтерактивної взаємодії з користувачем.

Практична значущість роботи полягає у створенні вебплатформи, яку може бути впроваджено в діяльність газопостачальних підприємств для підвищення ефективності збору та аналізу даних, контролю технічного стану мереж і поліпшення взаємодії з клієнтами.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

Перший розділ присвячений дослідженню предметної області газорозподільної сфери та визначенню основних особливостей управління даними в діяльності підприємств галузі. Аналіз існуючих підходів дозволяє окреслити ключові проблеми та обґрунтувати необхідність створення інформаційної системи. На цій основі формуються цілі та завдання роботи. Окрему увагу приділено визначенню акторів системи, сценаріїв взаємодії та вимог замовника ТОВ «Газмережі».

1.1 Аналіз предметної області

У сучасних умовах розвитку енергетичного сектору України підприємства газорозподільної галузі потребують високого рівня автоматизації процесів обліку, контролю та аналітичної обробки даних. ТОВ «Газмережі» здійснює діяльність, пов'язану із транспортуванням, розподілом та обліком природного газу для побутових і промислових споживачів [2]. У процесі роботи формується значний обсяг інформації, зокрема дані про стан лічильників, обсяги споживання, фінансові розрахунки, технічне обслуговування мереж, а також звітність для регуляторних органів.

Відсутність єдиної інформаційної системи призводить до фрагментованості даних, ускладнення внутрішніх комунікацій і підвищення ризику помилок при обробці показників. В умовах цифровізації економіки актуальною є розробка вебсистеми для збору, зберігання та аналітичної обробки інформації, що забезпечить централізований доступ до даних у реальному часі, підвищить ефективність прийняття управлінських рішень і зменшить навантаження на персонал.

Основними функціональними напрямками діяльності таких систем є:

- автоматизований збір показників лічильників і технічних даних;
- облік споживання природного газу;

- аналітична обробка інформації для формування звітів;
- моніторинг стану мереж і виявлення відхилень;
- підтримка прийняття управлінських рішень.

Управління даними обліку споживання газу може здійснюватися за допомогою різних підходів, кожен з яких має свої особливості. У таблиці 1.1 наведено порівняльну характеристику традиційного (паперового), частково автоматизованого та веб-орієнтованого підходів до організації обліку газових даних.

Таблиця 1.1 – Порівняльна характеристика підходів [3].

Підхід	Характеристика	Переваги	Недоліки
Традиційний (паперовий)	Дані фіксуються вручну, обробка здійснюється працівниками на місці	Простота впровадження, не потребує спеціального ПЗ	Висока ймовірність помилок, затримка в отриманні інформації, відсутність централізованого аналізу
Частково автоматизований	Використання Excel або локальних БД	Легка адаптація, можливість базової аналітики	Відсутність інтеграції, складність масштабування
Веб-орієнтований	Дані збираються та обробляються централізовано через веб-інтерфейс	Реальний час, доступність, зручність, аналітична обробка	Потреба у належному захисті та технічному супроводі

Як видно з таблиці 1.1, найефективнішим підходом є веб-орієнтована модель, що забезпечує комплексний облік, інтеграцію даних і можливість подальшого розширення функціональності.

На ринку присутні різноманітні системи, орієнтовані на автоматизацію процесів обліку та моніторингу в газорозподільній сфері. До найбільш поширених належать [4-6]:

- SCADA-системи (Supervisory Control and Data Acquisition), забезпечують збір технічних параметрів і диспетчеризацію об'єктів;
- ERP-рішення (наприклад, SAP for Utilities), використовуються для фінансового обліку, планування та звітності;

- CRM-платформи, підтримують взаємодію зі споживачами та облік звернень;
- Локальні розробки енергетичних компаній, орієнтовані на вузькі бізнес-процеси (зокрема, формування звітів чи виставлення рахунків).

В таблиці 1.2 наведено порівняльний аналіз ключових вище представлених рішень.

Таблиця 1.2 – Порівняльний аналіз існуючих рішень

Система	Основне призначення	Переваги	Недоліки
SAP for Utilities	ERP-платформа для енергетичних підприємств	Висока надійність, масштабованість, інтеграція з CRM	Висока вартість ліцензій, складність впровадження
SCADA-системи	Технічний моніторинг і контроль обладнання	Висока точність, автоматизація процесів	Обмежена бізнес-аналітика, потребує фахового обслуговування
GasControl Local	Вітчизняна система для обліку та звітності	Простота інтерфейсу, відповідність локальним нормам	Відсутність глибокої аналітики, слабка масштабованість

Незважаючи на значні переваги цифровізації, галузь газорозподілу стикається з низкою ризиків [7, с. 100]:

- інформаційна безпека, можливість несанкціонованого доступу до даних споживачів або порушення конфіденційності;
- надійність збереження даних, ризик втрати інформації через технічні збої або людський фактор;
- помилки при передачі показників, недостовірні дані можуть призвести до фінансових втрат;
- низький рівень цифрової грамотності персоналу, що ускладнює впровадження нових систем.

Зменшення цих ризиків можливе шляхом впровадження авторизованого доступу, хмарного зберігання, системи резервного копіювання та аудиту дій користувачів.

В результаті аналізу предметної області встановлено, що діяльність підприємств газорозподільного комплексу супроводжується значними обсягами даних, які потребують централізованої обробки та аналізу. Існуючі програмні рішення не забезпечують комплексного підходу до збору, зберігання, аналітики та візуалізації показників у межах однієї системи. Створення вебсистеми InfoGas для ТОВ «Газмережі» є доцільним, оскільки вона дозволить інтегрувати технічні, фінансові та аналітичні дані, підвищити точність розрахунків і забезпечити прозорість діяльності підприємства.

1.2 Постановка завдання та цілей

На основі аналізу предметної області визначено, що для ефективної діяльності ТОВ «Газмережі» необхідно створити єдину інтегровану вебсистему, яка дозволить автоматизувати процеси збору, оброблення та аналітичного аналізу даних, пов'язаних із роботою користувачів, лічильників і квитанцій.

Запровадження подібного інструменту сприятиме підвищенню точності обліку, скороченню часу на оброблення даних, мінімізації людського фактору та створенню єдиного інформаційного простору підприємства. Метою створення системи InfoGas є розробка вебплатформи, яка забезпечить централізоване зберігання інформації, аналітичну обробку показників споживання газу, відображення результатів у візуальній формі та надання адміністративних інструментів для управління процесами в межах діяльності підприємства.

Для досягнення зазначеної мети передбачено реалізацію таких основних завдань:

1. Забезпечити можливість авторизованого доступу адміністратора до даних усіх користувачів, перегляду та контролю стану їхніх лічильників і квитанцій. Передбачити інструменти пошуку, фільтрації, аналітики та формування статистичних звітів про загальну діяльність системи. Система повинна відображати ключові показники у вигляді діаграм і графіків, що характеризують динаміку оплат, кількість активних об'єктів та сумарні фінансові операції.

2. Реалізувати функціонал реєстрації, автентифікації та управління особистим профілем користувача. Передбачити можливість створення, редагування та видалення лічильників і квитанцій, а також формування звітів про обсяги споживання газу та оплату. Забезпечити автоматичний розрахунок сум до сплати з урахуванням пільг і субсидій, а також можливість візуалізації персональної статистики. Користувач повинен мати доступ до розділу документації підприємства та отримувати довідкову інформацію про тарифи та правила користування.

3. Розробити модуль статистичного та графічного відображення даних, який дозволяє аналізувати динаміку оплат, активність користувачів, статуси лічильників і квитанцій. Забезпечити синхронізацію даних у режимі реального часу та автоматичне оновлення показників без потреби у ручному перезавантаженні. Передбачити можливість експорту звітів у формат електронних таблиць для подальшої обробки.

4. Інтегрувати модуль штучного інтелекту, який надає користувачам роз'яснення щодо тарифів, принципів нарахування оплат і дозволяє отримувати консультації за попередньо сформульованими запитам. Цей компонент має виконувати функцію інтелектуального помічника, що сприяє підвищенню поінформованості користувачів і полегшує роботу з системою.

5. Забезпечити захищену авторизацію користувачів із розмежуванням прав доступу (адміністратор / користувач), централізоване хмарне зберігання даних і надійний механізм їх оновлення. Передбачити використання інтерактивних повідомлень для підтвердження дій і попередження помилкових операцій.

Постановка завдання охоплює комплекс робіт, спрямованих на створення сучасної вебсистеми, яка поєднує облік, аналітику та управління даними діяльності ТОВ «Газмережі».

Розроблена система InfoGas має забезпечити прозорість процесів, підвищення якості управлінських рішень, спрощення взаємодії між користувачами та адміністраторами, а також надання аналітичної підтримки для подальшого вдосконалення діяльності підприємства.

1.3 Пошук акторів та варіантів використання

На етапі постановки завдання визначено, що вебсистема InfoGas має дві основні ролі користувачів (акторів): адміністратор та звичайний користувач. Кожен із них виконує власні дії, які формують окремі сценарії використання системи. Пошук і класифікація акторів є важливим етапом моделювання, оскільки дозволяють структурувати функціональні вимоги та визначити межі взаємодії між користувачем і системою.

В системі передбачені наступні актори:

1. Адміністратор системи, користувач із розширеними правами доступу, який відповідає за моніторинг усіх користувачів, контроль за лічильниками та квитанціями, а також управління аналітичними показниками системи.
2. Звичайний користувач (клієнт), особа, що використовує систему для ведення власних даних: реєстрації, обліку лічильників, створення квитанцій, перегляду звітів та отримання консультацій від інтегрованого ШІ-модуля.
3. Система Firestore / Firebase (непрямий актор), виконує роль сховища даних і сервісу автентифікації, взаємодіючи з користувачами через інтерфейс застосунку.

Для опису функціональних можливостей кожного актора було сформовано таблицю 1.3, у якій наведено основні сценарії взаємодії користувачів із системою.

Таблиця 1.3 – Актори та варіанти використання системи InfoGas

№	Актор	Варіант використання	Короткий опис дії
1	Адміністратор	Авторизація в системі	Вхід до панелі керування з використанням облікових даних Firebase Authentication.
2	Адміністратор	Перегляд списку користувачів	Отримання переліку всіх зареєстрованих користувачів із можливістю сортування, пошуку та фільтрації.
3	Адміністратор	Перегляд детальної інформації користувача	Перехід на сторінку користувача, відображення його лічильників, квитанцій, загальної суми оплат та дати останніх операцій.
4	Адміністратор	Зміна статусу лічильника	Активування або деактивування лічильника безпосередньо в інтерфейсі.

Продовження таблиці 1.3

5	Адміністратор	Підтвердження квитанцій	Зміна статусу квитанції з “не підтверджено” на “підтверджено”.
6	Адміністратор	Аналіз статистики системи	Перегляд діаграм і графіків (кількість користувачів, стан лічильників, оплати).
7	Адміністратор	Вихід із системи	Завершення сеансу роботи адміністратора.
8	Користувач	Реєстрація в системі	Створення облікового запису через Firebase Authentication.
9	Користувач	Вхід / вихід із системи	Авторизація для доступу до персонального кабінету та завершення сеансу.
10	Користувач	Редагування профілю	Зміна персональних даних (ПІБ, пошта, телефон, адреса, пароль).
11	Користувач	Видалення профілю	Видалення облікового запису разом із пов’язаними лічильниками та квитанціями.
12	Користувач	Створення лічильника	Внесення даних про номер, тип, адресу, тариф і наявність пільг або субсидій.
13	Користувач	Пошук і фільтрація лічильників	Пошук за номером, типом, адресою, статусом або тарифом.
14	Користувач	Видалення лічильника	Повне видалення запису з бази даних.
15	Користувач	Створення квитанції	Заповнення форми (вибір лічильника, обсяг газу, автоматичний розрахунок суми).
16	Користувач	Перегляд та фільтрація квитанцій	Сортування за статусом, датою, обсягом споживання чи сумою до сплати.
17	Користувач	Перегляд аналітичних звітів	Відображення статистики (кількість лічильників, квитанцій, загальна сума оплат, графіки).
18	Користувач	Експорт звітів у Excel	Завантаження сформованих фінансових звітів у форматі таблиць.
19	Користувач	Використання ШІ-модуля	Отримання відповідей на типові питання про тарифи та власні дані.
20	Користувач	Перегляд документації	Ознайомлення з матеріалами про діяльність ТОВ «Газмережі» та правила користування системою.

Для кращого розуміння логіки роботи акторів на етапі проєктування були побудовані діаграми варіантів використання (Use Case Diagram) для кожної ролі користувача, що відображає основні зв’язки між користувачами та підсистемами (Додаток А).

Проведений аналіз акторів і варіантів використання показав, що система InfoGas має чітко структуровану архітектуру з двома рівнями взаємодії: адміністративним і користувацьким. Кожен актор виконує набір функцій, що взаємопов’язані через спільну базу даних і аналітичні модулі. Завдяки цьому

система забезпечує повний цикл управління інформацією, від введення та оброблення до візуалізації та прийняття управлінських рішень.

1.4 Опис ключових вимог замовника ТОВ «Газмережі»

У результаті аналізу вимог замовника ТОВ «Газмережі» було визначено, що основною метою створення вебсистеми InfoGas є автоматизація процесів збору, оброблення та аналітики даних щодо використання природного газу, спрощення взаємодії між користувачами і адміністратором компанії, а також підвищення ефективності контролю за станом лічильників і розрахунками споживачів.

Нижче наведено основні функціональні та технічні вимоги до системи.

1. Адміністративна частина, загальна мета: забезпечити повноцінну роботу адміністратора з користувацькими даними, лічильниками, квитанціями та аналітичною інформацією у зручному інтерфейсі.

1.1. Авторизація адміністратора

- Система має підтримувати авторизований доступ адміністратора через безпечну автентифікацію Firebase Authentication.
- Усі дії адміністратора фіксуються в базі даних із можливістю подальшого аудиту.

1.2. Робота з користувачами

- Надати можливість перегляду повного списку користувачів із бази даних.
- Реалізувати пошук і фільтрацію користувачів за полями: ПІБ, електронна пошта, номер телефону, адреса.
- Додатково, фільтрацію за активністю пов'язаних об'єктів (лічильників та квитанцій).
- Передбачити перехід на сторінку детального перегляду користувача.

1.3. Керування лічильниками та квитанціями, адміністратор повинен мати змогу:

- переглядати усі лічильники користувачів;
- змінювати статус лічильника (активний / неактивний);
- переглядати усі квитанції користувачів;
- підтверджувати квитанції (зміна статусу на «підтверджено»);

Кожна операція підтверджується інтерактивним повідомленням (SweetAlert).

1.4. Аналітична панель (Dashboard), система повинна відображати інтегровану статистику в реальному часі, що включає:

- кількість зареєстрованих користувачів;
- кількість активних / неактивних лічильників;
- кількість підтверджених / непідтверджених квитанцій;
- загальну суму оплат користувачів.

Дані мають візуалізуватись у вигляді:

- кругових діаграм розподілу статусів лічильників та квитанцій;
- лінійного графіка динаміки оплат у часі;
- гістограми «Топ користувачів» за сумою оплат.

2. Користувацька частина, загальна мета: надати користувачам можливість обліку власних лічильників і квитанцій, перегляду фінансової статистики та використання інтелектуальних функцій у єдиній системі.

2.1. Реєстрація та автентифікація, реалізувати механізм реєстрації, входу та виходу користувача через Firebase Authentication.

Під час реєстрації користувач вводить такі дані: ПІБ, електронну пошту, номер телефону, адресу, пароль. Для кожного користувача система генерує унікальний номер рахунку (наприклад, «WB-391»).

2.2. Профіль користувача

- Надати можливість перегляду та редагування персональних даних: ПІБ, пошта, телефон, адреса, пароль.
- Реалізувати функцію повного видалення профілю користувача з усіма пов'язаними лічильниками та квитанціями.

2.3. Робота з лічильниками, користувач може створювати нові лічильники, вказуючи:

- номер лічильника;
- тип (однотарифний, багатотарифний, промисловий, побутовий);
- адресу встановлення;
- наявність пільг та/або субсидій;
- тариф (грн/м³).

Передбачити можливість перегляду, фільтрації та видалення лічильників. Новий лічильник за замовчуванням має статус «неактивний», який може змінити лише адміністратор.

2.4. Робота з квитанціями, користувач створює квитанцію, вибираючи один зі своїх лічильників. Вводить обсяг спожитого газу (м³), після чого система автоматично обчислює суму до сплати з урахуванням пільг та субсидій. Квитанція отримує статус «не підтверджено» до моменту перевірки адміністратора. Користувач може переглядати, фільтрувати або видаляти квитанції.

2.5. Звіти та аналітика, надати користувачеві підсумкову статистику:

- кількість створених лічильників і квитанцій;
- загальну суму оплат;
- графіки динаміки оплат і розподілу за статусами.
- Реалізувати функцію експорту звітів у формат Excel.

2.6. Інтелектуальний модуль, забезпечити інтеграцію з модулем штучного інтелекту, який відповідає на підготовлені запитання користувачів щодо тарифів, оплат, умов обслуговування. Система має враховувати дані конкретного користувача при формуванні відповіді.

2.7. Документація, створити сторінку з довідковими матеріалами, тарифами, умовами користування та контактами компанії. Передбачити головну сторінку з коротким описом сервісу InfoGas та посиланням на офіційний сайт ТОВ «Газмережі».

3. Взаємодія системи та аналітика

3.1. Зберігання даних, всі дані мають зберігатися в хмарній базі Firebase Firestore, що забезпечує:

- синхронізацію в реальному часі;
- відсутність потреби у перезавантаженні сторінки для оновлення даних;
- надійність та масштабованість.

3.2. Безпека та рольовий доступ, визначено дві ролі користувачів:

- Адміністратор – повний доступ до системи;
- Користувач – лише до власних даних.

Доступ обмежується через токен Firebase Authentication.

3.3. Інтерактивність, всі ключові операції (створення, видалення, підтвердження, редагування) супроводжуються інтерактивними сповіщеннями (SweetAlert2) із коротким таймером та візуальними підтвердженнями. Зміни у базі даних відображаються динамічно без перезавантаження сторінки.

Ключові вимоги замовника ТОВ «Газмережі» формують комплексну основу для розробки вебсистеми InfoGas, яка поєднує функції обліку, моніторингу, аналітики та штучного інтелекту. Визначені модулі, адміністративний, користувацький та аналітичний, забезпечують повний цикл роботи з даними споживачів природного газу: від введення інформації до аналітичного опрацювання. Впровадження системи дозволить замовнику підвищити прозорість обліку, зменшити кількість ручних операцій та оптимізувати управління клієнтською базою.

РОЗДІЛ 2. ПРОЄКТУВАННЯ І РОЗРОБКА

Другий розділ зосереджений на питаннях проєктування вебсистеми InfoGas та виборі технологічних рішень. Розглядаються архітектурні підходи, структура бази даних і взаємодія основних компонентів системи. Для формалізації проєктних рішень використовуються UML-діаграми. Також описуються принципи побудови інтерфейсу користувача та логіки взаємодії із системою.

2.1. Вибір процесу розробки та обґрунтування вибору технологій

Процес створення вебсистеми InfoGas базувався на гнучкій методології розробки Agile з елементами підходу Scrum, що дало змогу забезпечити адаптивність до змін вимог замовника ТОВ «Газмережі» та поступове розширення функціональності. Основною причиною вибору саме Agile-підходу є його ітераційний характер, який дозволяє отримувати проміжні результати після кожного спринту, тестувати функції у реальному середовищі та вчасно коригувати технічні рішення [8].

Розробку системи було поділено на окремі фази:

- Аналіз предметної області – вивчення процесів газопостачання, моделі взаємодії користувачів і адміністраторів, визначення вимог до безпеки та аналітики.
- Проєктування архітектури – побудова логічної структури бази даних, моделі користувачів і ролей, створення прототипів інтерфейсу.
- Реалізація клієнтської частини (Front-End) – розроблення компонентів React, маршрутизації, інтеграції з Firebase.
- Розробка серверної логіки та взаємодії з базою – створення CRUD-операцій через Firebase SDK, реалізація аналітичних запитів.
- Тестування та впровадження – відлагодження функцій, перевірка безпеки доступу, перевірка коректності візуалізацій даних.

Для розроблення клієнтської частини системи було використано сучасний стек вебтехнологій, що забезпечує швидкодію, інтерактивність і привабливий користувацький інтерфейс. Основним фреймворком є React.js, який реалізує компонентно-орієнтований підхід до побудови інтерфейсу. Завдяки використанню віртуального DOM React забезпечує високу продуктивність оновлення елементів сторінки та спрощує розробку складних динамічних структур. Компонентна архітектура полегшує повторне використання елементів і масштабування системи [9].

Для стилізації інтерфейсу застосовано Tailwind CSS, що дає змогу створювати сучасний, адаптивний дизайн без написання великих обсягів CSS-коду. Використання утилітарних класів скоротило час верстки та забезпечило єдину стилістичну узгодженість інтерфейсу [10]. Для побудови аналітичних діаграм і візуалізації даних використано Recharts, що підтримує побудову кругових, лінійних і стовпчикових графіків [11]. Окрему увагу приділено інтерактивності та покращенню користувацького досвіду. Для цього у проєкті застосовано бібліотеку SweetAlert2, яка використовується для створення діалогових вікон, підтверджень дій і коротких повідомлень (toast-сповіщень) [12].

Серверна частина системи реалізована на основі хмарної платформи Firebase, яка поєднує в собі інструменти для зберігання даних, автентифікації користувачів і синхронізації у реальному часі. Основним сховищем даних є Firebase Firestore документно-орієнтована NoSQL база, яка забезпечує масштабоване та ефективне зберігання інформації про користувачів, лічильники та квитанції [13]. Firestore підтримує оновлення даних у реальному часі, що дозволяє адміністраторам та користувачам бачити актуальні зміни без перезавантаження сторінки. Така технологія є оптимальною для систем із великою кількістю взаємодій та високою частотою оновлень. Для безпечного входу в систему застосовано Firebase Authentication, який підтримує рольову модель доступу [14].

Для реалізації модуля інтелектуальної підтримки користувачів у системі InfoGas інтегровано Poe API, платформу, що надає доступ до сучасних мовних

моделей штучного інтелекту [15]. Цей модуль використовується для автоматизованого консультування користувачів.

Застосування сучасного технологічного стеку React + Firebase у поєднанні з інструментами аналітики та штучного інтелекту дало змогу створити ефективну, надійну та масштабовану систему. Обрані технології забезпечують високу швидкодію, автоматичне оновлення даних у реальному часі, безпечну авторизацію та гнучкість у подальшому розширенні функціоналу.

2.2. Архітектурна модель системи

Архітектурна модель вебсистеми InfoGas побудована за принципом багаторівневої клієнт–серверної архітектури, що забезпечує чітке розмежування між рівнями представлення, логіки та даних. Такий підхід сприяє масштабованості, гнучкості та простоті супроводу системи, дозволяючи ефективно реалізувати нові функціональні модулі без втручання у вже наявні компоненти.

Загальна структурна схема архітектури зображена на рисунку 2.1, який ілюструє взаємодію основних компонентів між собою.

На зображенні показано три основні рівні архітектури: Front-End (React.js), Application Layer (Firebase Cloud SDK) та Data Layer (Firebase).

Кожен із них виконує власні функції у процесі роботи системи: від обробки користувацьких дій до зберігання й аналізу інформації.

Фронтендна частина реалізує компонентно-орієнтований підхід до формування інтерфейсу користувача. Інтерактивність та реактивне оновлення елементів забезпечуються за допомогою Hooks та Context API, які використовуються для управління станом застосунку [16-17].

Рівень бізнес-логіки реалізовано через Firebase Cloud SDK, який виступає проміжною ланкою між інтерфейсом і базою даних. На цьому рівні обробляються основні бізнес-процеси: валідація даних, обчислення вартості спожитого газу, зміна статусів квитанцій і лічильників, а також побудова аналітичних показників. До складу прикладного рівня входить аналітичний модуль (Analytics Module), який

відповідає за формування зведених даних, звітів і графіків, а також взаємодіє з Рое API для реалізації інтелектуального консультування користувачів.

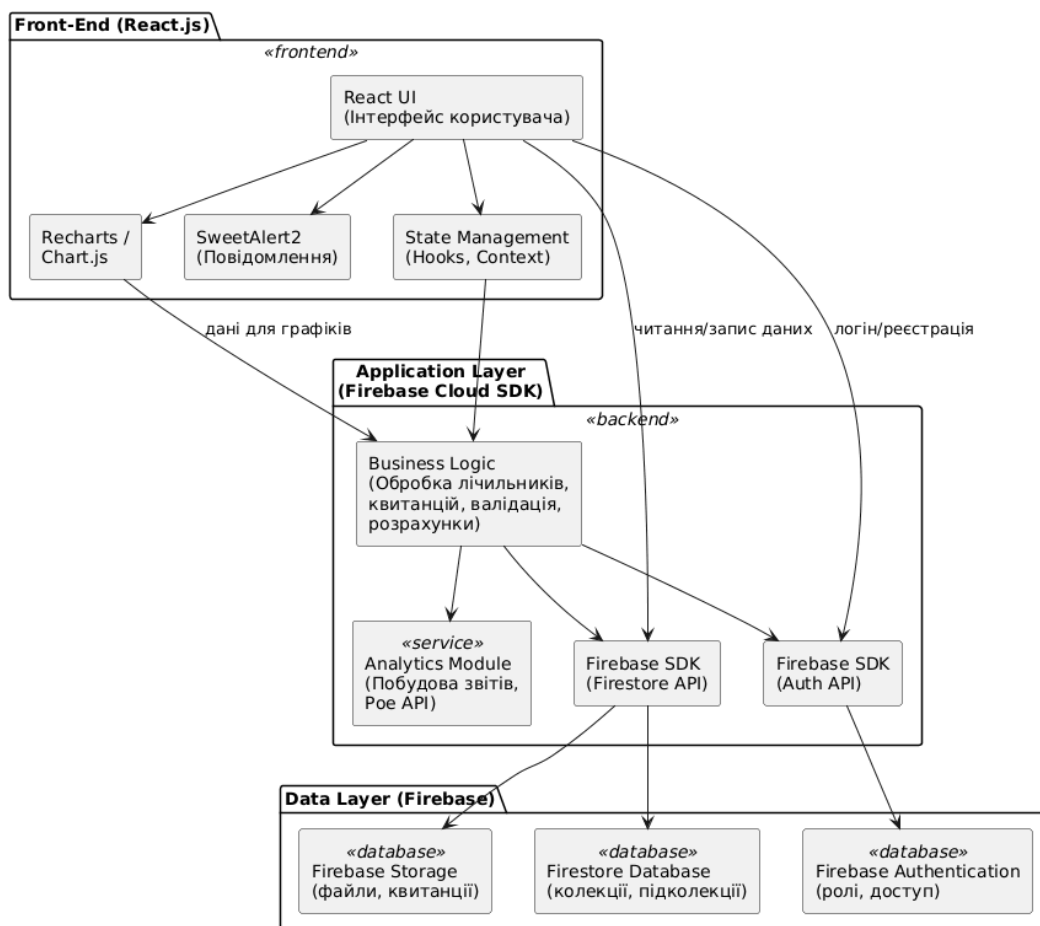


Рисунок 2.1 – Архітектурна модель вебсистеми InfoGas

Рівень даних забезпечується інфраструктурою Firebase, що включає в себе кілька сервісів: Firestore Database, основне сховище даних, що підтримує оновлення в реальному часі; Firebase Authentication, модуль автентифікації, який відповідає за безпечний вхід користувачів і розмежування прав доступу;

Ці компоненти працюють у тісній взаємодії з прикладним рівнем через API-запити Firebase SDK, забезпечуючи централізоване управління інформаційними потоками.

Обмін даними між рівнями здійснюється асинхронно. Користувацькі дії з інтерфейсу React (створення лічильників, генерація квитанцій, зміна профілю тощо) передаються до прикладного рівня через Firebase SDK, де відбувається

обробка логіки та подальший запис до бази даних. Зміни, що відбуваються в Firestore, автоматично оновлюються у фронтенді завдяки механізму підписки (onSnapshot), що забезпечує реактивність і миттєве відображення актуального стану системи без перезавантаження сторінки. Водночас модуль аналітики отримує дані з рівня логіки для формування статистики, графіків і звітів, а результати виводяться безпосередньо у React UI.

Архітектурна модель системи InfoGas побудована за принципом модульності, що дозволяє ефективно масштабувати застосунок, розширювати функціонал та інтегрувати зовнішні сервіси.

2.3. Проектування структури бази даних і взаємозв'язків компонентів

Для зберігання даних у вебсистемі InfoGas використано хмарну базу даних Firebase Firestore, що належить до класу NoSQL документно-орієнтованих систем [18]. Вона забезпечує масштабованість, високу швидкодію, а також підтримує синхронізацію в реальному часі, що є критично важливим для інтерактивних систем моніторингу.

Попри те, що Firestore є нереляційною базою, її логічну структуру можна подати у вигляді ER-діаграми, яка демонструє взаємозв'язки між ключовими сутностями: users, counters та receipts (рис. 2.2).

Колекція users є центральною в системі та містить інформацію про всіх зареєстрованих користувачів. Кожен документ у цій колекції відповідає одному користувачеві, який може мати декілька лічильників і квитанцій.

Основні поля:

- `accountNumber (string)` – унікальний номер облікового запису користувача;
- `address (string)` – поштова адреса проживання користувача;
- `createdAt (timestamp)` – дата й час створення облікового запису;
- `email (string)` – електронна адреса для входу до системи;

- fullName (string) – повне ім'я користувача (ПІБ);
- phone (string) – контактний номер телефону.

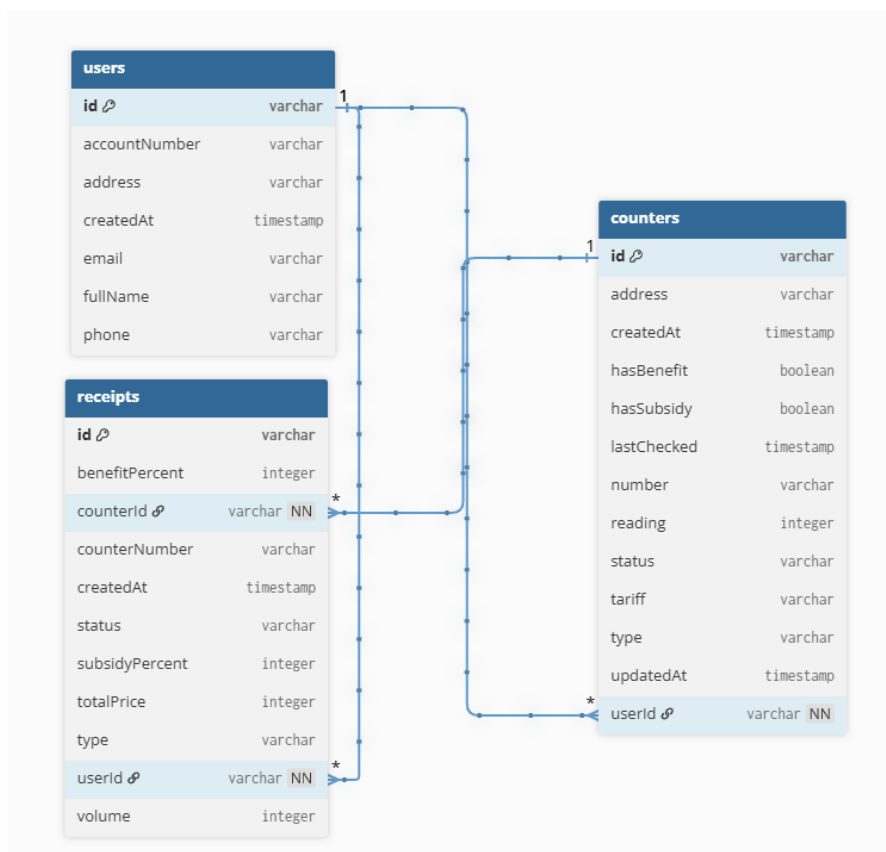


Рис. 2.2 – Логічна модель структури бази даних вебсистеми InfoGas

Колекція counters містить дані про лічильники газу, зареєстровані користувачами. Вона має зв'язок багато-до-одного з колекцією users, оскільки кожен користувач може мати кілька лічильників.

Основні поля:

- address (string) – місце встановлення лічильника;
- createdAt (timestamp) – дата реєстрації лічильника;
- hasBenefit (boolean) – наявність пільг у користувача;
- hasSubsidy (boolean) – наявність субсидії;
- lastChecked (timestamp / null) – дата останньої перевірки або null;
- number (string) – унікальний номер лічильника;
- reading (number) – поточні показники лічильника;

- status (string) – стан лічильника (активний / неактивний);
- tariff (string) – тариф за 1 м³ газу;
- type (string) – тип лічильника (однотарифний, багатотарифний тощо);
- updatedAt (timestamp) – час останнього оновлення даних;
- userId (string) – ідентифікатор користувача, з яким пов'язаний лічильник.

Ця колекція дозволяє відстежувати стан і параметри кожного пристрою обліку газу, а також визначати, чи підпадає користувач під пільгову категорію.

Колекція receipts містить інформацію про сформовані квитанції користувачів. Кожна квитанція пов'язана одночасно з конкретним користувачем і його лічильником, формуючи зв'язки типу багато-до-одного з обома сутностями.

Основні поля:

- benefitPercent (number) – відсоток знижки за пільгою;
- counterId (string) – ідентифікатор лічильника, до якого належить квитанція;
- counterNumber (string) – номер лічильника для відображення у звіті;
- createdAt (timestamp) – дата створення квитанції;
- status (string) – стан квитанції (підтверджено / не підтверджено);
- subsidyPercent (number) – відсоток субсидії;
- totalPrice (number) – сума до сплати;
- type (string) – тип лічильника, на основі якого створено квитанцію;
- userId (string) – ідентифікатор користувача;
- volume (number) – обсяг спожитого газу в м³.

Колекція receipts є основою для формування фінансових звітів, обчислення динаміки споживання та побудови графіків аналітики.

Зв'язки між колекціями мають такий вигляд: один користувач (users) може мати кілька лічильників (counters); кожен лічильник (counters) може бути пов'язаний із кількома квитанціями (receipts); кожна квитанція (receipts) належить одному користувачу та одному лічильнику.

2.4. Побудова UML-діаграм

Для формального опису структури, логіки роботи та поведінки вебсистеми InfoGas застосовано уніфіковану мову моделювання UML (Unified Modeling Language). UML-діаграми забезпечують візуальне подання взаємозв'язків між компонентами, послідовностей обміну даними та станів об'єктів у різних частинах програмного комплексу [19].

У межах проєкту InfoGas було розроблено чотири основні UML-діаграми:

- Діаграму класів;
- Діаграму послідовності;
- Діаграму станів;
- Діаграму розгортання.

Послідовність їх розміщення у тексті є логічною: від опису структурної моделі (класи), через динамічну взаємодію об'єктів (послідовність), зміни станів об'єктів (стати), до фізичної архітектури системи (розгортання).

Діаграма класів відображає статичну структуру системи, її основні сутності, атрибути, методи та зв'язки між об'єктами [20]. На рис. 2.3 подано UML-діаграму класів вебсистеми InfoGas.

Вона містить такі основні класи:

- User – зберігає персональні дані користувача: id, fullName, email, address, accountNumber;
- Counter – описує лічильники користувача (number, type, tariff, status, hasBenefit, hasSubsidy);
- Receipt – моделює квитанції (counterNumber, volume, totalPrice, status);
- FirestoreService – відповідає за обмін даними з базою (завантаження й збереження лічильників і квитанцій);
- AuthService – виконує функції автентифікації (вхід, вихід, отримання користувача);

- AIService – реалізує звернення до модуля штучного інтелекту (аналіз питань через Poe API).

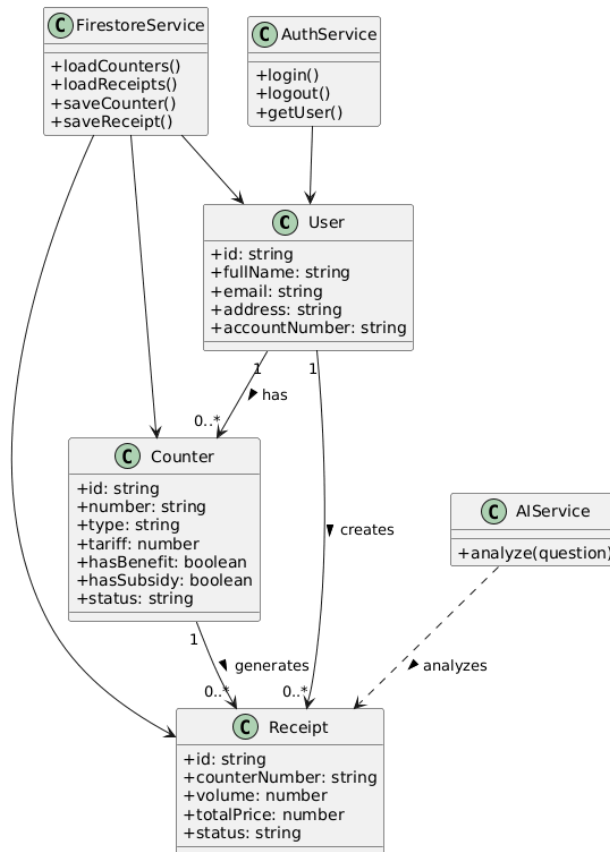


Рис. 2.3 – Діаграма класів системи InfoGas

Діаграма послідовності демонструє динамічну взаємодію об'єктів у часі, як відбувається обмін повідомленнями між компонентами системи під час виконання певного сценарію [21].

На рисунку 2.4 зображено сценарій взаємодії користувача з модулем штучного інтелекту.

Користувач через React UI (AIAnalysis Component) вводить запитання, після чого відбувається:

1. Надсилання POST-запиту /api/ai-analyze до Express Server;
2. Формування промпту, який включає контекст і запит користувача;
3. Передача запиту до Poe API (OpenAI-сумісного);
4. Отримання відповіді у форматі JSON;

5. Відображення результату у React-інтерфейсі.

Діаграма ілюструє типовий цикл запиту-відповіді між клієнтським інтерфейсом, сервером і зовнішнім API.

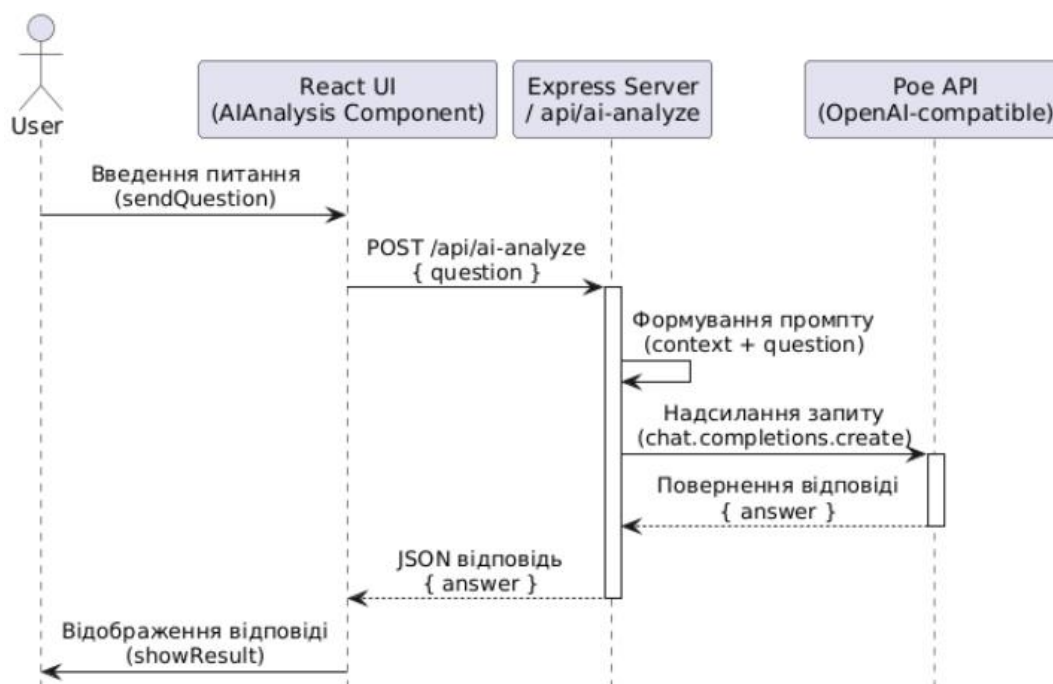


Рис. 2.4 – Діаграма послідовності взаємодії користувача з ШІ модулем

Діаграма станів (State Machine Diagram) описує життєвий цикл об'єктів системи, зокрема перехід між різними станами в процесі їх використання [22].

Для InfoGas такі діаграми доцільно побудувати для об'єктів лічильника та квитанції, оскільки вони мають чітко визначені етапи існування (рисю 3.5).

Лічильник, після створення користувачем, переходить у початковий стан «створений», у якому він ще не бере участі в обчисленнях і лише очікує перевірки адміністратора. Після затвердження він набуває статусу «активний» і може використовуватися у всіх розрахунках, що проводяться системою. У разі потреби, як з боку користувача, так і адміністратора, лічильник може бути переведений у стан «неактивний», що тимчасово виключає його з обробки даних, або у стан «тимчасово відключений», що зазвичай пов'язано з технічними збоями чи помилками показників. Завершальною стадією є «видалений», коли об'єкт остаточно виключається з системи.

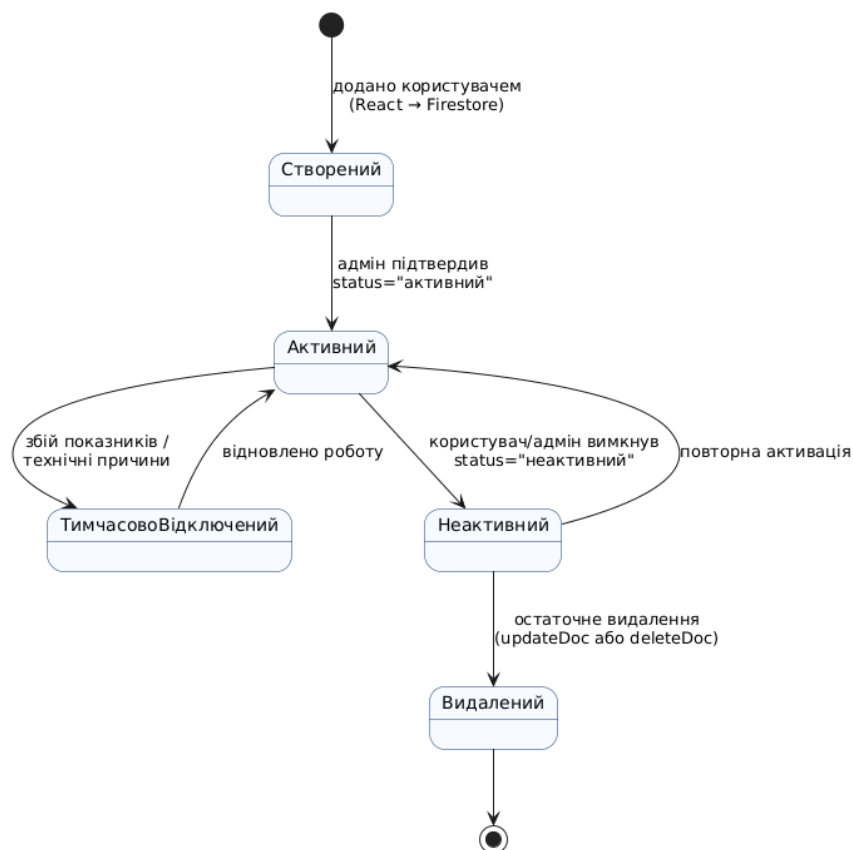


Рис. 2.5 – Діаграма станів об'єкта «Лічильник»

На рис. 2.6 представлено діаграму станів об'єкта Receipt (Квитанція), життєвий цикл якого також передбачає низку логічно впорядкованих етапів.

Спочатку квитанція існує у вигляді «чернетки», коли дані тільки вводяться та ще не збережені. Після успішної валідації вона переходить у стан «створена», тобто записи фіксуються у Firestore.

Далі квитанція очікує перевірки адміністратора, це стан «очікує підтвердження», під час якого може бути або погоджена, або відхилена залежно від коректності даних. Якщо адміністратор затверджує квитанцію, вона переходить у стан «підтверджена», після чого користувач має можливість здійснити оплату, відповідно, змінюється стан на «оплачена». Кінцевим етапом є «архівована», коли завершено звітний період, і квитанція переноситься до архіву.

Такий життєвий цикл квитанції дозволяє забезпечити контроль коректності введених даних, прозорість фінансових операцій та чітке відстеження стану кожної квитанції на всіх етапах оброблення.



Рис. 2.6 – Діаграма станів об'єкта «Лічильник»

Діаграма розгортання (Deployment Diagram) описує фізичну архітектуру системи, тобто як компоненти InfoGas розміщені в середовищі виконання та які між ними канали зв'язку [23]. На рис. 2.7 подано структуру розгортання, яка демонструє взаємодію між фізичними компонентами програмної інфраструктури.

Фронтенд на React.js працює у веббраузері користувача та здійснює запити до хмарних сервісів Firebase, які забезпечують зберігання даних, автентифікацію й обробку бізнес-логіки.

Окремий Express-сервер виступає проміжною ланкою для AI-аналізу, перенаправляючи запити з клієнта до моделі FreeGPT-4 через Poe API. Усі елементи системи обмінюються даними через захищені HTTPS-з'єднання. Представлені UML-діаграми охоплюють усі ключові аспекти функціонування вебсистеми InfoGas.

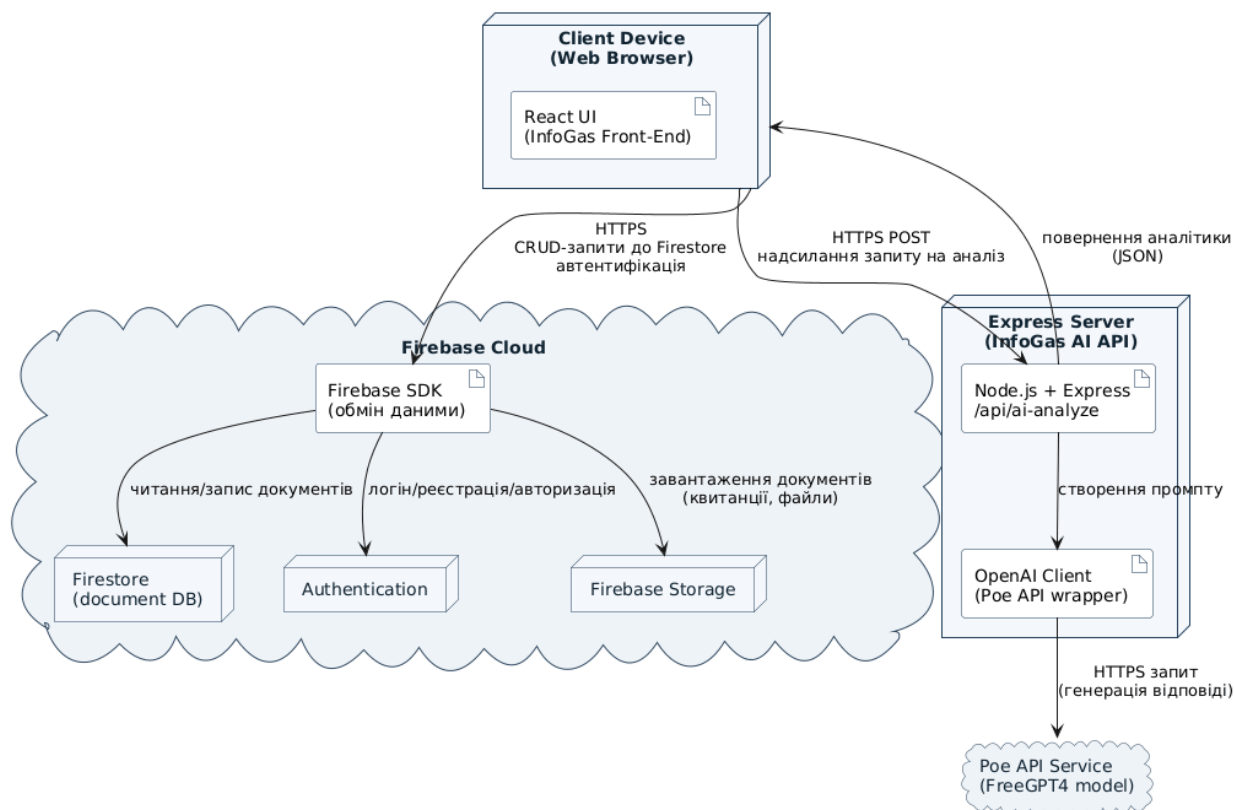


Рис. 2.7 – Діаграма розгортання вебсистеми InfoGas

Побудовані ключові UML-діаграми, комплексно відображають логіку та архітектуру вебсистеми InfoGas.

Діаграма класів структурує основні сутності та їх взаємозв'язки, забезпечуючи формальне бачення внутрішньої моделі даних. Діаграма послідовності демонструє покрокову взаємодію користувача з модулем штучного інтелекту, що дозволяє зрозуміти рух даних у системі. Діаграми станів розкривають життєвий цикл лічильників і квитанцій, відображаючи всі можливі зміни статусів. Діаграма розгортання узагальнює фізичну архітектуру, показуючи роботу системи у середовищі Firebase та зовнішніх сервісах. Сукупність цих моделей забезпечує цілісне уявлення про InfoGas, підвищує прозорість логіки, полегшує розробку, налагодження та подальше масштабування програмного комплексу. Застосування UML-діаграм дало змогу узгодити вимоги замовника з технічною реалізацією системи та зменшити ризики помилок на етапі програмування. Формалізація структури й поведінки компонентів спрощує командну розробку та забезпечує можливість подальшого розвитку і супроводу вебсистеми.

2.5. Проектування інтерфейсу користувача та логіки взаємодії

Інтерфейс вебсистеми InfoGas було спроектовано за принципами простоти, візуальної чистоти та інтуїтивності. Основою дизайну є сучасна кольорова палітра на базі градієнтів від темно-синього (#1e3a8a) до насиченого блакитного (#2563eb), що створює відчуття надійності та асоціюється з тематикою енергетичних сервісів. Додаткові відтінки білого та світло-сірого застосовані для розмежування блоків і підвищення читабельності.

Компонування інтерфейсу виконано у стилі «card-based design», де ключові дані подаються у вигляді окремих карток, а навігація реалізована через чітке верхнє меню, доступне на всіх сторінках (рис. 2.8). Використані анімації мінімальні та не відволікають користувача, зосереджуючи увагу на основному функціоналі.

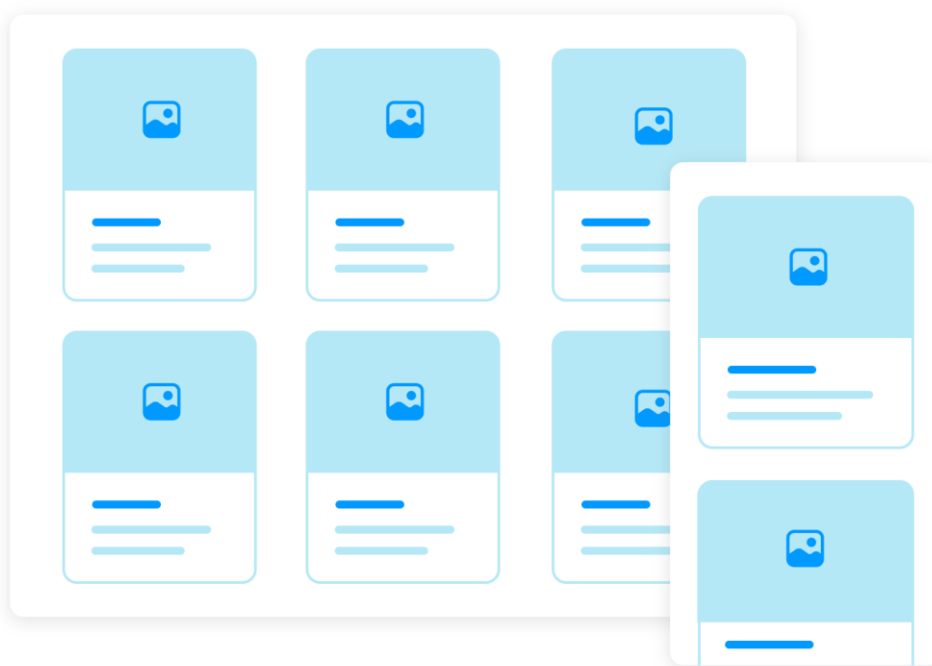


Рис. 2.8 – Приклад стилю сторінки «card-based design»

Логіка взаємодії побудована так, щоб користувач без додаткових пояснень міг додати лічильник, створити квитанцію або переглянути статистику. Кожен крок супроводжується підказками та візуальними акцентами, а підтвердження дій спливаючими повідомленнями SweetAlert2.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ СИСТЕМИ

Третій розділ присвячений реалізації функціональних можливостей вебсистеми та перевірці її працездатності. Описується структура проєкту, ключові програмні модулі та особливості інтеграції модуля штучного інтелекту. Значну увагу приділено питанням тестування та оцінювання коректності роботи системи. Завершальним етапом розділу є демонстрація отриманих результатів.

3.1. Структура проєкту та реалізація основних модулів

Структура проєкту побудована модульно: кожен компонент, сторінка та сервіс мають власну відповідальність, що забезпечує високу читабельність коду та легкість супроводу (рис. 3.1).

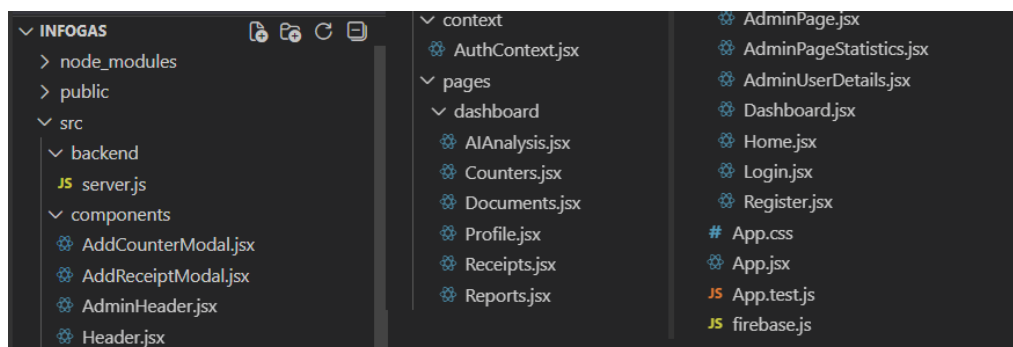


Рис. 3.1 – Структура проєкту та ключові файли

Кореневий каталог містить стандартні для React-проєкту папки `node_modules` (залежності) та `public` (статичні файли). Основний функціональний код розміщено в директорії `src`, де зосереджені всі модулі клієнтської частини, бекенд-сервер, сторінки та контекст авторизації.

Папка `backend` містить файл `server.js`, невеликий Express-сервер, що виступає проміжною ланкою між клієнтом і зовнішнім Рое API. Він обробляє запити модуля AI-аналізу та повертає готові відповіді в JSON-форматі. Така організація дозволяє винести роботу з ШІ в окремий сервіс та спростити інтеграцію з фронтом.

Каталог `components` включає багаторазові UI-компоненти системи:

- `Header.jsx` – навігаційний верхній бар для звичайного користувача;
- `AdminHeader.jsx` – окрема версія хедера для адміністратора;
- `AddCounterModal.jsx` та `AddReceiptModal.jsx` – модальні вікна для створення лічильника та квитанції відповідно.

Ці компоненти забезпечують модульність інтерфейсу та повторне використання елементів у різних частинах системи.

Папка `pages` містить повноцінні сторінки застосунку. Кожна сторінка відповідає окремому маршруту та відображає завершений функціональний блок. Директорія `pages/dashboard` об'єднує всі внутрішні сторінки користувача: профіль, лічильники, квитанції, AI-аналіз, документи та звіти. Окремо винесені `AdminPage.jsx`, `AdminUserDetails.jsx` та `AdminPageStatistics.jsx`, сторінки адміністратора, що реалізують перегляд користувачів, детальну інформацію та статистичні звіти.

Контекст авторизації (`AuthContext.jsx`) відповідає за зберігання інформації про поточного користувача та керування його станами входу/виходу. Це забезпечує централізований доступ до `Firebase Authentication` по всій системі.

Файли `App.jsx` та `App.test.js` містять логіку маршрутизації та базове налаштування застосунку. Конфігурація `Firebase` зосереджена у `firebase.js`, що спрощує взаємодію з `Firestore`, `Authentication` та іншими сервісами.

Така структура забезпечує логічне розділення відповідальностей, масштабованість системи та зручність подальшого розвитку. Кожна частина коду відповідає за чітко визначений функціонал, що сприяє стабільності та передбачуваності проєкту.

3.2. Програмна реалізація ключових функцій системи

Нижче наведено практичну реалізацію основних компонентів вебсистеми `InfoGas`, яка забезпечує автентифікацію користувачів, взаємодію з базою даних

Firestore, створення та управління лічильниками й квитанціями, а також відображення аналітичної інформації в адміністративній панелі.

Модуль автентифікації є ключовим елементом системи, оскільки забезпечує захищений доступ до персональних даних користувачів та розмежування прав між ролями: адміністратором і звичайним користувачем.

Для реалізації авторизації використано Firebase Authentication, що дозволяє виконувати операції входу, реєстрації, виходу й перевірки статусу поточного користувача.

Реєстрація реалізована у компоненті Register.jsx. Після введення даних користувача викликається функція createUserWithEmailAndPassword(), а також виконується створення нового документа у колекції users з персональними даними.

Лістинг 3.1 – Реєстрація нового користувача у системі:

```
const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    const userCredential = await
    createUserWithEmailAndPassword(
      auth,
      form.email,
      form.password
    );
    await setDoc(doc(db, "users",
    userCredential.user.uid), {
      fullName: form.fullName,
      email: form.email,
      phone: form.phone,
      address: form.address,
      accountNumber: `WB-${Math.floor(100 +
    Math.random() * 900)} `,
      createdAt: serverTimestamp(),
    });
  };
};
```

У наведеному коді виконується створення нового облікового запису та запис користувача у Firestore. Система автоматично генерує унікальний номер рахунку (поле accountNumber) і зберігає дату створення через serverTimestamp().

Процес входу реалізовано в компоненті Login.jsx, який перевіряє достовірність даних через Firebase Authentication і, залежно від ролі користувача, перенаправляє його до відповідного інтерфейсу.

Лістинг 3.2 – Авторизація користувача:

```
const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    await signInWithEmailAndPassword(auth, form.email,
form.password);
    Swal.fire({ icon: "success", title: "Вхід
успішний", timer: 1000 });
    navigate("/profile");
  } catch {
    Swal.fire("Помилка входу", "Невірна пошта або
пароль", "error");
  }
};
```

Після успішного входу користувач отримує повідомлення про авторизацію, а далі відбувається перенаправлення на персональну сторінку профілю. Для оновлення даних профілю застосовується компонент Profile.jsx. Користувач має змогу змінювати ПІБ, адресу, телефон або пароль. Зміни одразу відображаються в базі Firestore.

Лістинг 3.3 – Оновлення профілю:

```
export const updateUser = async (userId, updatedData) => {
  try {
    await updateDoc(doc(db, "users", userId),
updatedData);
    Swal.fire({ icon: "success", title: "Оновлено!",
timer: 1200 });
  } catch {
    Swal.fire("Помилка оновлення");
  }
};
```

Видалення користувача здійснюється через deleteDoc() у Firestore і видалення акаунту в Authentication.

Лістинг 3.4 – Видалення користувача:

```
export const removeUser = async (userId) => {
  try {
    await deleteDoc(doc(db, "users", userId));
    await deleteUser(auth.currentUser);
  } catch (err) {
    console.error("Помилка видалення:", err);
  }
};
```

Завершення сесії реалізоване через функцію `signOut()` Firebase, після чого користувача автоматично повертає на сторінку авторизації.

Лістинг 3.5 – Функція виходу користувача із системи:

```
import { signOut } from "firebase/auth";
import { auth } from "../firebase";
import { useNavigate } from "react-router-dom";

export default function LogoutButton() {
  const navigate = useNavigate();
  const handleLogout = async () => {
    await signOut(auth);
    navigate("/login");
  };
}
```

Реалізований модуль автентифікації забезпечує повноцінний цикл управління користувачами, від реєстрації та входу до редагування та видалення облікових записів.

Одним із основних компонентів вебсистеми InfoGas є хмарна база даних Firebase Firestore, що забезпечує зберігання користувацьких профілів, лічильників і квитанцій.

Для зчитування даних використовується функція `onSnapshot()`, що створює підписку на зміни у вказаній колекції Firestore. Цей підхід дозволяє автоматично оновлювати таблиці в інтерфейсі при кожному додаванні, зміні або видаленні документів.

Лістинг 3.6 – Отримання списку лічильників поточного користувача:

```
useEffect(() => {
  const user = auth.currentUser;
  if (!user) return;

  const q = query(
    collection(db, "counters"),
    where("userId", "==", user.uid)
  );

  const unsub = onSnapshot(q, (snap) => {
    setCounters(
      snap.docs.map((d) => ({ id: d.id, ...d.data()
    })))
  });

  return () => unsub();}, []);
```

У цьому прикладі реалізовано підписку на зміни у колекції counters. Компонент автоматично оновлює свій стан при будь-яких змінах у базі даних, що забезпечує реактивність інтерфейсу користувача. Для створення записів у Firestore використовується функція addDoc(). У системі InfoGas це застосовується для створення нових лічильників і квитанцій. При цьому до документа автоматично додаються поля createdAt та updatedAt із часовою міткою сервера.

Лістинг 3.7 – Додавання нового лічильника користувачем:

```
await addDoc(collection(db, "counters"), {
  number: data.number,
  address: data.address,
  type: data.type,
  tariff: data.tariff,
  userId: auth.currentUser.uid,
  status: "неактивний",
  createdAt: serverTimestamp(),
  updatedAt: serverTimestamp(),
});
```

Створення лічильника здійснюється через централізований виклик функції, що забезпечує узгодженість структури документів і полегшує подальшу аналітику.

Аналогічно функція `addDoc()` використовується для створення квитанцій у компоненті `AddReceiptModal.jsx`.

Для зміни статусів, оновлення показників або тарифів використовується метод `updateDoc()`. Це дозволяє адміністратору підтверджувати квитанції, а користувачу, оновлювати власні дані.

Лістинг 3.8 – Оновлення статусу лічильника адміністратором:

```
import { doc, updateDoc, serverTimestamp } from
"firebase/firestore";
import { db } from "../firebase";

export const updateCounterStatus = async (counterId,
newStatus) => {
  await updateDoc(doc(db, "counters", counterId), {
    status: newStatus,
    updatedAt: serverTimestamp(),
  });
};
```

Завдяки використанню часової мітки `updatedAt` система може відстежувати останні зміни та використовувати їх для побудови аналітичних графіків активності.

Видалення елементів реалізовано через метод `deleteDoc()`. У системі `InfoGas` ця функція застосовується як для користувача (видалення власних лічильників), так і для адміністратора (видалення некоректних записів).

Лістинг 3.9 – Видалення лічильника:

```
export const deleteCounter = async (counterId) => {
  try {
    await deleteDoc(doc(db, "counters", counterId));
    Swal.fire({
      icon: "success",
      title: "Лічильник видалено!",
      timer: 1200,
      showConfirmButton: false,
    });
  } catch (error) {
    Swal.fire({
      icon: "error",
      title: "Помилка при видаленні",
    });
  }
};
```

```
    }  
  };
```

Модуль взаємодії з Firestore забезпечує повний набір CRUD-операцій (створення, зчитування, оновлення, видалення), реалізує реактивне оновлення інтерфейсу та підтримує структуровану взаємодію між користувачем і базою даних.

Такий підхід мінімізує кількість серверних запитів, спрощує підтримку коду та підвищує швидкодію системи.

Для зручності користувача реалізовано механізм пошуку та фільтрації за номером, типом або статусом лічильника. Пошук виконується у стані клієнта без повторного запиту до Firestore, що підвищує швидкодію системи.

Лістинг 3.10 – Фільтрація лічильників:

```
const filteredCounters = counters.filter((c) => {  
  const matchesSearch =  
    c.number.toLowerCase().includes(search.toLowerCase())  
  ||  
  
    c.type.toLowerCase().includes(search.toLowerCase()) ||  
  
    c.address.toLowerCase().includes(search.toLowerCase());  
  const matchesType = filterType ? c.type ===  
    filterType : true;  
  const matchesStatus = filterStatus ? c.status ===  
    filterStatus : true;  
  return matchesSearch && matchesType &&  
    matchesStatus;  
});
```

Завдяки поєднанню пошуку та подвійної фільтрації користувач може швидко знайти потрібний прилад навіть за великої кількості записів.

Адміністративна частина системи InfoGas реалізує повноцінну панель управління користувачами, лічильниками, квитанціями та статистичними показниками діяльності підприємства ТОВ «Газмережі». Метою цього модуля є забезпечення адміністратора засобами контролю, аналітичного спостереження та прийняття рішень на основі актуальних даних.

Архітектурно панель адміністратора побудована за компонентним принципом, де кожен функціональний елемент (таблиця користувачів, графіки, фільтри) є окремим компонентом React.

Адміністратор має змогу переглядати список усіх користувачів системи, фільтрувати їх за активністю лічильників і квитанцій, а також переходити до сторінки детальної інформації. Дані підтягуються з Firestore у реальному часі через `onSnapshot()`.

Лістинг 3.11 – Отримання та фільтрація користувачів:

```
useEffect(() => {
    const unsubscribe = onSnapshot(collection(db, "users"),
    (snapshot) => {
        const data = snapshot.docs.map((doc) => ({ id:
        doc.id, ...doc.data() }));
        setUsers(data);
    });
    return () => unsubscribe();
}, []);

const filteredUsers = users.filter(
    (u) =>

    u.fullName.toLowerCase().includes(search.toLowerCase()) ||

    u.email.toLowerCase().includes(search.toLowerCase()) ||
    u.phone.includes(search)
);
```

Пошук виконується локально на клієнтській стороні, що дозволяє швидко знаходити потрібного користувача без повторних запитів до бази даних.

Після пошуку адміністратор може переглянути загальну кількість лічильників і квитанцій кожного користувача. Реалізовано автоматичне підрахування активних і неактивних об'єктів. Для переходу на сторінку детальної інформації використовується компонент `Link` [24].

Лістинг 3.12 – Відображення користувачів із кількістю лічильників і квитанцій:

```
{filteredUsers.map((u) => (
  <tr key={u.id} className="border-b hover:bg-blue-50
transition">
    <td className="px-4 py-2">{u.fullName}</td>
    <td className="px-4 py-2">{u.email}</td>
    <td className="px-4 py-2">{u.phone}</td>
    <td className="px-4 py-2 text-green-600">
      {u.activeCounters}/{u.totalCounters}
    </td>
    <td className="px-4 py-2 text-blue-600">
      {u.confirmedReceipts}/{u.totalReceipts}
    </td>
    <td className="text-center">
      <Link
        to={` /admin/user/${u.id}`}
        className="text-[#2563eb] hover:underline"
      >
        Більше →
      </Link>
    </td>
  </tr>
)})}
```

Після переходу адміністратор може не лише переглянути інформацію користувача, але й змінити статус лічильників або підтвердити квитанції. Зміна статусів лічильників і квитанцій виконується аналогічно до попереднього розділу (через `updateDoc()`), але ініціюється саме з адміністративного інтерфейсу.

Лістинг 3.13 – Підтвердження квитанції в панелі адміністратора:

```
export const confirmReceipt = async (id) => {
  await updateDoc(doc(db, "receipts", id), { status:
"підтверджено" });
  Swal.fire({
    icon: "success",
    title: "Квитанцію підтверджено!",
    timer: 1000,
    showConfirmButton: false,
  });
};
```

Ключовим елементом є аналітичний dashboard, який відображає агреговані показники системи в реальному часі. Для побудови графічних елементів використовується бібліотека Recharts, що дозволяє створювати лінійні, стовпчикові та кругові діаграми.

Лістинг 3.14 – Побудова кругової діаграми:

```
export default function CountersPie() {
  return (
    <PieChart width={300} height={250}>
      <Pie
        data={data}
        cx="50%"
        cy="50%"
        outerRadius={90}
        dataKey="value"
        label
      >
        {data.map((entry, i) => (
          <Cell key={i} fill={COLORS[i %
COLORS.length]} />
        ))}
      </Pie>
      <Tooltip />
      <Legend />
    </PieChart>
  );
}
```

Для аналітики фінансових показників використовується лінійний графік, який відображає динаміку оплат користувачів за датами. Це дозволяє оцінити активність споживачів і прогнозувати майбутні надходження.

Лістинг 3.15 – Лінійний графік оплат у часі:

```
export default function PaymentsChart() {
  return (
    <LineChart width={600} height={300}
      data={chartData}>
      <CartesianGrid strokeDasharray="3 3" />
      <XAxis dataKey="date" />
      <YAxis />
      <Tooltip />
    </LineChart>
  );
}
```

```

        <Line type="monotone" dataKey="total"
stroke="#2563eb" strokeWidth={2} />
    </LineChart>
    );
}

```

Для аналізу фінансової активності реалізовано гістограму найактивніших користувачів, хто здійснює найбільше оплат.

Лістинг 3.16 – Гістограма активних користувачів:

```

export default function TopUsersChart() {
  return (
    <BarChart width={500} height={300}
data={topUsers}>
      <XAxis dataKey="name" />
      <YAxis />
      <Tooltip />
      <Bar dataKey="total" fill="#60a5fa" />
    </BarChart>
  );
}

```

Адміністративна панель системи InfoGas об'єднує інструменти моніторингу, управління користувачами та аналітики. Завдяки поєднанню реактивної роботи з базою даних та наочних засобів візуалізації адміністратор отримує цілісне уявлення про стан системи в реальному часі. Аналітична панель дозволяє оперативно виявляти відхилення, оцінювати активність користувачів і контролювати фінансові показники без необхідності ручної обробки даних.

3.2.1. Інтеграція модуля штучного інтелекту

Модуль штучного інтелекту є одним із найінноваційніших компонентів вебсистеми InfoGas, який реалізує автоматизоване консультування користувачів щодо тарифів, умов користування газопостачанням, розрахунку пільг і субсидій.

В основі його функціонування лежить інтеграція з Рое API, що забезпечує взаємодію з мовною моделлю, здатною формувати природномовні відповіді на запити користувачів. На серверному рівні використовується Node.js + Express, що

дозволяє безпечно взаємодіяти з Poe API [25-26]. Модуль експортує єдиний ендпоінт `/api/ask`, який приймає текст запиту користувача, відправляє його до API і повертає відповідь.

Лістинг 3.17 – Node.js ендпоінт для AI-запитів:

```
app.post("/api/ask", async (req, res) => {
  try {
    const response = await
    fetch("https://api.poe.com/v1/chat", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        Authorization: `Bearer
        ${process.env.POE_API_KEY}`,
      },
      body: JSON.stringify({
        model: "claude-instant",
        messages: [{ role: "user", content:
        req.body.prompt }],
      })),));
```

Сервер обробляє кожен запит незалежно, забезпечуючи безпеку токенів і стабільну роботу навіть при високому навантаженні. Також додано CORS для можливості обміну даними між клієнтським React-додатком і Node.js-сервером.

Фронтенд реалізовано у вигляді компонента `AIAssistant.jsx`, який містить історію діалогу, поле введення та обробник запитів до API [27]. Кожне повідомлення зберігається у стані `messages`, що дозволяє підтримувати послідовність діалогу.

Лістинг 3.18 – Логіка надсилання запитів:

```
const handleSend = async () => {
  const message = input;
  setMessages((m) => [...m, { role: "user", content:
  message }]);
  const res = await
  fetch("http://localhost:3001/api/ask", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ prompt: message })),));
  const data = await res.json();
```

```

    setMessages((m) => [...m, { role: "assistant",
content: data.answer }]);
    setInput("");});

```

Компонент підтримує діалогову взаємодію, де повідомлення користувача і відповідь моделі виводяться у різних кольорах. AI-модуль може аналізувати дані користувача (наприклад, тариф, кількість лічильників, статус квитанцій) і формувати персоналізовані рекомендації. Це досягається через попередній збір контексту з Firestore перед запитом до Poe API.

Лістинг 3.19 – Формування контексту перед відправкою запиту

```

const preparePrompt = (userData, question) => {
  return `
    Користувач: ${userData.fullName}, тариф
    ${userData.tariff} грн/м³.
    Активних лічильників: ${userData.activeCounters}.
    Питання: ${question}.
    Відповідай українською коротко і по суті.`;
};

```

Відповіді системи є не лише загальними, а й контекстно-залежними, адаптованими під кожного користувача. Це підвищує ефективність консультування та скорочує навантаження на технічну підтримку. Застосування алгоритмічних методів оброблення даних у поєднанні з трансформерними нейронними мережами відповідає концепціям Industry 4.0 та підтверджується сучасними науковими дослідженнями [28-29]. Використання персоналізованих відповідей на основі реальних даних забезпечує більш точні рекомендації та сприяє підвищенню довіри користувачів до сервісу. У перспективі такий підхід створює передумови для подальшого розвитку інтелектуальної аналітики та автоматизації консультаційних процесів.

3.4 Методика тестування системи

Тестування програмного забезпечення InfoGas ґрунтується на технології формалізованих сценаріїв використання (Use Case), яка є одним із найбільш ефективних підходів до оцінювання функціональної коректності інформаційних

систем. Use Case виступає логічною моделлю взаємодії між актором (користувачем або адміністратором) та системою, відображаючи точні умови виконання певної дії, очікувані результати, можливі виняткові ситуації та поведінку системи у відповідь на кожен із запитів [30].

Саме деталізація альтернативних ситуацій дозволяє перевірити коректність обробки помилок, стабільність роботи системи та готовність до непередбачуваних дій користувача.

В процесі тестування InfoGas Use Case моделює реальні сценарії роботи системи, максимально наближені до поведінки користувача у повсякденних умовах. Це дозволяє не лише підтвердити працездатність окремих функцій, але й оцінити цілісність взаємодії між підсистемами автентифікацією, обробкою даних Firestore, логікою розрахунків, валідаційними механізмами, інтеграцією з AI-модулем, інтерфейсами аналітики тощо.

Механізм входу до системи є базовим елементом безпеки та необхідною умовою для доступу до персонального функціоналу. Авторизація визначає права користувача, забезпечує захист даних та дозволяє системі коректно ідентифікувати суб'єкта взаємодії.

У таблиці 3.1 наведено формалізований сценарій використання, який охоплює всі ключові етапи автентифікації.

Таблиця 3.1 – Use Case «Авторизація користувача»

Поле	Опис
Дійові особи	Користувач, Система
Ціль	Увійти в систему та отримати доступ до персонального кабінету
Передумови	Користувач має зареєстрований обліковий запис
Успішний сценарій	1. Користувач вводить пошту і пароль. 2. Система перевіряє облікові дані у Firebase. 3. Успішний вхід → перехід до кабінету. 4. Подія фіксується у логах.
Альтернативні сценарії	Невірні дані → відмова; помилка Firebase → технічне повідомлення.
Результат	Користувач отримує доступ до системи

Функція авторизації працює стабільно та коректно обробляючи усі сценарії.

Створення облікового запису є ключовою функцією, що формує персональний профіль, унікальний номер рахунку й забезпечує подальшу ідентифікацію в системі. У таблиці 3.2 представлено сценарій реєстрації з усіма необхідними перевітками та логікою створення записів.

Таблиця 3.2 – Use Case «Реєстрація користувача»

Поле	Опис
Дійові особи	Користувач, Система
Ціль	Створити новий обліковий запис
Передумови	Відсутність існуючого профілю
Успішний сценарій	1. Користувач вводить ПІБ, пошту, телефон, адресу, пароль. 2. Система виконує валідацію. 3. Firebase створює обліковий запис. 4. Система генерує номер рахунку. 5. Дані зберігаються у Firestore. 6. Користувач автоматично авторизується.
Альтернативні сценарії	Пошта уже використовується; некоректні дані; збій Firebase.
Результат	Новий користувач успішно створений

Процес реєстрації реалізовано коректно: усі необхідні валідації та інтеграції працюють стабільно.

Робота з лічильниками є одним із головних функціональних модулів. Процес додавання нового лічильника має однакову структуру для всіх типів, різниця полягає лише у параметрах. У таблиці 3.3 наведено сценарій створення об'єкта цього типу.

Таблиця 3.3 – Use Case «Реєстрація користувача»

Поле	Опис
Дійові особи	Користувач, Система
Ціль	Створити новий лічильник
Передумови	Користувач авторизований
Успішний сценарій	1. Користувач відкриває «Лічильники». 2. Натискає «Додати» та вводить дані (номер, тип, адреса, тариф, пільги). 3. Система перевіряє коректність. 4. Лічильник зберігається у Firestore зі статусом «неактивний». 5. Лічильник з'являється в таблиці лічильників користувача.
Альтернативні сценарії	Порожні поля; некоректний формат номеру; збій БД.
Результат	Лічильник успішно створений

Логіка створення лічильників працює стабільно; помилки введення даних коректно обробляються.

Операція видалення у системі є уніфікованою для лічильників і квитанцій різниці полягає лише у типі об'єкта. У таблиці 3.4 наведено сценарій для видалення лічильника.

Таблиця 3.4 – Use Case «Видалення лічильника»

Поле	Опис
Дійові особи	Користувач, Система
Ціль	Видалити наявний лічильник
Передумови	Лічильник належить користувачу
Успішний сценарій	1. Користувач відкриває список лічильників. 2. Обирає об'єкт. 3. Натискає «Видалити». 4. Підтверджує SweetAlert. 5. Система видаляє запис із Firestore. 6. Показує повідомлення про видалення, та видаляє лічильник з таблиці.
Альтернативні сценарії	Користувач скасовує дію; збій БД; відсутній об'єкт.
Результат	Лічильник видалено

Створення квитанції базується на тих самих принципах, що й створення лічильника, але містить додаткову логіку автоматичного розрахунку оплати. У таблиці 3.5 наведено відповідний сценарій.

Таблиця 3.5 – Use Case «Створення квитанції»

Поле	Опис
Дійові особи	Користувач, Система
Ціль	Створити квитанцію з автоматичним розрахунком
Передумови	Є хоча б один лічильник
Успішний сценарій	1. Користувач обирає лічильник. 2. Вводить обсяг спожитого газу (м³). 3. Система автоматично обчислює суму (враховуючи пільги та субсидії). 4. Квитанція зберігається зі статусом «не підтверджено». 5. Квитанція з'являється в таблиці квитанцій.
Альтернативні сценарії	Невалідний обсяг; відсутні лічильники.
Результат	Квитанцію успішно створено

Алгоритм розрахунку та створення квитанцій працює коректно й стабільно.

AI-помічник у системі InfoGas надає відповіді на 16 типових питань, аналізуючи дані конкретного користувача. У таблиці 3.6 представлено сценарій взаємодії з інтелектуальним модулем.

Таблиця 3.6 – Use Case «Взаємодія з AI-помічником»

Поле	Опис
Дійові особи	Користувач, Система (AI-модуль)
Ціль	Отримати персоналізовану відповідь
Передумови	Користувач авторизований
Успішний сценарій	1. Користувач обирає одне з 16 запитань. 2. Система передає запит до AI. 3. AI аналізує дані Firestore. 4. Формує відповідь. 5. Відповідь з'являється в інтерфейсі.
Альтернативні сценарії	AI недоступний; нестача даних.
Результат	Користувач отримує відповідь, яка враховує його дані

Процес видалення акаунта є найбільш критичною операцією, яка вимагає каскадного очищення всіх пов'язаних даних. У таблиці 3.7 наведено відповідний Use Case.

Таблиця 3.7 – Use Case «Видалення профілю»

Поле	Опис
Дійові особи	Користувач, Система
Ціль	Повністю видалити обліковий запис і всі дані
Передумови	Користувач авторизований
Успішний сценарій	1. Користувач натискає «Видалити профіль». 2. Підтверджує у формі видалення SweetAlert. 3. Система видаляє лічильники, квитанції, профіль. 4. Firebase видаляє акаунт. 5. Користувача перенаправляє на головну.
Альтернативні сценарії	Відмова від підтвердження; збій БД; збій Firebase.
Результат	Профіль успішно видалено

Перегляд профілю користувача є базовою адміністративною функцією, що дозволяє аналізувати інформацію про обліковий запис, лічильники, квитанції та історію взаємодії. Це необхідно для контролю коректності даних та виконання подальших адміністративних дій. Формалізований сценарій подано в таблиці 3.8.

Таблиця 3.8 – Use Case «Перегляд профілю користувача адміністратором»

Поле	Опис
Дійові особи	Адміністратор, Система
Ціль	Переглянути детальну інформацію про користувача
Передумови	Адміністратор авторизований
Успішний сценарій	1. Адміністратор відкриває розділ «Користувачі». 2. Обирає конкретного користувача зі списку. 3. Система завантажує інформацію профілю, лічильників та квитанцій. 4. Відображається загальна статистика: кількість лічильників, кількість квитанцій, сума оплат. 5. Адміністратор може перейти до керування об'єктами користувача.
Альтернативні сценарії	Дані недоступні → повідомлення про помилку.
Результат	Адміністратор бачить повну інформацію про користувача

Функціонал перегляду профілю працює коректно та дозволяє адміністратору виконувати подальші дії над об'єктами користувача.

Активація або деактивація лічильника є ключовою адміністративною операцією, яка впливає на можливість користувача створювати квитанції та вести облік.

В таблиці 3.9 наведено сценарій зміни статусу лічильника.

Таблиця 3.9 – Use Case «Активація лічильника адміністратором»

Поле	Опис
Дійові особи	Адміністратор, Система
Ціль	Змінити статус лічильника (активний / неактивний)
Передумови	Лічильник існує у Firestore
Успішний сценарій	1. Адміністратор відкриває профіль користувача. 2. Переходить до списку лічильників. 3. Натискає перемикач статусу лічильника. 4. Система оновлює поле isActive у Firestore. 5. З'являється SweetAlert із результатом операції.
Альтернативні сценарії	Помилка БД → статус не змінено.
Результат	Статус лічильника успішно оновлено

Модуль активації лічильника працює стабільно, забезпечуючи синхронізацію із Firestore і коректне відображення змін.

Усі наведені Use Case охоплюють ключовий функціонал системи InfoGas: автентифікацію, реєстрацію, операції з лічильниками та квитанціями, роботу з AI-модулем та видалення профілю. Інші функції, не включені до таблиць, мають аналогічну структуру бізнес-процесів або є похідними від тих, що вже описані. Вони повторюють логіку CRUD-операцій, відрізняючись лише набором полів або контекстом застосування.

Проведене сценарне моделювання підтверджує, що система поводить себе стабільно у всіх критичних точках взаємодії. Ключові модулі працюють згідно з вимогами, а обробка помилкових ситуацій реалізована коректно, що свідчить про готовність InfoGas до практичної експлуатації.

3.5. Демонстрація отриманих результатів

Головна сторінка InfoGas реалізована у вигляді сучасного адаптивного інтерфейсу з акцентом на швидкому доступі до системи. У верхній частині розміщено навігацію для входу та реєстрації, а центральний блок містить короткий опис призначення сервісу та кнопку переходу до роботи, виконані у фірмовій синій колірній палітрі (рис. 3.2).

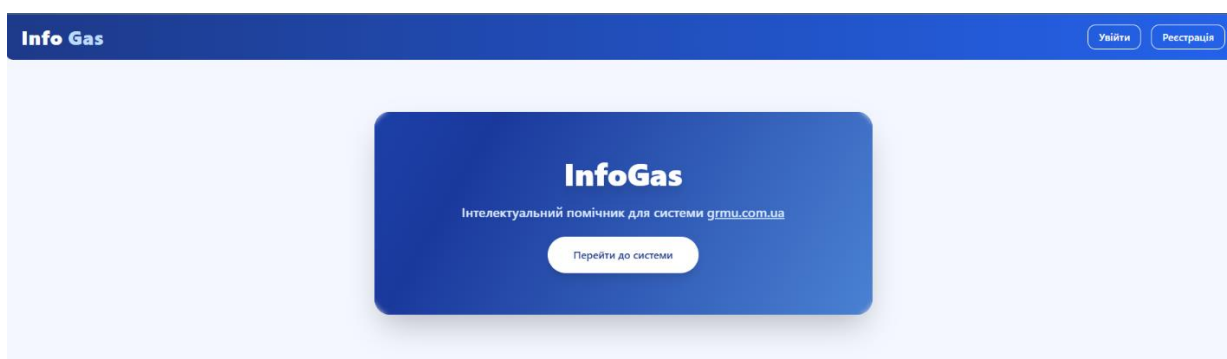


Рис. 3.2 – Фрагмент головної сторінки InfoGas

Другий фрагмент головної сторінки демонструє основні можливості системи у вигляді структурованих інформаційних карток. Такий формат забезпечує швидке ознайомлення користувача з ключовими функціями сервісу, зокрема профілем,

аналітикою, звітами, документацією та AI-модулем, що підвищує зручність навігації (рис. 3.3).

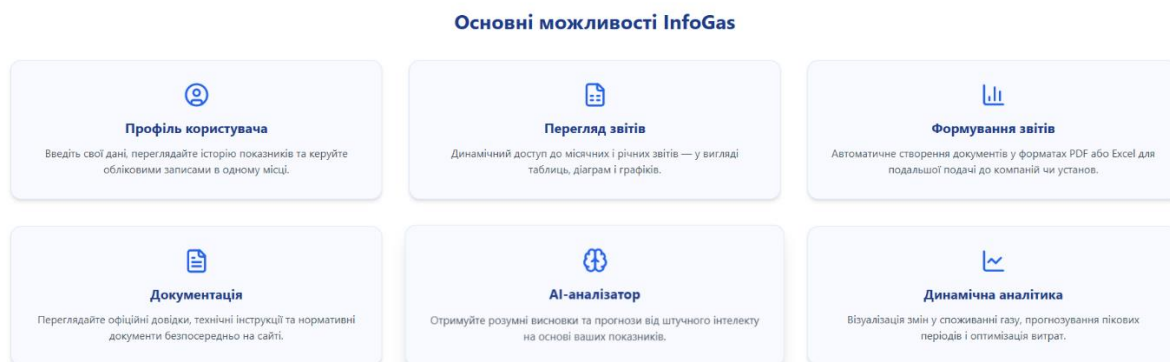


Рис. 3.3 – Основні функціональні модулі, представлені на головній сторінці

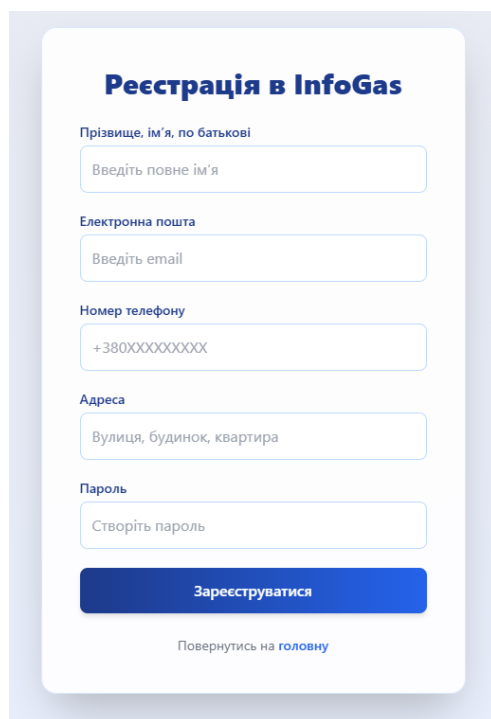
Форма входу до системи представлена у вигляді компактного блоку з полями для електронної пошти та пароля.

Інтерфейс виконано в мінімалістичному стилі з використанням світлого фону та акцентної кнопки входу, що забезпечує комфортне використання та швидке введення даних користувачем (рис. 3.4).

Зареєструйтесь!'."/>

Рис. 3.4 – Форма входу до системи InfoGas

Форма реєстрації містить усі необхідні поля для створення облікового запису, включно з ПІБ, електронною поштою, телефоном, адресою та паролем. Вона оформлена у стилістиці всієї системи та передбачає зручне заповнення завдяки чіткому розташуванню елементів і візуальній легкості компоновання (рис. 3.5).

The image shows a registration form titled "Реєстрація в InfoGas". It contains five input fields: "Прізвище, ім'я, по батькові" with placeholder "Введіть повне ім'я", "Електронна пошта" with placeholder "Введіть email", "Номер телефону" with placeholder "+380XXXXXXXX", "Адреса" with placeholder "Вулиця, будинок, квартира", and "Пароль" with placeholder "Створіть пароль". Below the fields is a blue button labeled "Зареєструватися" and a link "Повернутись на головну".

Реєстрація в InfoGas

Прізвище, ім'я, по батькові
Введіть повне ім'я

Електронна пошта
Введіть email

Номер телефону
+380XXXXXXXX

Адреса
Вулиця, будинок, квартира

Пароль
Створіть пароль

Зареєструватися

[Повернутись на головну](#)

Рис. 3.5 – Форма реєстрації нового користувача InfoGas

Після успішної авторизації користувач автоматично переходить на сторінку власного профілю, де може редагувати персональні дані, змінювати пароль та переглядати унікальний номер рахунку.

Інтерфейс виконаний у мінімалістичному стилі та забезпечує швидкий доступ до ключових полів, включно з можливістю повного видалення облікового запису (рис. 3.6).

Рис. 3.6 – Інтерфейс персонального профілю користувача InfoGas

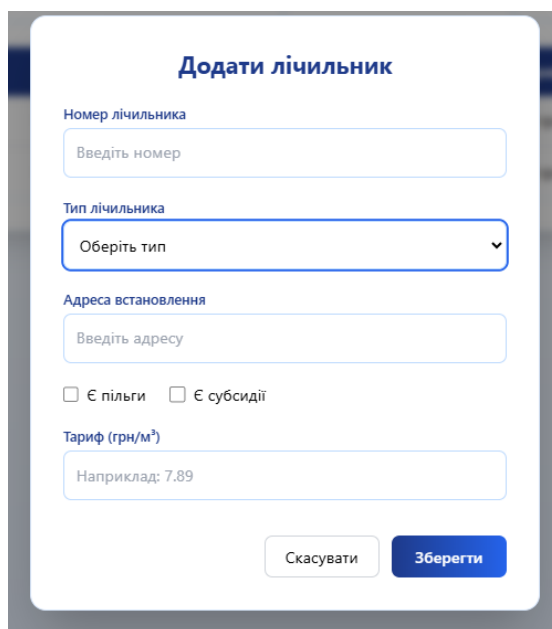
Сторінка лічильників дозволяє користувачу переглядати всі створені записи, фільтрувати їх за статусом та типом, а також швидко знаходити потрібний лічильник через поле пошуку.

В таблиці відображається ключова інформація про кожен лічильник, включно з адресою, тарифом, наявністю пільг та поточним статусом, а в правій частині доступні дії для редагування або видалення запису (рис. 3.7).

№	Тип	Адреса	Пільги	Субсидія	Тариф	Статус	Дії
JH90428	побутовий	Київ, Нехалежності 10а	Ні	Так	36 грн/м³	неактивний	
2222	однотарифний	Київ, вул. Героїв, буд3\кв2	Так	Так	25 грн/м³	неактивний	

Рис. 3.7 – Список лічильників із можливістю фільтрації та керування

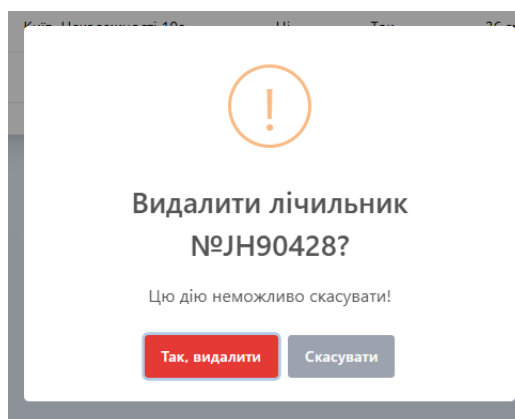
При натисканні на кнопку «Додати лічильник» відкривається модальна форма, яка дозволяє ввести тип лічильника, адресу, тариф та інші параметри. Чітка структура полів, підписи та підказки забезпечують коректність введення даних, а всі введені значення проходять валідацію перед збереженням у базі даних (рис. 3.8).



The image shows a modal form titled "Додати лічильник" (Add meter). It contains several input fields: "Номер лічильника" (Meter number) with a placeholder "Введіть номер", "Тип лічильника" (Meter type) with a dropdown menu showing "Оберіть тип", "Адреса встановлення" (Installation address) with a placeholder "Введіть адресу", and "Тариф (грн/м³)" (Tariff) with a placeholder "Наприклад: 7.89". There are also two checkboxes: "Є пільги" (There are discounts) and "Є субсидії" (There are subsidies). At the bottom, there are two buttons: "Скасувати" (Cancel) and "Зберегти" (Save).

Рис. 3.8 – Форма додавання нового лічильника

Під час спроби видалити лічильник система виводить форму підтвердження, що запобігає випадковому видаленню даних користувача. Повідомлення містить попередження про незворотність операції та пропонує два варіанти дії: підтвердити або скасувати видалення (рис. 3.9).



The image shows a confirmation modal for deleting a meter. It features a large orange exclamation mark icon at the top. Below the icon, the text reads "Видалити лічильник №JH90428?". Underneath, a warning message states "Цю дію неможливо скасувати!". At the bottom, there are two buttons: a red button labeled "Так, видалити" (Yes, delete) and a gray button labeled "Скасувати" (Cancel).

Рис. 3.9 – Форма підтвердження видалення лічильника

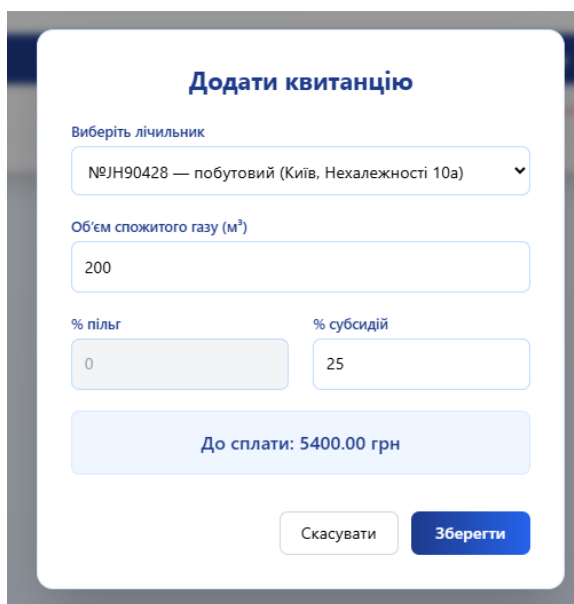
Сторінка квитанцій реалізована за тією ж структурною моделлю, що й модуль лічильників: користувач може переглядати всі створені квитанції, фільтрувати їх за статусом та знаходити потрібний запис за допомогою пошуку.

В таблиці відображаються основні параметри: обсяг газу, відсоток субсидій, розрахована сума до сплати та дата створення, що дозволяє швидко оцінити свою фінансову активність (рис. 3.10).

№ лічильника	Тип	Об'єм (м³)	% пільг	% субсидій	До сплати	Статус	Дата	Дії
JH90428	побутовий	100	0%	25%	2700 грн	не підтверджено	20.11 / 14:09	

Рис. 3.10 – Список квитанцій із можливістю фільтрації та перегляду

При натисканні кнопки «Додати квитанцію» відкривається модальна форма, у якій користувач обирає один із наявних лічильників, вводить обсяг спожитого газу та, за потреби, коригує значення пільг або субсидій (рис. 3.11).



Додати квитанцію

Виберіть лічильник

№JH90428 — побутовий (Київ, Нехалежності 10а)

Об'єм спожитого газу (м³)

200

% пільг

0

% субсидій

25

До сплати: 5400.00 грн

Скасувати Зберегти

Рис. 3.11 – Форма створення нової квитанції з автоматичним розрахунком

Система одразу виконує автоматичний перерахунок суми до сплати, що дозволяє уникнути помилок та підвищує зручність формування квитанції.

У розділі «Звіти» користувач отримує доступ до зведеної статистики щодо кількості створених лічильників, підтверджених квитанцій та загальної суми оплат (рис. 3.12).

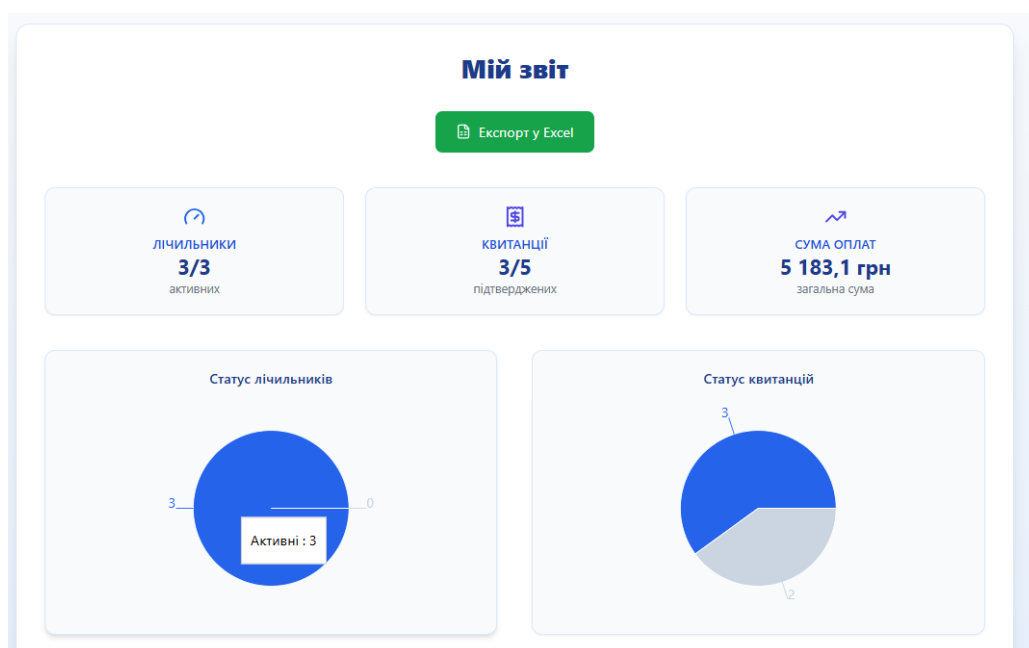


Рис. 3.12 – Аналітичний звіт користувача з круговими діаграмами та можливістю експорту

Тут доступна кнопка експорту в Excel, що дозволяє сформувати офіційний документ для подальшого аналізу або подачі в організації. Дві кругові діаграми відображають співвідношення статусів лічильників та квитанцій, забезпечуючи швидке візуальне сприйняття даних.

Нижня частина сторінки звітів містить інтерактивний лінійний графік, який відображає динаміку оплат користувача за певний період.

Кожна точка на графіку містить підказку з конкретною датою та значенням, що забезпечує детальне вивчення фінансової активності та дозволяє відстежувати зміни у споживанні газу або оплатах у часі (рис. 3.13).

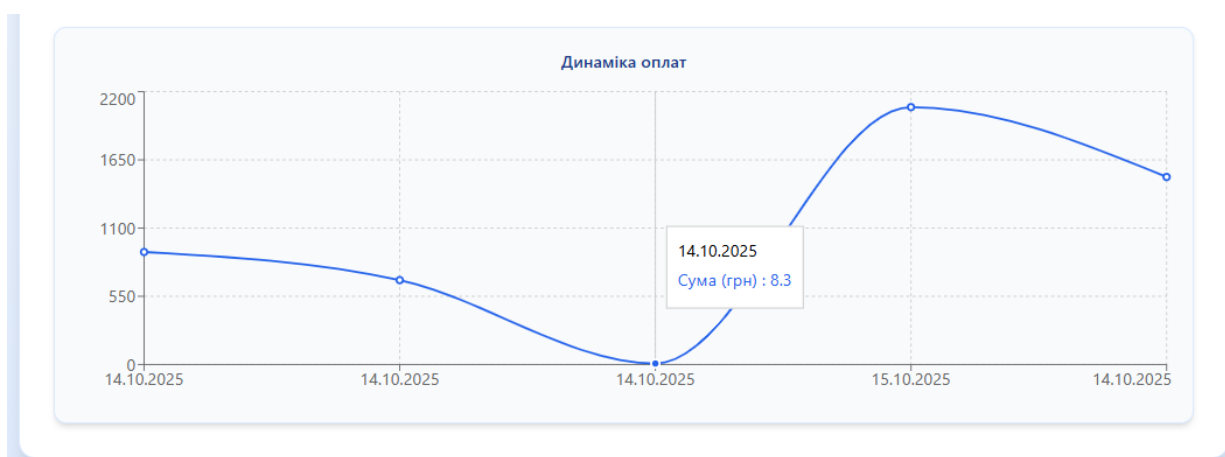


Рис. 3.13 – Лінійний графік динаміки оплат користувача

Після відкриття сторінки «AI Аналіз споживання газу» користувач бачить набір із 16 готових аналітичних питань. Для запуску персоналізованого аналізу достатньо натиснути одне з них (рис. 3.14).

Після вибору питання система автоматично формує аналітичний запит з урахуванням індивідуальних даних користувача, збережених у базі даних. У процесі аналізу використовуються історичні показники споживання, інформація про тарифи, наявність пільг і статуси квитанцій.

Результат оброблення подається у вигляді текстових рекомендацій, зрозумілих для користувача без спеціальної технічної підготовки.

 **AI Аналіз споживання газу**

Оберіть питання, щоб отримати персоналізовану аналітику на основі ваших даних.

☐ Який середній обсяг споживання газу за останній місяць?

☐ Як змінилося споживання газу порівняно з попереднім місяцем?

☐ Чи перевищую я середнє споживання серед користувачів?

☐ Як мені оптимізувати використання газу вдома?

☐ Чи вигідно мені перейти на інший тариф?

☐ Який прогноз споживання газу на наступний місяць?

☐ Як пільги або субсидії впливають на мої витрати?

☐ Які фактори найбільше впливають на моє споживання газу?

☐ Як зменшити рахунок без втрати комфорту?

☐ Які поради з енергоефективності ви можете надати мені?

☐ Чи є сезонні коливання у моєму споживанні газу?

☐ Який період року є найбільш витратним для мене?

☐ Як встановлення лічильника з дистанційною передачею даних може допомогти мені економити?

☐ Які прилади у домі споживають найбільше газу і як це контролювати?

☐ Як зміна температури опалення впливає на мій місячний рахунок?

☐ Які інноваційні технології можуть допомогти зменшити споживання газу?

Оберіть питання для аналізу.

Рис. 3.14 – Питання в AI-модулі

Після того як користувач обрав питання, система формує персоналізовану відповідь з урахуванням тарифів, історії квитанцій та інших параметрів (рис. 3.15).

Аналіз споживання газу для Тернушак Віктора Івановича

За наданими даними, можна провести наступний аналіз споживання газу для Тернушак Віктора Івановича:

1. Динаміка споживання газу:

- Загальний обсяг споживання газу склав **711 м³** за період, що охоплює **5** квитанцій.
- Середнє споживання на квитанцію становить **142.2 м³**, що може слугувати орієнтиром для порівняння з іншими періодами.

2. Можливі перевитрати та вплив тарифів:

- Лічильник **1** та Лічильник **2** мають однакові тарифи, проте різні адреси. Тут може бути корисно перевірити, чи впливає місце проживання на споживання газу.
- Лічильник **3** має вищий тариф, що може впливати на загальні витрати.

3. Пільги та субсидії:

- Усі три лічильники мають пільги та субсидії, що може зменшити загальні витрати на оплату газу.
- Важливо враховувати, як ці пільги та субсидії впливають на кінцеві витрати при різних тарифах.

4. Оптимізація споживання:

- Розгляньте можливі шляхи оптимізації споживання газу, такі як утеплення будинку, встановлення енергоефективного обладнання або раціоналізація використання газу.

З урахуванням вищезазначеного, перехід на інший тариф може бути вигідним, але потребує додаткового аналізу. Рекомендується порівняти тарифи, враховуючи пільги та субсидії, а також розглянути можливість оптимізації споживання газу для зменшення витрат.

Рис. 3.15 – Відповідь AI-модуля

На сторінці «Документація» користувач отримує офіційні відомості про технічне обслуговування, нормативні правила, способи оплати та додаткові матеріали (рис. 3.16).

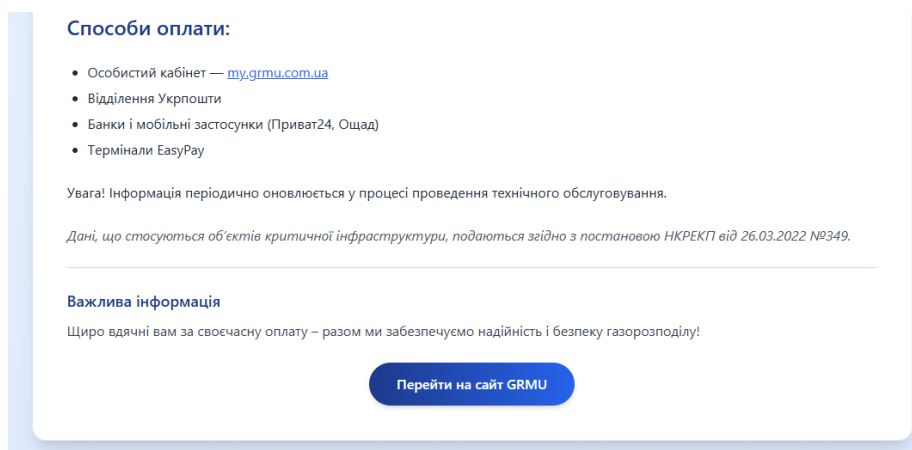


Рис. 3.16 – Частина сторінки «Документація»

Після авторизації адміністратор переходить на сторінку користувачів системи. Де отримує доступ до повного списку профілів, які можна фільтрувати за активністю лічильників і квитанцій, а також шукати за будь-яким атрибутом користувача. Це дозволяє швидко знаходити необхідні облікові записи та переглядати їхній статус у системі (рис. 3.17).

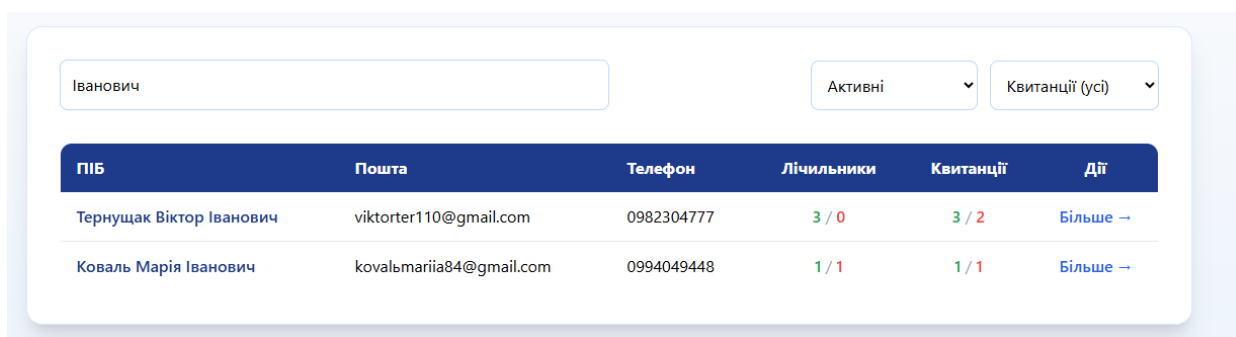


Рис. 3.17 – Сторінка керування користувачами з пошуком і фільтрами

Після вибору конкретного користувача відкривається розділ розширеної статистики, де адміністратор бачить персональні дані, кількість активних лічильників і квитанцій, а також графік змін об'єму газу. Наявність діаграм

забезпечує швидке візуальне сприйняття основних параметрів споживання (рис. 3.18).

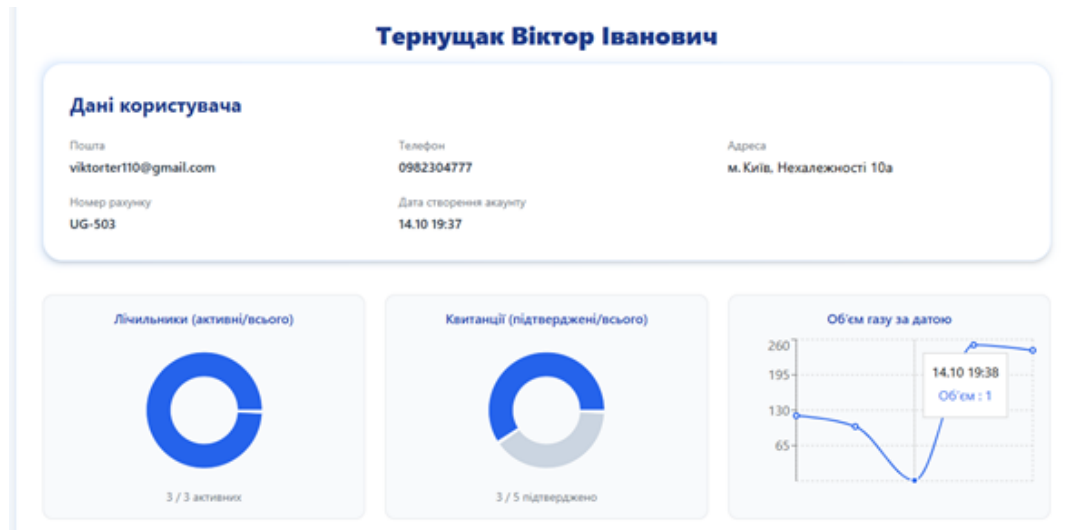


Рис. 3.18 – Персональна статистика користувача з діаграмами та графіками

Нижче розміщено перелік лічильників і квитанцій користувача, де адміністратор може змінювати статуси, активувати або скасовувати підтвердження. Після зміни статусу дані автоматично оновлюються в інтерфейсі, адміністратор отримує відповідне сповіщення про успішне виконання дії (рис. 3.19).

Лічильники							
№	Тип	Тариф	Пільга	Субсидія	Статус	Дія	
JH90428	багатотарифний	8.40 грн/м³	Так	Так	активний	Змінити	
QG90428	однотарифний	8.30 грн/м³	Так	Так	неактивний	Змінити	
JH90500	побутовий	10 грн/м³	Так	Так	активний	Змінити	

Квитанції							
№ лічильника	Об'єм	% пільг	% субсидій	До сплати	Статус	Дата	Дія
JH90428	120 м³	10%	0%	907.2 грн	підтверджено	14.10 20:08	Змінити
QG90428	100 м³	10%	8%	680.6 грн	підтверджено	14.10 20:07	Змінити
QG90428	1 м³	0%	0%	8.3 грн	підтверджено	14.10 19:38	Змінити
JH90500	250 м³	10%	7%	2075 грн	неактивний	15.10 17:19	Змінити
JH90428	240 м³	25%	0%	1512 грн	не підтверджено	14.10 20:07	Змінити

Рис. 3.19 – Керування статусами лічильників і квитанцій користувача

У розділі «Загальна статистика системи» наводяться узагальнені показники всієї платформи (рис. 3.20).

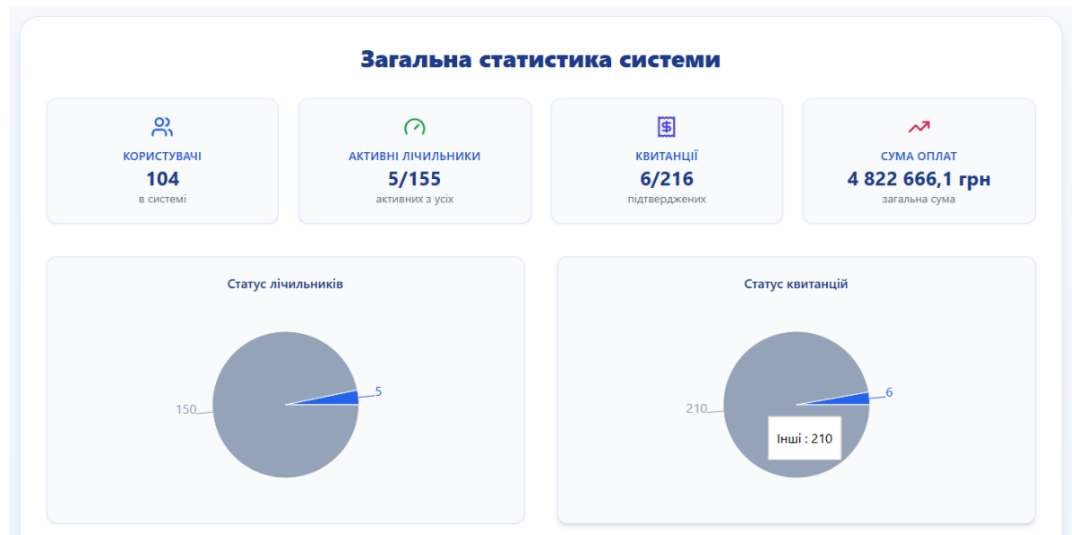


Рис. 3.20 – Зведена статистика функціонування системи InfoGas

Дві кругові діаграми демонструють співвідношення статусів усіх лічильників і квитанцій у системі.

Додаткові графіки відображають динаміку оплат по всій системі та рейтинг користувачів за сумою оплат, що дозволяє адміністратору аналізувати активність і виявляти ключові тенденції. Такі інструменти сприяють ефективному моніторингу всієї інфраструктури платформи (рис. 3.21).

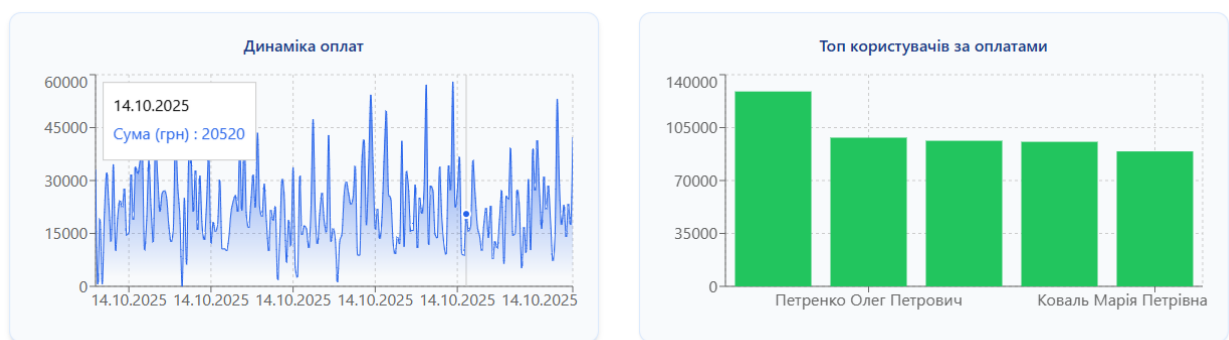


Рис. 3.21 – Динаміка оплат та топ користувачів за фінансовими показниками

Наведені вище інтерфейси відображають основні сценарії взаємодії: управління профілем, лічильниками та квитанціями, перегляд аналітики, використання AI-асистента, а також адміністрування користувачів і загальної статистики. Візуалізація даних у вигляді діаграм, таблиць та графіків підтверджує коректність реалізованого функціоналу та демонструє стабільність роботи системи при обробці реальних даних. Сукупність представлених скріншотів свідчить про завершеність, цілісність та практичну готовність програмного продукту до використання.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Четвертий розділ присвячений питанням охорони праці та забезпечення безпеки в надзвичайних ситуаціях під час розробки та експлуатації програмного забезпечення. У розділі проаналізовано основні небезпечні та шкідливі фактори, що можуть виникати в процесі роботи з комп'ютерною технікою, а також розглянуто нормативно-правові вимоги у сфері охорони праці. На цій основі визначено заходи щодо забезпечення безпечних умов праці, збереження здоров'я працівників та мінімізації ризиків у разі виникнення надзвичайних ситуацій.

4.1. Охорона праці

При розробці та експлуатації вебсистеми збору, оброблення та аналізу даних для ТОВ «Газмережі» враховувалися вимоги охорони праці та пожежної безпеки, оскільки основна діяльність користувачів системи пов'язана з тривалою роботою за персональними комп'ютерами та іншими електронними засобами оброблення інформації. З огляду на специфіку діяльності підприємства та використання інформаційних технологій у виробничо-управлінських процесах, створення безпечних умов праці розглядалося як важливий чинник збереження працездатності персоналу та запобігання професійним ризикам.

Організація безпечних умов праці здійснювалася відповідно до чинних нормативно-правових актів України, зокрема Закону України «Про охорону праці» № 2694-ХІІ, Закону України «Про охорону здоров'я» № 2801-ХІІ, Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, затверджених наказом Міністерства соціальної політики України № 207 від 14.02.2018, а також загальних норм Кодексу законів про працю України [31–35]. Зазначені документи визначають актуальні вимоги до організації робочих місць із використанням комп'ютерної техніки, режимів праці та відпочинку, а також заходів із запобігання негативному впливу екранних пристроїв на здоров'я працівників в офісних і виробничих приміщеннях. Під час проєктування інтерфейсу користувача

вебсистеми враховувалися ергономічні вимоги, спрямовані на зниження навантаження на зоровий апарат і нервову систему оператора. Зокрема, інтерфейс реалізовано з використанням помірної контрастної кольорової гами без різких переходів, оптимального розміру та міжрядкового інтервалу шрифтів, логічної ієрархії елементів керування та мінімізації надлишкової візуальної інформації. Такі рішення сприяють зменшенню зорової втоми, підвищенню концентрації уваги користувачів і зниженню психоемоційного напруження під час тривалої роботи з інформаційною системою.

Відповідно до Закону України «Про охорону праці» № 2694-XII, Закону України «Про охорону здоров'я» № 2801-XII та Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, затверджених наказом Міністерства соціальної політики України № 207 від 14.02.2018, у процесі експлуатації вебсистеми передбачено раціональну організацію робочого процесу користувачів. Зазначені нормативні документи регламентують необхідність дотримання оптимального режиму праці та відпочинку, що спрямований на зниження функціонального перенапруження органів зору, опорно-рухового апарату та нервової системи працівників. Робочі місця користувачів вебсистеми повинні бути облаштовані з урахуванням ергономічних вимог, зокрема використання регульованих крісел, робочих столів відповідної висоти та забезпечення правильного розташування екранних пристроїв у полі зору працівника. Освітлення робочих зон має відповідати санітарно-гігієнічним нормам та забезпечувати комфортні умови для тривалої роботи з інформаційними ресурсами без створення додаткового зорового навантаження. Особлива увага приділялася питанням електробезпеки та пожежної безпеки під час використання комп'ютерної техніки, на якій здійснюється робота з вебсистемою. Згідно з вимогами НПАОП 40.1-1.21-98, робочі місця повинні бути оснащені справними електроприладами із заземленням, без пошкоджених кабелів та з дотриманням правил недопущення перевантаження електромереж. Забороняється використання несправних розеток, подовжувачів та саморобних електричних з'єднань, що можуть стати джерелом короткого замикання або пожежі. Комп'ютерна техніка розміщується на безпечній

відстані від джерел тепла та легкозаймистих матеріалів, що знижує ризик виникнення пожежонебезпечних ситуацій.

Вимоги до мікроклімату, вентиляції та рівня шуму робочих приміщень, у яких використовується розроблена вебсистема, регламентуються ДСН 3.3.6.042-99. Температурний режим повинен підтримуватись у межах 18-24 °С, відносна вологість повітря на рівні, що забезпечує комфортні умови праці, рівень шуму не перевищувати 50 дБ, а приміщення мають бути забезпечені природною або примусовою вентиляцією для підтримання належного повітрообміну та запобігання перевтомі персоналу.

Під час розробки та впровадження вебсистеми для ТОВ «Газмережі» було враховано комплекс вимог охорони праці та пожежної безпеки, що забезпечує безпечні, комфортні та ефективні умови роботи користувачів. Реалізація зазначених заходів сприяє зниженню рівня професійних ризиків, запобіганню нещасним випадкам і професійним захворюванням, а також повністю відповідає нормам чинного законодавства України.

4.2. Забезпечення безпеки життєдіяльності при роботі з ПК

Забезпечення безпеки життєдіяльності при роботі з персональними комп'ютерами та інформаційними системами є важливою складовою організації безпечних умов праці в сучасних цифрових середовищах. Активне впровадження інформаційних технологій у виробничі, управлінські та сервісні процеси зумовлює необхідність комплексного підходу до захисту користувачів, інформаційних ресурсів і технічної інфраструктури. Сучасні інформаційні системи виступають не лише інструментом автоматизації, а й критично важливим елементом діяльності підприємств, тому порушення їх функціонування може мати негативні наслідки як для персоналу, так і для безперервності виробничих процесів.

У процесі експлуатації програмного забезпечення та комп'ютерної техніки працівник зазнає впливу низки потенційно небезпечних і шкідливих факторів. До них належать інформаційні перевантаження, тривала робота з екранними

пристроями, підвищене психоемоційне напруження, висока відповідальність за прийняття управлінських рішень, а також ризики, пов'язані з помилками оброблення даних. Окрему групу небезпек становлять ризики порушення інформаційної безпеки, що можуть призвести до витоку конфіденційної інформації, неправомірного доступу до системи або зупинки її роботи. Крім того, небезпеки можуть бути пов'язані з надзвичайними ситуаціями техногенного, природного чи воєнного характеру, які впливають на стабільність роботи комп'ютерної техніки та інформаційних систем.

Відповідно до чинного законодавства України, питання безпеки життєдіяльності при роботі з комп'ютерною технікою та інформаційними системами регламентуються Законом України «Про інформацію» № 2657-ХІІ, Законом України «Про захист персональних даних» № 2297-VI, Законом України «Про основні засади забезпечення кібербезпеки України» № 2163-VIII, а також міжнародними стандартами у сфері управління інформаційною безпекою, зокрема ДСТУ ISO/IEC 27001:2015 та ДСТУ ISO/IEC 27002:2015 [36–38]. Зазначені нормативні документи визначають правові, організаційні та технічні засади безпечної експлуатації інформаційних систем, вимоги до оброблення, зберігання та передавання інформації, відповідальність користувачів і власників систем, а також загальні принципи побудови захищеного інформаційного середовища.

У контексті розробленої вебсистеми для ТОВ «Газмережі» безпека життєдіяльності розглядається як сукупність заходів, спрямованих на збереження життя і здоров'я користувачів, захист інформаційних ресурсів та забезпечення стійкої і безперебійної роботи цифрової інфраструктури підприємства. З огляду на те, що система використовується для роботи з виробничими, технологічними та персональними даними, її надійне функціонування є важливим чинником загальної безпеки підприємства. Порушення доступності або цілісності інформації може призвести до помилок в управлінських рішеннях, збоїв у виробничих процесах та створення небезпечних ситуацій для персоналу.

Особлива увага в межах системи приділяється мінімізації ризиків, пов'язаних із людським фактором, помилками користувачів, несанкціонованим доступом та

технічними відмовами обладнання. Згідно з положеннями Закону України «Про захист персональних даних», оброблення інформації про клієнтів та об'єкти газорозподільної інфраструктури повинно здійснюватися з дотриманням принципів законності, цільового використання, конфіденційності та обмеження доступу. Реалізація цих принципів у вебсистемі досягається шляхом розмежування прав доступу між різними категоріями користувачів, застосування механізмів автентифікації та авторизації, а також фіксації та контролю дій користувачів, що знижує ймовірність помилкових або зловмисних дій. Важливим складником безпеки життєдіяльності є захист користувачів від кіберзагроз, які можуть спричинити не лише втрату або спотворення даних, але й порушення нормального функціонування інформаційної системи, зупинку робочих процесів та виникнення аварійних ситуацій. Закон України «Про основні засади забезпечення кібербезпеки України» передбачає необхідність впровадження комплексу організаційних і технічних заходів для запобігання кіберінцидентам, що можуть створювати загрозу безпеці людей, інформаційним ресурсам та об'єктам критичної інфраструктури. У процесі експлуатації вебсистеми це реалізується через використання захищених каналів зв'язку, регулярне оновлення програмного забезпечення, резервне копіювання даних, моніторинг подій безпеки та контроль доступу до інформаційних ресурсів.

Безпека життєдіяльності при роботі з персональними комп'ютерами також тісно пов'язана з питаннями цивільної безпеки та підготовки персоналу до дій у надзвичайних ситуаціях. Як зазначено у методичному посібнику В. С. Стручка «Безпека в надзвичайних ситуаціях», сучасні інформаційні технології повинні експлуатуватися з урахуванням можливих аварійних ситуацій, зокрема відмов електроживлення, пожеж, техногенних аварій, кібератак та надзвичайних подій воєнного характеру, що є особливо актуальним для підприємств, діяльність яких пов'язана з критичною інфраструктурою [39]. У таких умовах важливого значення набуває наявність резервних джерел живлення, чітко визначених процедур аварійного відновлення та інструкцій для персоналу.

У навчальному посібнику В. С. Стручка «Техноекологія та цивільна безпека. Частина “Цивільна безпека”» підкреслюється, що безпека життєдіяльності в інформаційному середовищі повинна ґрунтуватися на принципі превентивності, тобто попередження небезпек шляхом правильного планування робочих процесів, інформування персоналу, проведення навчань і впровадження чітких алгоритмів дій у разі виникнення надзвичайних ситуацій [40]. Зазначений підхід є важливим елементом організації роботи користувачів із комп’ютерною технікою та інформаційними системами, а також формування культури безпечної праці в цифровому середовищі.

Безпека життєдіяльності при роботі з персональними комп’ютерами та інформаційними системами є комплексним завданням, що поєднує правові, організаційні та технічні заходи. Її забезпечення створює передумови для стабільної та безпечної експлуатації розробленої вебсистеми для ТОВ «Газмережі».

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було проведено комплексне дослідження предметної області газорозподільної сфери, визначено ключові проблеми та сформульовано цілі створення програмної системи InfoGas. На основі вимог замовника проаналізовано основні сценарії взаємодії користувачів і адміністратора, виявлено всі необхідні функціональні можливості та сформовано детальну модель майбутнього програмного продукту.

У процесі розроблення системи обґрунтовано вибір сучасних технологій і засобів розробки, побудовано архітектурну модель та логічну структуру, визначено ролі компонентів і їх взаємодію. База даних була спроектована з урахуванням вимог до цілісності, масштабованості та надійності зберігання даних, а для системної візуалізації створено UML-діаграми, які відобразили структуру, поведінку та основні функціональні процеси. Було розроблено інтерфейси користувача з акцентом на зрозумілість, естетичність та зручність роботи, що дозволило сформувати інтуїтивну логіку взаємодії.

На етапі програмної реалізації створено повноцінний вебзастосунок з розмежуванням функцій між користувачем та адміністратором, розроблено модулі для роботи з лічильниками, квитанціями, профілями, статистикою та документацією. Особливу увагу приділено інтеграції модуля штучного інтелекту, який забезпечує персоналізований аналіз споживання газу на основі даних користувача. Проведено тестування функціональних модулів, перевірено коректність бізнес-логіки, надійність обробки даних та відповідність очікуваним сценаріям використання. Отримані результати продемонстровано за допомогою реальних інтерфейсів, графіків, таблиць і модульних сценаріїв, що підтвердило якість реалізації системи.

Також виконано аналіз вимог охорони праці та безпеки життєдіяльності, пов'язаних з експлуатацією комп'ютерної техніки та інформаційних систем. Було досліджено нормативно-правові засади, що забезпечують безпечні умови роботи, захист інформації та мінімізацію ризиків під час взаємодії з цифровими ресурсами.

Результати кваліфікаційної роботи демонструють завершеність проєкту InfoGas як сучасної, функціональної та безпечної системи, яка відповідає вимогам замовника та може бути впроваджена для реального використання у сфері обліку й аналітики газоспоживання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до виконання кваліфікаційної роботи магістра для здобувачів спеціальності 121 – Інженерія програмного забезпечення, всіх форм навчання / укладачі: Михалик Д.М., Цуприк Г.Б., Бревус В.М., Мудрик І.Я. – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2024. – 44 с. URL: <http://elartu.tntu.edu.ua/handle/lib/50316>
2. Про компанію GRMU. URL: <https://grmu.com.ua/about/>
3. Монографія ТНТУ ім. І. Пулюя. Монографія. С. 20–21. URL: https://elartu.tntu.edu.ua/bitstream/lib/46522/1/Monohrafiya_2024.pdf
4. Що таке SCADA-система та як вона працює. Sinsmarts. URL: <https://www.sinsmarts.com/uk/blog/what-is-a-scada-system-and-how-does-it-work>
5. SAP TechEd Virtual. SAP. URL: <https://www.sap.com/events/teched/virtual.html>
6. CRM Integration: Guide & Tools. Semrush Blog. URL: <https://www.semrush.com/blog/crm-integration>
7. Василюк А. В. Інтегровані інформаційні системи підприємств: особливості впровадження та розвитку. Вісник Хмельницького національного університету. Економічні науки. С. 100–104. URL: <https://heraldes.khmnu.edu.ua/index.php/heraldes/article/view/1890/1927>
8. Agile: що це таке. Brainrain. URL: <https://brainrain.com.ua/uk/chto-takoe-agile-ua>
9. React Documentation. URL: <https://react.dev>
10. TailwindCSS Documentation. URL: <https://tailwindcss.com>
11. Recharts Documentation. URL: <https://recharts.github.io>
12. SweetAlert2 Documentation. URL: <https://sweetalert2.github.io>
13. Firebase Firestore Documentation. URL: <https://firebase.google.com/docs/firestore>
14. Firebase Authentication Documentation. URL: <https://firebase.google.com/docs/auth>

15. Poe API: Introducing the Poe API. URL: <https://poe.com/blog/introducing-the-poe-api>
16. React Hooks API Reference. URL: <https://react.dev/reference/react/hooks>
17. React Context API (Legacy). URL: <https://legacy.reactjs.org/docs/context.html>
18. Що таке бази даних NoSQL. Freehost. URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-bazi-dannih-nosql>
19. UML-діаграми: огляд. Foxminded. URL: <https://foxminded.ua/uml-diagramy>
20. UML Class Diagram: що це таке. MindOnMap. URL: <https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram>
21. Sequence Diagrams: приклади та пояснення. Макс Зосім. URL: <https://www.maxzosim.com/sequence-diagrams>
22. Методичні вказівки з дисципліни «ЕМТД». Житомирська політехніка. URL: <https://emtd.ztu.edu.ua/view/word-3021/0>
23. Deployment Diagram – UML Example. Guru99. URL: <https://www.guru99.com/uk/deployment-diagram-uml-example.html>
24. What is a Use Case and What Are They For? QATestLab. URL: <https://training.qatestlab.com/blog/technical-articles/what-is-a-use-case-and-what-are-they-for>
25. Methods of constructing algorithms for comparative test statistical verification of mathematical models of bioobject responses to low-intensity stimuli / Bohdan Yavorskyu, Evhenia Yavorska, Halyna Tsupryk, Roman Kinash // Scientific Journal of TNTU. — Tern.: TNTU, 2023. — Vol 112. — No 4. — P. 82–90.
26. Олянін, Д., Цуприк, Г. (2025) Transformer Neural Networks in Industry 4.0 / Д. Олянін, Г. Цуприк, Т. Говорущенко, О. Багрій-Заяць, І. Андрущак // Computer Information Technologies in Industry 4.0: proceedings of the 3rd International Workshop (CITI-2025), Ternopil, Ukraine, 11–12 June 2025. – Ternopil : Ternopil Ivan Puluj National Technical University, 2025 (Scopus) <https://ceur-ws.org/Vol-4057/>

27. Tsupryk H. LLM-based Extraction from Resumes / D. Olianin, H. Tsupryk // *Advanced Technologies in Scientific Research: collection of scientific papers with proceedings of the 1st International Scientific and Practical Conference, Rotterdam, Netherlands, 20–22 August 2025.* – International Scientific Unity, 2025. – 72-76

28. Methods of constructing algorithms for comparative test statistical verification of mathematical models of bioobject responses to low-intensity stimuli / Bohdan Yavorsky, Evhenia Yavorska, Halyna Tsupryk, Roman Kinash // *Scientific Journal of TNTU.* — Tern.: TNTU, 2023. — Vol 112. — No 4. — P. 82–90.

29. Олянін, Д., Цуприк, Г. (2025) Transformer Neural Networks in Industry 4.0 / Д. Олянін, Г. Цуприк, Т. Говорущенко, О. Багрій-Заяць, І. Андрущак // *Computer Information Technologies in Industry 4.0: proceedings of the 3rd International Workshop (CITI-2025), Ternopil, Ukraine, 11–12 June 2025.* – Ternopil : Ternopil Ivan Puluj National Technical University, 2025 (Scopus) <https://ceur-ws.org/Vol-4057/>

30. React Router: Link API. URL: https://api.reactrouter.com/v7/functions/react_router.Link.html

31. Наказ Міністерства соціальної політики України № 207 від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». URL: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>

32. Закон України «Основи законодавства України про охорону здоров'я» № 2801-XII. URL: <https://zakon.rada.gov.ua/laws/show/2801-12#Text>

33. Кодекс законів про працю України. URL: <https://zakon.rada.gov.ua/laws/show/322-08#Text>

34. Наказ Міністерства охорони здоров'я України № 400 від 31.03.2010 «Про затвердження ДСанПіН 3.3.2-007-09». URL: <https://zakon.rada.gov.ua/laws/show/z0293-10>

35. Закон України «Про основні засади забезпечення кібербезпеки України» № 2163-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2163-19#Text>

36. Нормативно-правова база у сфері податкового законодавства. ДПС України. URL: <https://tax.gov.ua/dlya-gromadskosti/dpa-i-gromadskist/normativno-pravova-baza-u-sferi/arhiv-normativno-pravova-baza/53366.html>

37. Закон України «Про захист персональних даних» № 2297-VI. URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>

38. ДСТУ ISO/IEC 27001:2015. Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=66910

39. Стручок В.С. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» / Тернопіль: ФОП Паляниця В. А., –156 с. Отримано з <https://elartu.tntu.edu.ua/handle/lib/39196>.

40. Стручок В.С. Навчальний посібник «ТЕХНОЕКОЛОГІЯ ТА ЦИВІЛЬНА БЕЗПЕКА. ЧАСТИНА «ЦИВІЛЬНА БЕЗПЕКА»» / Тернопіль: ФОП Паляниця В. А., – 156 с. Отримано з <http://elartu.tntu.edu.ua/handle/lib/39424>.

ДОДАТКИ

Додаток А. Апробація

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

ХІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



17-18 грудня 2025 року

**ТЕРНОПІЛЬ
2025**

УДК 004.42

Т. Соловій; Г. Цуприк, к. т. н., доц.

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

**РОЗРОБКА СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ
З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ REACT
ТА УДОСКОНАЛЕННЯМ РБД ТЕХНОЛОГІЄЮ FIREBASE**

UDC 004.42

T. Soloviy; H. Tsupryk, PhD., Assoc.Prof.

**DEVELOPMENT OF A DATA PROCESSING SYSTEM USING REACT TECHNOLOGIES
AND IMPROVEMENT OF RDB WITH FIREBASE**

Галузь газопостачання є одним із ключових сегментів енергетичної інфраструктури, де ефективне управління інформаційними системами безпосередньо впливає на стабільність і безпеку енергопостачання. Сучасні цифрові трансформації сприяють переходу від традиційних систем баз даних до інтелектуальних платформ, що інтегрують технології штучного інтелекту для аналізу, прогнозування та підтримки прийняття рішень.

Метою роботи є узагальнення можливостей використання технологій штучного інтелекту для підвищення ефективності управління даними в системах газопостачання.

Проведений аналіз показав, що впровадження технологій штучного інтелекту та робота з великими даними у сферу газопостачання сприяє комплексній оптимізації управління даними [1–4]. Зокрема, алгоритми машинного навчання забезпечують автоматичне виявлення аномалій у показниках споживання, прогнозування витрат і попиту, а також формування рекомендацій для операторів систем. Це мінімізує людський фактор, підвищує точність рішень і скорочує час реакції на відхилення у споживанні. Інтеграція аналітичних модулів штучного інтелекту у внутрішні бази даних дає змогу здійснювати статистичний аналіз у режимі, наближеному до реального часу. Такі системи можуть автоматично створювати звіти для різних рівнів управління, формувати попередження про критичні стани обладнання та прогнозувати ризики аварій чи порушень постачання.

Використання технологій Big Data у поєднанні з хмарними платформами дозволяє масштабувати обробку великих потоків даних, що надходять із IoT-сенсорів, встановлених у мережах постачання. Це створює єдине інформаційне середовище, де дані з різних джерел (лічильники, SCADA-системи, клієнтські інтерфейси) об'єднуються для побудови інтегрованих моделей прогнозування [1–2]. Згідно з дослідженнями Abdelkader Baaziz і Luc Quoniam, ефективність управління зростає завдяки використанню алгоритмів прогнозувальної аналітики, що дозволяють не лише аналізувати історичні показники, а й виявляти закономірності, які дають змогу попереджати перевитрати та аварійні ситуації [3–4]. Таким чином, аналітичні системи на основі ШІ формують базу для прийняття стратегічних рішень щодо модернізації інфраструктури, енергоефективності та автоматизації контролю.

Інтелектуалізація управління даними в галузі газопостачання визначається як стратегічний напрям цифрової трансформації, що забезпечує ефективність, прозорість і безпеку функціонування галузі. Поєднання оптимізованих баз даних, технологій Big Data і штучного інтелекту створює основу для формування гнучких інформаційних систем, які здатні до самоаналізу, адаптації та прогнозування.

Такі системи відкривають можливості для створення інтелектуальних консультаційних сервісів, підвищення енергоефективності та зниження операційних ризиків у газовій інфраструктурі.

Література

1. Олянін, Д., Цуприк, Г. (2025) Transformer Neural Networks in Industry 4.0 / Д. Олянін, Г. Цуприк, Т. Говорущенко, О. Багдій-Заяць, І. Андрушак // Computer Information Technologies in Industry 4.0: proceedings of the 3rd International Workshop (CITI-2025), Ternopil, Ukraine, 11–12 June 2025. – Ternopil : Ternopil Ivan Puluj National Technical University, 2025 (Scopus)

2. Tsupryk, H., Olianin, D. (2025). Vydobuvannia danyh z tekstu vykorystovuiuchy transformerni neironni merezhi [Data extraction from text using Transformer Neural Networks]. Information Technology: Computer Science, Software Engineering and Cyber Security, 125–130, DOI: <https://doi.org/10.32782/IT/2025-2-13>
3. Tsupryk, H., Olianin, D. (2025). Overview of transformers role in data mining from unstructured data. International Scientific-technical journal «Measuring and computing devices in technological processes» 2025, Issue 2, 125–130, DOI: 10.31891/2219-9365-2025-82-52
4. Tsupryk H. LLM-based Extraction from Resumes / D. Olianin, H. Tsupryk // Advanced Technologies in Scientific Research: collection of scientific papers with proceedings of the 1st International Scientific and Practical Conference, Rotterdam, Netherlands, 20–22 August 2025. – International Scientific Unity, 2025. – 72-76
5. Hussain, M. (2023). Adoption of big data analytics for energy pipeline condition monitoring: Oil and gas sector. Energy Policy, (XX). DOI: <https://www.sciencedirect.com/science/article/pii/S0308016123001783>
6. Abdelkader Baaziz, Luc Quoniam: "How to use Big Data technologies to optimize operations in Upstream Petroleum Industry".

Рисунок А1.Збірник Тез

УДК 004.42

Тарас Соловій, Галина Цуприк, канд. техн. наук, доц.

Тернопільський національний технічний університет імені Івана Пулюя

РОЗРОБКА СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ REACT ТА УДОСКОНАЛЕННЯМ РБД ТЕХНОЛОГІЄЮ FIREBASE

Taras Soloviy, Halyna Tsupryk, PhD, Assoc.Prof.

DEVELOPMENT OF A DATA PROCESSING SYSTEM USING REACT TECHNOLOGIES AND IMPROVEMENT OF RDB WITH FIREBASE

Галузь газопостачання є одним із ключових сегментів енергетичної інфраструктури, де ефективне управління інформаційними системами безпосередньо впливає на стабільність і безпеку енергопостачання. Сучасні цифрові трансформації сприяють переходу від традиційних систем баз даних до інтелектуальних платформ, що інтегрують технології штучного інтелекту для аналізу, прогнозування та підтримки прийняття рішень.

Метою роботи є узагальнення можливостей використання технологій штучного інтелекту для підвищення ефективності управління даними в системах газопостачання.

Проведений аналіз показав, що впровадження технологій штучного інтелекту та робота з великими даними у сферу газопостачання сприяє комплексній оптимізації управління даними [1-4]. Зокрема, алгоритми машинного навчання забезпечують автоматичне виявлення аномалій у показниках споживання, прогнозування витрат і попиту, а також формування рекомендацій для операторів систем. Це мінімізує людський фактор, підвищує точність рішень і скорочує час реакції на відхилення у споживанні. Інтеграція аналітичних модулів штучного інтелекту у внутрішні бази даних дає змогу здійснювати статистичний аналіз у режимі, наближеному до реального часу. Такі системи можуть автоматично створювати звіти для різних рівнів управління, формувати попередження про критичні стани обладнання та прогнозувати ризики аварій чи порушень постачання.

Використання технологій Big Data у поєднанні з хмарними платформами дозволяє масштабувати обробку великих потоків даних, що надходять із IoT-сенсорів, встановлених у мережах постачання. Це створює єдине інформаційне середовище, де дані з різних джерел (лічильники, SCADA-системи, клієнтські інтерфейси) об'єднуються для побудови інтегрованих моделей прогнозування [1-2]. Згідно з дослідженнями Abdelkader Baaziz і Luc Quoniam, ефективність управління зростає завдяки використанню алгоритмів прогнозової аналітики, що дозволяють не лише аналізувати історичні показники, а й виявляти закономірності, які дають

змогу попереджати перевитрати та аварійні ситуації [3-4]. Таким чином, аналітичні системи на основі ШІ формують базу для прийняття стратегічних рішень щодо модернізації інфраструктури, енергоефективності та автоматизації контролю.

Інтелектуалізація управління даними в галузі газопостачання визначається як стратегічний напрям цифрової трансформації, що забезпечує ефективність, прозорість і безпеку функціонування галузі. Поєднання оптимізованих баз даних, технологій Big Data і штучного інтелекту створює основу для формування гнучких інформаційних систем, які здатні до самоаналізу, адаптації та прогнозування.

Такі системи відкривають можливості для створення інтелектуальних консультаційних сервісів, підвищення енергоефективності та зниження операційних ризиків у газовій інфраструктурі.

Література

1. Олянін, Д., Цуприк, Г. (2025) Transformer Neural Networks in Industry 4.0 / Д. Олянін, Г. Цуприк, Т. Говорущенко, О. Багрій-Заяць, І. Андрущак // Computer Information Technologies in Industry 4.0: proceedings of the 3rd International Workshop (CITI-2025), Ternopil, Ukraine, 11–12 June 2025. – Ternopil : Ternopil Ivan Puluj National Technical University, 2025 (Scopus)

2. Tsupryk, H., Olianin, D. (2025). Vydobuvannia danyh z tekstu vykorystovuiuchy transformerni neironni merezhi [Data extraction from text using Transformer Neural Networks]. Information Technology: Computer Science, Software Engineering and Cyber Security, 125–130, DOI: <https://doi.org/10.32782/IT/2025-2-13>

3. Tsupryk, H., Olianin, D. (2025). Overview of transformers role in data mining from unstructured data. International Scientific-technical journal «Measuring and computing devices in technological processes» 2025, Issue 2, 125–130, DOI: 10.31891/2219-9365-2025-82-52

4. Tsupryk H. LLM-based Extraction from Resumes / D. Olianin, H. Tsupryk // Advanced Technologies in Scientific Research: collection of scientific papers with proceedings of the 1st International Scientific and Practical Conference, Rotterdam, Netherlands, 20–22 August 2025. – International Scientific Unity, 2025. – 72-76

5. Hussain, M. (2023). Adoption of big data analytics for energy pipeline condition monitoring: Oil and gas sector. Energy Policy, (XX). DOI: <https://www.sciencedirect.com/science/article/pii/S0308016123001783>

6. Abdelkader Baaziz, Luc Quoniam: "How to use Big Data technologies to optimize operations in Upstream Petroleum Industry".

Додаток Б. Діаграми варіантів використання

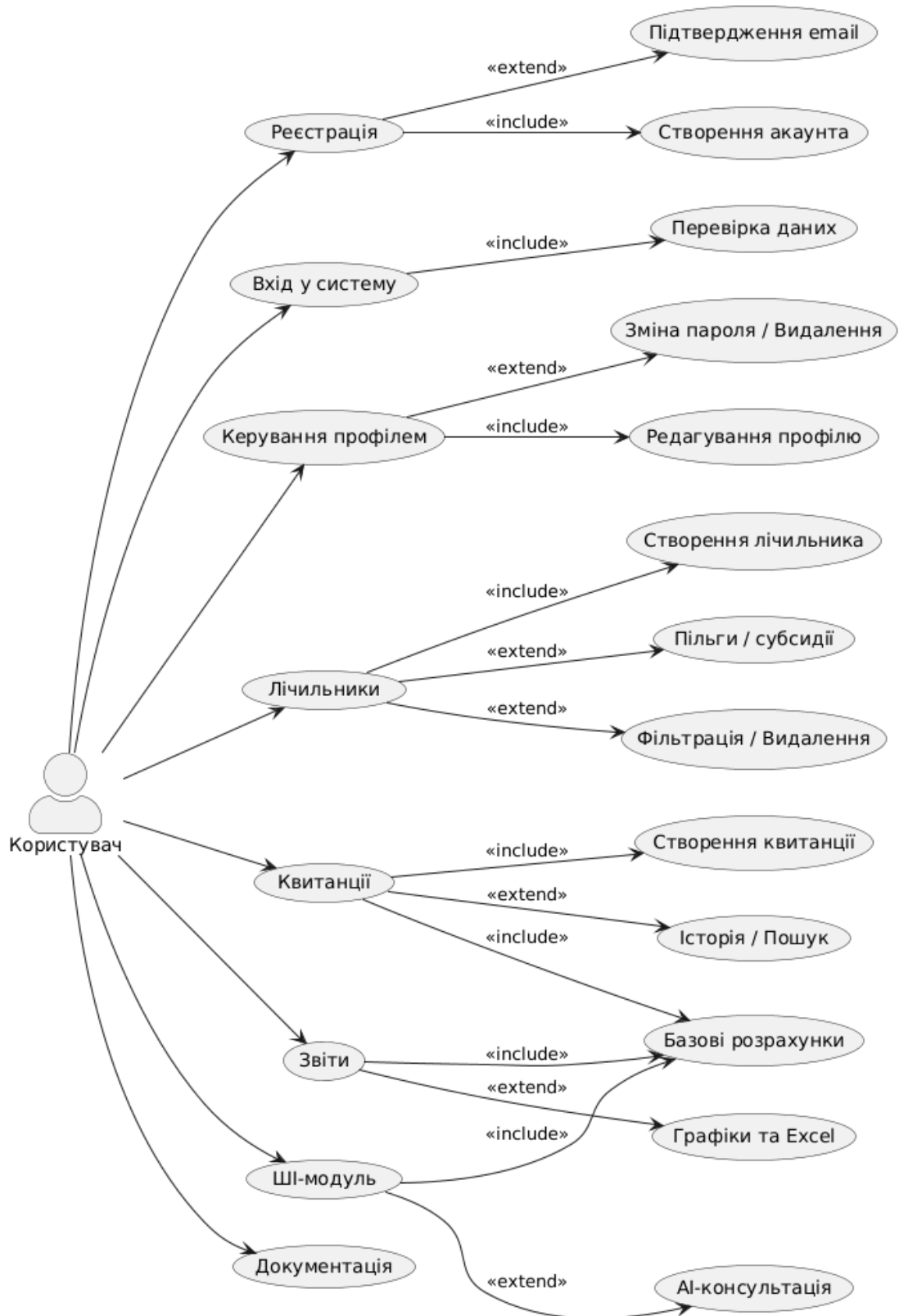


Рис. Б.1. – Діаграма варіантів використання для користувача

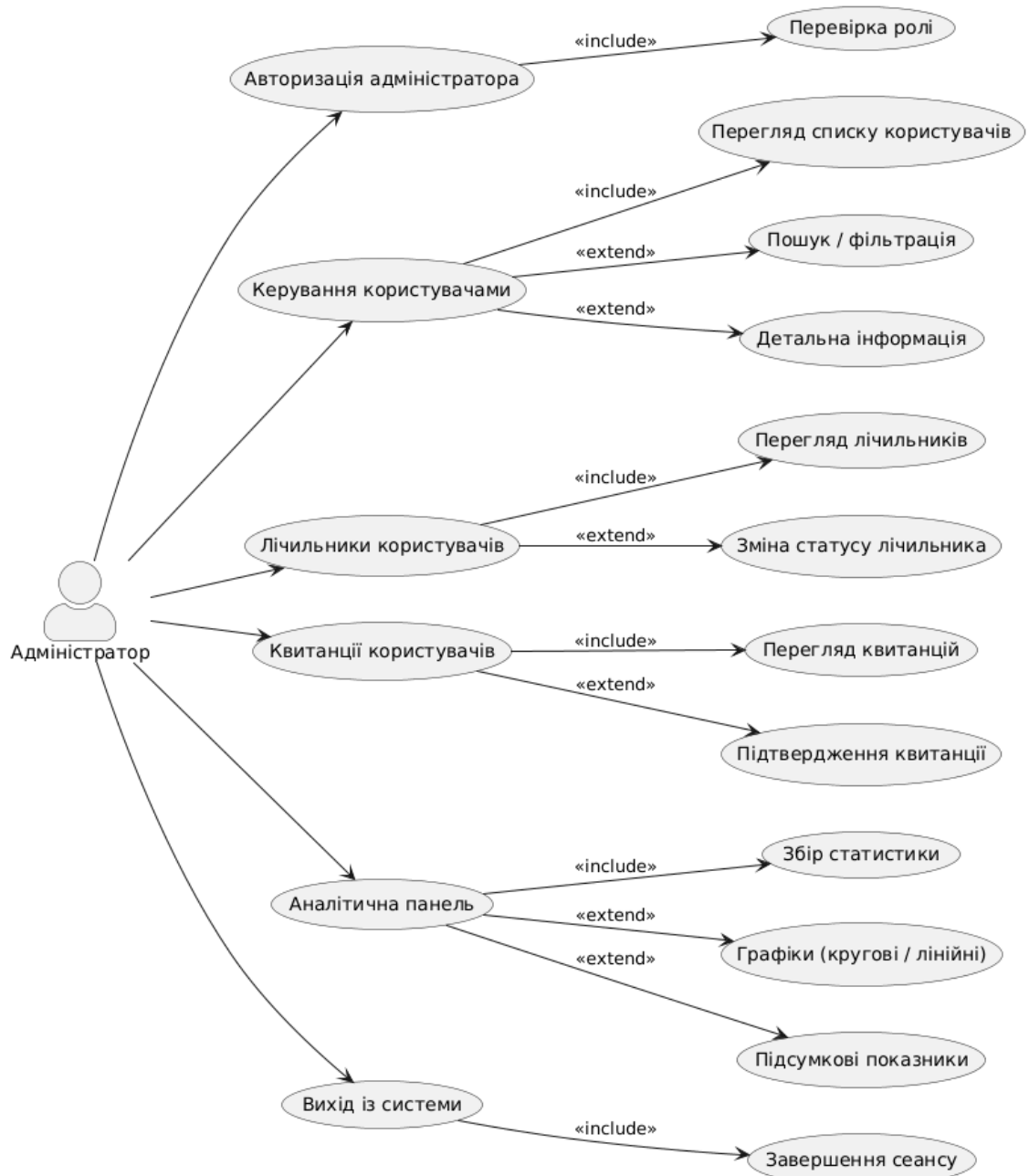


Рис. Б.2. – Діаграма варіантів використання для адміністратора

Додаток В. Лістинг ключових компонентів системи

Лістинг коду розміщено на Github за посиланням:
<https://github.com/tarassolovey/infogas.git>