

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут управління, технологій та правових наук
Кафедра інформаційних технологій

МАГІСТЕРСЬКА РОБОТА

на тему:

БІЛІНГОВА СИСТЕМА CALL-ЦЕНТРУ

Спеціальність: 121 «Інженерія програмного забезпечення»
ОПП «Інженерія програмного забезпечення»

Розробив: Анастасія ЗІНЧЕНКО

(ім'я, ПРІЗВИЩЕ)

(підпис)

Керівник: к.ф.-м.н., доцент Олександр ТКАЧЕНКО

(науковий ступінь, вчене звання) (ім'я, ПРІЗВИЩЕ)

(підпис)

Нормоконтроль: к.ф.-м.н., доцент Олександр ТКАЧЕНКО

(науковий ступінь, вчене звання) (ім'я, ПРІЗВИЩЕ)

(підпис)

Допущений до захисту

«_____» _____ 2025р.

В.о. завідувача кафедри: д.т.н., професор Олег НІКОНОВ

(науковий ступінь, вчене звання) (ім'я, ПРІЗВИЩЕ)

(підпис)

Рецензент: к.т.н., доцент Ірина МАШКІНА

(науковий ступінь, вчене звання) (ім'я, ПРІЗВИЩЕ)

(підпис)

Київ – 2025

Міністерство освіти і науки України
Національний транспортний університет
Навчально-науковий інститут управління, технологій та правових наук
Кафедра інформаційних технологій

Рівень вищої освіти – другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
ОПП «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
інформаційних технологій
д.т.н., проф. _____ Олег НІКОНОВ

«_____» _____ 2025 р.

**Завдання
на магістерську роботу здобувачу вищої освіти
Анастасія ЗІНЧЕНКО**

(ім'я, прізвище)

1. Тема роботи: Білінгова система Call-центру

керівник роботи: к.ф.-м.н., доцент Олександр ТКАЧЕНКО

Затверджені наказом від «17» жовтня 2025 року № 1031

2. Строк подання здобувачем роботи 19 грудня 2025 року

3. Вихідні дані до роботи: Білінгова система Call-центру, база даних системи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Програмна реалізація роботи білінгової системи сутність сучасних проблем в білінгових системах створення інноваційної системи білінга проєктування бази даних, опис предметної області, аналіз проблеми на існуючому аналозі, створення графічного інтерфесу

5. Перелік графічного матеріалу: презентація магістерської роботи та скріншорти функціонування програмного продукту “Billing”

6. Консультанти роботи: к.ф.-м.н., доцент Олександр ТКАЧЕНКО

7. Календарний план виконання роботи

№ з/п	Найменування етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітки
1	Огляд та аналіз джерел тематичної області	17.10.25 – 22.10.25	
2	Огляд та аналіз шляхів розв'язання проблем роботи	23.10.25 – 28.10.25	
3	Опис власного підходу до вирішення проблеми	29.10.25 – 04.11.25	
4	Визначення вимог до програмного продукту та його структури	05.11.25 – 07.11.25	
5	Опис та реалізація основних функцій програмного продукту	08.11.25 – 28.11.25	
6	Оформлення пояснювальної записки	29.11.25 – 19.12.25	
7	Захист		

8. Дата видачі завдання «17» жовтня 2025 року

Здобувач вищої освіти _____ Зінченко А.М.
 (підпис) (прізвище та ініціали)

Керівник роботи _____ Ткаченко О.А.
 (підпис) (прізвище та ініціали)

РЕФЕРАТ

Тема: Білінгова система Call-центру

Магістерська робота: 74 с., 25 рис., 2 таблиць, 26 літературних джерела, 3 додатки.

Магістерська робота присвячена розробці білінгової системи для оптимізації роботи в цій сфері. Метою роботи є дослідження предметної області та розробка білінгової системи на основі технічних і соціальних даних. Система реалізує функціонал для ведення абонентських даних, автоматичного розрахунку вартості послуг відповідно до тарифних планів, а також контролю стану підключення користувачів. Для реалізації програмного продукту застосовано сучасні веб-технології, зокрема HTML, CSS, мову програмування JavaScript та систему керування базами даних.

У процесі розробки проведено аналіз вимог до білінгових систем щодо точності фінансових розрахунків, обробки платежів, ведення історії змін та інтеграції з мережевим обладнанням. В роботі сформульовано функціональні вимоги до системи та визначено взаємодію із зовнішніми сервісами та мережевими протоколами для забезпечення актуальності даних про стан підключення абонентів.

Результатом роботи є білінгова система, яка надає можливість провайдеру ефективно керувати абонентською базою, здійснювати контроль платежів та нарахувань, отримувати інформацію про технічний і фінансовий стан абонентів. Система забезпечує зручний інтерфейс користувача для навігації між основними модулями та швидкого доступу до необхідних даних.

Розроблена система має потенціал для подальшого розширення та вдосконалення, шляхом інтеграції аналітичних модулів, автоматизованого виявлення заборгованостей, а також використання інтелектуальних алгоритмів для прогнозування фінансових показників і підвищення ефективності управління абонентськими послугами.

Ключові слова: білінгова система, Call-центр, послуги, база даних.

ABSTRACT

Topic: Call Center Billing System

Master's Thesis: 74 p., 25 figures, 2 tables, 25 references, 3 appendices.

The master's thesis is devoted to the development of a billing system for improving work in this area. The purpose of the work is to study the subject area and develop a billing system based on technical and social data.

The research focuses on the development of a billing system for a network service provider that provides subscriber accounting, billing control, and technical operations management. The system implements the main functionality for maintaining subscriber data, automatically calculating the cost of services according to tariff plans, and monitoring user connection status. Modern web technologies, including HTML, CSS, the JavaScript programming language, and a database management system, were used to implement the software product.

During the development process, an analysis of providers' requirements for billing systems was conducted, in particular regarding the accuracy of financial calculations, payment processing, maintaining a history of changes, and integration with network equipment. Based on the results obtained, key functional requirements for the system were formulated and the need for interaction with external services and network protocols was determined to ensure the relevance of data on subscriber connection status.

The result of the work is a billing system that enables provider employees to effectively manage the subscriber base, control payments and billing, and promptly receive information about the technical and financial status of subscribers. The system provides a convenient user interface for navigation between the main modules and quick access to the necessary data.

The developed system has the potential for further expansion and improvement, in particular through the integration of analytical modules, automated

detection of debts, as well as the use of intelligent algorithms for forecasting financial indicators and improving the efficiency of subscriber service management.

Keywords: billing system, Call Center, services, database, system.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕМАТИЧНОЇ ОБЛАСТІ БІЛІНГОВИХ СИСТЕМ CALL-ЦЕНТРУ	10
1.1. Сучасний стан цифрових систем обліку й тарифікації в Call-центрах.....	10
1.2. Аналіз існуючих підходів до тарифікації дзвінків і послуг	12
1.3. Огляд програмних рішень для білінгу	13
1.4. Проблеми та обмеження сучасних білінгових систем	15
РОЗДІЛ 2. АНАЛІЗ І ШЛЯХИ РОЗВ’ЯЗАННЯ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО БІЛІНГУ CALL-ЦЕНТРУ	23
2.1. Постановка задачі.....	24
2.2. Аналіз алгоритмів тарифікації та обліку дзвінків	27
2.3. Розробка алгоритму роботи білінгової підсистеми	32
2.4. Проектування архітектури білінгового програмного продукту	33
РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ “Billing”	39
3.1 Функціональні та нефункціональні вимоги до програмного продукту	39
3.2 Структурна модель системи «Billing».....	45
3.3 Опис функціонування веб-орієнтованої системи «Billing»	50
ВИСНОВОК	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	73
ДОДАТКИ	76
Додаток А: Діаграми прецедентів	76

ВСТУП

Актуальність проблеми. У сучасних умовах інтенсивного розвитку телекомунікаційних послуг ефективне управління ресурсами call-центру стає вирішальним чинником його конкурентоспроможності. Особливого значення набувають системи автоматизованого білінгу, які забезпечують точний облік, тарифікацію та аналіз телекомунікаційних послуг. Від їх надійності та функціональності безпосередньо залежать фінансові результати діяльності компанії, якість обслуговування клієнтів та оперативність прийняття управлінських рішень.

Існуючі білінгові системи часто не відповідають сучасним вимогам гнучкості, масштабованості та інтеграції з іншими бізнес-процесами. Це обумовлює необхідність розробки нових рішень, здатних ефективно обробляти великі обсяги даних, адаптуватися до мінливих тарифних планів та забезпечувати прозорість фінансових розрахунків.

Зв'язок роботи з науковими дослідженнями. Дослідження базуються на фундаментальних працях у галузі телекомунікацій, систем реального часу, обробки великих даних та проектування програмного забезпечення.

Метою дослідження є розробка автоматизованої білінгової системи для call-центру, здатної підвищити ефективність обліку та тарифікації телекомунікаційних послуг.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Провести аналіз сучасного стану та визначити недоліки існуючих білінгових систем.
- Дослідити алгоритми тарифікації та методології обробки деталізованих записів про дзвінки (CDR).
- Розробити білінгову систему Call-центру.
- Реалізувати механізми тарифікації, формування звітності та моніторингу.

Об'єкт дослідження: процеси обліку та тарифікації телекомунікаційних послуг у call-центрах.

Предмет дослідження: методи, алгоритми та програмні рішення для автоматизації білінгових систем.

Наукова новизна дослідження полягає в удосконаленні архітектури білінгових систем шляхом розробки гнучких механізмів тарифікації, оптимізації процесів обробки CDR-даних та реалізації комплексної системи моніторингу та аналітики.

Практична цінність дослідження полягає в розробці програмного продукту, який може бути впроваджений у роботі call-центрів для автоматизації фінансових розрахунків, підвищення точності тарифікації та зменшення операційних витрат.

Робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі досліджено теоретичні основи білінгових систем. У другому розділі представлено розробку автоматизованої білінгової системи. У третьому розділі описано реалізацію та апробацію системи "Billing1".

Особистий внесок магістра. Магістерська робота є самостійним завершеним дослідженням автора.

Публікації: дослідження були апробовані:

– Ткаченко О.І., Зінченко А.М. Білінгова система Call-центру – інструмент оптимізації роботи та аналітики обслуговування клієнтів.

Актуальні питання розвитку галузей: збірник наукових праць з матеріалами VI Міжнародної наукової конференції (м. Київ, 26 листопада, 2025), 2025. С.28 – 32.[27]

– результати дослідження опубліковані в науковій статті Ткаченко О.І., Ткаченко К.О., Зінченко А.М. (2024). Вебзастосунок «Медична послуга». *IT SYNERGY*. 1. С. 74-91. <https://doi.org/10.53920/ITS-2024-1-6> [26]

1. ОГЛЯД ТА АНАЛІЗ ТЕМАТИЧНОЇ ОБЛАСТІ БІЛІНГОВИХ СИСТЕМ CALL-ЦЕНТРУ

Розглянемо сучасний стан та особливості функціонування білінгових систем у call-центрах, здійснюючи систематизацію теоретичних знань та практичних рішень у галузі автоматизованого обліку та тарифікації телекомунікаційних послуг [8].

У рамках розділу проводиться критичний огляд існуючих технологічних рішень, аналізуються переваги та недоліки різних підходів до організації білінгових процесів. Особлива увага приділяється вивченню архітектурних особливостей сучасних систем, аналізу методів тарифікації, а також оцінці проблемних аспектів, що виникають при їх експлуатації в реальних умовах роботи call-центрів.

Проведене дослідження дозволяє ідентифікувати ключові виклики, що стоять перед розробниками білінгових систем, та сформулювати вимоги до сучасних рішень, здатних ефективно функціонувати в умовах динамічно змінюючогося ринку телекомунікаційних послуг. Отримані результати створюють теоретичний фундамент для подальшої розробки власної білінгової системи, орієнтованої на потреби сучасних українських підприємств.

1.1 Сучасний стан цифрових систем обліку й тарифікації в Call-центрах

Сучасні системи обліку та тарифікації в call-центрах зазнали значних змін під впливом цифровізації та розвитку телекомунікаційних технологій. Від простого підрахунку тривалості дзвінків вони перетворилися на складні

багатофункціональні комплекси, які інтегрують облік, аналітику та управління клієнтськими взаємодіями [2]. Основними тенденціями розвитку є:

- Перехід від локальних рішень до хмарних архітектур, що забезпечують масштабованість та гнучкість

- Інтеграція з CRM-системами та іншими бізнес-додатками
- Автоматизація процесів тарифікації на основі штучного інтелекту
- Реалізація систем реального часу для миттєвого обліку та контролю

Аналіз бізнес-моделей сучасних провайдерів зв'язку (на прикладі «фРІНЕТ»)

Компанія «фРІНЕТ» реалізує інноваційну бізнес-модель, що поєднує традиційні телекомунікаційні послуги з сучасними цифровими рішеннями [25]. Основні характеристики моделі включають:

Гібридну тарифікацію:

- Поєднання фіксованої абонентської плати за базові послуги
- Гнучка система оплати за фактичним використанням додаткових сервісів

- Можливість комбінування різних тарифних пакетів
- Сервіс-орієнтований підхід:
- Інтеграція послуг IP-телефонії з інфраструктурою передачі даних
- Розробка індивідуальних рішень для корпоративних клієнтів
- Впровадження віртуальних АТС з розширеним функціоналом
- Технологічні інновації:
- Використання хмарних технологій для забезпечення доступності послуг

- Впровадження AI-інструментів для оптимізації тарифних пропозицій

- Автоматизація процесів обліку та звітності

Вплив технологій Big Data та AI на розвиток білінгових систем

Технології Big Data та штучного інтелекту значно розширили функціональні можливості сучасних білінгових систем:

Аналітичні можливості:

- Обробка великих масивів даних про використання послуг у реальному часі
- Виявлення паттернів поведінки клієнтів
- Прогнозування навантаження на основі історичних даних
- Інтелектуальна тарифікація:
- Автоматичне виявлення аномалій та потенційного шахрайства
- Адаптація тарифних планів на основі аналізу поведінки користувачів
- Персоналізація пропозицій з використанням машинного навчання
- Оптимізація бізнес-процесів:
- Автоматизація процесів виставлення рахунків
- Інтелектуальна обробка звернень клієнтів
- Підвищення точності прогнозування доходів

Впровадження цих технологій дозволяє створювати проактивні системи, здатні не лише фіксувати факти надання послуг, але й аналізувати їх ефективність, пропонувати оптимізації та покращувати якість обслуговування клієнтів [1].

1.2 Аналіз існуючих підходів до тарифікації дзвінків і послуг

Сучасна практика тарифікації в call-центрах охоплює різноманітні підходи, що формуються під впливом технологічного прогресу та ринкових вимог. Основними моделями є похвилинна тарифікація, яка забезпечує пряму кореляцію між тривалістю розмови та її вартістю; пакетна модель, що передбачає попередню оплату пакету хвилин з подальшим списанням по мірі використання; та абонентська система з фіксованою платою незалежно від інтенсивності використання [10].

Окрім цих класичних підходів, все більшого поширення набувають спеціалізовані моделі, такі як тарифікація за результатом, де вартість послуги залежить від досягнення певного бізнес-результату, та динамічна тарифікація, що враховує поточне навантаження call-центру, часові інтервали та пріоритетність обслуговування [11]. Також варто відзначити багаторівневу тарифікацію, яка диференціює вартість за типом обслуговування, категорією складності запиту та необхідною кваліфікацією оператора.

Також варто відзначити багаторівневу тарифікацію, яка диференціює вартість за типом обслуговування, категорією складності запиту та необхідною кваліфікацією оператора.

Вибір конкретної моделі тарифікації залежить від низки факторів, включаючи цільовий сегмент клієнтів (B2B чи B2C), специфіку бізнес-процесів, рівень конкуренції на ринку та технологічні можливості самої білінгової системи.

Для B2B-сегменту typically використовуються індивідуальні тарифні плани з комбінацією фіксованої та змінної складових, тоді як для масового B2C-ринку перевага часто віддається стандартизованим пакетним пропозиціям [2].

1.3 Огляд програмних рішень для білінгу

Сучасний ринок програмних рішень для білінгу в call-центрах пропонує широкий спектр продуктів, які можна класифікувати за рівнем складності, вартістю та функціональними можливостями.

Аналіз вбудованих білінгових модулів у сучасних АТС

Вбудовані білінгові модулі в сучасних АТС забезпечують базові функції тарифікації без необхідності інтеграції зовнішніх систем [15]. Найпоширенішими рішеннями є:

- Asterisk Billing – open-source рішення з відкритим кодом, що дозволяє глибоку кастомізацію під специфічні потреби бізнесу
- FreePBX – пропонує інтуїтивний веб-інтерфейс для налаштування тарифних правил і генерації звітів
- 3CX Phone System – включає вбудовані інструменти для тарифікації та аналітики викликів
- Avaya IP Office – забезпечує комплексне рішення для малих і середніх підприємств

Таблиця 1.1. Порівняльна характеристика вбудованих білінгових модулів

Параметр	Asterisk Billing	FreePBX	3CX	Avaya IP Office
Ліцензування	Відкрите	Відкрите	Комерційне	Комерційне
Складність налаштування	Висока	Середня	Низька	Середня
Гнучкість тарифікації	Дуже висока	Висока	Середня	Середня
Інтеграція з CRM	Обмежена	Середня	Добра	Добра
Вартість володіння	Низька	Низька	Середня	Висока
Підтримка VAS	Так	Так	Обмежена	Так

Огляд ринку CRM-систем з інтегрованими білінговими функціями

Сучасні CRM-системи все частіше включають білінгові модулі як частину єдиного інформаційного простору для управління клієнтськими взаємодіями [6]. Серед найпопулярніших рішень:

- Salesforce Service Cloud - пропонує комплексне рішення з інтегрованими функціями тарифікації та управління платежами
- Zendesk - забезпечує єдиний інтерфейс для обробки запитів, управління клієнтською базою та білінгу
- HubSpot Service Hub - поєднує інструменти обслуговування клієнтів з базовими функціями білінгу
- Freshdesk - включає модуль тарифікації для автоматизації процесів виставлення рахунків

Таблиця 1.2 Порівняння CRM-систем з білінговими функціями

Критерій	Salesforce	Zendesk	HubSpot	Freshdesk
Білінгові функції	Розширені	Середні	Базові	Середні
Інтеграція з АТС	Добра	Добра	Обмежена	Добра
Автоматизація рахунків	Так	Так	Так	Так
Аналітика витрат	Розширена	Базова	Базова	Середня
Вартість	Висока	Середня	Низька	Середня

Перевагою інтеграції білінгу з CRM-системами є створення єдиного інформаційного простору, що дозволяє аналізувати взаємозв'язок між якістю обслуговування клієнтів та фінансовими результатами діяльності call-центру.

1.4 Проблеми та обмеження сучасних білінгових систем

Сучасні білінгові системи, незважаючи на стрімкий технологічний розвиток, стикаються з низкою системних проблем та обмежень, що істотно впливають на їх ефективність та функціональність [19]. Ці проблеми особливо

виражені в умовах швидкої диджиталізації телекомунікаційних послуг та зростання вимог клієнтів до гнучкості та якості обслуговування;

- Архітектурні та технологічні обмеження:

Однією з ключових проблем сучасних білінгових систем є їхня архітектурна спадковість. Багато систем було розроблено десятиліття тому і вони базуються на застарілих технологіях, що ускладнює їхню інтеграцію з сучасними хмарними платформами та API. Монолітна архітектура обмежує гнучкість системи та ускладнює її масштабування. Існують значні проблеми з обробкою великих обсягів даних у реальному часі, особливо в умовах зростаючої кількості телекомунікаційних послуг та клієнтської бази.

- Операційні та функціональні проблеми:

Багато сучасних білінгових систем мають обмежену функціональність щодо гнучкості тарифікації [11]. Вони не дозволяють швидко адаптуватися до змін ринкових умов, впроваджувати нові тарифні моделі або створювати персоналізовані пропозиції для клієнтів [4]. Складність конфігурування та обмежені можливості кастомізації ускладнюють процес налаштування системи під конкретні бізнес-процеси компанії.

- Проблеми інтеграції та сумісності:

Істотною проблемою є складність інтеграції білінгових систем з іншими корпоративними системами, такими як CRM, ERP та бухгалтерські платформи [6]. Відсутність стандартизованих інтерфейсів та протоколів обміну даними призводить до необхідності розробки складних та вартісних адаптерів. Крім того, існують проблеми з сумісністю різних версій програмного забезпечення та оновленням систем без перерв у роботі.

- Управлінські та організаційні складнощі:

Багато білінгових систем мають складний та неінтуїтивний інтерфейс, що ускладнює їх використання та вимагає спеціальної підготовки персоналу. Відсутність зручних інструментів аналітики та звітності обмежує можливості прийняття обґрунтованих управлінських рішень. Крім того, існують проблеми

з якістю даних та їхньою консистентністю, що впливає на точність тарифікації та звітів.

– Економічні обмеження:

Вартість володіння сучасними білінговими системами залишається високою, особливо для малих та середніх підприємств. Ліцензійні витрати, витрати на обслуговування та оновлення системи часто перевищують бюджетні можливості компаній [5]. Крім того, існують проблеми з окупністю інвестицій у білінгові системи, особливо в умовах швидких технологічних змін.

– Безпекові виклики:

Сучасні білінгові системи стикаються з численними загрозами безпеці, включаючи несанкціонований доступ до даних, кібератаки та витоки конфіденційної інформації [1]. Відповідність вимогам регуляторних органів та стандартам захисту даних (таким як GDPR) вимагає додаткових інвестицій у безпеку системи.

– Масштабованість та продуктивність:

З ростом обсягів даних та кількості користувачів багато білінгових систем демонструють зниження продуктивності. Обмеження щодо масштабування та обробки пікових навантажень можуть призводити до затримок у обробці даних та формуванні звітів [16]. Це особливо критично для великих телекомунікаційних компаній із мільйонами абонентів.

– Підтримка нових бізнес-моделей:

Сучасний ринок телекомунікацій вимагає підтримки нових бізнес-моделей, таких як послуги за підпискою, спільне використання ресурсів, динамічне ціноутворення. Багато традиційних білінгових систем не мають необхідної функціональності для ефективної підтримки цих моделей, що обмежує конкурентні позиції компаній на ринку [22].

Проблема обліку та тарифікації VAS (Value-Added Services):

Архітектурні виклики інтеграції. Складність інтеграції різномірних сервісів у єдину білінгову екосистему становить значну проблему. Кожен

додатковий сервіс вимагає розробки індивідуальних модулів обліку, що призводить до фрагментації системи [16]. Особливі труднощі виникають при об'єднанні legacy-сервісів з сучасними хмарними рішеннями, де необхідно забезпечити консистентність даних між різними технологічними платформами.

Динамічність тарифних моделей. Сучасні VAS характеризуються нелінійними моделями споживання, де традиційні підходи до ціноутворення виявляються неефективними. Виникає необхідність реалізації адаптивних тарифних структур, здатних враховувати:

- Комбіноване використання послуг (bundling)
- Еластичність попиту в реальному часі
- Контекстне ціноутворення на основі аналізу поведінки користувачів

Специфіка обліку складних сервісів. Окремі категорії VAS демонструють унікальні характеристики, що ускладнює їх стандартизацію:

- Інтерактивні сервіси (IVR, голосові асистенти) вимагають обліку складності сценаріїв та якості NLP-обробки
- Послуги передачі даних потребують диференціації за пріоритетністю та класом якості (QoS)
- Медійні сервіси залежать від параметрів якості контенту та формату доставки

Операційні складнощі моніторингу. Відсутність уніфікованих інструментів моніторингу створює проблеми в оцінці ефективності VAS. Критичними залишаються питання:

- Верифікації фактичного надання послуг
- Контролю відповідності SLA
- Ідентифікації технічних збоїв та їх впливу на якість сервісів

Адаптація до регуляторних вимог. Стрімкий розвиток VAS випереджає формування нормативної бази, що створює додаткові виклики:

- Відповідність вимогам захисту даних для різних юрисдикцій

- Забезпечення прозорості тарифікації складних сервісів
- Виконання вимог щодо локалізації даних для окремих категорій послуг

Оптимізація витрат на облік. Собівартість обліку окремих категорій VAS часто перевищує економічний ефект від їх надання [20]. Це потребує реалізації інтелектуальних систем пріоритезації, здатних автоматично визначати оптимальний рівень деталізації обліку для кожного типу послуг [9].

Синхронізація з клієнтськими системами. Складність забезпечення коректної взаємодії з системами клієнтів для автоматизації процесів:

- Онлайн-моніторингу витрат
- Контролю лімітів споживання
- Інтеграції з корпоративними платіжними системами

Складності інтеграції з системами бухгалтерського обліку та фінансового звітності

Концептуальна невідповідність структур даних. Фундаментальна проблема полягає в різних підходах до організації інформації [17]. Білінгові системи використовують реляційні моделі, орієнтовані на операційну ефективність, тоді як бухгалтерські платформи базуються на принципах подвійного запису та балансової звітності [22]. Ця різниця проявляється в:

- Неможливості прямого відображення транзакцій
- Необхідності складних трансформацій даних
- Втраті семантичного контексту при передачі інформації

Дисонанс у часових періодах та циклах обробки. Критичною проблемою є розбіжність у часових інтервалах між системами:

- Різні дати закриття звітних періодів
- Неспівпадіння моментів визнання доходів і витрат
- Відмінності в графіках проведення переоцінок та амортизації

Специфіка обліку телекомунікаційних операцій. Багато операцій у телекомунікаційній галузі не мають аналогів у стандартних бухгалтерських системах:

Облік міжоператорських розрахунків за складними тарифами

- Відображення доходів від наданих послуг з відстроченим визнанням
- Облік зобов'язань за передплатними тарифними планами

Проблеми нормативної відповідності. Інтеграція ускладнюється через суворі вимоги регуляторних органів:

- Необхідність дотримання міжнародних стандартів фінансової звітності
- Виконання вимог податкового законодавства різних країн
- Забезпечення відповідності вимогам фінансового моніторингу

Складність забезпечення консистентності даних. Основні проблеми узгодження інформації між системами:

- Розбіжності в залишках та оборотах за рахунками
- Різні алгоритми округлення та конвертації валют
- Необхідність багаторазових перевірок та коригувань

Технологічні бар'єри інтеграції. Сучасні системи часто використовують різні технологічні стеки:

- Несумісність форматів даних та протоколів обміну
- Відмінності в системах безпеки та автентифікації
- Проблеми з продуктивністю при великих обсягах даних

Вимоги до масштабованості та надійності. Великі обсяги операцій створюють додаткові складнощі:

- Забезпечення стабільності передачі даних у реальному часі
- Обробка мільйонів транзакцій щодоби
- Забезпечення відмовостійкості інтеграційних рішень

Обмеження існуючої системи «ФРІНЕТ» щодо гнучкості тарифних планів

Існуюча білінгова система «ФРІНЕТ», яка використовується компанією для обліку послуг, тарифікації та управління абонентами, має низку функціональних обмежень, що впливають на можливість гнучкого формування тарифних планів [8].

В умовах сучасного ринку телекомунікацій гнучкість тарифів є важливим конкурентним фактором, однак поточна система не повністю задовольняє потреби компанії щодо швидкої адаптації послуг під різні категорії користувачів [10].

Одним із основних обмежень є жорстка структура тарифів, яка не дозволяє оперативно змінювати параметри, такі як швидкість, додаткові послуги, знижки або періоди дії. Для внесення будь-яких змін потрібні ручні маніпуляції адміністраторів та іноді навіть доопрацювання програмної частини, що значно ускладнює процес оновлення тарифної сітки.

Ще однією проблемою є обмежена підтримка персоналізованих тарифів. У сучасних умовах багато компаній переходять на моделі, де абонент може самостійно формувати пакет послуг, обираючи швидкість, додаткові опції, типи доступу та персональні умови [4]. Система «ФРІНЕТ» не забезпечує такої можливості, оскільки працює переважно з фіксованими тарифними наборами, які мають жорстку логіку розрахунків.

Також система має низьку адаптивність до акційних або тимчасових тарифів. Для запуску акційного тарифу необхідно створювати окрему тарифну категорію, що призводить до дублювання даних та збільшення обсягу ручної роботи.

Відсутність автоматизованих інструментів для налаштування сезонних акцій, знижок для нових абонентів або пакетів «інтернет + додаткові послуги» зменшує гнучкість компанії у реагуванні на ринкову конкуренцію [5].

Крім того, система має обмеження щодо інтеграції з сторонніми сервісами та CRM-системами, що ускладнює реалізацію динамічних тарифів, прив'язаних до поведінки абонента. Наприклад, неможливо автоматично

формувати пропозиції на основі обсягу використання, терміну співпраці або залишку трафіку, що ускладнює впровадження сучасних моделей тарифікації.

Загалом, обмеження гнучкості тарифних планів у системі «ФРІНЕТ» знижують ефективність управління тарифною політикою та стримують можливості компанії щодо впровадження інноваційних і персоналізованих тарифних рішень.

Це створює потребу в модернізації системи або переході на нову платформу, яка забезпечуватиме більшу адаптивність, автоматизацію та інтеграцію з іншими сервісами [1].

Проведене дослідження дозволяє ідентифікувати ключові виклики, що стоять перед розробниками білінгових систем, та сформулювати вимоги до сучасних рішень, здатних ефективно функціонувати в умовах динамічно змінюючогося ринку телекомунікаційних послуг.

Отримані результати створюють теоретичний фундамент для подальшої розробки власної білінгової системи, орієнтованої на потреби сучасних українських підприємств.

2 АНАЛІЗ І ШЛЯХИ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО БІЛІНГУ CALL-ЦЕНТРУ

Автоматизований білінг у call-центрі є критично важливим елементом ефективного управління абонентськими послугами, тарифами, платежами та обліком взаємодії з клієнтами. Сучасні telecommunication-компанії, такі як «ФРІНЕТ», обслуговують значну кількість абонентів, що вимагає швидкого, точного та масштабованого обліку кожної операції.

Традиційні системи, які виконують білінгові функції частково або з використанням ручного втручання, нерідко призводять до помилок, затримок, неузгодженостей у тарифах та складності інтеграції з іншими сервісами [25].

У цьому розділі розглядаються ключові проблеми існуючих білінгових рішень, фактори, що ускладнюють роботу call-центру, а також можливі шляхи модернізації системи з урахуванням сучасних вимог до автоматизації, безпеки, стабільності й масштабованості.

Особлива увага приділяється оптимізації процесів тарифікації, обліку звернень клієнтів, автоматизації підключень та відключень послуг, а також підвищенню прозорості взаємодії між операторами call-центру та абонентами [24].

Основна мета аналізу - визначити, як саме можна модернізувати білінгову систему таким чином, щоб вона:

- мінімізувала людський фактор,
- скорочувала час обробки запитів,
- підвищувала точність тарифікації,
- забезпечувала зручність користування,
- легко інтегрувалася з іншими модулями та сервісами компанії.

Шляхи розв'язання проблем у цьому розділі охоплюють удосконалення архітектури білінгу, застосування сучасних API, автоматизацію з використанням скриптів і мікросервісів, а також впровадження гнучкішої

системи тарифних планів, яка здатна пристосовуватися до змін ринку й попиту клієнтів.

2.1 Постановка задачі

Автоматизація білінгових процесів у call-центрі є центральним елементом підвищення ефективності обслуговування абонентів та зменшення операційних витрат компанії. Існуюча система «фРІНЕТ» частково виконує функції обліку, однак має низку обмежень, що ускладнюють роботу операторів, технічного відділу та абонентів [1].

Серед ключових проблем - недостатня гнучкість тарифних планів, відсутність інтеграції між білінгом та CRM, складність внесення змін у параметри послуг, потреба у ручних операціях під час підключення чи відключення абонентів, а також низька прозорість тарифікації [8].

Завданням розробки нової білінгової підсистеми є створення цілісного програмного рішення, яке забезпечить автоматизоване управління тарифами, коректну тарифікацію послуг, облік взаємодій з клієнтами, можливість масштабування та інтеграцію з іншими внутрішніми системами компанії.

Нова білінгова підсистема повинна мінімізувати людський фактор, прискорити обробку заявок, забезпечити точність і прозорість нарахувань, а також створити гнучку платформу для подальшого розвитку сервісів.

Формалізація вимог стейкхолдерів.

Розробка потребує врахування інтересів та вимог декількох груп стейкхолдерів. Їхні очікування визначають функціональність, архітектуру та експлуатаційні характеристики майбутнього продукту [19].

1) Керівництво компанії

- Підвищення ефективності роботи call-центру та техпідтримки.
- Скорочення витрат за рахунок автоматизації рутинних процесів.
- Підвищення прозорості фінансових операцій.

- Можливість оперативно впроваджувати нові тарифні плани без необхідності доопрацювання ядра системи.
- Зменшення кількості помилок у білінгу, що впливають на лояльність клієнтів.

2) Оператори call-центру

- Швидкий доступ до даних абонента: стан рахунку, тариф, швидкість, історія звернень.
- Ясна й автоматизована система зміни тарифів, активації та деактивації послуг.
- Мінімізація ручних операцій, що зменшує ризик помилок.
- Зручний інтерфейс для пошуку й внесення даних.

3) Абоненти

- Прозоре та зрозуміле нарахування платежів.
- Гнучкі тарифні плани, можливість переходу між ними без тривалих процедур.
- Оперативне підключення або відключення додаткових послуг.
- Доступ до особистого кабінету з повною інформацією про стан рахунку.

4) Технічний відділ / адміністратори системи

- Легка інтеграція з мережевим обладнанням та CRM-платформами.
- Підтримка API для автоматизації технічних операцій (наприклад, через SSH, Ansible, Expect).
- Масштабованість та модульність системи.
- Можливість швидкого внесення змін у логіку тарифікації через конфігураційні файли або адміністративну панель.

5) Фінансово-бухгалтерський відділ

- Коректний та стабільний облік платежів.
- Генерація фінансових звітів за періоди.

- Експорт даних у системи обліку (1С, BAS, власні модулі).

Формалізація вимог дає змогу проектувати систему так, щоб вона одночасно задовольняла інтереси всіх груп і забезпечувала комплексне рішення для компанії [7].

Визначення критеріїв успіху для нової білінгової підсистеми.

Для оцінки ефективності створення нової підсистеми білінгу необхідно визначити чіткі критерії успіху, які дозволять виміряти, наскільки розроблена система відповідає очікуванням стейкхолдерів та технічним вимогам [9].

1) Функціональні критерії

- Коректна тарифікація усіх видів послуг.
- Підтримка гнучкої системи тарифів (динамічні параметри: швидкість, обсяг, ціна, початкові періоди).
- Повна автоматизація підключення/відключення послуг.
- Інтеграція з CRM, техпідтримкою та мережею (через API або скрипти).

2) Технічні критерії

- Висока продуктивність при роботі з багатьма абонентами.
- Масштабованість під час розширення інфраструктури.
- Стійкість до збоїв і наявність аварійного відновлення.
- Підтримка журналювання всіх операцій для аудиту.

3) Критерії зручності та взаємодії

- Інтуїтивний інтерфейс для операторів та адміністраторів.
- Наявність особистого кабінету абонента з повною інформацією.
- Мінімальна кількість помилок оператора при роботі з системою.

4) Фінансово-економічні критерії

- Зменшення операційних витрат за рахунок автоматизації.
- Зниження кількості помилкових нарахунків.
- Скорочення часу обробки одного клієнтського запиту.

5) Організаційні критерії

- Можливість швидкого внесення правок у тарифну політику без зміни коду.
- Наявність документації та інструментів для адміністрування.
- Забезпечення надійного доступу з контролем прав користувачів.

2.2 Аналіз алгоритмів тарифікації та обліку дзвінків

Облік дзвінків і тарифікація - центральні функції білінгової підсистеми. Процес включає прийом записів про дзвінки (CDR), їх уніфікацію, перевірку, класифікацію за типом послуги, застосування тарифних правил, нарахування вартості, облік акцій та знижок, агрегацію для рахунків і ведення журналів для аудиту [11].

Алгоритми повинні забезпечувати надійність, відтворюваність рішень, ідентифікацію дублікатів і можливість змінювати тарифну політику без втручання в код (через конфігурації або правила).

Ключові етапи обробки дзвінка:

- прийом і нормалізація запису про дзвінок (уніфікація полів);
- валідація і обробка дублікатів;
- класифікація послуги за напрямком і типом;
- розрахунок вартості згідно тарифних правил і часових коефіцієнтів;
- застосування знижок, бонусів та акцій;
- агрегація результатів по абоненту;

Аналіз алгоритмів розпізнавання типу послуги.

Розпізнавання типу послуги визначає, за якою тарифною логікою оплачується дзвінок. Помилкова класифікація призводить до некоректних нарахувань і скарг, тому алгоритм має бути точним і прозорим [20].

Основні підходи та елементи логіки класифікації:

- Префіксна логіка – порівняння номера абонента призначення з таблицею префіксів (локальні, міжміські, міжнародні, спеціальні номери). Використовується найбільш довгий співпадаючий префікс для остаточної класифікації. Перевага - швидкість і простота; недолік - потреба постійного оновлення префіксної бази.

- Перевірка перенесення номера (LRN) – у ринках з номероперенесенням додатковий запит до бази перенесення дозволяє визначити фактичного оператора призначення і правильно віднести виклик до on-net або off-net. Це підвищує точність тарифікації.

- Аналіз сигналізації та джерела виклику – поля протоколів (SIP/ISDN), ідентифікатори транків або POP, інформація про маршрут можуть вказувати на спеціальні послуги (наприклад, платні короткі номери, служби підтримки, переадресацію).

- Часові та контекстні правила – тариф може залежати від часу доби, дня тижня або свят, а також від тарифного плану абонента (наприклад, «нічний тариф» чи «тариф для бізнес-клієнтів»).

- Піраміда правил і пріоритетів – механізм, який визначає порядок застосування джерел рішення: префікси → LRN → trunk/маршрут → контекстні правила → запасний (default) клас. Важливо зберігати детальне пояснення (лог) того, чому дзвінок отримав саме такий клас - це необхідно для спорів і аудиту.

Вимоги до реалізації:

- можливість оновлення префіксних таблиць і правил без зупинки системи;
- збереження журналу рішень зі «слідом аудиту» (яке правило застосовано, які дані використано);
- підтримка механізмів корекції у випадку зміни інформації (наприклад, оновлення LRN після первинного нарахування).

Алгоритми обробки помилкових та невдалих викликів.

Невдалі дзвінки (наприклад: no answer, busy, rejected, cancelled, помилки сигналізації) вимагають чіткої політики нарахувань. Різні підприємства застосовують різні підходи: повна відсутність нарахувань, нарахування фіксованої плати за встановлення з'єднання, або часткова тарифікація за час до моменту відключення [13].

Логіка обробки таких випадків повинна включати:

- Категоризацію результату виклику - однозначне віднесення запису до категорій: Billinglable (успішний), non-Billinglable (невдалий без нарахувань), частково Billinglable (наприклад, нарахування за установку). Це робиться на основі кодів причин у SIP/ISDN або полів CDR.

- Агрегація пов'язаних записів - у випадках, коли АТС/мережа генерує кілька записів для одного сеансу (setup, answer, disconnect), необхідно корелювати їх за ідентифікаторами сесії та формувати один узагальнений білінговий запис.

- Політика щодо дублювання і пізніх записів - система повинна ідентифікувати дублікати та забезпечувати ідемпотентність (однакова обробка при повторному надходженні запису). Для пізніх CDR має бути процедура корекції: або відкладене включення у наступний період, або створення коригувального запису (adjustment).

- Конфігуровані тарифи для failed calls - адміністратор має мати можливість задавати, чи нараховувати setup-fee, чи враховувати «платну мелодію» під час дзвінка, чи робити виключення для певних напрямків.

- Логування причин і рішень - запис інформації про причину невдачі й прийняте рішення щодо нарахувань, щоб можна було відтворити обчислення у разі спірних ситуацій.

- Процедури для спорів і повернень - механізм автоматичної або ручної перевірки й виправлення помилкових нарахувань з відкатом використаних бонусів/знижок, якщо необхідно.

Практичні рекомендації:

- визначити типові політики (повний non-Billinglable, charge_setup, charge_for_ringing) і впровадити їх як параметри, що змінюються через панель адміністрування;
- забезпечити щоденну звірку статистики failed vs successful для виявлення аномалій;
- зберігати докази (cdr + лог рішення) для юридичного захисту та розгляду скарг абонентів.

Алгоритми застосування знижок, бонусів та акційних пакетів

Застосування промоцій і знижок значно ускладнює білінг: виникають питання пріоритетів, сумування знижок, лімітів, часових рамок і тригерів активації. Алгоритм має бути прозорим, детермінованим і легко конфігурованим.

Ключова логіка застосування промоцій:

- Опис промо як набору умов і дій - кожна акція містить умови (сегментація абонентів, час дії, напрямок дзвінків, мінімальні поповнення) та ефект (фіксована знижка, відсоток, безкоштовні хвилини, кешбек, бонусні бали).
- Тригери активації - промо може активуватися автоматично (наприклад, при першому поповненні), по промокоду, або вручну оператором; також можливі умовні активації (виконання певної умови - «якщо абонент витратив більше X грн»).
- Порядок і політика комбінування - для більшості систем потрібно чітко визначити правила: чи сумуються знижки (stackable), чи є взаємовиключні пропозиції (exclusive), і у якому порядку вони застосовуються (наприклад, спочатку безкоштовні хвилини, потім процентна знижка). Політика має бути детермінованою, щоб у разі спору результат можна було відтворити.

- Ліміти та облік використання - для промо важливі ліміти (наприклад, «100 безкоштовних хвилин на місяць»), тому потрібно вести стан залишку для кожного абонента (remaining_free_minutes, remaining_credit тощо).

- Часові рамки та TTL - промо може мати початок/кінець, а також часові умови (тільки вечірні дзвінки, тільки вихідні). Необхідно враховувати часові зони і перехід на літній/зимовий час.

- Пріоритети й детермінізм - при наявності декількох застосовних промо система має застосувати їх у визначеному порядку або вибрати «best-offer» згідно політики.

- Rollback і корекції - у разі повернення платежу або виправлення помилок потрібно коректно відновити стан промо (наприклад, повернути витрачені безкоштовні хвилини або поповнити кредит).

- Аудит і звітність - зберігати повні записи про застосовані промо (яка акція, скільки хвилин/сум застосовано, у якому періоді), щоб можна було генерувати звіти й відновлювати розрахунки.

Приклад типової логіки (словесно):

1. На момент обробки дзвінка система визначає всі активні та застосовні промо для даного абонента та виклику.

2. Фільтрує промо за умовами (напрямок, тариф, час).

3. Визначає порядок застосування (наприклад: безкоштовні хвилини → фіксовані кредити → процентні знижки).

4. Застосовує по черзі, слідкуючи за лімітами і залишками, фіксує витрати промо у відповідних записах.

5. Після повного застосування обчислює фінальну вартість дзвінка і зберігає лог із деталізацією.

Рекомендації щодо впровадження:

- зберігати правила промо у вигляді конфігурованих даних (адмінпанель), щоб маркетинг міг налаштовувати акції без релізів;

- вести автоматичні тести для комбінацій промо (unit/integration) і періодичні звірки використання промо з фінансовими записами;
- документувати правила комбінування і пріоритети, робити їх публічними для внутрішніх процесів, щоб уникнути невизначеності при спорах.

2.3 Розробка алгоритму роботи білінгової підсистеми

Розробка алгоритму роботи білінгової підсистеми передбачає створення логічної послідовності дій, яка забезпечує коректний облік дзвінків, їх тарифікацію, фіксацію результатів і подальше відображення інформації у системі обслуговування абонентів.

Основне завдання алгоритму – автоматично й безперервно опрацьовувати дані про виклики, мінімізуючи залежність від оператора call-центру та виключаючи можливість помилок, пов'язаних з людським фактором [1].

Базовий алгоритм починається з моменту отримання системою події про дзвінок: його ініціація, тип (вхідний, вихідний, внутрішній), джерело та пункт призначення. Після цього підсистема виконує визначення категорії дзвінка, встановлює відповідні тарифні параметри та перевіряє чинність тарифного плану абонента.

Далі алгоритм визначає тривалість виклику, враховує можливі технічні особливості (наприклад, короткі або обірвані дзвінки) і застосовує правила тарифікації згідно з політикою компанії.

У процесі обробки також передбачається перевірка наявності додаткових умов – бонусів, акційних хвилин, індивідуальних пакетів та пільг. Якщо такі опції активні, система коректно перераховує кінцеву вартість або списання одиниць пакету.

Після обчислення фінального результату формуються білінгові записи, які зберігаються у базі даних та стають доступними для аналітики, звітності та перегляду співробітниками компанії [2].

На завершальному етапі алгоритм передає інформацію в інші модулі: CRM, внутрішній портал, систему інкасації платежів або в інструменти контролю якості обслуговування. Такий підхід забезпечує прозорість, узгодженість даних та стабільну роботу всієї інфраструктури call-центру.

Такий підхід забезпечує прозорість, узгодженість даних та стабільну роботу всієї інфраструктури call-центру.

2.4 Проектування архітектури білінгового програмного продукту

Проектування архітектури білінгового продукту має забезпечити надійність, масштабованість, високу продуктивність та простоту інтеграції з мережею та суміжними системами (CRM, система платежів, мерчандайзинг, аналітика) [14].

Архітектура повинна розділяти відповідальність між компонентами (інгест CDR, нормалізація, рейтинг, промо-движок, агрегатор, API, сховище) і передбачати можливість горизонтального масштабування, обробки як поточкових, так і пакетних даних, управління конфігурацією тарифів та повний аудит рішень.

Вибір моделі даних та проектування бази даних

Основні вимоги до моделі даних:

- Цілісність і відтворюваність: зберігати вхідні CDR у незмінному вигляді (raw), а також трансформовані/нормалізовані записи і фінальні білінгові івенти.
- Аудит і історія змін: фіксувати версії тарифів, застосовані правила та лог рішень; вести історію корекцій (adjustments).

- Швидкий доступ і агрегації: підтримка запитів для звітності (по абоненту, по періоду, по напрямку) з мінімальною затримкою.
- Масштабованість і архівування: політика зберігання CDR - online (оперативний шар) та cold-storage/архів (довгострокове зберігання).

Рекомендована схема зберігання (логічний шар):

1. Raw CDR store - сховище необроблених записів (файлове сховище або колонкова БД/об'єктне сховище). Призначення: джерело істини, можливість повторної обробки.
2. Normalized CDR table - реляційна таблиця або колонковий шар з уніфікованими полями (call_id, caller_id, callee, start_time, end_time, duration, cause_code, source_id, normalizer_version). Індеси: call_id, subscriber_id, start_time.
3. Billing events / Rated records - фінальні івенти після рейтингу (subscriber_id, event_id, service_class, base_cost, discounts_applied, final_cost, applied_promos, rating_version).
4. Promos / Credits state - таблиці/колекції, що зберігають стан промо для абонента (remaining_minutes, remaining_credit, expiry).
5. Audit logs - детальні логи прийняття рішень (rule_id, reason, input_snapshot, timestamp, operator_id_if_manual).
6. Archive - холодне сховище (S3/об'єктне сховище або окрема колонкова БД) з політиками TTL/ретеншн.

Вибір технологій (архітектурні помічники - приклади):

- Реляційні БД (Postgres, MySQL) - для фінальних білінг-записів, транзакцій, звітів; сильні сторони: консистентність, SQL-звітність.
- Колонкові/аналітичні сховища (ClickHouse, BigQuery) - для великих обсягів CDR та швидких аналітичних агрегацій.
- Об'єктні сховища (S3, MinIO) - для raw CDR та архівів.
- NoSQL (key-value, document) - для зберігання стану промо, кешу залишків, швидкого читання.

Індексація та партиціювання:

- Партиціювання по даті (month/day) для CDR та Billingled events.
- Індeksi на полях subscriber_id, start_time, call_id, service_class.
- Використання TTL/архівних політик для старих даних.

Безпека та відповідність:

- Шифрування даних у спокої й під час передачі.
- Розмежування доступів (RBAC), аудит доступів до конфіденційних даних.

- Політики зберігання відповідно до регуляторних вимог (законодавство про персональні дані).

Проектування API для інтеграції з іншими системами

Принципи проектування API:

- Ясність і простота: чіткі контрактні інтерфейси з добре документованими схемами (OpenAPI/Swagger).
- Сумісність та версіонування: версії API для запобігання наривам при змінах (v1, v2).
- Ідемпотентність при прийомі CDR (повторні запити не мають дублювати нарахування).
- Безпека: TLS, OAuth2 / JWT для аутентифікації/авторизації, rate limiting.
- Стандарти обробки помилок: зрозумілі коди помилок і повідомлення.

Ключові API-контракти:

1. Ingest API (CDR ingestion)

Пакетна і потокова підтримка - POST для пачок CDR та streaming (gRPC, Kafka connector).

- Підтвердження прийому з унікальним receipt_id; підтримка retry/idempotent.

2. Normalization / Validation callbacks

- Ендпоінти для повідомлення про проблемні/необроблені CDR або webhook для асинхронної обробки.

3. Rating API

- Запит на рейтинг одного або групи CDR (при real-time charging / on-demand rating).
- Параметри: subscriber_id, call_id, timestamp, requested_version; відповідає rated_event.

4. Promotion / Promo Engine API

- Запити на перевірку/активацію промо, отримання стану залишків, блокування/списання ресурсів.

5. Billing events API / Export

- Підписка на фінальні івенти (webhook/Kafka) або запити на пакетний експорт (CSV, Parquet).

6. Admin API

- Управління префіксними таблицями, тарифами, політиками failed call, ролями користувачів.

7. Audit & Query API

- Пошук за event_id, call_id, subscriber_id; отримання повного логу рішень для спорів.

Особливості інтеграції:

- Асинхронність: більшість інтеграцій повинні підтримувати асинхронну обробку (через черги, webhook, stream).
- Контракти доступу: SLA на latency, формат повідомлень, гарантії доставки.
- Тести контрактів: automated contract tests (consumer-driven) між білінгом і системами-споживачами.

Архітектура системи збору даних CDR (Call Detail Record)

Завдання підсистеми збору CDR:

Забезпечити надійний і масштабований прийом великого потоку CDR зі шлюзів, АТС, SIP-серверів; гарантувати доставку, відсутність втрат і можливість повторної обробки [6].

Компоненти та потік даних:

- Collector / Forwarder
 - Легкі агенти на межі мережі або логічні шлюзи, що приймають CDR у рідному форматі від АТС/шлюзів (SMPP, SIP accounting, RADIUS, syslog).
 - Мінімізують затримки, забезпечують підпис/метадані про джерело.
- Ingest Layer (Message Queue / Stream Broker)
 - Використання черги повідомлень (Kafka, Pulsar, RabbitMQ) для надійної доставки та буферизації.
 - Переваги: масштабованість, можливість повторного відтворення топіку для ре-рейту.
- Normalizer / Enricher
 - Сервіси, які читають з черги, нормалізують формат, додають метадані (trunk id → region), роблять LRN lookup, збагачують дані для класифікації.
- Validator / Deduplicator
 - Перевірка обов'язкових полів, виявлення дублікатів (call_id+source_id+timestamp), маркування/пересилання аномалій у окрему чергу для ручної перевірки.
- Rating/Processing pipeline
 - Після нормалізації записи потрапляють до рейтингового блоку (онлайн або батч), де застосовується тарифікація та промо.
- Storage
 - Raw CDR зберігаються в об'єктному сховищі; normalized і Billigled events - у БД.

- Monitoring & Replay
 - Метрики по лагу ingestion, кількості помилок, ступені backlog; можливість replay з топіку для повторної обробки.

3 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ “Billing”

3.1 Функціональні та нефункціональні вимоги до програмного продукту

Функціональні вимоги веб-системи на основі сторінок програми можуть включати наступне:

- Авторизація з ролями та правами доступу систему. Зареєстровані користувачі отримують доступ до системи за допомогою логіну та паролю. Система перевіряє введені дані та надає доступ до білінгової системи. Користувач отримує доступ лише до своєї програмної частини в залежності від відділу роботи та посади яку займає. Захист персональних даних абонентів. Користувач без належного статусу доступу не отримує допуск до картки клієнта згідно з договором [21].

- Головна сторінка. На головній сторінці представлений доступ для пошуку згідно з такими умовами – Робота з особистими даними абонента. Система повинна забезпечувати перегляд і редагування основних персональних даних користувача, таких як прізвище, ім'я, по батькові, номер телефону та електронна пошта. У разі юридичної особи відображається код ЄДРПОУ та тип укладеного договору. Усі зміни мають зберігатися в базі даних із фіксацією історії оновлень.

- Перегляд балансу. На сторінці абонента система відображає його поточний фінансовий баланс, а також може надавати додаткову інформацію щодо заборгованості або переплат.

- Відображення адресної інформації. Для кожного абонента система забезпечує збереження адреси проживання чи підключення: населений пункт, вулиця, номер будинку та квартири. Дані використовуються для технічних перевірок та формування заявок [24].

- Управління технічними параметрами. Система повинна відображати технічні дані, що стосуються підключення абонента, зокрема внутрішній ID,

SID, логін (наприклад, PPPoE), виділену IP-адресу, MAC-адресу обладнання, тип активного сервісу (Інтернет, IPTV, VoIP тощо) та поточний тарифний план. Також має бути доступна можливість змінювати тариф чи вид сервісу, якщо це дозволяє політика оператора.

- Фільтрація списку абонентів. Система надає розширені можливості фільтрації для швидкого пошуку потрібних клієнтів. Доступні фільтри можуть включати: показ закритих договорів, відображення лише фізичних осіб або лише юридичних осіб, вибірка VIP-абонентів, фільтрація приватного сектору або комерційних користувачів. Також можливий пошук абонентів без заповнених паспортних даних або перегляд старих сервісів [19].

- Розмежування типів абонентів. Система підтримує різні категорії користувачів (фізичні особи, юридичні особи, VIP-клієнти), що дозволяє застосовувати різні бізнес-правила, тарифні моделі та обробку даних.

- Картка абонента (договору) –Має забезпечувати такі показники як . Система повинна надавати можливість переглядати повну інформацію про абонента, Починаючи з персональних даних що надавались при реєстрації договору особою яка хотіла стати абонентом. Персональні дані такі як ПІБ, адреса проживання під якою буде зареєстрований договір даними договору контактні номери телефонів, електронна пошта, можливі контактні особи які мають право отримати інформацію за телефоном який вніс власник договору балас договору, типом клієнта фізичне чи юридична особа, статусом послуги та самого договору.

- Дані договору в перелік яких входить тип послуги підключення в залежності від багатоквартирного будинку та приватного сектор) дата підключення та початок виконання умов з надання послуги номер договору для зручності знаходження в системі та коментарі по підключенню [18].

- Перегляд технічних даних підключення. Користувач (оператор або технічний спеціаліст) може переглядати поточний стан обладнання абонента чи доступне обладнання якщо не доступне з якої причини не доступне, Система повинна забезпечувати зручний доступ до діагностики лінії та

моніторингу параметрів мережі Перевірка відповідності налаштувань до заданих договором вимог по наданню послуг. Відповідності надходження швидкості до первинного обладнання абонента та фізичного зв'язку по лінії підключення абонента.

- Робота із заявками. На сторінці абонента система відображає всі активні та виконані заявки. Виконані заявки та активні відрізняються кольоровою гамою для гручності надходження Користувач може створювати нові заявки наприклад, ремонт або купівля обладнання, призначати виконавців, змінювати статуси та переглядати Тако можливе переглядання попередніх заявок та перевірка як сама була вирішена проблема з коментарів назначених відповідній інженерів по ній [25].

- Обробка інцидентів та звернень. Система дозволяє реєструвати звернення клієнтів, фіксувати дзвінки в call-центр чи звернення з чату чи пошти, переглядати історію комунікацій персоналу та абонента. відображати інцидентів які були зафіксовані за існування договору та по самому абоненту. Уся інформація доступна через відповідні вкладки у картці абонента. Кожен інцедент та звернення зафіксовані окремо та відповідно вітображається інформація хто створив інцедент та хто з ним працював. Поки інцедент активен оператори різних ввіділів можуть залишати додаткову інформацію для відповідних віділів чи фіксувати звернення абоненту згідно з цього інцеденту доповнюючи інформацію для можливості повного та детального аналізу ситуації яка наведена в певному інциденті. Прошу зауважити що фінансові звернення обролююлише віділ фінансів та стакр. А технічний відділ не має потрібного доступу [21].

- Управління фінансовими операціями. В блоку фінансів автоматично відображені всі нарахування та платежі абонента також списання відповідно до послуг які були надані. Фінансові послуги такі як щомісячна абоненська плата та виклик інженерів рахуються окремо. Користувач може переглядати історію змін балансу відповідно до послуг.

– Додавання коментарів. Працівники компанії можуть створювати службові нотатки та коментарі щодо абонента. Система повинна зберігати повну історію коментарів. Коментарі неможливо видалити з договору оператором всіх відділів. Коментарі залишені для додаткової інформації по абоненту наприклад під'їзд чи поверху чи нестандартного підключення [3].

– Використання панелі інструментів. На сторінці абонента доступні швидкі дії: підключення та зміна тарифного плану, Привязка портів та обладнання абонета згідно з його mac-адреси яку оператор бачить при перевірці лінії, відправлення повідомлень SMS до прив'язаного месенджера та перехід до інших модулів системи заявки, інциденти, фінанси.

– Співробітники Сторінка дозволяє переглянути всіх співробітників та їх статус відповідно до таких критеріїв. Усі співробітники виводить співробітників які були внесені в базу даних за весь час та сортиру по дирекції. Звільнені це заблоковані користувачі яких вилучили з бази даних та заблокували доступ до системи їх ні акаунти не допускаються в систему. Розклад виводить всіх співробітників та їх робочі години згідно з обраним днем перевірки. При загальному пошуку співробітника за його нікнеймом ми бачило нікнейм ПІБ, електрону пошту телефон та дирекцію до якої належить співробітник [1].

– Ліди – Створення нового ліда. Користувач має змогу створити новий лід. Створення відбувається за допомогою заповнення персональні дані: прізвище, ім'я та по батькові, контактний номер телефону та за потреби - додаткові номери та e-mail. Система перевіряє правильність формату введених даних. Введення адресної інформації. Під час створення ліда користувач може вибрати населений пункт і вулицю із довідника або ввести адресу вручну. Система повинна підтримувати пошук по довіднику населених пунктів та вулиць. Зазначення деталей заявки на підключення. Користувач може ввести тип активації, попередньо обрати тарифний план, тип ліда, джерело реклами та канал надходження звернення. Усі ці параметри мають зберігатися в базі даних для подальшого аналізу. Призначення дати сповіщення. Під час

реєстрації ліда оператор може вибрати дату та час нагадування. Система повинна автоматично створити внутрішню нотифікацію для відповідального працівника. Введення додаткової інформації. Інтерфейс дає змогу вказати тип обладнання, коментар щодо технічних або організаційних особливостей, а також примітки для відділу продажів або монтажно́ї бригади. Збереження ліда. Після заповнення всіх обов'язкових полів система зберігає інформацію у модулі “Продажі” та відображає новий лід у загальному списку лідів із відповідним статусом. Перевірка доступності покриття. У випадку, якщо вибраний населений пункт або вулиця відсутні в зоні покриття, система повинна дозволяти ручне введення адреси і позначати такого ліда як потенційного або потребуючого перевірки покриття.

- Сторінка Лідів Центральна частина сторінки створена для зручної роботи зі всіма зверненнями потенційних майбутніх клієнтів згідно з їх особистим зверненням за доступними каналами зв'язку з компанією. Дає можливість швидко знаходити, переглядати та обробляти ліди, контролювати їхній стан та історію взаємодій. Сторінка має панель фільтрів над таблицею вана створена для швидкого сортування лідів за їхнім статусом. Дозволяє обирати відображення однієї з категорій таких як:

- всіх звернень, нових лідів, оброблених, в роботі, неуспішних чи відхилених. Можливість сортирування значно пришвидшує роботу операторів та менеджерів. Наступна можливість сторінки це пошук по лідам. Створений для полегшення знаходження конкретного звернення за іменем, номером телефону, адресою або ID. Поле пошуку мінімізує час обробки лідів та дозволяє уникнути дублювання даних. Основна частина сторінки це таблиця лідів, де відображаються всі ліди у зручному табличному вигляді. Таблиця надає працівнику повну інформацію для оперативного прийняття рішень та подальшої обробки звернення. ID Унікальний номер створеного ліда, за яким його легко знайти або відслідкувати історію. Дата Показує час створення звернення, що допомагає визначити його актуальність та пріоритетність. Адреса Містить дані про населений пункт, вулицю та будинок.

Адреса необхідна для подальшої технічної перевірки можливості підключення та передачі заявки інженерам. ПІБ Вказує прізвище та ім'я потенційного абонента. Дозволяє точно ідентифікувати людину та уникнути дублювання лідів. Контакти Відображає номер телефону або інші способи зв'язку, необхідні для дзвінків операторів, уточнення даних та підтвердження заявки. Агент Показує, хто працював із лідом - оператор, менеджер або система. Дозволяє контролювати відповідальних за обробку звернень. Історія Містить усі дії, виконані з лідом: дзвінки, уточнення даних, зміни статусів, коментарі та час виконання. Це допомагає відслідковувати повний шлях обробки та уникати помилок або повторної роботи. Статус Відображає поточний стан ліда: новий, у роботі, успішно оброблений, відхилений або помилковий.

- Ці функціональні вимоги були реалізовані на сторінках роботи згідно з першим концептом проєкту й можуть бути розширені чи змінені відповідно до потреб користувачів і потреб проєкту в майбутньому використанні.

- Нефункціональні вимоги були спрямовані на покращення взаємодії з користувачем, забезпечення безпеки даних, продуктивності та стабільності системи "Billing".

Серед нефункціональних вимог, виділимо, зокрема, такі, як:

- *Простота використання інтерфейсу.* Система має забезпечувати інтуїтивно зрозумілий та легкий у використанні інтерфейс, який дозволяє користувачам швидко орієнтуватися й виконувати необхідні дії без зайвих зусиль. Мати чітку структуру та логічність для людського розуміння.

- *Доступність.* Система має бути придатною для використання декількома користувачами одночасно

- *Швидкість і продуктивність.* Система повинна працювати швидко і ефективно, забезпечуючи швидкий перегляд і обробку даних. Наприклад, щоб мінімізувати затримки під час завантаження сторінок.

- *Безпека даних.* Система повинна гарантувати захист персональних даних користувачів, обладнання, мережевих налаштувань та самого коду

системи. Це можливо при автоматичному шифруванні даних, контроль доступу до інформації та інші заходи для запобігання несанкціонованому доступу або втраті даних.

- *Надійність.* Система повинна бути стабільною і надійною, вільною від помилок і збоїв. На випадок проблем слід передбачити резервне копіювання даних і механізми відновлення системи.

- *Сумісність з обладнанням.* Веб-система повинна коректно взаємодіяти з мережевим обладнанням провайдера: OLT, ONU/ONT, маршрутизаторами, BRAS, DHCP-серверами та Radius-інфраструктурою. Підтримуються стандартизовані протоколи (PPPoE, DHCP, SNMP, TR-069).

- *Інтегрованість з іншими сервісами.* Програмний продукт повинен підтримувати інтеграцію з зовнішніми модулями - білінгом, CRM, системами моніторингу, SMS-шлюзами, VoIP-платформами та фінансовими сервісами.

- *Відмовостійкість мережесих інтеграцій.* У випадку недоступності зовнішніх систем (Radius, DHCP, SNMP) веб-система повинна коректно повідомляти користувача про помилку, а при відновленні роботи автоматично синхронізувати дані.

3.2 Структурна модель системи «Billing»

Розроблена система побудована на принципі роботи в архітектурі «Користувач-сервер». Система «Billing» зафіксована на використанні виділених серверів, які зберігають усі дані проєкту, але потребує встановлення додаткових компонентів на комп'ютері клієнта для доступу до роботи з нею. Це пов'язано з мірами безпеки для підтримки конфіденційних даних користувачів та самої системи загалом [18].

Клієнтська частина системи «Billing» складається з наступних компонентів:

- інтуїтивно зрозумілий інтерфейс

- підсистема перегляду та зміни даних
- підсистема розрахунку з бази даних
- підсистема введення фактичних даних
- підсистема виведення даних

Серверна частина веб-орієнтованої системи «Billing» складається з:

- підсистеми інформаційної складової даних абонентів та договорів;
- підсистеми доступних адрес покриття
- підсистеми інформації про обладнання та його технічні властивості
- підсистема розрахунку та інформування стану обладнання

Всі сторінки системи «Billing» поділяються на три компоненти:

- header (шапка);
- aside (бічна панель)
- main (головна частина);
- footer (підвал).

Для коректної роботи системи та збереження всіх даних було створено базу даних системи «Billing».

Сутності – це окремі об'єкти, які збирають дані. Кожна сутність має унікальний ідентифікатор, який може однозначно ідентифікувати запис, пов'язаний з цією сутністю в базі даних [17].

У предметній області системи «Billing» визначені наступні сутності, що представляють різні об'єкти та аспекти системи. Зараз я наведу для розбору основні:

- «Знижки» - зберігається інформація про всі знижки що були призначені на договорі (рис. 3.1).
- «Проплати» - зберігається інформація про всі проплати що були за відповідним договором (рис. 3.2).
- «Звернення» - зберігається інформація про звернення та питання (рис. 3.3).
- «Сервіси» - зберігаються посилання на фотографії лікарів (рис. 3.4).

Атрибут - це іменовані характеристики сутностей, імена яких мають бути специфічними для певного типу сутності, але можуть бути однаковими для всіх сутностей.

Визначимо наступні властивості для кожної сутності на основі наведених таблиць

Таблиця «Знижки»:

- Сервіси – тип підключеної послуги на яку видається знижка;
- Тип оплати - назва тарифу по якому видається послуга
- Дата початку - дата початку дії знижки
- Дата закінчення - дата кінця дії знижки
- Знижка - значення знижки у відсотка
- Хто - хто саме видав знижку

Знижки					
Сервіс	Тип оплати	Дата початку	Дата закінчення	Знижка (%)	Хто
Інтернет	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	2025-01-01 00:00:00	2025-04-03 00:00:00	15%	starlord
Інтернет	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	2019-09-01 00:00:00	2019-11-01 00:00:00	25%	_LOYAL2018_
Інтернет	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	2018-11-01 00:00:00	2019-01-01 00:00:00	25%	_LOYAL_
Інтернет	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	2018-08-01 00:00:00	2018-09-01 00:00:00	25%	_LOYAL_
Інтернет	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	2016-09-01 00:00:00	2016-11-01 00:00:00	100%	e.levchenko (Приведи друга)
Інтернет	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	2015-09-01 00:00:00	2015-11-01 00:00:00	100%	e.levchenko
Інтернет	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	2015-08-01 00:00:00	2015-09-01 00:00:00	40%	_LOYAL_

Рисунок 3.1 – Таблиця «Знижки» з заповненням по певному абоненту

Таблиця «Проплати»:

- Дата дата надходження коштів;
- Банк - банк з якого надійшов платіж якщо віє був через касу банку
- Хто вніс - з якої платіжної системи надійшли кошти на рахунок
- Примітка - код самої операції

Проплати (Не показувати минулі проплати)				
Дата	Банк	На рахунок	Хто вніс	Примітки
2025-12-05 17:00:06		250 грн	EasySoft	BillID: 15819018 Date: 2025-12-05 17:00:06
2025-11-11 17:24:49		255 грн	EasySoft	BillID: 15745045 Date: 2025-11-11 17:24:49
2025-10-03 00:00:00		250 грн	portmone	BillID: 2117509191 Auth: 117123
2025-09-02 18:19:05		250 грн	EasySoft	BillID: 15536687 Date: 2025-09-02 18:19:05
2025-08-05 16:28:39		250 грн	EasySoft	BillID: 15470170 Date: 2025-08-05 16:28:39
2025-07-04 07:21:05		250 грн	EasySoft	BillID: 15375764 Date: 2025-07-04 07:21:05
2025-06-03 19:58:44		40 грн	EasySoft	BillID: 15279663 Date: 2025-06-03 19:58:44
2025-06-03 19:33:57		200 грн	Privat	BillID: 15279144 Date: 2025-06-03 19:33:57
2025-05-08 15:02:01		225 грн	EasySoft	BillID: 15196433 Date: 2025-05-08 15:02:01

©

Рисунок 3.2 – Таблиця «Проплати» з заповненням по певному абоненту

Таблиця «Звернення»:

- Дата - дата проведення операції
- Менеджер - хто саме фіксував звернення
- Тип - тип звернення та платформа звернення
- Причина - причина звернення абонента або додаткова інформація

з приводу звернення для наступних користувачів

Дата	Менеджер	Тип	Причина
2021-06-10 12:49:36	System User	Звонок абоненту	Переїзд (немає тех. можливості)
2021-02-11 14:29:59	System User	Звонок абоненту	Вже перейшли на «Vodafone Gigabit Net»
2021-03-11 11:50:12	System User	Звонок абоненту	Висока вартість послуг у порівнянні з конкурентом
2021-10-12 12:18:00	System User	Звонок абоненту	Висока вартість послуг у порівнянні з конкурентом
2021-12-13 11:16:09	System User	Звонок абоненту	Висока вартість послуг у порівнянні з конкурентом
2021-11-10 15:53:18	System User	Звонок абоненту	Переїзд (є тех. можливість)
2021-05-12 15:50:55	System User	Звонок абоненту	Переїзд (є тех. можливість)
2021-04-12 13:06:02	System User	Звонок абоненту	Переїзд (немає тех. можливості)
2021-06-10 12:49:36	System User	Звонок абоненту	Переїзд (немає тех. можливості)
2021-02-11 14:29:59	System User	Звонок абоненту	Вже перейшли на «Vodafone Gigabit Net»
2021-03-11 11:50:12	System User	Звонок абоненту	Висока вартість послуг у порівнянні з конкурентом

©

Рисунок 3.3 – Таблиця «Звернення» з заповненням по певному абоненту

Таблиця «Сервіси»:

- IDSid - унікальний код
- Тип оплати - назва тарифу по якому видається послуга
- Логін - номер договору користувача
- Старт - дата початку послуги
- Стоп - дата закінчення надавання послуги
- Знижка - розмір знижки
- За сервіс - ціна тарифу за надання
- Абонплата - фіксована ціна тарифу
- Ір - ір адреса яка надається користувачу
- Примітки - додаткові записи

Сервіси										
Id	Sid	Тип оплати	Логін	Старт	Стоп	Знижки	За сервіс	Абонплата	Ір	Примітки
Інтернет (Не показувати минулі оплати)										
32041824	9001051	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	41343101850	2025-12-01 00:00:00	0000-00-00 00:00:00		250.00грн	250 грн	100.64.62.48	
31904487	9001051	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	41343101850	2025-11-01 00:00:00	2025-12-01 00:00:00		250.00грн	250 грн	100.64.62.48	
31768372	9001051	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	41343101850	2025-10-01 00:00:00	2025-11-01 00:00:00		250.00грн	250 грн	100.64.62.48	
31662593	9001051	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	41343101850	2025-09-01 00:00:00	2025-10-01 00:00:00		250.00грн	250 грн	100.64.62.48	
31442162	9001051	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	41343101850	2025-08-01 00:00:00	2025-09-01 00:00:00		250.00грн	250 грн	100.64.62.48	
31227222	9001051	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	41343101850	2025-07-01 00:00:00	2025-08-01 00:00:00		250.00грн	250 грн	100.64.62.48	
31180061	9001051	F - 100 (ПС)(100 Мбіт/с - 250 грн./міс.)	41343101850	2025-06-01 00:00:00	2025-07-01 00:00:00		250.00грн	250 грн	100.64.62.48	
30883903	9001051	(10.24) 100 Мбіт/с - 225 грн./міс.	41343101850	2025-05-01 00:00:00	2025-06-01 00:00:00		225.00грн	225 грн	100.64.62.48	
30786489	9001051	(10.24) 100 Мбіт/с - 225 грн./міс.	41343101850	2025-04-01 00:00:00	2025-05-01 00:00:00	15%	222.75грн	225 грн	100.64.62.48	
30637299	9001051	(10.24) 100 Мбіт/с - 225 грн./міс.	41343101850	2025-03-01 00:00:00	2025-04-01 00:00:00	15%	191.25грн	225 грн	100.64.62.48	
30442196		(10.24) 100 Мбіт/с - 225 грн./міс.	41343101850	2025-02-01 00:00:00	2025-03-01 00:00:00	15%	191.25грн	225 грн	100.64.62.48	

Рисунок 3.4 – Таблиця «Звернення» з заповненням по певному абоненту

3.3 Опис функціонування веб-орієнтованої системи «Billing»

Робота системи «Billing» передбачає взаємодію роботу користувачів з різним типом доступу у своїх секторах відповідальності з системою через мережевий інтерфейс.

Ключові загальні функції системи базуються на обробці запитів різних категорій але всі вони можливі лише на основі даних які користувачі отримують з бази даних. Всі дії проведені зберігаються та фіксуються [16].

Як тільки користувач отримав доступ до системи згідно зі своїм рівнем допуску, він може розпочати роботу з системою та її функціональними елементами такими як навігаційні меню на бічній панелі, форми введення даних, кнопки та посилання [23].

Користувачі взаємодіють із системою «Billing», за допомогою виконання логічних дій, як заповнення форм, вибір пунктів меню або натискання кнопок.

Система «Billing» обробляє запит користувача, перевіряє правильність введених даних, виконує необхідні операції та зберігає результати в базі

даних. Наприклад, якщо користувач заповнює новий лід для підключення абонента система перевіряє по перше наявність технічної можливості підключення до адреси майбутнього абонента, наявність вільних портів на обладнанні та технологію підключення, зберігає персональні дані в базі даних і повідомляє користувача про збереження та фіксацію ліда [25].

Система забезпечує вивод інформації по певному запиту користувача. Це включає виведення результатів запитів до бази даних системи «Billing», відображення списків даних, графіків та звітів.

Користувачі можуть переглядати договори абонентів, створені ліди та їх статус, адреси де можливо підключення, список співробітників та їх посади.

Функціонування веб-орієнтованої системи «Billing» також може включати ряд додаткових функцій, що сприяють зручності та ефективності роботи системи, таких як зручна зміна паролів в особистому кабінеті, перегляд розкладу всіх співробітників за днем роботи.

Робота системи «Billing» базується на взаємодії користувачів із системою через веб-інтерфейс, обробці запитів, зберіганні та отриманні даних із баз даних, наданні інформації та виконанні дій, які відповідають потребам користувачів системи [21].

Коли користувач відкриває вебсайт системи «Billing», він потрапляє на головну сторінку. (рис. 3.5). Одразу на ній розташовані елементи керування для переходу на:

- сторінку входу в існуючий профіль (рис. 3.5 – рис. 3.8);
- головну сторінку системи (рис. 3.9);
- сторінку Нового ліда (рис. 3.12);
- сторінку Лідів (рис. 3.17);
- сторінку договору (рис. 3.19);

Перша сторінка, яку зустрічає користувач це «Авторизація» (рис. 3.5), ця сторінка є входом у систему та першою перевіркою безпеки. Вхід відбувається за алгоритмом вводу даних таких як логін та пароль у відповідні їм поля.

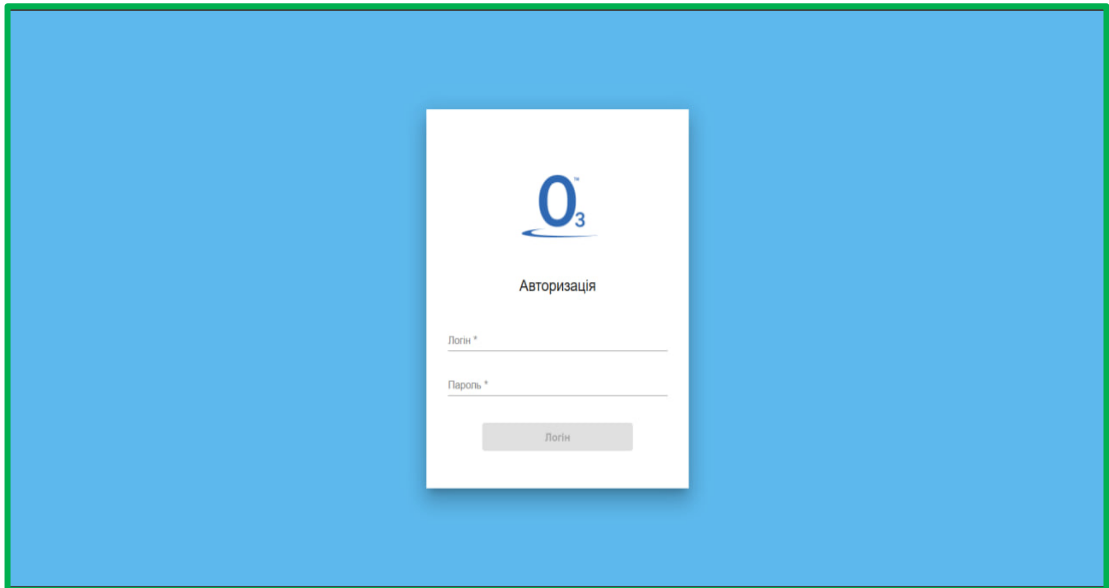


Рисунок 3.5 – Форма входу в профіль системи «Billing»

Форма має перевірку на заповненість та вірність введених даних (рис. 3.6, рис. 3.8). Наступним буде представлена помилка заповнення полів. Коли користувач залишає поля не заповненими.

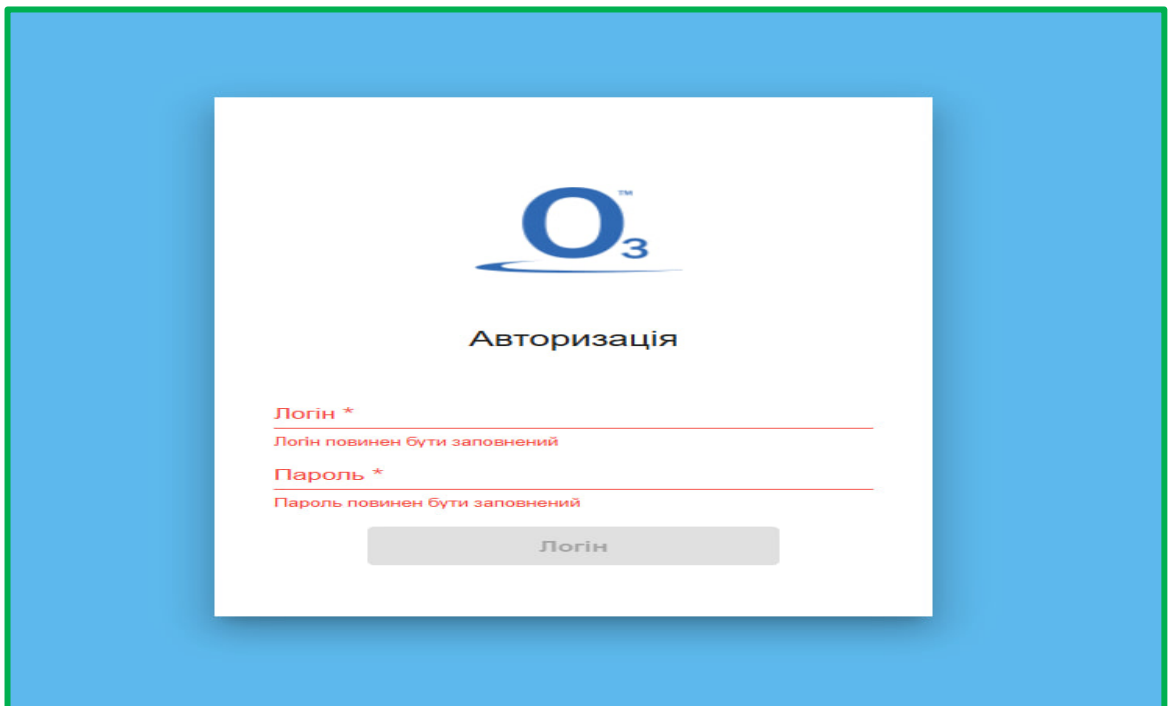


Рисунок 3.6 – Приклад помилки формами входу облікового запису

Форма входу (рис. 3.7). має представлену одну з можливих помилок доспуа користувача який не зафіксован в системі. Система перевіряє одразу введення двох значень.

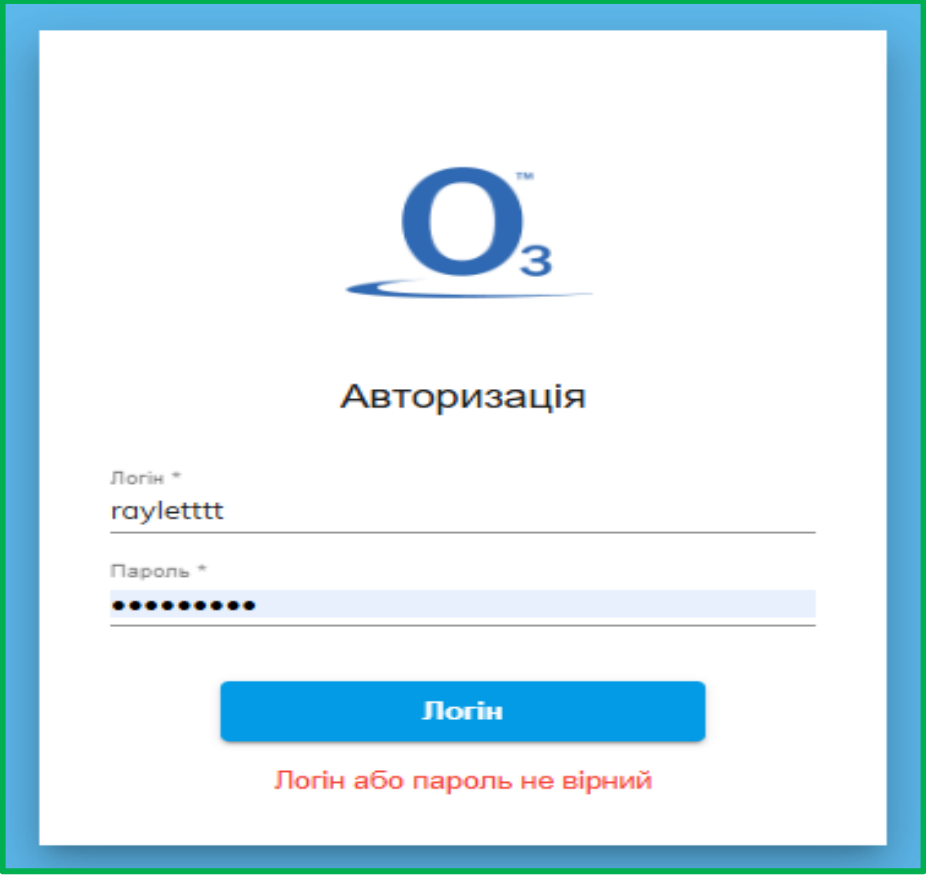
The image shows a login form for a system called 'Billing'. At the top is a logo consisting of a blue 'O' with a subscript '3' and a trademark symbol. Below the logo is the title 'Авторизація' (Authorization). There are two input fields: 'Логін *' (Login *) with the text 'rayleTTTT' and 'Пароль *' (Password *) with masked characters. A blue button labeled 'Логін' (Login) is positioned below the fields. At the bottom, a red error message reads 'Логін або пароль не вірний' (Login or password is incorrect). The entire form is enclosed in a blue and green border.

Рисунок 3.7 – Форма входу з помилковим логіном з системи «Billing»

Коректне заповнення форми дозволяє доступ до системи. Приклад коректного заповнення (рис. 3.8)

Після правильного заповнення полів (рис. 3.8) користувача автоматично переправить на головну сторінку сторінку (рис. 3.9).

О₃TM

Авторизація

Логін *

raylet

Пароль *

.....

Логін

Рисунок 3.8 – Сповіщення про некоректне заповнення

Після введення коректних даних користувач потрапляє на головну сторінку системи яка дозволяє проводити пошук за різними критеріями. Система дозволяє знайти абонента за різними критеріями:

Особисті дані:

- ПІБ – Абонента який зазначений в системі
- Телефон – телефон абонента який зазначений в системі
- e-mail – пошта що зазначена в системі, якщо пошта не зазначена система не знайде відповідни договір
- Баланс – точний баланс рахунку. Виведе всі договори з таким балансом що є в системі
- ЄДРПОУ (для юр. осіб) - тільки для Юридичних договорів

- Тип договору- тип договору такий як фізичний чи юридичний договір

Адреса

- Населений пункт – населений пункт який зазначений в договорі
- Вулиця – вулиця що зазначена у новому та колишньому значені
- Будинок- номер будинку та корпус якщо такий є
- Квартира – номер квартири якщо присутня

Технічні дані

- ID- унікальний номер абонента
- Логін – номер договору
- IP-адреса – ір адреса що привязана до договору
- MAC-адреса- адреса пристрою під якою абонент виходить в інтернет

за певним підключенням

- Тип сервісу – тип послуги що надається
- Тарифний план – тариф за яким підключено

Фільтри

- показувати закриті договори – відображає також минулі договори що вже в статусі закритих

- тільки фіз. Особи – окремо виводить тільки фіз особи
- тільки юр. Особи – окремо виводить тільки юридичні договори
- тільки VIP – окремо виводить тільки договори типу VIP
- приватний сектор – виводить тільки підключення за приватним сектором

сектором

- відсутні паспортні дані – договори з відсутніми паспортними даними

На ведені вижче критерії зумовлюють точне знаходження абонента при наявності в наступного коректних даних для можливої співпраці.

Це дуже зручно для оператора call-центру, техпідтримки, технічного відділу інженерів та абоненського відділа та відділю підключення

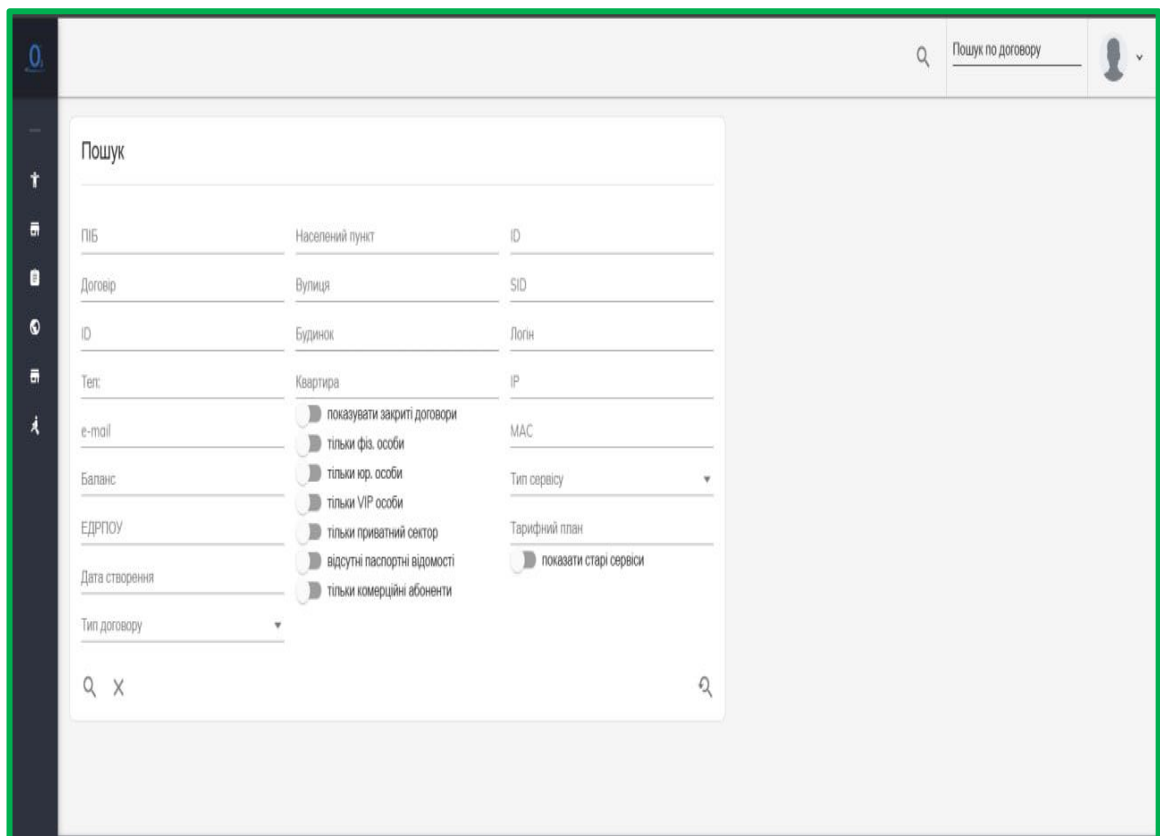


Рисунок 3.9 – Головна сторінка веб-орієнтованої системи «Billing»

Для зручності роботи з лівої сторони сторінки є меню швидкого переходу.

Дане меню для ергономічності сторінки можна згорнути або залишити в розгорнутому вигляді (рис. 3.10)

- Абоненти – робота з клієнтами, що вже внесені в систему.
- Продажі – підключення нових користувачів через систему лідів (рис. 3.11) та створення нових лідів (рис. 3.10)
- Заявки – заявки на ремонт, підключення, техпідтримку.
- DHCP – управління IP-адресами.
- Будинки та вулиці – адресна база.
- Персонал – співробітники та їх доступи.

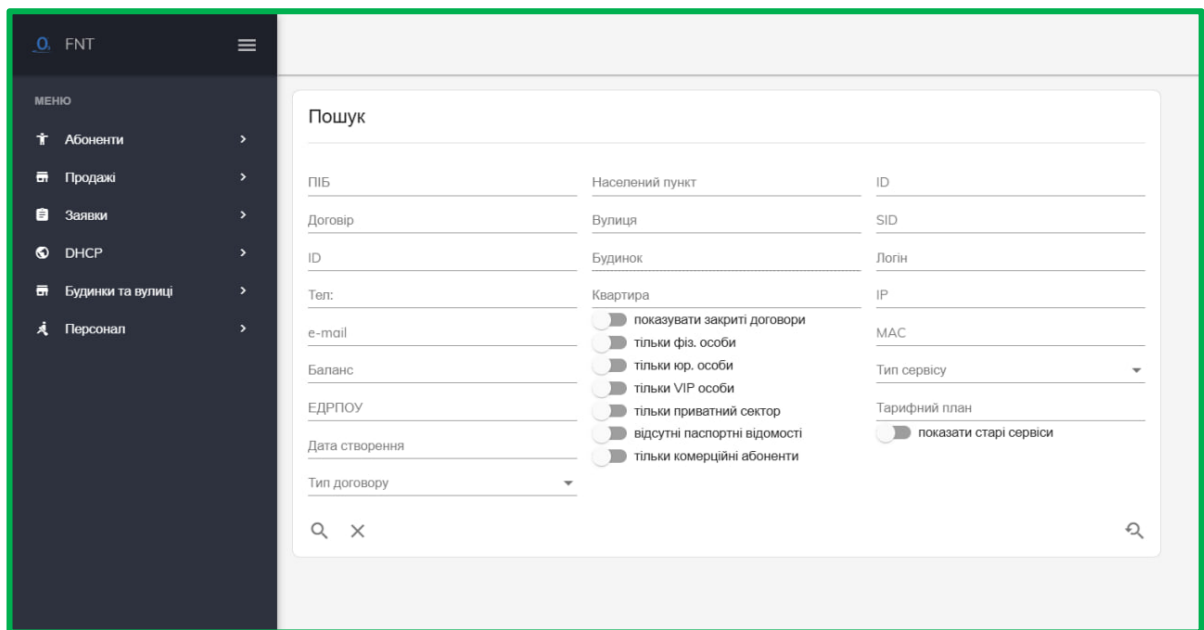


Рисунок 3.10 – Головна сторінка з розгорнутим боковим меню

Ліва панель дозволяє перейти до різних пунктів але більш використані це пункт «Продажі». (рис. 3.9). Цей пункт надає користувачу змогу зробити новий лід на підключення чи знайти вже існуючий.

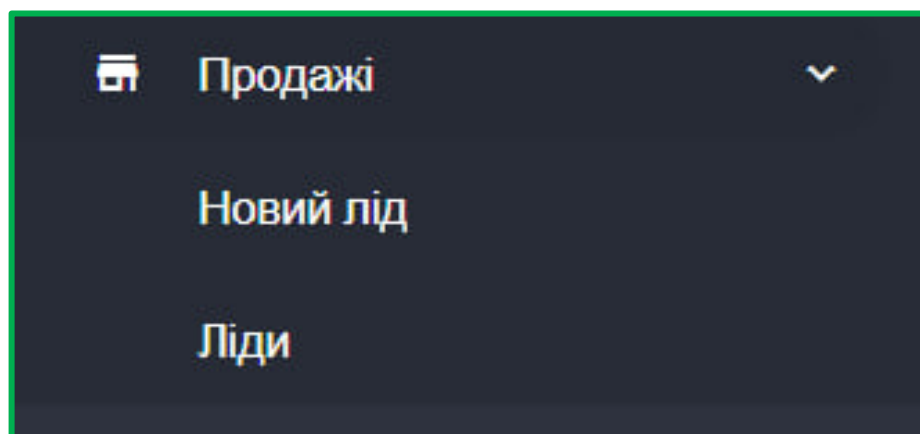


Рисунок 3.11 – Сторінка «Продажі» список

Після натискання на кнопку «Новий лід.» користувач бачить електронну форму для заповнення ліда(рис. 3.12).

Електронна форма дозволяє перевірити технічну можливість підключення до мережі покриття провайдера та заповнити основні дані для майбутнього договору та співпраці.

Рисунок 3.12 – Сторінка «Новий лід» вибору дати та лікаря

Якщо всі відповідні пункти були заповнені можливо створити новий лід який буде зафіксовано в системі для цього треба натиснути кнопку створити (рис. 3.13).

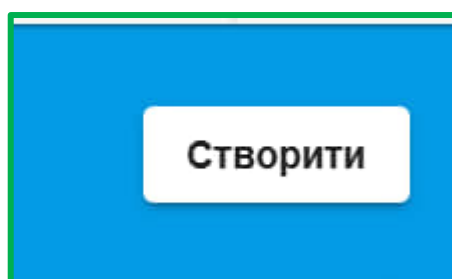


Рисунок 3.13 – Зовнішній вигляд кнопки створення

При написканні на кнопку користувач бачить відповідне повідомлення про успішне створення ліда (рис. 3.14). Повідомлення відображається 5 секунд та зникає з екрану [25].

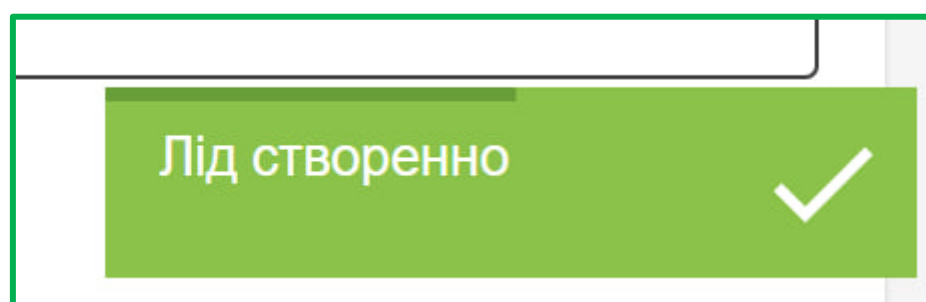


Рисунок 3.14 – Вигляд повідомлення про успішне створення ліда

Після створення лід можливо редагувати для подальшої роботи з ним. Кнопка створити тепер змінена на кнопку зберегти (рис. 3.15).

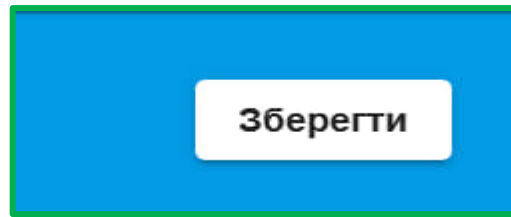


Рисунок 3.15 – Зовнішній вигляд кнопки збереження

Повідомлення про збереження змін у відповідному ліді після натискання кнопки(рис. 3.16).

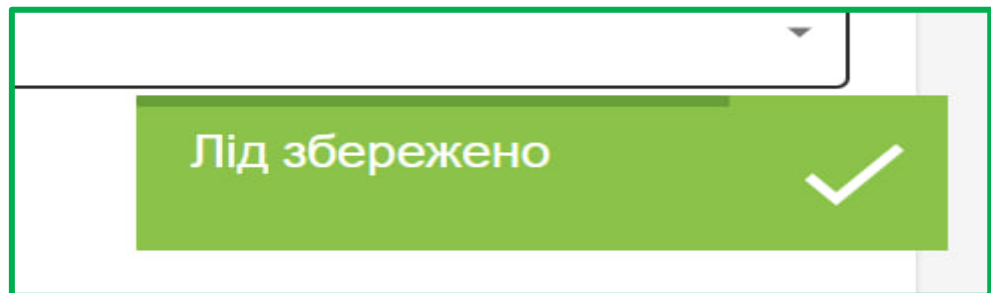


Рисунок 3.16 – Вигляд повідомлення про успішне створення ліда

Після створення лід знаходиться в загальному списку лідів (рис. 3.17).

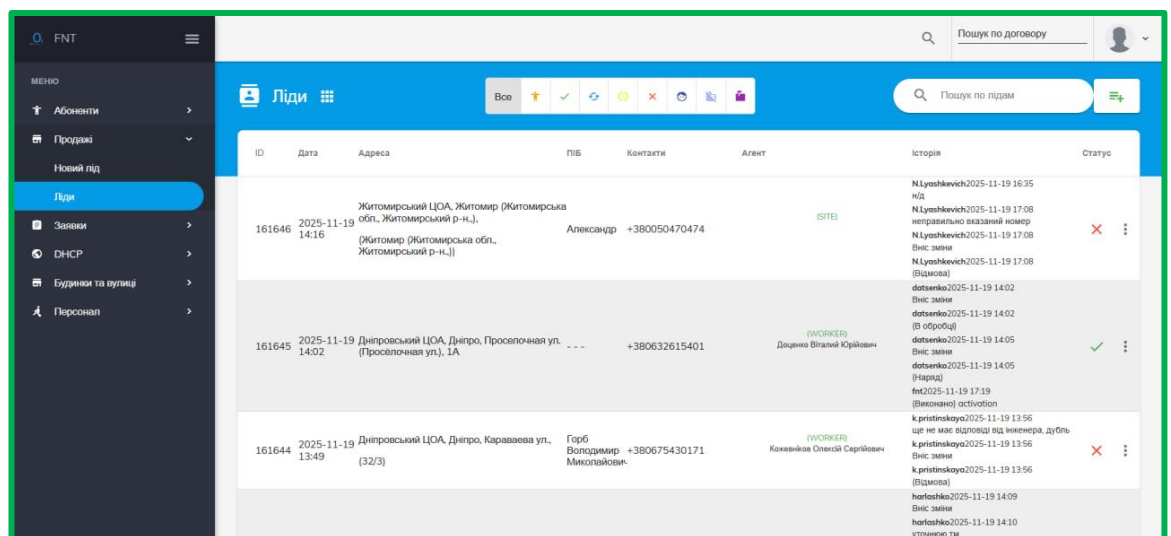


Рисунок 3.17 – Сторінка лідів

Сторінка «Ліди» має зручне поле фільтрації яке оформлене як іконки (рис. 3.18).

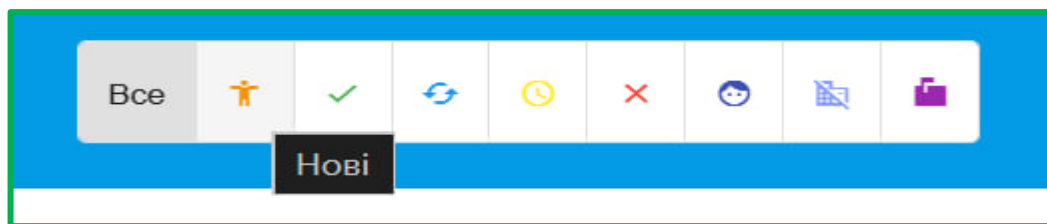


Рисунок 3.18 – Зовнішній вигляд поля фільтрації для лідів

Після обрання одної з категорій список буде відфільтровано по вибраній категорії для зручності створено такі категорії [7]:

- Всі - показує всі наявні ліди
- Нові - нові або необроблені звернення
- Успішні - оброблені та підтверджені ліди
- В роботі - очікують або в роботі у оператора
- Відмінені - показують ліди помилкові чи недійсні
- Відкриті - показує ліди з якими зараз працюють
- Експорт - можливість експорту списку
- Імпорт - імпорт залежно від системи

Після ознайомлення та підключення створюється договір. Сторінка «Договір» (рис. 3.19) це основна сторінка для взаємодії з ним

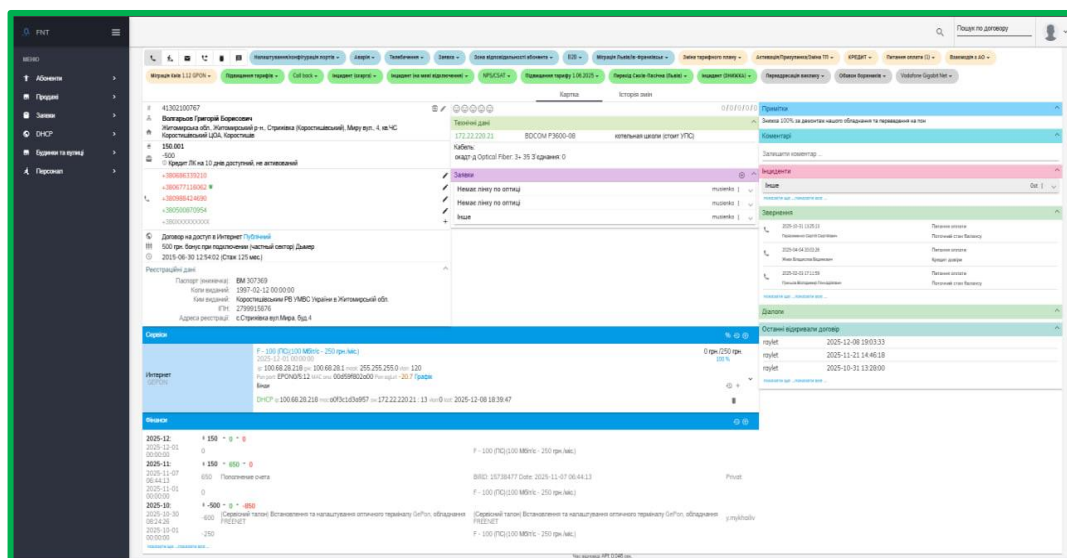


Рисунок 3.19 – Зовнішній вигляд сторінки «Договір»

Перед початком роботи з договором та надані інформації треба провести перевірку основних даних абонента чи первинну верифікація.

Починати треба з перевірки інформації в лівій частині картки – це базова інформація про клієнта так називаний блок Особистих даних по договору

Обов'язково призвернені проводиться зв'язка наступні дані при кожному дзвінку або зверненні:

- Повне ім'я (ПІБ): чи відповідає особі, яка звернулася.
- Адреса підключення: коректність будинку та квартири.
- Контактні телефони: можливість зворотного зв'язку.
- Статус договору: найчастіше «Активний», «Призупинений», «Не активований», «Розірваний». Це критично важливо для розуміння можливості надання послуг.

Договір та активні послуги (Тариф)

1. Знайдіть на картці блок, присвячений послугам, найчастіше він називається “Інтернет”, “Послуги” або “Тарифний план”. Можливо, потрібно буде прокрутити сторінку вниз.

2. У цьому блоці перевірте:

- Назву тарифного плану (наприклад, “Світло 100 Мбіт/с”).
- Заявлену швидкість підключення (до 100 Мбіт/с, до 1 Гбіт/с).
- Щомісячну вартість послуги.
- Статус оплати: чи внесені кошти за поточний період.

3. Діагностика: перевірка технічного стану

Для виявлення причин відсутності зв'язку перейдіть до технічного блоку, який зазвичай знаходиться в правій частині екрану.

- Перейдіть до блоку “Технічні дані”, “Обладнання” або “Мережеві параметри”.
- Знайдіть інформацію про пристрій абонента:

- Тип обладнання (ONU/ONT/роутер): наприклад, Huawei HG8245H, ZTE F670.
- Серійний номер (S/N) або MAC-адреса пристрою.
- Аналіз параметрів підключення (якщо система відображає в реальному часі):
 - Оптичний порт (PON ID): номер порту на обладнанні провайдера, до якого підключений абонент.
 - Рівень оптичного сигналу (Rx Power): ключовий параметр. Норма зазвичай в діапазоні від -8 до -26 dBm. Сигнал гірший за -27 dBm може свідчити про проблеми на лінії (перегини, поганий зварювальний стик).
 - Статус авторизації пристрою в мережі (Online/Offline/Registering).

Практична користь: Якщо абонент скаржиться на відсутність інтернету, а пристрій в статусі “Offline” або рівень сигналу критично низький – причина, швидше за все, в технічній проблемі на лінії, а не в налаштуваннях його комп’ютера [25].

1. *Контроль процесів:* активні заявки

Переконайтеся, що проблема абонента ще не знаходиться в опрацюванні.

- Знайдіть на картці вкладку або блок “Заявки”, “Тікети” або “Завдання”.
- Уважно перегляньте список. Особливу увагу звертайте на заявки з типами:
 - “Немає лінку” / “Немає оптики” – проблеми з фізичним підключенням.
 - “Немає інтернету” / “Не працює інтернет” – загальна проблема з доступом.
 - “Заміна обладнання”, “Консультація”, “Інше”.

- Перевірте статус заявки: “Відкрита”, “В роботі”, “Виконана”, “Відкладена”.
- Зверніть увагу на виконавця: це може бути “Мультилінк” (черга обробки), “Технік Іванов” або інша служба.
- Важливе правило: Якщо відкрита заявка на ту ж проблему вже існує – не створюйте дубля. Краще додайте коментар до існуючої або повідомте відповідного виконавця [21].

2. *Фінансовий аналіз*: перевірка стану рахунку

Одна з найпоширеніших причин проблем з доступом – негативний баланс.

- Перейдіть до вкладки “Фінанси”, “Баланс”, “Рахунок” або “Платежі”.
- Основний показник – поточний баланс, який зазвичай відображається великими цифрами на початку блоку.
 - Баланс позитивний (наприклад, +150 грн): зазвичай виділяється зеленим кольором. Фінансових перешкод для роботи послуг немає.
 - Баланс негативний (наприклад, -50 грн): зазвичай виділяється червоним кольором. Це борг. При певному розмірі боргу система може автоматично обмежити або призупинити надання послуг.
- Для глибшого аналізу перегляньте історію операцій: останні нарахування (рахунки), списання (оплата послуг), платежі (внесені кошти).

Ключова логіка: Якщо абонент каже “не працює інтернет”, а баланс негативний – спочатку поясніть необхідність погашення боргу і лише потім переходите до технічної діагностики [23].

3. *Контекст спілкування*: історія звернень

Перед тим, як почати діалог, дізнайтеся, чи звертався клієнт раніше і з яким питанням.

- Відкрийте блок “Звернення”, “Історія дзвінків” або “Лог контактів”.
- У списку ви побачите попередні записи:
 - Дата та час звернення.
 - Оператор або відділ, який приймав дзвінок/заявку (наприклад, “Кол-центр”, “Техпідтримка”).
 - Тема або короткий опис питання (“Скарга на швидкість”, “Консультація з роутером”).
- Як це допомагає: Це дозволяє уникнути повторних питань, зрозуміти “хронічні” проблеми абонента та забезпечити послідовність в комунікації.

4. Загальні проблеми: перевірка інцидентів

Завжди перевіряйте, чи не є проблема масовою.

- Перейдіть до блоку “Інциденти”, “Аварії” або “Масові проблеми” (іноді це окремий елемент меню для всієї системи).
- Система може показувати активні аварії з прив’язкою до районів, вулиць або конкретних вузлів зв’язку.
- Чіткий алгоритм дій:
 - Якщо в районі абонента є активний інцидент – не створюйте індивідуальну заявку. Занотуйте номер інциденту та орієнтовний час відновлення, вказаний інженерами.
 - Повідомте про це клієнта: “На вашій вулиці проводиться ремонтні роботи, послуга буде відновлена до 18:00”.

5. Документування: додавання коментаря (примітки)

Кожна ваша дія або отримана інформація має бути зафіксована.

- Знайдіть поле або блок “Примітки”, “Коментарі”, “Нотатки”.
- Впишіть службову інформацію в вільну форму, але структуровано. Зразок:

- [12.05.2024 14:30] Дзвінок від абонента. Скарга на низьку швидкість вечорами.
- Клієнт перевіряв швидкість по Wi-Fi: 15 Мбіт/с при тарифі 100 Мбіт/с.
- Пояснено про можливе завантаження Wi-Fi-каналу. Запропоновано перевірити через кабель.
- Створено заявку №450128 для перевірки лінії.
- Обов'язково натисніть кнопку “Зберегти”, “Додати” або “Відправити”.

6. *Ескалація проблеми*: створення нової заявки для техпідтримки

Якщо проблему не вдалося вирішити віддалено, її потрібно передати технічним фахівцям.

- Детально опишіть проблему в полі “Опис” або “Коментар”:
 - Час виникнення: “Не працює з 10:00 сьогодні”.
 - Симптоми: “Інтернет повністю відсутній, світлодіод на роутері блимає червоним”.
 - Виконані дії: “Абонент перезавантажував роутер. Обладнання в системі Offline, рівень сигналу -30 dBm”.
 - Контакт для зворотного зв'язку.
- Натисніть “Зберегти” або “Створити”. Система присвоїть заявці унікальний номер, який можна повідомити абоненту для відстеження.

7. *Аудит*: перегляд історії змін

Цей блок дає повну картину всіх взаємодій з клієнтом.

- Відкрийте вкладку “Діалоги”, “Історія дій”, “Аудит-лог”.
- Тут системно фіксується хто, коли і що робив в картці:
 - Зміна тарифного плану (хто і коли).
 - Додані коментарі (хто і які).
 - Створені та закриті заявки.
 - Зміни в технічних параметрах (MAC-адреси, порти).

- Фінансові операції (нарахування, списання, кредити).
- Це головний інструмент для контролю якості, навчання та розбору складних ситуацій.

8. *Завершення роботи:* фінальний контроль

Перш ніж закрити картку абонента і перейти до наступного клієнта, виконайте короткий фінальний чек-лист:

Перевірено баланс (наявність боргу чи позитивний баланс).

За допомогою стрілочки можна прочитати всі відгуки що були залишені.



Рисунок 3.20 – Вид кнопок по кольоровому сегменту

Всі основні кнопки були розділені за кольоровою гамою для зручності роботи з різними зверненнями та різними відділами. Сині технічні питання, жовті питання обслуговування фінансів чи тарифів, зелені скарги чи невдоволення або перевірка якості, сірі технічні внутрішньої компанії.

Рисунок 3.21 – Зовнішній вигляд форми звернення

Всі звернення фіксуються двома формами. Це причина та коментар до неї (рис. 3.21).

При натисканні на форму причини буде видно список можливих причин, які найчастіше були в даній категорії та шаблони для них. Даний метод прискорює роботу користувача (рис. 3.22).

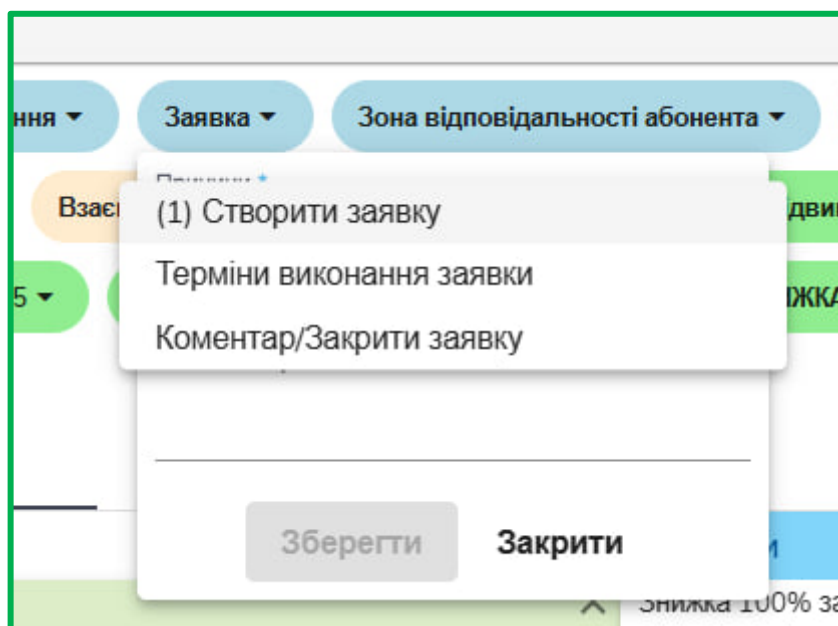


Рисунок 3.22 – Приклад заповнення форми звернення з кнопки «Заявка»

Коли натискається кнопка «Зона відповідальності абонента», ми дачимо інший список доступних причин.

Для прикладу оберемо причину «Погіршення якості/Відсутність послуги» (рис. 3.23).

В даному пункті ми дачи підказку для користувача в вигляді шаблону для швидшого заповнення з цього звернення.

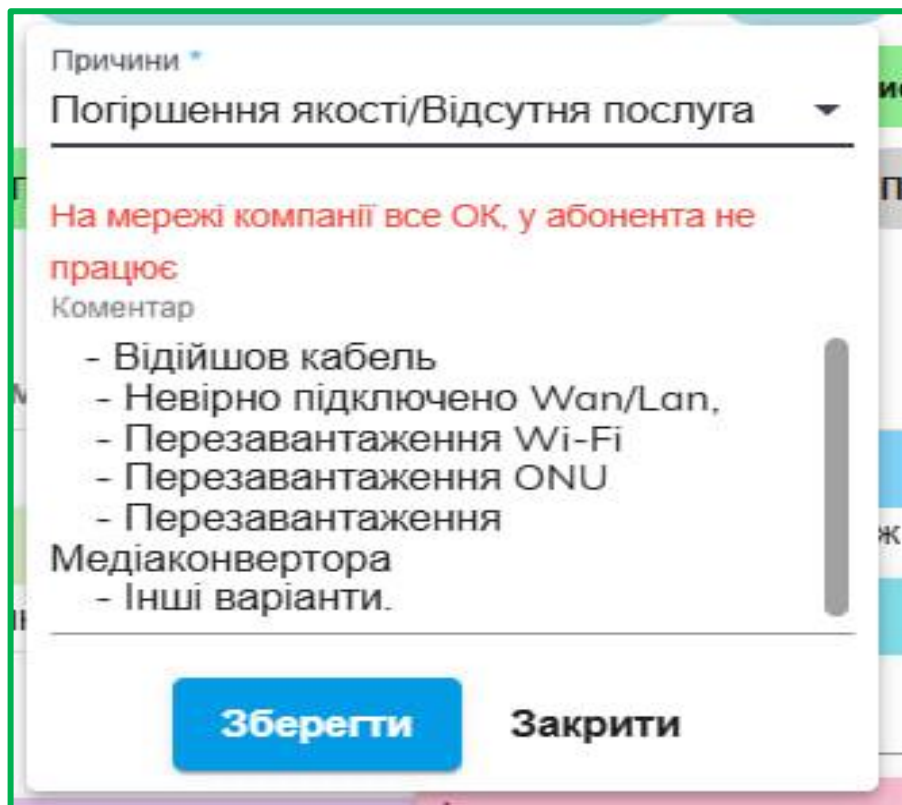


Рисунок 3.23 – Повідомлення про успішне надсилання даних

При видорі фінансової кнопки «Питання оплати»(рис. 3.24). Ми бачимо список фінансових питань та кількість звернень по ним.

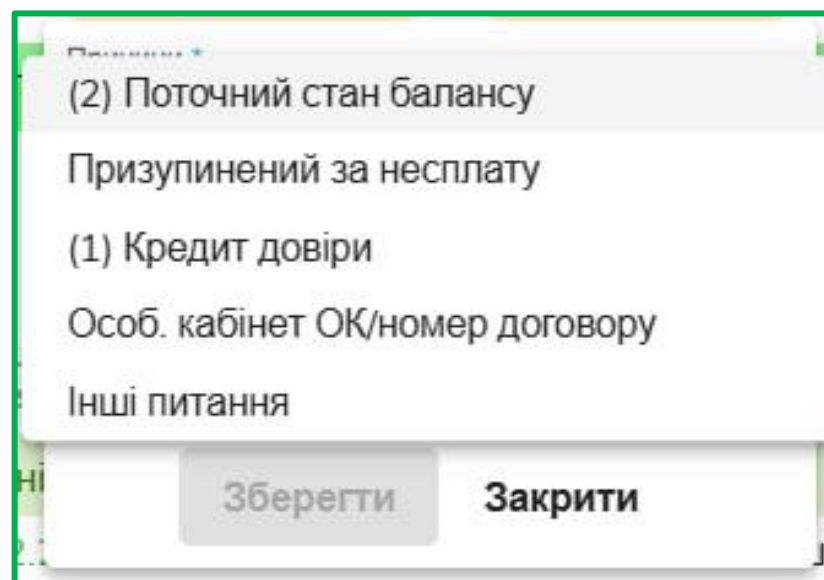
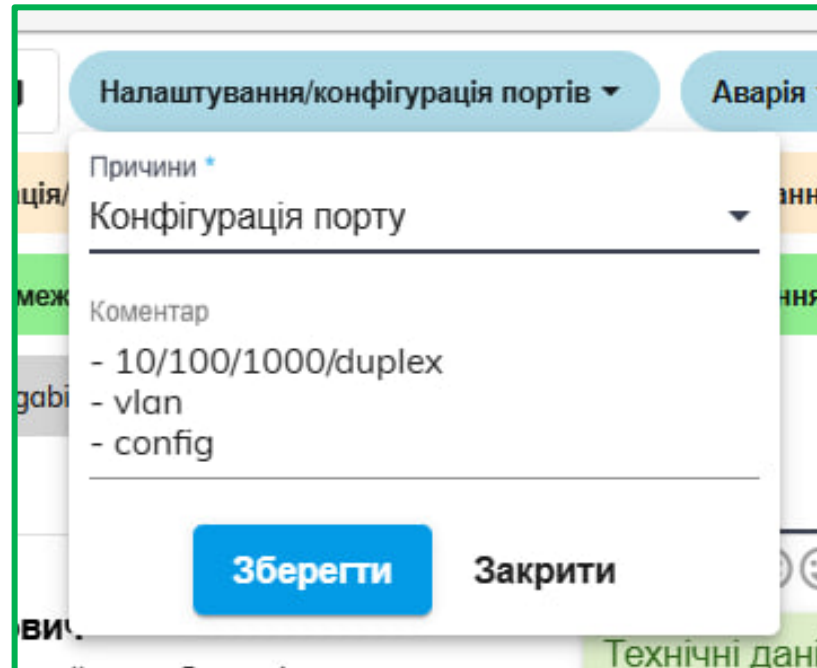


Рисунок 3.24 – Вигляд у базі даних тестового прикладу

Технічне звернення «Налаштування/конфігурація портів»(рис.3.25), демонструє нам підпункт «Конфігурація порту». Де користувач має обрати які саме налаштування він змінив. Перший рядок представляє швидкість порту, другий vlan, третій config налаштування порту як його статус та тип роботи.



Налаштування/конфігурація портів ▾ Аварія ▾

Причини *

Конфігурація порту ▾

Коментар

- 10/100/1000/duplex
- vlan
- config

Зберегти Закрити

Технічні дані

Рисунок 3.25 – Сторінка з контактною інформацією

ВИСНОВКИ

У ході виконання магістерської дипломної роботи було проведено комплексне дослідження теоретичних, методологічних та прикладних аспектів розроблення сучасної білінгової системи для телекомунікаційних компаній та сервісних центрів.

У роботі проаналізовано особливості функціонування білінгових платформ, їх роль у забезпеченні точного обліку послуг, автоматизації фінансових операцій, інтеграції з CRM-системами та системами обробки викликів, а також у підвищенні ефективності бізнес-процесів підприємства.

Проведений огляд сучасного стану білінгових систем засвідчив, що ринок активно трансформується у бік комплексних рішень, здатних забезпечувати не лише нарахування платежів, а й взаємодію з абонентськими базами, підтримку інтеграцій через API, аналітику, звітність та обробку великих обсягів телекомунікаційних даних.

Окрема увага була приділена аналізу CRM-платформ (Salesforce, Zendesk, HubSpot, Freshdesk), які частково реалізують білінгові функції, але не покривають потрібного рівня деталізації та автоматизації для повноцінного управління тарифами, послугами та CDR-записами. Це підтвердило необхідність створення окремого, спеціалізованого білінгового рішення.

У процесі роботи було спроектовано архітектуру білінгового програмного продукту, що включає модуль управління абонентами, модуль тарифікації, систему збору та обробки CDR-записів, інтерфейс адміністрування, API для інтеграції з іншими сервісами, а також базу даних, оптимізовану для високонавантажених операцій. Особлива увага приділялася вибору моделі даних: у роботі було доведено доцільність використання реляційної архітектури для забезпечення транзакційності, консистентності та можливості проведення складних вибірок у режимі реального часу.

Проектування API дозволило створити універсальний механізм взаємодії системи з зовнішніми платформами, зокрема з АТС, CRM,

мобільними застосунками та адміністративними сервісами. У роботі описано принципи побудови REST-інтерфейсу, механізми авторизації та безпеки, а також схеми стандартних викликів для отримання та обробки білінгових даних. Такий підхід забезпечує масштабованість, гнучкість та можливість інтеграції з будь-якими сучасними ІТ-системами.

Окремим результатом є розроблення архітектури системи збору CDR (Call Detail Record). У роботі проаналізовано структуру CDR-записів, принципи їх отримання, збереження та обробки, можливі проблеми синхронізації, а також методи оптимізації обчислень під час розрахунку вартості викликів. У результаті створено модель, здатну працювати з великими потоками даних і забезпечувати точність тарифікації.

Практична частина роботи продемонструвала можливість реалізації білінгового рішення на сучасних технологічних стекових інструментах, зокрема Java, Spring Boot, Gradle, REST, реляційних СУБД.

Реалізація ключових модулів підтвердила коректність теоретичних рішень, адаптивність архітектури та можливість подальшого масштабування системи.

Узагальнюючи результати дослідження, можна зробити такі висновки:

1. Білінгові системи залишаються критично важливим елементом телекомунікаційної інфраструктури та потребують постійного розвитку у зв'язку зі зростанням обсягів даних і складністю тарифікації.

2. Існуючі CRM-системи не можуть повністю замінити спеціалізований білінг, що обґрунтовує необхідність створення власних технологічних продуктів.

3. Розроблена архітектура білінгового рішення є гнучкою, масштабованою та адаптованою до потреб сучасних компаній, що працюють з великими масивами телекомунікаційних даних.

4. Проєктована база даних та API забезпечують надійну основу для інтеграції з іншими сервісами, такими як АТС, мобільні додатки чи платіжні системи.

5. Створена модель обробки CDR забезпечує високу точність та оптимальний розподіл навантаження, що є ключовим аспектом для телеком-галузі.

6. Практична реалізація підтвердила життєздатність запропонованих технічних рішень та можливість їх подальшого впровадження на підприємствах.

Таким чином, магістерська робота досягла поставленої мети: створено науково обґрунтований, сучасний та технологічно ефективний підхід до проєктування та розроблення білінгового програмного продукту, який здатний суттєво підвищити ефективність роботи сервісних та телекомунікаційних компаній.

Отримані результати можуть бути використані як основа для подальшого розвитку системи, її комерційного впровадження, розширення функціоналу, інтеграції з новими інформаційними платформами та впровадження у реальних бізнес-процесах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Таненбаум Е., Ветеринг Х. Комп'ютерні мережі. Київ : Вільямс, 2020. 960 с.
2. Кормен Т., Лейзерсон Ч., Рівест Р., Штайн К. Алгоритми. Вступ до проектування та аналізу. Київ : ДІАЛОГ, 2019. 1320 с.
3. Stallings W. Data and Computer Communications. 10th ed. Pearson, 2021.
4. Fogelström E., Li W., Bertin E. Billing Systems for Telecommunication Networks: Architecture and Implementation. IEEE Communications Surveys & Tutorials, 2019.
5. ETSI. Digital cellular telecommunications system (Phase 2+); Call Detail Record (CDR) structures. ETSI TS 129 002. 2021.
6. RFC 4006. Diameter Credit-Control Application. IETF, 2020. URL: <https://datatracker.ietf.org/doc/html/rfc4006>
7. O'Reilly Media. RESTful Web APIs: Services for a Changing World. 2019.
8. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley, 2022.
9. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley, 2019.
10. Beaulieu A. Learning SQL. 3rd ed. O'Reilly, 2021.
11. MySQL Documentation. MySQL 8.0 Developer Guide. Oracle, 2024. URL: <https://dev.mysql.com/doc/>
12. PostgreSQL Global Development Group. PostgreSQL Documentation 16. 2024. URL: <https://www.postgresql.org/docs/16/>
13. Spring Framework Documentation. Spring Boot Reference Guide. 2024. URL: <https://docs.spring.io/spring-boot/docs/current/reference/html/>
14. OpenAPI Initiative. OpenAPI Specification 3.1. 2023. URL: <https://spec.openapis.org/oas/3.1.0>
15. Twilio. Call Detail Records (CDR) API Documentation. 2024. URL: <https://www.twilio.com/docs>

16. Cisco Systems. Call Accounting and Billing for VoIP Networks. Whitepaper. 2023.
17. Freshdesk. Freshdesk API & Billing Functions. 2024. URL: <https://developers.freshdesk.com>
18. HubSpot. CRM & Billing API Overview. 2024. URL: <https://developers.hubspot.com>
19. Salesforce. Salesforce Billing API Guide. 2024. URL: <https://developer.salesforce.com>
20. Zendesk. Zendesk Billing, CRM, and Integration APIs. 2024. URL: <https://developer.zendesk.com>
21. Особливості побудови білінгових систем у телекомунікаційних компаніях України. Сучасні інформаційні системи. №2. 2022.
22. Martin R. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2018.
23. Коваленко С., Марченко П. Проектування веб-систем: архітектури та технології. Київ : КНЕУ, 2021.
24. ISO/IEC 27001:2022. Information Security Management Systems.
25. ПрАТ «ФРІНЕТ». Офіційний сайт. Особливості білінгу інтернет-операторів. 2024. URL: <https://www.o3.ua>
26. Зінченко А.М. ВЕБЗАСТОСУНОК «МЕДИЧНА ПОСЛУГА». *Науковий журнал «IT SYNERGY»*, 2024, випуск 1 (6). № 10(51). С. 74-91. 2024. URL: <https://doi.org/10.53920/ITS-2024-1-6>
27. Ткаченко О.І., Зінченко А.М. Білінгова система Call-центру – інструмент оптимізації роботи та аналітики обслуговування клієнтів. *Актуальні питання розвитку галузей: збірник наукових праць з матеріалами VI Міжнародної наукової конференції* (м. Київ, 26 листопада, 2025), 2025. С.28 – 32

ДОДАТКИ

Додаток А: Код сторінки нового ліда

```
<!DOCTYPE html><html lang="uk"><head>

  <title>Billing::ISP 03</title>

  <base href="/spa/">

  <meta charset="utf-8">

  <meta name="description" content="ISP billing">

  <meta name="keywords" content="Activation, billing">

  <meta name="author" content="Rybachuk Dmytro">

  <meta      name="viewport"      content="width=device-width,
initial-scale=1.0, maximum-scale=5.0">

  <meta name="msapplication-TileColor" content="#2b5797">

  <meta                                     name="msapplication-config"
content="browserconfig.xml">

  <meta name="theme-color" content="#1976d2">

  <link      rel="apple-touch-icon"      sizes="114x114"
href="assets/icons/apple-touch-icon.png">

  <link      rel="icon"      type="image/png"      sizes="32x32"
href="assets/icons/favicon-32x32.png">

  <link      rel="icon"      type="image/png"      sizes="16x16"
href="assets/icons/favicon-16x16.png">

  <link rel="mask-icon" href="assets/icons/safari-pinned-
tab.svg" color="#5bbad5">

  <link rel="manifest" href="manifest.json">

  <noscript>
```

<p>Робота з біллінгом потребує JavaScript. Скорегуйте налаштування Вашого браузера для роботи з JavaScript.</p>

</noscript

<!-- APP Splash Screen CSS -->

<style type="text/css">

```
#app-splash-screen {  
    display: block;  
    position: absolute;  
    top: 0;  
    left: 0;  
    right: 0;  
    bottom: 0;  
    background-color: #2D323D;  
    z-index: 99999;  
    pointer-events: none;  
  
#app-splash-screen .center {  
    display: block;  
    width: 100%;  
    position: absolute;  
    top: 50%;  
    transform: translateY(-50%);  
  
#app-splash-screen .logo {  
    width: 128px;  
    margin: 0 auto;
```

```

    }

    #app-splash-screen .logo img {

        filter: drop-shadow(0px 10px 6px rgba(0, 0, 0,
0.2))

    }

    #app-splash-screen .spinner-wrapper {

        display: block;

        position: relative;

        width: 100%;

        min-height: 100px;

        height: 100px;


    }

    #app-splash-screen .spinner-wrapper .spinner {

        position: absolute;

        overflow: hidden;

        left: 50%;

        margin-left: -50px;

        animation: outer-rotate 2.91667s linear infinite;

    }

    #app-splash-screen .spinner-wrapper .spinner .inner

    {

        width: 100px;

        height: 100px;

        position: relative;

        animation:    sporadic-rotate    5.25s    cubic-
bezier(0.35, 0, 0.25, 1) infinite;

```

```

        #app-splash-screen .spinner-wrapper .spinner .inner
.gap {

    position: absolute;

    left: 49px;

    right: 49px;

    top: 0;

    bottom: 0;

    border-top: 10px solid;

    box-sizing: border-box;


    #app-splash-screen .spinner-wrapper .spinner .inner
.left,

    #app-splash-screen .spinner-wrapper .spinner .inner
.right {

    position: absolute;

    top: 0;

    height: 100px;

    width: 50px;

    overflow: hidden;


    #app-splash-screen .spinner-wrapper .spinner .inner
.left .half-circle,

    #app-splash-screen .spinner-wrapper .spinner .inner
.right .half-circle {

    position: absolute;

    top: 0;

```

```

        width: 100px;

        height: 100px;

        box-sizing: border-box;

        border: 10px solid #4285F4;

        border-bottom-color: transparent;

        border-radius: 50%;

        #app-splash-screen .spinner-wrapper .spinner .inner
.left {

        left: 0;

        #app-splash-screen .spinner-wrapper .spinner .inner
.left .half-circle {

        left: 0;

        border-right-color: transparent;

        animation:      left-wobble      1.3125s      cubic-
bezier(0.35, 0, 0.25, 1) infinite;

        -webkit-animation: left-wobble 1.3125s cubic-
bezier(0.35, 0, 0.25, 1) infinite;

        #app-splash-screen .spinner-wrapper .spinner .inner
.right {

        right: 0;

        #app-splash-screen .spinner-wrapper .spinner .inner
.right .half-circle {

        right: 0;

```

```

        border-left-color: transparent;

        animation:      right-wobble      1.3125s      cubic-
bezier(0.35, 0, 0.25, 1) infinite;

        -webkit-animation: right-wobble 1.3125s cubic-
bezier(0.35, 0, 0.25, 1) infinite;

@keyframes outer-rotate {
    0% {
        transform: rotate(0deg) scale(0.5);
    }
    100% {
        transform: rotate(360deg) scale(0.5);
    }
}

@keyframes left-wobble {
    0%, 100% {
        transform: rotate(130deg);
    }
    50% {
        transform: rotate(-5deg);
    }
}

@keyframes right-wobble {
    0%, 100% {
        transform: rotate(-130deg);
    }
}

```

```
    50% {  
        transform: rotate(5deg);  
    }  
}  
  
@keyframes sporadic-rotate {  
    12.5% {  
        transform: rotate(135deg);  
    }  
    25% {  
        transform: rotate(270deg);  
    }  
    37.5% {  
        transform: rotate(405deg);  
    }  
    50% {  
        transform: rotate(540deg);  
    }  
    62.5% {  
        transform: rotate(675deg);  
    }  
    75% {  
        transform: rotate(810deg);  
    }  
    87.5% {  
        transform: rotate(945deg);  
    }  
}
```

```

    }

    100% {

        transform: rotate(1080deg);

    }

}

</style>

<!-- / APP Splash Screen CSS -->

<style>@charset "UTF-8";html{line-height:1.15;-ms-text-size-
adjust:100%;-webkit-text-size-
adjust:100%}body{margin:0}img{border-style:none}body:not(.is-
mobile)::-webkit-scrollbar{width:12px;height:12px;background-
color:#0000}body:not(.is-mobile)::-webkit-
scrollbar:hover{background-color:#0000001f}body:not(.is-mobile)
::-webkit-scrollbar-thumb{border:2px solid transparent;box-
shadow:inset 0 0 0 12px #00000005e;border-radius:12px}body:not(.is-
mobile)::-webkit-scrollbar-thumb:active{box-shadow:inset 0 0 0
12px #00000008a;border-radius:12px}html,body{display:flex;flex:1 0
auto;width:100%;height:100%;max-height:100%;min-
height:100%;margin:0;padding:0;overflow:hidden}*{text-
rendering:optimizeLegibility;-o-text-
rendering:optimizeLegibility;-ms-text-
rendering:optimizeLegibility;-moz-text-
rendering:optimizeLegibility;-webkit-text-
rendering:optimizeLegibility;-webkit-tap-highlight-
color:transparent;box-sizing:border-box}*:before,*:after{box-
sizing:border-box}*:focus{outline:none}img{max-
width:100%;height:auto;vertical-align:top;border:none}html{font-
size:62.5%;font-family:Muli,Helvetica Neue,Arial,sans-
serif;letter-spacing:-.1px!important}body{font-
size:14px}html,body{-webkit-font-smoothing:auto;-moz-osx-font-
smoothing:auto}@media
print{html,body{height:auto!important;overflow:visible!important
;overflow:initial!important}}@font-face{font-family:Muli;font-

```

```

style:normal;font-weight:300;font-
display:swap;src:url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-
300.eot);src:local("Muli                                Light"),local("Muli-
Light"),url(/assets/fonts/google-font-muli/muli-v12-latin_latin-
ext/muli-v12-latin_latin-ext-300.eot?#iefix)        format("embedded-
opentype"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.woff2)
format("woff2"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.woff)
format("woff"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.ttf)
format("truetype"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.svg#Muli)
format("svg")}@font-face{font-family:Muli;font-
style:normal;font-weight:400;font-
display:swap;src:url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-
regular.eot);src:local("Muli                                Regular"),local("Muli-
Regular"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-regular.eot?#iefix)
format("embedded-opentype"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.woff2)        format("woff2"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.woff)        format("woff"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.ttf)        format("truetype"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.svg#Muli) format("svg")}@font-face{font-family:Muli;font-
style:normal;font-weight:600;font-
display:swap;src:url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-
600.eot);src:local("Muli                                SemiBold"),local("Muli-
SemiBold"),url(/assets/fonts/google-font-muli/muli-v12-

```

```

latin_latin-ext/muli-v12-latin_latin-ext-600.eot?#iefix)
format("embedded-opentype"),url(/assets/fonts/google-font-
mul/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
600.woff2)          format("woff2"),url(/assets/fonts/google-font-
mul/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-600.woff)
format("woff"),url(/assets/fonts/google-font-mul/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-600.ttf)
format("truetype"),url(/assets/fonts/google-font-mul/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-600.svg#Muli)
format("svg") }@font-face{font-family:Muli;font-
style:normal;font-weight:700;font-
display:swap;src:url(/assets/fonts/google-font-mul/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-
700.eot);src:local("Muli          Bold"),local("Muli-
Bold"),url(/assets/fonts/google-font-mul/muli-v12-latin_latin-
ext/muli-v12-latin_latin-ext-700.eot?#iefix)    format("embedded-
opentype"),url(/assets/fonts/google-font-mul/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.woff2)
format("woff2"),url(/assets/fonts/google-font-mul/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.woff)
format("woff"),url(/assets/fonts/google-font-mul/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.ttf)
format("truetype"),url(/assets/fonts/google-font-mul/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.svg#Muli)
format("svg") }</style><link          rel="stylesheet"
href="styles.b44bc4897f6f17a4.css"  crossorigin="use-credentials"
media="print"          onload="this.media='all'"><noscript><link
rel="stylesheet"
href="styles.b44bc4897f6f17a4.css"></noscript></head>

<body

<!-- APP Splash Screen -->

<app-splash-screen id="app-splash-screen"

    <div class="center"

```

```

        <div class="logo">

        </div>

        <!-- Material Design Spinner -->

        <div class="spinner-wrapper">

            <div class="spinner">

                <div class="inner">

                    <div class="gap"></div>

                    <div class="left">

                        <div class="half-circle"></div>

                    </div>

                    <div class="right">

                        <div class="half-circle"></div>

                    </div>

                </div>

            </div>

        </div>

        <!-- / Material Design Spinner --

    </div>

</app-splash-screen>

<!-- / APP Splash Screen -->

<fnt-app></fnt-app>

    <script    src="runtime.7a47ea7d14a600d9.js"    type="module"
crossorigin="use-credentials"></script><script
src="polyfills.cc7779c60254bdd5.js"                type="module"
crossorigin="use-credentials"></script><script

```

```
src="main.4bf4e631861aa829.js" type="module" crossorigin="use-  
credentials"></script  
  
</body></html>
```

Додаток Б: Код сторінки договору

```

<!DOCTYPE html><html lang="uk"><head>

<title>Billing::ISP 03</title>

<base href="/spa/">

<meta charset="utf-8">

<meta name="description" content="ISP billing">

<meta name="keywords" content="Activation, billing">

<meta name="author" content="Rybachuk Dmytro">

<meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=5.0">

<meta name="msapplication-TileColor" content="#2b5797">

<meta name="msapplication-config" content="browserconfig.xml">

<meta name="theme-color" content="#1976d2">

<link rel="apple-touch-icon" sizes="114x114"
href="assets/icons/apple-touch-icon.png">

<link rel="icon" type="image/png" sizes="32x32"
href="assets/icons/favicon-32x32.png">

<link rel="icon" type="image/png" sizes="16x16"
href="assets/icons/favicon-16x16.png">

<link rel="mask-icon" href="assets/icons/safari-pinned-tab.svg"
color="#5bbad5">

<link rel="manifest" href="manifest.json">

<noscript>

<p>Робота з білінгом потребує JavaScript. Скорегуйте
налаштування Вашого браузера для роботи з JavaScript.</p>

</noscript>

<!-- APP Splash Screen CSS -->

<style type="text/css">

#app-splash-screen {

```

```
display: block;

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background-color: #2D323D;

z-index: 99999;

pointer-events: none;

}

#app-splash-screen .center {

display: block;

width: 100%;

position: absolute;

top: 50%;

transform: translateY(-50%);

}

#app-splash-screen .logo {

width: 128px;

margin: 0 auto;

}

#app-splash-screen .logo img {

filter: drop-shadow(0px 10px 6px rgba(0, 0, 0, 0.2))

}

#app-splash-screen .spinner-wrapper {

display: block;

position: relative;
```

```

width: 100%;
min-height: 100px;
height: 100px;
}

#app-splash-screen .spinner-wrapper .spinner {
position: absolute;
overflow: hidden;
left: 50%;
margin-left: -50px;
animation: outer-rotate 2.91667s linear infinite;
}

#app-splash-screen .spinner-wrapper .spinner .inner {
width: 100px;
height: 100px;
position: relative;
animation: sporadic-rotate 5.25s cubic-bezier(0.35, 0, 0.25, 1)
infinite;
}

#app-splash-screen .spinner-wrapper .spinner .inner .gap {
position: absolute;
left: 49px;
right: 49px;
top: 0;
bottom: 0;
border-top: 10px solid;
box-sizing: border-box;
}

#app-splash-screen .spinner-wrapper .spinner .inner .left,

```

```

#app-splash-screen .spinner-wrapper .spinner .inner .right {
position: absolute;
top: 0;
height: 100px;
width: 50px;
overflow: hidden;
}

#app-splash-screen .spinner-wrapper .spinner .inner .left .half-
circle,
#app-splash-screen .spinner-wrapper .spinner .inner .right
.half-circle {
position: absolute;
top: 0;
width: 100px;
height: 100px;
box-sizing: border-box;
border: 10px solid #4285F4;
border-bottom-color: transparent;
border-radius: 50%;
}

#app-splash-screen .spinner-wrapper .spinner .inner .left {
left: 0;
}

#app-splash-screen .spinner-wrapper .spinner .inner .left .half-
circle {
left: 0;
border-right-color: transparent;
animation: left-wobble 1.3125s cubic-bezier(0.35, 0, 0.25, 1)
infinite;

```

```

-webkit-animation: left-wobble 1.3125s cubic-bezier(0.35, 0,
0.25, 1) infinite;

}

#app-splash-screen .spinner-wrapper .spinner .inner .right {
right: 0;

}

#app-splash-screen .spinner-wrapper .spinner .inner .right
.half-circle {

right: 0;

border-left-color: transparent;

animation: right-wobble 1.3125s cubic-bezier(0.35, 0, 0.25, 1)
infinite;

-webkit-animation: right-wobble 1.3125s cubic-bezier(0.35, 0,
0.25, 1) infinite;

}

@keyframes outer-rotate {

0% {

transform: rotate(0deg) scale(0.5);

}

100% {

transform: rotate(360deg) scale(0.5);

}

}

@keyframes left-wobble {

0%, 100% {

transform: rotate(130deg);

}

50% {

transform: rotate(-5deg);

```

```
}  
  
}  
  
@keyframes right-wobble {  
  0%, 100% {  
    transform: rotate(-130deg);  
  }  
  
  50% {  
    transform: rotate(5deg);  
  }  
}  
  
@keyframes sporadic-rotate {  
  12.5% {  
    transform: rotate(135deg);  
  }  
  
  25% {  
    transform: rotate(270deg);  
  }  
  
  37.5% {  
    transform: rotate(405deg);  
  }  
  
  50% {  
    transform: rotate(540deg);  
  }  
  
  62.5% {  
    transform: rotate(675deg);  
  }  
  
  75% {
```

```

transform: rotate(810deg);

}

87.5% {

transform: rotate(945deg);

}

100% {

transform: rotate(1080deg);

}

}

</style>

<!-- / APP Splash Screen CSS -->

<style>@charset "UTF-8";html{line-height:1.15;-ms-text-size-
adjust:100%;-webkit-text-size-
adjust:100%}body{margin:0}img{border-style:none}body:not(.is-
mobile)::-webkit-scrollbar{width:12px;height:12px;background-
color:#0000}body:not(.is-mobile)::-webkit-
scrollbar:hover{background-color:#0000001f}body:not(.is-mobile)
::-webkit-scrollbar-thumb{border:2px solid transparent;box-
shadow:inset 0 0 0 12px #00000005e;border-
radius:12px}body:not(.is-mobile)::-webkit-scrollbar-
thumb:active{box-shadow:inset 0 0 0 12px #00000008a;border-
radius:12px}html,body{display:flex;flex:1 0
auto;width:100%;height:100%;max-height:100%;min-
height:100%;margin:0;padding:0;overflow:hidden}{text-
rendering:optimizeLegibility;-o-text-
rendering:optimizeLegibility;-ms-text-
rendering:optimizeLegibility;-moz-text-
rendering:optimizeLegibility;-webkit-text-
rendering:optimizeLegibility;-webkit-tap-highlight-
color:transparent;box-sizing:border-box}*:before,*:after{box-
sizing:border-box}*:focus{outline:none}img{max-
width:100%;height:auto;vertical-align:top;border:none}html{font-
size:62.5%;font-family:Muli,Helvetica Neue,Arial,sans-
serif;letter-spacing:-.1px!important}body{font-
size:14px}html,body{-webkit-font-smoothing:auto;-moz-osx-font-
smoothing:auto}@media
print{html,body{height:auto!important;overflow:visible!important
;overflow:initial!important}}@font-face{font-family:Muli;font-
style:normal;font-weight:300;font-
display:swap;src:url(/assets/fonts/google-font-muli/muli-v12-

```

```

latin_latin-ext/muli-v12-latin_latin-ext-
300.eot);src:local("Muli Light"),local("Muli-
Light"),url(/assets/fonts/google-font-muli/muli-v12-latin_latin-
ext/muli-v12-latin_latin-ext-300.eot?#iefix) format("embedded-
opentype"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.woff2)
format("woff2"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.woff)
format("woff"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.ttf)
format("truetype"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-300.svg#Muli)
format("svg")}@font-face{font-family:Muli;font-
style:normal;font-weight:400;font-
display:swap;src:url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-
regular.eot);src:local("Muli Regular"),local("Muli-
Regular"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-regular.eot?#iefix)
format("embedded-opentype"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.woff2) format("woff2"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.woff) format("woff"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.ttf) format("truetype"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
regular.svg#Muli) format("svg")}@font-face{font-
family:Muli;font-style:normal;font-weight:600;font-
display:swap;src:url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-
600.eot);src:local("Muli SemiBold"),local("Muli-
SemiBold"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-600.eot?#iefix)
format("embedded-opentype"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-
600.woff2) format("woff2"),url(/assets/fonts/google-font-
muli/muli-v12-latin_latin-ext/muli-v12-latin_latin-ext-600.woff)
format("woff"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-600.ttf)
format("truetype"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-600.svg#Muli)
format("svg")}@font-face{font-family:Muli;font-
style:normal;font-weight:700;font-
display:swap;src:url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-
700.eot);src:local("Muli Bold"),local("Muli-
Bold"),url(/assets/fonts/google-font-muli/muli-v12-latin_latin-
ext/muli-v12-latin_latin-ext-700.eot?#iefix) format("embedded-

```

```

opentype"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.woff2)
format("woff2"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.woff)
format("woff"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.ttf)
format("truetype"),url(/assets/fonts/google-font-muli/muli-v12-
latin_latin-ext/muli-v12-latin_latin-ext-700.svg#Muli)
format("svg")}</style><link rel="stylesheet"
href="styles.b44bc4897f6f17a4.css" crossorigin="use-credentials"
media="print" onload="this.media='all'"><noscript><link
rel="stylesheet"
href="styles.b44bc4897f6f17a4.css"></noscript></head>

<body>

<!-- APP Splash Screen -->

<app-splash-screen id="app-splash-screen">

<div class="center">

<div class="logo">



</div>

<!-- Material Design Spinner -->

<div class="spinner-wrapper">

<div class="spinner">

<div class="inner">

<div class="gap"></div>

<div class="left">

<div class="half-circle"></div>

</div>

<div class="right">

<div class="half-circle"></div>

</div>

</div>

```

```

</div>

</div>

<!-- / Material Design Spinner -->

</div>

</app-splash-screen>

<!-- / APP Splash Screen -->

<fnt-app></fnt-app>

<script src="runtime.7a47ea7d14a600d9.js" type="module"
crossorigin="use-credentials"></script><script
src="polyfills.cc7779c60254bdd5.js" type="module"
crossorigin="use-credentials"></script><script
src="main.4bf4e631861aa829.js" type="module" crossorigin="use-
credentials"></script>

</body></html>

```

Джава

```

{
  "name": "FNT",
  "short_name": "FNT",
  "theme_color": "#1976d2",
  "background_color": "#fafafa",
  "display": "standalone",
  "scope": "/spa/",
  "start_url": "/spa/",
  "icons": [
    {
      "src": "assets/icons/favicon-16x16.png",

```

```

    "sizes": "16x16",
    "type": "image/png"
  },
  {
    "src": "assets/icons/favicon-32x32.png",
    "sizes": "32x32",
    "type": "image/png"
  },
  {
    "src": "assets/icons/android-chrome-96x96.png",
    "sizes": "96x96",
    "type": "image/png"
  },
  {
    "src": "assets/icons/icon-16.png",
    "sizes": "16x16",
    "type": "image/png"
  },
  {
    "src": "assets/icons/icon-32.png",
    "sizes": "32x32",
    "type": "image/png"
  },
  {
    "src": "assets/icons/icon-72.png",
    "sizes": "72x72",
    "type": "image/png"
  },
  {
    "src": "assets/icons/icon-88.png",
    "sizes": "88x88",
    "type": "image/png"
  }

```

```

    },
    {
      "src": "assets/icons/icon-128.png",
      "sizes": "128x128",
      "type": "image/png"
    },
    {
      "src": "assets/icons/icon-144.png",
      "sizes": "144x144",
      "type": "image/png"
    },
    {
      "src": "assets/icons/icon-152.png",
      "sizes": "152x152",
      "type": "image/png"
    },
    {
      "src": "assets/icons/icon-196.png",
      "sizes": "196x196",
      "type": "image/png"
    },
    {
      "src": "assets/icons/icon-256.png",
      "sizes": "256x256",
      "type": "image/png"
    },
    {
      "src": "assets/icons/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]

```

}

Додаток В: Діаграми прецедентів

