

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» інформаційної системи інтернет-провайдеру
(тема)

Виконав:

здобувач 4 року навчання,
групи ІТУЗ-21-1

Валерія ЮРЧЕНКО

(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології управління
(повна назва освітньої програми)

Керівник доц. каф. ІУС Олена ДОЛЯ
(прізвище, власне ім'я, по батькові)

Допускається до захисту

Зав. кафедри ІУС



(підпис)

Костянтин ПЕТРОВ

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Навчально-науковий центр заочної форми навчання


Кафедра Інформаційних управляючих систем

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології управління
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ: 
Зав. кафедри _____
(підпис)
«19» травня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Юрченко Валерії Євгенівні
(прізвище, власне ім'я, по батькові)

1. Тема роботи Розробка модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» інформаційної системи інтернет-провайдеру
затверджена наказом університету від «19» травня 2025 р. № 82Стз
2. Термін подання студентом роботи до екзаменаційної комісії «16» червня 2025 р.
3. Вихідні дані до роботи Опис об'єкта автоматизації, схема організаційної структури, альбом документів, що затверджені.
4. Перелік питань, що потрібно опрацювати у роботі Дослідження та оцінка поточного стану проблеми, а також аналіз існуючих методів і підходів до її вирішення в межах кваліфікаційної роботи, детальний опис та аналіз структурних і функціональних характеристик об'єкта дослідження, а також ключових компонентів його забезпечення, формулювання основних завдань кваліфікаційної роботи, обґрунтування мети розробки та критеріїв оцінки ефективності запропонованого рішення, Проектування та обґрунтування інформаційного забезпечення системи, вибір, розробка та обґрунтування технічного забезпечення, розробка програмного забезпечення.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вибрання теми кваліфікаційної роботи	19.05.2025	Виконано
2	Детальний аналіз структурних та функціональних характеристик об'єкта дослідження та основних компонентів його забезпечення	20.05.2025	Виконано
3	Огляд та аналіз поточного стану розглянутої проблеми, а також наявних методів і засобів її вирішення	22.05.2025	Виконано
4	Постановка завдання на розробку	23.05.2025	Виконано
5	Обґрунтування мети розробки модуля та визначення критеріїв його ефективності	26.05.2025	Виконано
6	Проектування та обґрунтування інформаційного забезпечення модуля	27.05.2025	Виконано
7	Вибір, розробка та обґрунтування технічного забезпечення модуля	28.05.2025	Виконано
8	Розробка та впровадження програмного забезпечення модуля	29.05.2025	Виконано
9	Тестування та оцінка надійності функціонування програмних і технічних компонентів модуля	30.05.2025	Виконано
10	Оформлення пояснювальної записки до кваліфікаційної роботи	01.06.2025	Виконано
11	Проходження перевірки на оригінальність кваліфікаційної роботи	06.06.2025	Виконано
12	Отримання допуску до захисту від завідувача кафедри	10.06.2025	Виконано
13	Захищення кваліфікаційної роботи	23.06.2025	Виконано

Дата видачі завдання 19 травня 2025 р.

Здобувач

(підпис)

Керівник роботи

(підпис)

доц. каф. ІУС Олена ДОЛЯ

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 75 с., 13 табл., 13 рис., 1 дод., 11 джерел.

БАЗА ДАНИХ, ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРНЕТ-ПРОВАЙДЕР.

Об'єктом дослідження кваліфікаційної роботи є система автоматизації процесів нагадувань про оплату рахунків клієнтами в інформаційній системі Інтернет-провайдера.

Предмет дослідження - методи та технології автоматизації процесів нагадувань, способи їх інтеграції в білінгові системи та підходи до підвищення ефективності взаємодії з клієнтами.

Мета роботи - розробка модуля автоматизації процесів системи нагадувань про оплату рахунків, що дозволить підвищити рівень своєчасних платежів, мінімізувати прострочення та покращити комунікацію між провайдером і клієнтами.

Методи досліджень базуються на аналізі існуючих систем автоматизації нагадувань, вивченні архітектури білінгових платформ, моделюванні бізнес-процесів, а також застосуванні сучасних підходів до розробки програмного забезпечення.

Практичною частиною роботи є розробка та впровадження модуля автоматизації нагадувань, що включає створення алгоритмів обробки платіжної інформації, налаштування механізмів сповіщення клієнтів через канал зв'язку та забезпечення безпечної обробки даних.

ABSTRACT

The explanatory note to the qualification paper contains: 75 pages, 13 tables, 11 reference sources, 1 appendix.

DATABASE, INFORMATION SYSTEM, INTERNET PROVIDER.

The object of the study is the system for automating the process of payment reminders for customers within the information system of an Internet service provider.

The subject of the study is the methods and technologies for automating reminder processes, approaches to their integration into billing systems, and strategies for improving the efficiency of customer interaction.

The goal of the study is to develop a module for automating the payment reminder system, which will increase the rate of on-time payments, minimize overdue payments, and enhance communication between the provider and customers.

The research methods are based on analyzing existing automated reminder systems, studying the architecture of billing platforms, modeling business processes, and applying modern software development approaches.

The practical part of the work involves the development and implementation of an automated reminder module, including the creation of payment information processing algorithms, configuration of customer notification mechanisms via communication channels, and ensuring secure data processing.

ЗМІСТ

Скорочення та умовні позначки	8
Вступ.....	9
1 Опис та аналіз структурних і функціональних особливостей нагадування про оплату для провайдера інтернет-послуг та основних забезпечуючих систем	11
1.1 Аналіз та характеристика об'єкту дослідження.....	11
1.2 Організаційна структура системи обліку та управління платежами Інтернет-провайдера	12
2 Огляд і аналіз існуючих методів і засобів автоматизації нагадування про оплату для провайдера інтернет-послуг	16
2.1 Аналіз потреби в автоматизації нагадувань про оплату рахунків.....	16
2.2 Огляд існуючих систем автоматизації нагадувань	17
2.3 Порівняльний аналіз систем автоматизації	18
3 Опис вимог до модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам». формулювання завдання розробки	20
3.1 Опис функціональних вимог до об'єкта розробки	20
3.1.1 Обґрунтування мети і критеріїв ефективності модуля нагадування.....	22
3.2 Опис нефункціональних вимог до об'єкта розробки.....	23
3.3 Мета створення системи та оцінка її ефективності	25
4 Опис архітектури модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам»	27
5 Розробка елементів інформаційного забезпечення модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам»	31
5.1 Обґрунтування вибору моделі даних та системи управління базами даних	31
5.2 Опис сутностей та зв'язків модуля	31

5.3	Опис атрибутів сутностей.....	34
5.4	Розробка логічної та фізичної схеми бази даних модуля.....	39
6	Розробка і обґрунтування вибору елементів програмного забезпечуючого модуля.....	45
7	Тестування та оцінка надійності функціонування програмних і технічних рішень модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам»	48
8	Опис впровадження та експлуатації модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам»	60
	Висновки	63
	Перелік джерел посилання	64
	Додаток А Графічний матеріал кваліфікаційної роботи.....	66

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних;

ІС – інформаційна система;

ОС – операційна система;

ПЗ – програмне забезпечення;

API – Application Programming Interface (інтерфейс прикладного програмування)

JSON – JavaScript Object Notation (текстовий формат обміну даними)

SOA – Service-Oriented Architecture (сервісно-орієнтована архітектура)

UI – User Interface (користувацький інтерфейс)

UX – User Experience (користувацький досвід)

ВСТУП

У сучасному світі інформаційні технології розвиваються дуже швидко, і багато компаній прагнуть автоматизувати свої бізнес-процеси, щоб працювати ефективніше. Одним із важливих аспектів управління фінансами є контроль оплат клієнтами. Несвоєчасні платежі можуть викликати фінансові втрати, проблеми з грошовими потоками та додаткові витрати для компаній. Саме тому важливо створювати автоматизовані системи, які допоможуть нагадувати клієнтам про необхідність оплати рахунків та зменшувати ризик прострочених платежів.

Тема цієї дипломної роботи актуальна, оскільки багато компаній, зокрема Інтернет-провайдери, стикаються з труднощами в управлінні платежами. Якщо цей процес виконувати вручну, це займає багато часу і потребує значних ресурсів. Автоматизація дозволяє зменшити вплив людського фактору, підвищити точність та зробити процес нагадувань ефективнішим. Завдяки сучасним технологіям, таким як автоматизовані повідомлення і інтегровані платіжні сервіси, можна значно покращити комунікацію між компанією та клієнтом.

У цій роботі досліджується процес автоматизації нагадувань про оплату рахунків. Основний акцент зроблено на методах і технологіях, які використовуються для створення таких систем. Головною метою є розробка та впровадження автоматизованого модуля, який допоможе зменшити кількість прострочених платежів, покращити взаємодію з клієнтами та оптимізувати фінансові процеси Інтернет-провайдера.

Для досягнення цієї мети потрібно дослідити існуючі рішення, визначити ключові вимоги до системи, розробити її архітектуру та протестувати запропоновані технічні рішення. У практичній частині роботи буде реалізовано функціональний модуль нагадувань, який можна

використовувати для автоматизації платежів у компаніях. Очікується, що така система допоможе зменшити навантаження на персонал, підвищити рівень своєчасних оплат та покращити фінансові показники підприємства.

Отже, у кваліфікаційній роботі розглядається процес проєктування та реалізації модуля автоматизованих нагадувань, який дозволить знизити фінансові ризики компаній та зробити управління платежами більш ефективним.

Кваліфікаційна робота виконана відповідно до методичних вказівок з організації виконання та захисту кваліфікаційної роботи [1].

1 ОПИС ТА АНАЛІЗ СТРУКТУРНИХ І ФУНКЦІОНАЛЬНИХ ОСОБЛИВОСТЕЙ НАГАДУВАННЯ ПРО ОПЛАТУ ДЛЯ ПРОВАЙДЕРА ІНТЕРНЕТ-ПОСЛУГ ТА ОСНОВНИХ ЗАБЕЗПЕЧУЮЧИХ СИСТЕМ

1.1 Аналіз та характеристика об'єкта дослідження

Об'єктом дослідження є система автоматизації процесів нагадувань про оплату рахунків клієнтами інформаційної системи Інтернет-провайдера. Вона є важливою складовою сучасних білінгових платформ, оскільки забезпечує оперативне інформування клієнтів про наявність заборгованості та терміни здійснення платежів. Автоматизація цього процесу дозволяє суттєво знизити кількість прострочених платежів, підвищити рівень дисципліни оплат серед клієнтів та зменшити навантаження на персонал компанії.

Система автоматизованих нагадувань є частиною загальної білінгової інфраструктури Інтернет-провайдера, яка включає такі ключові компоненти:

- білінгова система – відповідає за облік фінансових операцій, формування рахунків та контроль за платежами;
- база даних користувачів – містить інформацію про клієнтів, їхні контактні дані, історію оплат та поточні заборгованості;
- механізм генерації та відправлення повідомлень – забезпечує автоматичне формування та доставку нагадувань через канал зв'язку;
- модуль інтеграції – дозволяє системі взаємодіяти з іншими інформаційними сервісами провайдера та зовнішніми платіжними платформами;
- аналітичний модуль – відповідає за збір та аналіз даних щодо ефективності нагадувань, що дозволяє оптимізувати процес взаємодії з клієнтами [6].

Основною функцією системи нагадувань є забезпечення своєчасного інформування клієнтів про необхідність оплати рахунків та надання зручних

способів для швидкого проведення платежів. Ефективність такої системи безпосередньо впливає на фінансові показники компанії, оскільки дозволяє зменшити кількість несплачених рахунків та покращити взаємодію з користувачами. Зокрема, правильне налаштування алгоритмів нагадувань дозволяє сегментувати клієнтів за рівнем платіжної дисципліни та застосовувати персоналізовані стратегії сповіщень.

Для Інтернет-провайдерів автоматизація процесу нагадувань є критично важливою, оскільки забезпечує стабільний фінансовий потік компанії та запобігає накопиченню заборгованості. Відсутність ефективної системи нагадувань може призвести до значних фінансових втрат, необхідності витратити додаткові ресурси на роботу з боржниками, а також до зниження рівня довіри клієнтів. Крім того, автоматизація допомагає мінімізувати вплив людського фактору, зменшуючи ризик помилок та підвищуючи точність обробки платіжної інформації.

У рамках дослідження будуть розглянуті сучасні підходи до реалізації таких систем, проведено аналіз існуючих технологічних рішень та розроблено оптимальну модель для впровадження у білінгові процеси Інтернет-провайдера.

Запропонований модуль буде орієнтований на забезпечення стабільної та безперебійної роботи, інтеграцію з наявними сервісами та високий рівень безпеки обробки платіжних даних.

1.2 Організаційна структура системи обліку та управління платежами Інтернет-провайдера

На рисунку 1.1 наведено схему організаційної структури компанії Інтернет-провайдера.

Система обліку та управління платежами Інтернет-провайдера є складним багатокомпонентним механізмом, який забезпечує безперервний контроль фінансових потоків, виставлення рахунків, проведення оплат і обробку фінансових транзакцій. Ефективність роботи провайдера значною мірою залежить від злагодженості функціонування цієї системи та її здатності до інтеграції з іншими бізнес-процесами компанії.

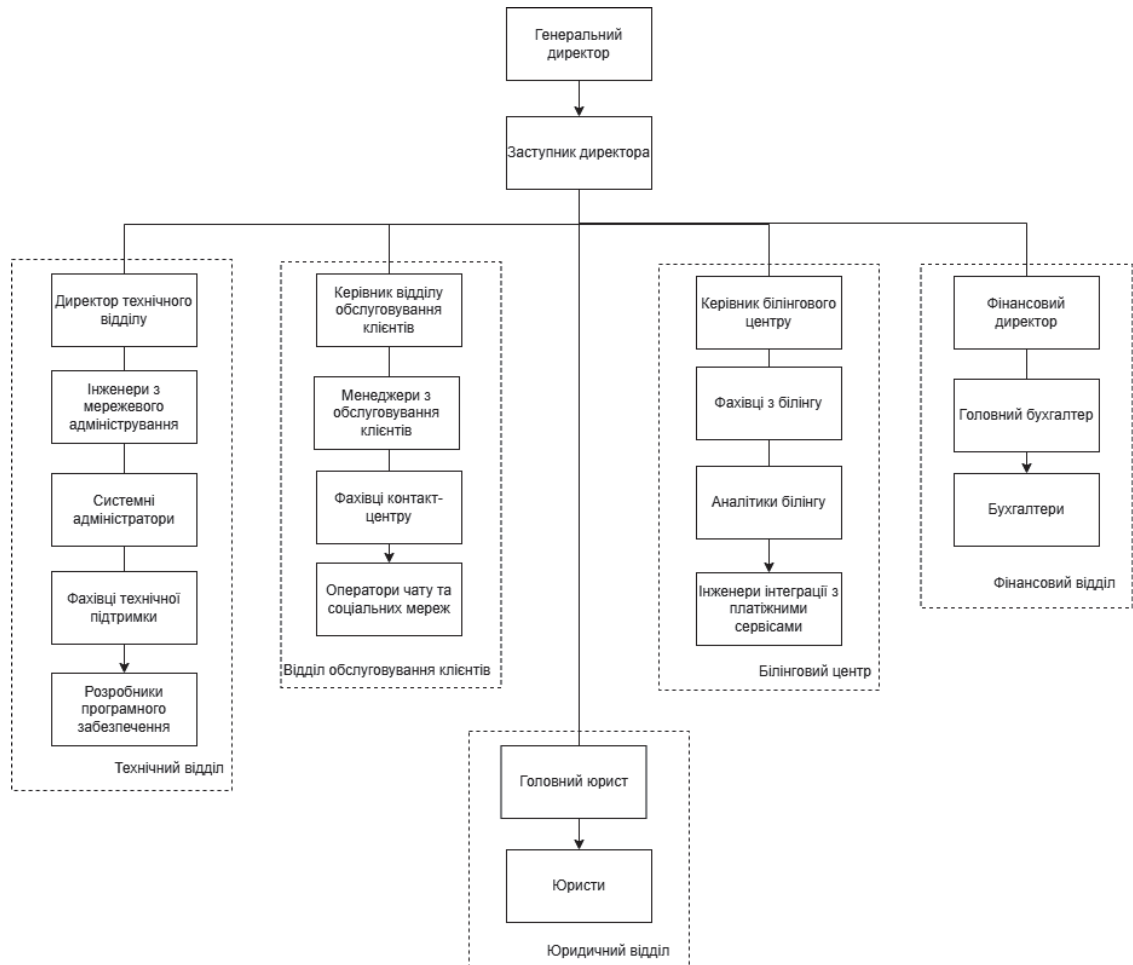


Рисунок 1.1 – Схема організаційної структури компанії Інтернет-провайдера

Організаційна структура Інтернет-провайдера включає кілька ключових відділів, кожен з яких виконує свої функції для забезпечення стабільної роботи компанії.

На чолі організації знаходиться генеральний директор, який приймає стратегічні рішення та контролює діяльність усіх підрозділів. Йому

підпорядковується заступник директора, який координує роботу відділів і відповідає за реалізацію внутрішніх процесів.

Технічний відділ займається підтримкою та розвитком технічної інфраструктури компанії. Його очолює директор технічного відділу, під керівництвом якого працюють інженери з мережевого адміністрування, що відповідають за налаштування та обслуговування мережі, системні адміністратори, які забезпечують роботу серверів та внутрішніх сервісів, а також фахівці технічної підтримки, що допомагають клієнтам усувати технічні проблеми. У відділі працюють розробники програмного забезпечення, які створюють і підтримують внутрішні системи.

Відділ обслуговування клієнтів забезпечує ефективну комунікацію з користувачами, допомагає з підключенням та вирішенням питань, пов'язаних із послугами провайдера. Керівник відділу координує роботу менеджерів з обслуговування клієнтів, фахівців контакт-центру та операторів онлайн-чату, які взаємодіють з клієнтами через телефонні дзвінки, соціальні мережі та месенджери.

Білінговий центр займається виставленням рахунків, обліком платежів та аналізом фінансових потоків. Його очолює керівник білінгового центру, який контролює діяльність фахівців з білінгу, що формують рахунки та стежать за своєчасністю оплат. Аналітики білінгу аналізують тенденції платежів і прогнозують фінансові ризики, а інженери інтеграції з платіжними сервісами відповідають за технічну взаємодію з банками та онлайн-оплатами.

Юридичний відділ забезпечує правову підтримку діяльності компанії. Головний юрист відповідає за відповідність компанії чинному законодавству, контролює ведення договірної документації, взаємодіє з регуляторними органами та займається судовими справами. Юристи цього відділу також слідкують за дотриманням норм захисту персональних даних.

Фінансовий відділ забезпечує контроль за фінансовими потоками та ведення бухгалтерського обліку. Його очолює фінансовий директор, який керує роботою головного бухгалтера та бухгалтерів. Головний бухгалтер

відповідає за облік доходів і витрат компанії, а бухгалтери ведуть податкову звітність, готують фінансові звіти та контролюють витрати.

Організаційна структура провайдера вибудована таким чином, щоб забезпечити ефективну взаємодію між відділами, підтримку стабільної роботи мережі, своєчасне нарахування платежів та якісне обслуговування клієнтів.

2 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ І ЗАСОБІВ АВТОМАТИЗАЦІЇ НАГАДУВАННЯ ПРО ОПЛАТУ ДЛЯ ПРОВАЙДЕРА ІНТЕРНЕТ-ПОСЛУГ

2.1 Аналіз потреби в автоматизації нагадувань про оплату рахунків

У сучасному цифровому середовищі стабільність грошових надходжень для будь-якої сервісної компанії є визначальним чинником її функціональної спроможності. Особливо це стосується провайдерів інтернет-послуг, для яких регулярна оплата абонентських рахунків - це не лише джерело доходу, а й запорука безперебійного забезпечення користувачів доступом до послуг зв'язку, підтримки технічної інфраструктури та модернізації мережі.

Однак у практиці обслуговування клієнтів поширеною є проблема несвоєчасної оплати, що призводить до касових розривів, необхідності постійного моніторингу заборгованостей та втрати частини прибутків.

У більшості випадків затримки пов'язані не з відмовою від оплати, а з банальним забуттям або технічними труднощами на боці користувача. За таких умов своєчасне інформування клієнтів про наявність рахунків, терміни їх оплати та можливі наслідки затримки дозволяє суттєво зменшити кількість проблемних випадків.

Традиційно нагадування про оплату здійснювались у ручному режимі: оператори або менеджери зв'язувались із клієнтами телефоном, надсилали індивідуальні листи або повідомлення. Цей процес є трудомістким, залежить від людського фактора, та не може масштабуватись при збільшенні кількості абонентів. Тому актуальним стає впровадження автоматизованих систем нагадувань, які без участі оператора надсилають клієнтам повідомлення на електронну пошту або через інші канали комунікації у визначений час.

Автоматизація процесу нагадування дає змогу:

- скоротити витрати часу працівників на обробку прострочених рахунків;
- зменшити кількість втраченої дебіторської заборгованості;
- підвищити рівень фінансової дисципліни абонентів;
- формалізувати процес взаємодії з клієнтами та забезпечити однакові умови обслуговування для всіх.

Додатковою перевагою є можливість інтеграції модулів автоматизації із CRM- або білінговими системами, що дозволяє у режимі реального часу відслідковувати статус оплати, історію повідомлень та швидко реагувати на відхилення від встановлених процедур [7].

Крім того, системи автоматичного нагадування легко адаптуються до змін у політиці компанії - наприклад, змін термінів оплати, введення нових шаблонів комунікації або каналів доставки повідомлень.

Таким чином, необхідність у впровадженні автоматизованого модуля нагадувань зумовлена не лише бажанням підвищити ефективність обслуговування, а й стратегічною потребою підприємства підтримувати стабільний фінансовий потік, оптимізувати операційну діяльність та зміцнити клієнтську лояльність.

2.2 Огляд існуючих систем автоматизації нагадувань

Сучасний ринок автоматизованих систем нагадувань про оплату рахунків представлений великою кількістю рішень, що відрізняються за функціоналом, технологічною основою та рівнем інтеграції з іншими фінансовими та білінговими платформами. У зв'язку з розвитком інформаційних технологій і зростаючою потребою компаній у підвищенні

ефективності управління платежами, автоматизовані системи нагадувань стали важливим елементом білінгових процесів у багатьох сферах діяльності, включаючи телекомунікації, комунальні послуги та фінансовий сектор.

Автоматизовані системи нагадувань можна розділити на три основні групи: вбудовані білінгові модулі, сторонні CRM-системи та спеціалізовані сервіси для автоматизації платежів [8]. Кожен з цих підходів має свої переваги та обмеження, які варто враховувати при виборі оптимального рішення для конкретного бізнесу.

Вбудовані білінгові модулі є частиною фінансових систем компаній і інтегруються безпосередньо в існуючий білінговий процес. Вони забезпечують автоматичне надсилання нагадувань на основі даних про оплату клієнтів, що дозволяє мінімізувати затримки у виставленні рахунків та покращити контроль за платежами.

CRM-системи, такі як Salesforce або Zoho CRM, включають модулі автоматизованого нагадування про платежі та надають розширені можливості управління клієнтськими взаємовідносинами [2].

Вони дозволяють налаштовувати різні сценарії комунікації з клієнтами, персоналізувати повідомлення та використовувати аналітику для прогнозування й оптимізації білінгових процесів.

2.3 Порівняльний аналіз систем автоматизації

Для визначення найкращого підходу до автоматизації нагадувань у білінговій системі Інтернет-провайдера проведено порівняльний аналіз найбільш поширених рішень. Нижче представлена порівняльна таблиця 2.1 характеристик різних систем.

Таблиця 2.1 – Порівняльна характеристика різних типів систем автоматизації нагадувань про оплату рахунків

Характеристика	Вбудовані білінгові модулі	CRM-системи	Спеціалізовані сервіси
Інтеграція з білінгом	Висока	Середня	Висока
Гнучкість налаштувань	Обмежена	Висока	Висока
Можливість персоналізації	Середня	Висока	Висока
Аналітика та звітність	Базова	Розширена	Середня
Канали комунікації	Email, SMS	Email, SMS, push	Email, SMS, push
Вартість впровадження	Відносно низька	Висока	Середня

Як видно з таблиці, кожен підхід має свої переваги. Вбудовані білінгові модулі підходять для компаній, які хочуть мінімізувати витрати та використовувати вже існуючі білінгові системи.

CRM-системи забезпечують широкий функціонал і гнучкі можливості персоналізації, що робить їх особливо привабливими для бізнесів із багаторівневими або складними сценаріями взаємодії з клієнтами. Вони дозволяють не лише зберігати інформацію про користувача, а й аналізувати історію спілкування, автоматизувати нагадування про платежі, сегментувати клієнтську базу за поведінковими та фінансовими показниками, а також будувати персоналізовані ланцюжки комунікацій. У CRM-середовищі менеджери можуть легко змінювати параметри сценаріїв, адаптувати шаблони повідомлень під окремі групи користувачів, а також налаштовувати умови тригерів для надсилання. Це особливо важливо для компаній, які прагнуть покращити клієнтський досвід, підвищити рівень утримання клієнтів і зменшити кількість відмов від послуг.

Аналіз існуючих систем автоматизації нагадувань про оплату рахунків показує, що вибір оптимального рішення залежить від специфіки бізнесу, рівня інтеграції з іншими системами та необхідності персоналізації комунікацій із клієнтами.

Для Інтернет-провайдерів, які працюють із великою кількістю клієнтів і потребують швидкої та безперебійної обробки платежів, найбільш ефективним варіантом є використання вбудованих білінгових модулів із можливістю інтеграції з CRM-системами або спеціалізованими сервісами. Це дозволить забезпечити автоматизацію процесів нагадувань, покращити контроль за платежами та зменшити рівень прострочених оплат.

3 ОПИС ВИМОГ ДО МОДУЛЯ «АВТОМАТИЗАЦІЯ ПРОЦЕСІВ СИСТЕМИ НАГАДУВАНЬ ПРО ОПЛАТУ РАХУНКІВ КЛІЄНТАМ».

ФОРМУЛЮВАННЯ ЗАВДАННЯ РОЗРОБКИ

3.1 Опис функціональних вимог до об'єкта розробки

Метою розробки модуля автоматизації процесів системи нагадувань про оплату рахунків є створення ефективного та інтегрованого рішення для Інтернет-провайдера, яке дозволить мінімізувати кількість прострочених платежів, підвищити рівень фінансової дисципліни клієнтів і зменшити навантаження на службу підтримки. Автоматизація цього процесу забезпечить своєчасне інформування клієнтів про необхідність оплати.

Об'єктом автоматизації є білінгова система Інтернет-провайдера, що відповідає за облік фінансових операцій, виставлення рахунків та контроль оплат. Інтеграція модуля нагадувань у білінгову систему дозволить покращити процес управління платежами, забезпечити безперервний контроль заборгованостей та автоматично генерувати повідомлення відповідно до визначених бізнес-правил. Основними цілями автоматизації є зменшення кількості прострочених оплат, підвищення рівня комунікації з клієнтами та оптимізація фінансових потоків компанії.

Система повинна функціонувати безперебійно відповідно до графіка роботи білінгової платформи, що передбачає обробку платежів у режимі 24/7. Нагадування про оплату повинні формуватися автоматично відповідно до заданих сценаріїв, зокрема: за кілька днів до дедлайну платежу, у день прострочення та через певний період після несплаченої заборгованості. Також передбачена можливість налаштування персоналізованих сценаріїв сповіщень для різних категорій клієнтів (регулярні платники, боржники тощо).

Функціонування модуля може бути порушене у наступних випадках:

- збої в роботі білінгової системи, які призводять до неможливості отримання актуальної інформації про стан рахунків;
- відсутність стабільного інтернет-з'єднання або проблеми з поштовими сервісами;
- некоректне налаштування модуля або його інтеграції з іншими системами;
- технічні помилки у сторонніх платіжних сервісах, які можуть впливати на обробку платежів;
- недостатній рівень підготовки персоналу щодо використання автоматизованої системи.

Система передбачає два основних типи користувачів: фінансовий відділ та клієнти провайдера. Фінансовий відділ має доступ до керування налаштуваннями нагадувань, формування звітів про ефективність сповіщень та адміністрування білінгової платформи. Клієнти отримують автоматизовані нагадування та можуть взаємодіяти із системою через особистий кабінет, де можна налаштовувати сповіщення та отримувати інформацію про стан рахунку.

На рисунку 3.1 наведено Use-case діаграму, що відображає основні функціональні можливості модуля автоматизації нагадувань для кожного типу користувачів.

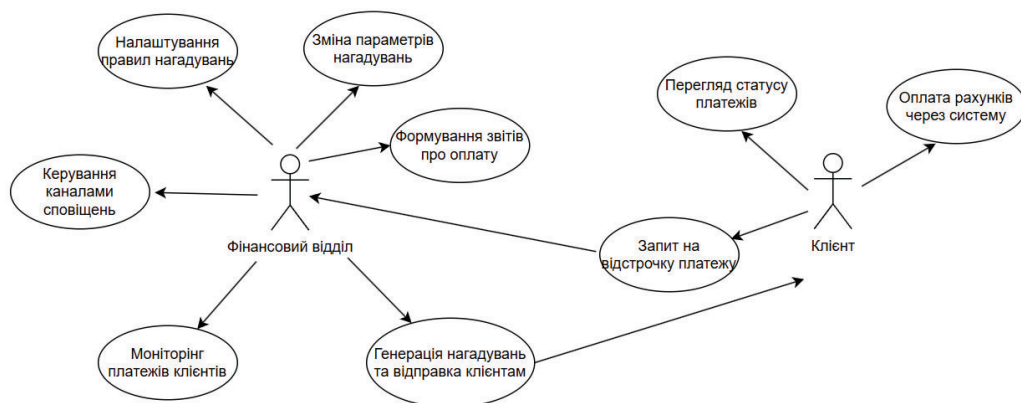


Рисунок 3.1 – Use-case діаграма для системи нагадувань про оплату рахунків

Діаграма відображає взаємодію фінансового відділу та клієнта в процесі автоматизації нагадувань про оплату рахунків. Фінансовий відділ виконує ключові дії, пов'язані з налаштуванням правил нагадувань, формуванням звітів про оплату, керуванням каналами сповіщень, моніторингом платежів клієнтів і зміною параметрів нагадувань. Ці процеси дозволяють системі ефективно контролювати фінансову дисципліну клієнтів і зменшити кількість прострочених платежів.

Одним із найважливіших етапів є передача нагадувань клієнтам, що є перехідною точкою між роботою фінансового відділу та взаємодією з користувачами. Після передачі нагадувань клієнти отримують повідомлення через доступні канали зв'язку, мають можливість переглядати статус своїх платежів, здійснювати оплату рахунків через систему або подавати запити на відстрочку платежу.

Ця структура дозволяє автоматизувати фінансові процеси, скоротити витрати на ручне опрацювання боргів та підвищити рівень задоволеності клієнтів за рахунок своєчасного сповіщення та доступу до необхідної інформації.

3.1.1 Обґрунтування мети і критеріїв ефективності модуля нагадування

Модуль автоматизації нагадувань про оплату рахунків має забезпечувати ефективну взаємодію між білінговою системою Інтернет-провайдера та клієнтами. Основними функціональними вимогами до системи є наступні:

- автоматичне формування та надсилання нагадувань через канали комунікації;

- інтеграція з білінговою системою для отримання актуальної інформації про статус платежів;
- гнучке налаштування сценаріїв нагадувань на основі стану рахунків клієнтів;
- підтримка персоналізованих повідомлень, що включають інформацію про суму до сплати, крайній термін та доступні способи оплати;
- реалізація механізму повторних нагадувань у випадку, якщо оплата не була здійснена після першого сповіщення;
- надання клієнтам можливості змінювати налаштування отримання нагадувань у своєму особистому кабінеті;
- генерація аналітичних звітів для фінансового відділу щодо ефективності нагадувань та динаміки оплат;
- висока надійність роботи системи з мінімальним рівнем затримок при надсиланні повідомлень;
- гарантований захист персональних даних клієнтів, відповідно до вимог чинного законодавства;
- забезпечення масштабованості рішення для підтримки збільшення кількості користувачів без втрати продуктивності.

Запропоновані функціональні вимоги спрямовані на підвищення ефективності управління платежами та забезпечення безперебійної комунікації між Інтернет-провайдером та його клієнтами.

3.2 Опис нефункціональних вимог до об'єкта розробки

Модуль автоматизації нагадувань про оплату рахунків повинен відповідати ряду нефункціональних вимог, що гарантують його ефективність, безпеку, продуктивність та зручність використання.

Система має бути високонадійною, працювати безперебійно в режимі 24/7, забезпечуючи безперервний доступ до нагадувань та оновлення інформації про платежі [3]. Для цього необхідно використовувати сервери з високою відмовостійкістю, а також резервне копіювання даних для запобігання втратам у разі технічних збоїв.

Продуктивність системи має відповідати вимогам швидкої обробки запитів. Вона повинна мати можливість обробляти десятки тисяч нагадувань одночасно без помітного впливу на швидкість білінгових операцій. Час реакції на запит користувача не повинен перевищувати 2 секунд, а відправка нагадувань має здійснюватися в межах 5-10 секунд після генерації запиту.

Захист даних є пріоритетним аспектом. Всі персональні дані клієнтів та фінансові операції повинні бути зашифровані відповідно до сучасних стандартів безпеки (наприклад, AES-256) [4]. Доступ до даних має контролюватися через механізми автентифікації та авторизації, з використанням двофакторної автентифікації для фінансових операцій.

Система повинна бути зручною для використання як для фінансового відділу, так і для клієнтів. Інтерфейс має бути інтуїтивно зрозумілим, з підтримкою адаптивного дизайну для роботи на мобільних пристроях. Для клієнтів повинна бути реалізована можливість швидко змінювати налаштування отримання нагадувань, переглядати історію оплат і отримувати автоматизовані підказки щодо термінів сплати.

Масштабованість системи є критично важливою. Вона повинна легко адаптуватися до зростання кількості користувачів та підтримувати інтеграцію з іншими сервісами Інтернет-провайдера. Для цього необхідно використовувати модульну архітектуру, що дозволяє безболісно додавати нові функціональні можливості.

Управління логами та моніторинг стану системи є обов'язковими вимогами для забезпечення стабільної роботи. Логування всіх подій та дій користувачів допоможе виявляти помилки та аномальні ситуації в системі.

Автоматизовані засоби моніторингу повинні сповіщати адміністраторів про можливі проблеми та гарантувати швидке реагування.

Таким чином, нефункціональні вимоги до системи визначають її продуктивність, надійність, безпеку, масштабованість та зручність у використанні. Їх дотримання дозволить забезпечити ефективну роботу автоматизованої системи нагадувань у довгостроковій перспективі.

3.3 Мета створення системи та оцінка її ефективності

Мета розробки модуля автоматизації нагадувань про оплату рахунків полягає у впровадженні надійного, масштабованого та зручного програмного компонента, який інтегрується в існуючу білінгову систему Інтернет-провайдера. Завдяки цьому модулю можна підвищити рівень платіжної дисципліни клієнтів, зменшити кількість прострочених платежів та автоматизувати рутинні операції, пов'язані з контролем заборгованостей і комунікацією з користувачами.

Проектований модуль має забезпечити оперативний моніторинг стану рахунків та ініціювати надсилання нагадувань у відповідний момент часу. Він повинен функціонувати як гнучке розширення наявної ІТ-інфраструктури, яке не потребує повної перебудови білінгової платформи. Його впровадження дозволить суттєво знизити витрати на обробку боргів, зменшити кількість помилок, пов'язаних з людським фактором, та покращити якість обслуговування клієнтів.

Ефективність модуля оцінюється за низкою якісних і кількісних показників. До якісних відносяться: зручність та зрозумілість інтерфейсу для фінансового персоналу, стабільність у роботі, відповідність логіки нагадувань бізнес-процесам компанії, швидкість адаптації працівників до використання інструменту, а також простота оновлення і супроводу.

Кількісні показники охоплюють: зменшення кількості прострочених оплат щонайменше на 30% протягом перших місяців після запуску, підвищення своєчасної оплати рахунків до 90% від загального обсягу, зменшення кількості звернень до контакт-центру на 20–25%, а також підвищення рівня задоволеності клієнтів.

Таким чином, модуль вважається ефективним, якщо він надійно виконує покладені функції, легко інтегрується в існуючу інфраструктуру провайдера, підвищує фінансову дисципліну клієнтів і сприяє покращенню взаємодії між компанією та її абонентами.

4 ОПИС АРХІТЕКТУРИ МОДУЛЯ «АВТОМАТИЗАЦІЯ ПРОЦЕСІВ СИСТЕМИ НАГАДУВАНЬ ПРО ОПЛАТУ РАХУНКІВ КЛІЄНТАМ»

Модуль автоматизації нагадувань про оплату рахунків клієнтами є окремим функціональним компонентом інформаційної системи Інтернет-провайдера. Його основне завдання полягає в автоматичному відстеженні строків оплати рахунків, генерації повідомлень про наближення або наявність заборгованості та надсиланні відповідних нагадувань клієнтам через обрані канали комунікації.

Архітектура модуля побудована за принципами сервісно-орієнтованої архітектури (SOA) та складається з декількох логічних компонентів, що взаємодіють між собою [9]. В основі модуля - ядро обробки сценаріїв, яке відповідає за запуск та керування логікою надсилання нагадувань відповідно до встановлених правил.

Інформація про клієнтів та їхні рахунки надходить із білінгової системи або окремої БД, яка виступає як джерело правдивих даних. За допомогою спеціального адаптера або АРІ-з'єднання, модуль отримує актуальні відомості про статус рахунків та дату їхнього виставлення. Ці дані обробляються відповідно до налаштованих тригерів, які ініціюють створення повідомлень, якщо клієнт не здійснив оплату у визначений строк або строк наближається.

Генерація повідомлень виконується шаблонізатором, який підставляє актуальні значення (ПШБ, сума, дата оплати, спосіб оплати тощо) у заздалегідь створені шаблони.

Крім основної логіки, модуль включає підсистему логування подій та журналу повідомлень. Вона дозволяє відстежувати, кому, коли та яке нагадування було відправлено, а також які статуси мали ці операції (успішно,

помилка, відкладено тощо). Це дає змогу вести аудит процесів та виявляти проблеми.

Архітектура модуля є розширюваною - тобто допускає додавання нових сценаріїв, каналів сповіщення або інтеграцій без необхідності переробки існуючої логіки. Основу гнучкості становить розподілення на окремі логічні блоки (сервіси), кожен з яких відповідає за свою область - обробка даних, генерація повідомлень, відправка, логування, конфігурація.

Таким чином, модуль автоматизації є самостійним елементом, який легко інтегрується в інфраструктуру Інтернет-провайдера, функціонує автономно, виконує свої задачі за заданим розкладом та дозволяє налаштовувати бізнес-логіку без змін у коді ядра білінгової системи.

Контекстна діаграма на рисунку 4.1 відображає місце модуля автоматизації нагадувань про оплату рахунків у структурі інформаційної системи Інтернет-провайдера, його взаємозв'язки з іншими елементами системи та зовнішніми об'єктами.

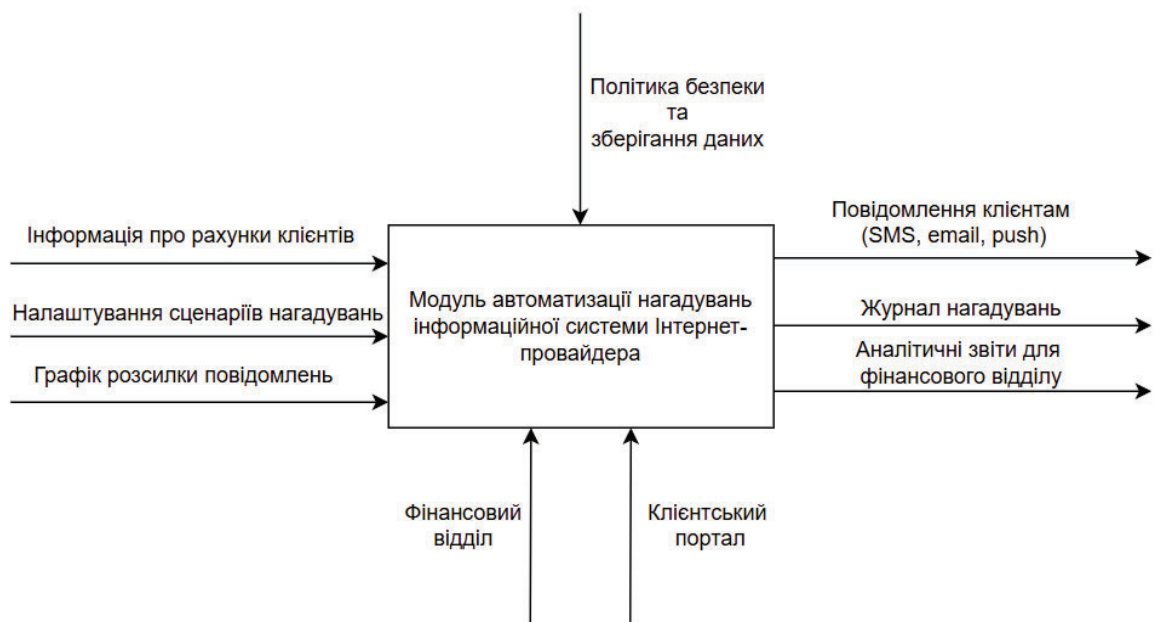


Рисунок 4.1 – Контекстна діаграма модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» інформаційної системи інтернет-провайдера

Перелік та опис вихідних та вхідних повідомлень модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» інформаційної системи інтернет-провайдеру представлено у таблиці 4.1 та 4.2.

Таблиця 4.1 – Перелік та опис вихідних повідомлень модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам»

Назва	Форма представлення	Періодичність видачі	Допустимий час затримки	Отримувач інформації
Повідомлення про прострочений рахунок	Email-повідомлення у форматі тексту або html	Щоденно о 09:00 (за розкладом сгон-задачі)	До 5 секунд	Клієнт
Журнал надсилян	Таблиця в інтерфейсі та документ .pdf з можливістю експорту	За запитом користувача	До 3 секунд	Менеджер, адміністратор
Рахунок до сплати	Файл .pdf з інформацією про суму, клієнта та послугу	Після створення нового рахунку	До 2 секунд	Клієнт, адміністратор (копія)
Інформація про шаблони повідомлень	Таблиця з назвами, текстами та каналами в UI	За запитом (панель адміністрування)	До 1 секунди	Адміністратор

Таблиця 4.2 – Перелік та опис вхідних повідомлень модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам»

Назва	Форма представлення	Частота одержання	Джерело
Дані про клієнтів	Відеограма	Після реєстрації клієнта або оновлення	БД модуля «Клієнти»
Інформація про рахунки	Відеограма	Після створення або зміни стану рахунку	БД модуля «Формування рахунків»
Шаблони повідомлень	Відеограма	За запитом адміністратора	БД модуля «Керування шаблонами»
Дані про канали надсилання	Відеограма	При створенні/редагуванні шаблону	БД модуля «Канали повідомлень»
Статуси оплати	Відеограма	Після оновлення інформації у рахунку	БД модуля «Формування рахунків»
Інформація про розклад надсилання	Відеограма	Щоденно (cron-задача)	Файл конфігурації модуля cron

5 РОЗРОБКА ЕЛЕМЕНТІВ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ МОДУЛЯ «АВТОМАТИЗАЦІЯ ПРОЦЕСІВ СИСТЕМИ НАГАДУВАНЬ ПРО ОПЛАТУ РАХУНКІВ КЛІЄНТАМ»

5.1 Обґрунтування вибору моделі даних та системи управління базами даних

Проектування інформаційної складової модуля автоматизації нагадувань про оплату рахунків є одним із ключових етапів створення повноцінного функціонального рішення. Інформаційна модель визначає, як саме система буде зберігати, структурувати, обробляти та пов'язувати дані між собою. У цьому розділі сформовано логічну модель даних, описано сутності, зв'язки, атрибути та варіанти реалізації на рівні бази даних.

Розробка модуля автоматизації базується на кількох групах інформаційних об'єктів:

- користувачі системи - персонал, що має доступ до модуля (наприклад, оператор, адміністратор, фінансовий менеджер);
- клієнти - особи або організації, які отримують послуги та сплачують за них рахунки;
- рахунки - фінансові документи з сумами до сплати, датами виставлення, строками оплати тощо;
- нагадування - повідомлення, які формуються та відправляються клієнтам;
- шаблони повідомлень - заздалегідь сформовані тексти із змінними, що підставляються під час відправки;
- канали зв'язку - способи надсилання нагадувань;
- журнал повідомлень - історія відправлених нагадувань із фіксацією часу, результату, каналу доставки.

5.2 Опис сутностей та зв'язків модуля

Інформаційна модель модуля автоматизації нагадувань про оплату рахунків клієнтам побудована з урахуванням типових сценаріїв взаємодії з базою даних білінгової системи Інтернет-провайдера. Основу структури становлять логічні сутності, що відображають ключові об'єкти предметної області: клієнти, рахунки, повідомлення, шаблони, канали сповіщень, а також журнал виконаних дій.

Центральною сутністю є Invoice (Рахунок), що містить інформацію про суму до сплати, клієнта-одержувача, дату створення та термін дії рахунку. Рахунок пов'язаний зовнішнім ключем із сутністю Customer (Клієнт), яка зберігає персональні дані клієнта: ім'я, контактну інформацію, адресу проживання тощо. Один клієнт може мати кілька рахунків, тому між цими сутностями реалізовано зв'язок типу один-до-багатьох.

Для кожного рахунку може бути згенероване одне або кілька Notification (Повідомлень). Повідомлення, у свою чергу, пов'язане із шаблоном повідомлення (NotificationTemplate) і каналом доставки (NotificationChannel). Такий підхід забезпечує гнучке формування повідомлень відповідно до правил, заданих у системі.

Факт надсилання повідомлення фіксується в таблиці NotificationLog, яка є журналом активності системи. Записи журналу включають час відправлення, статус доставки, а також відомості про користувача, який ініціював дію. NotificationLog пов'язаний зовнішніми ключами із Notification, Invoice, а також із UserAccount, що представляє адміністратора чи оператора системи.

Таблиця 5.1 – Відомості про сутності розроблюваної БД

Ім'я сутності	Призначення	Опис сутності
Customer	Інформація про клієнтів провайдера	Містить основну інформацію про клієнтів (фізичних або юридичних осіб), які отримують послуги провайдера
Invoice	Дані про виставлені рахунки	Містить записи про виставлені рахунки, їх суми, дати створення та терміни сплати
PaymentStatus	Довідник статусів оплати	Є довідником статусів рахунків (наприклад, Оплачено, Прострочено, Очікується).
Notification	Об'єкт нагадування, пов'язаний із рахунком	Представляє собою нагадування про необхідність оплати конкретного рахунку.
NotificationTemplate	Шаблон повідомлення, що використовується для нагадувань	Містить текстові шаблони повідомлень, які використовуються для створення нагадувань.
NotificationChannel	Канали зв'язку для розсилки повідомлень	Зберігає типи каналів комунікації (наприклад, Email).
NotificationLog	Історія всіх здійснених надсилок повідомлень	Фіксує історію надсилок повідомлень клієнтам та їхній результат.
UserAccount	Системні користувачі модуля	Містить облікові записи персоналу, що працює з модулем нагадувань.
ActionRule	Правила активації сценаріїв нагадування	Містить умови та логіку активації сценаріїв нагадувань.

Після визначення основних сутностей необхідно проаналізувати взаємозв'язки між ними, а також встановити типи їхньої кардинальності.

Узагальнена інформація про типи зв'язків між сутностями наведена в таблиці 5.2.

Таблиця 5.2 – Відомості про зв'язки між сутностями розроблюваної БД

Назва першої сутності	Назва другої сутності	Кардинальність
Customer	Invoice	1: N
Invoice	PaymentStatus	N:1
Invoice	Notification	1: N
Notification	NotificationTemplate	N:1
Notification	NotificationChannel	N:1
NotificationLog	Notification	1: N
UserAccount	NotificationLog	1: N

У розроблюваній інформаційній системі модуль нагадувань базується на взаємозв'язаних сутностях, які логічно підтримують повний цикл обробки рахунків та комунікації з клієнтами. Клієнт (Customer) може мати кілька рахунків (Invoice), кожен з яких має певний статус (PaymentStatus) - оплачено, прострочено або очікується. До кожного рахунку може бути створено кілька нагадувань (Notification), які генеруються відповідно до заданих правил. Для кожного нагадування визначається шаблон повідомлення (NotificationTemplate) і канал доставки (NotificationChannel), що забезпечує гнучкість у форматі комунікації.

Подальша обробка кожного повідомлення фіксується у журналі (NotificationLog), де зберігаються дата й час надсилання, статус доставки, а також користувач системи (UserAccount), який виконував операцію або запустив сценарій. Таким чином, зв'язки між сутностями мають кардинальність переважно типу «один до багатьох» (1:N або N:1), що забезпечує масштабованість системи та дозволяє точно фіксувати взаємодію

на кожному етапі - від створення рахунку до надсилання повідомлення клієнту та реєстрації його статусу.

5.3 Опис атрибутів сутностей

Після визначення взаємозв'язків між сутностями наступним етапом є уточнення складу атрибутів, що описують кожну з них. Деталізовану інформацію про атрибути сутностей представлено в таблицях 5.3 – 5.10.

Таблиця 5.3 – Відомості про атрибути сутності «Customer»

Атрибут	Опис	Тип даних	NULL	Ключ
CustomerID	Унікальний ідентифікатор клієнта	INT	Ні	PK
FullName	Повне ім'я клієнта	VARCHAR(100)	Ні	
Email	Адреса електронної пошти	VARCHAR(100)	Так	
Phone	Контактний номер телефону	VARCHAR(20)	Так	
Address	Адреса клієнта	VARCHAR(150)	Так	
RegistrationDate	Дата реєстрації в системі	DATE	Ні	

Customer – основна таблиця, що містить унікальні дані про клієнтів Інтернет-провайдера, які отримують послуги. Зберігає такі дані, як ПІБ, контактну інформацію, адресу, дату реєстрації. Є вихідною точкою для пов'язування клієнта з усіма фінансовими операціями та комунікаціями в системі. Усі рахунки, нагадування та історія взаємодії з клієнтом формуються на основі цього запису.

Таблиця 5.4 – Відомості про атрибути сутності «Invoice»

Атрибут	Опис	Тип даних	NUL L	Ключ
InvoiceID	Унікальний ідентифікатор рахунку	INT	Hi	PK
CustomerID	Зовнішній ключ до клієнта	INT	Hi	FK
Amount	Сума рахунку до сплати	DECIMAL(10,2)	Hi	
DateIssued	Дата створення рахунку	DATE	Hi	
DueDate	Кінцева дата оплати	DATE	Hi	
PaymentStatus ID	Зовнішній ключ до статусу рахунку	INT	Hi	FK

Таблиця 5.5 – Відомості про атрибути сутності «Notification»

Атрибут	Опис	Тип даних	NULL	Ключ
NotificationID	Унікальний ідентифікатор нагадування	INT	Hi	PK
InvoiceID	Зовнішній ключ до рахунку, до якого стосується нагадування	INT	Hi	FK
CreatedAt	Дата та час створення нагадування	DATETIME	Hi	
ChannelID	Зовнішній ключ до каналу комунікації (email)	INT	Hi	FK
TemplateID	Зовнішній ключ до шаблону повідомлення, що використовується	INT	Hi	FK

Таблиця 5.6 – Відомості про атрибути сутності «NotificationLog»

Атрибут	Опис	Тип даних	NULL	Ключ
LogID	Унікальний ідентифікатор запису в журналі	INT	Ні	PK
NotificationID	Зовнішній ключ до нагадування, з яким пов'язаний цей запис	INT	Ні	FK
SentAt	Дата та час надсилання повідомлення	DATETIME	Ні	
SentBy	Зовнішній ключ до користувача, який ініціював надсилання	INT	Ні	FK
DeliveryStatus	Поточний статус доставки повідомлення (успішно, помилка тощо)	VARCHAR(50)	Так	

Таблиця 5.6 – Відомості про атрибути сутності «UserAccount»

Атрибут	Опис	Тип даних	NULL	Ключ
UserID	Унікальний ідентифікатор користувача	INT	Ні	PK
FullName	Повне ім'я користувача	VARCHAR(100)	Ні	
Login	Логін для авторизації у системі	VARCHAR(50)	Ні	
PasswordHash	Хеш пароля користувача	VARCHAR(100)	Ні	

Таблиця 5.7 – Відомості про атрибути сутності «NotificationTemplate»

Атрибут	Опис	Тип даних	NULL	Ключ
TemplateID	Унікальний ідентифікатор шаблону повідомлення	INT	Ні	PK
Name	Назва шаблону	VARCHAR(100)	Ні	
Content	Основний текст шаблону з параметрами (наприклад: {Ім'я}, {Сума})	TEXT	Ні	
ChannelID	Зовнішній ключ до каналу, з яким пов'язано цей шаблон	INT	Ні	FK

NotificationTemplate – таблиця, в якій зберігаються шаблони повідомлень. Шаблони містять уніфіковані тексти з параметрами (наприклад: {Ім'я}, {Сума}, {Дата}). Їх використання дає змогу стандартизувати зовнішню комунікацію й полегшує масштабування системи.

Таблиця 5.8 – Відомості про атрибути сутності «NotificationChannel»

Атрибут	Опис	Тип даних	NULL	Ключ
ChannelID	Унікальний ідентифікатор каналу сповіщення	INT	Ні	PK
Name	Назва каналу	VARCHAR(50)	Ні	
Description	Опис каналу або його технічні особливості	VARCHAR(150)	Так	

Таблиця 5.9 – Відомості про атрибути сутності «PaymentStatus»

Атрибут	Опис	Тип даних	NULL	Ключ
StatusID	Унікальний ідентифікатор статусу	INT	Ні	PK
StatusName	Назва статусу (наприклад: «Оплачено», «Прострочено»)	VARCHAR(50)	Ні	

PaymentStatus – довідкова таблиця, яка містить перелік можливих статусів рахунку: «Оплачено», «Прострочено», «Очікується оплата» тощо. Дає змогу уніфікувати зберігання статусів і застосовувати їх до великої кількості рахунків. Вона також використовується для визначення логіки спрацьовування нагадувань.

Таблиця 5.10 – Відомості про атрибути сутності «ActionRule»

Атрибут	Опис	Тип даних	NULL	Ключ
RuleID	Унікальний ідентифікатор правила	INT	Ні	PK
Name	Назва правила	VARCHAR(100)	Ні	
TriggerCondition	Умова спрацювання (наприклад, «3 дні до дедлайну»)	TEXT	Ні	
IsActive	Активність правила (true/false)	BOOLEAN	Ні	

ActionRule – таблиця з описом автоматичних правил, що визначають умови формування нагадувань. Наприклад: «за 3 дні до дати оплати» або «якщо рахунок не сплачено протягом 7 днів після нагадування». Кожне правило може бути активоване або вимкнене, що дозволяє гнучко керувати сценаріями.

5.4 Розробка логічної та фізичної схеми бази даних модуля

База даних є ключовим та критично важливим компонентом програмного модуля автоматизації нагадувань про оплату рахунків, оскільки вона забезпечує не лише зберігання, а й швидкий та структурований доступ до всієї необхідної інформації, яка використовується під час обробки, формування та відправки повідомлень клієнтам. Саме база даних виконує роль централізованого сховища, що акумулює відомості про клієнтів, їхні рахунки, історію оплат, шаблони повідомлень, статуси надсилання, журнали комунікацій, зареєстрованих адміністраторів та правила, які визначають умови для генерації нагадувань. Надійність та структурованість бази даних напряму впливають на стабільність роботи всієї системи.

На етапі логічного проектування бази даних було визначено перелік основних сутностей, які відображають ключові об'єкти предметної області, та побудовано логічну модель взаємозв'язків між ними. В логічну структуру увійшли таблиці: Customer - для зберігання детальної інформації про клієнтів, таких як ПІБ, email, номер телефону, адреса реєстрації; Invoice - яка фіксує дані про виставлені рахунки, їхню суму, дату створення, дату сплати та статус; PaymentStatus - що містить довідник можливих станів оплати (наприклад, «не оплачено», «оплачено», «прострочено»); NotificationTemplate - для зберігання шаблонів повідомлень із текстом та каналом надсилання; NotificationChannel - що відображає доступні способи комунікації (email,

SMS, push); Notification - де фіксуються сформовані повідомлення із прив'язкою до конкретного рахунку; NotificationLog - що відображає історію всіх відправлень із фіксацією дати, часу, користувача та результату; UserAccount - таблиця з обліковими даними адміністраторів; ActionRule - яка описує умови запуску певних повідомлень; а також CustomerCategory - для сегментації клієнтів за рівнем пріоритету чи платіжної дисципліни.

Логічна модель реалізована у вигляді діаграми типу «сутність-зв'язок» (ER-діаграма), яка дозволяє наочно представити усі взаємозв'язки між таблицями. Наприклад, один запис у таблиці Customer може бути пов'язаний із багатьма рахунками в таблиці Invoice (зв'язок «один до багатьох»), кожен рахунок має статус з таблиці PaymentStatus, а кожне повідомлення в Notification базується на шаблоні з NotificationTemplate та надсилається через конкретний канал з NotificationChannel. Журнал NotificationLog, у свою чергу, посилається на Notification, Invoice, UserAccount, тим самим зберігаючи повну історію виконаних дій і дозволяючи здійснювати аудит усіх взаємодій системи.

Після логічного моделювання було розроблено фізичну схему бази даних, яку реалізовано в середовищі PostgreSQL. Під час фізичного проєктування особливу увагу приділяли коректному визначенню типів даних, встановленню зовнішніх ключів для забезпечення цілісності інформації, а також розміщенню індексів на найчастіше використовувани поля - зокрема, email, invoiceID, notificationID, які активно використовуються під час вибірок для розсилки повідомлень чи формування аналітики. Усі таблиці структуровано відповідно до вимог третьої нормальної форми (3NF), що дозволяє мінімізувати дублювання інформації та зменшити надмірність даних, підвищуючи цілісність і узгодженість у процесі роботи.

Таблиці створювалися за допомогою SQL-інструкцій, із використанням конструкцій CREATE TABLE, REFERENCES, CONSTRAINT тощо. Для зручності адміністрування та візуального контролю використовувалося середовище pgAdmin, яке дозволило перевірити структуру, виявити можливі

помилки на ранньому етапі, а також запустити початкові запити на вставку тестових даних. Крім того, було передбачено створення резервної копії бази для захисту від втрати даних у разі системного збою або неправомірного втручання.

Особливістю фізичного проекту є також адаптація структури під майбутнє масштабування. У разі розширення функціональності модуля - наприклад, додавання нових каналів сповіщень, сегментів клієнтів або локалізацій - поточна структура дозволяє безболісно додавати нові таблиці або змінювати існуючі, зберігаючи цілісність і гнучкість. Усі відношення побудовані таким чином, щоб можна було використовувати каскадне оновлення/видалення записів, що спрощує модифікацію й обробку пов'язаних об'єктів.

Загалом, база даних для модуля нагадувань спроектована з урахуванням сучасних вимог до захищеного зберігання персональних і фінансових даних, із забезпеченням чіткої логіки зв'язків, високої продуктивності та можливості ефективної обробки великого обсягу транзакцій. Це створює надійне підґрунтя для стабільної роботи автоматизованої системи нагадувань про оплату рахунків в рамках білінгової інфраструктури інтернет-провайдера.

Логічну та фізичну моделі даних наведено на рисунках 5.1 та 5.2.

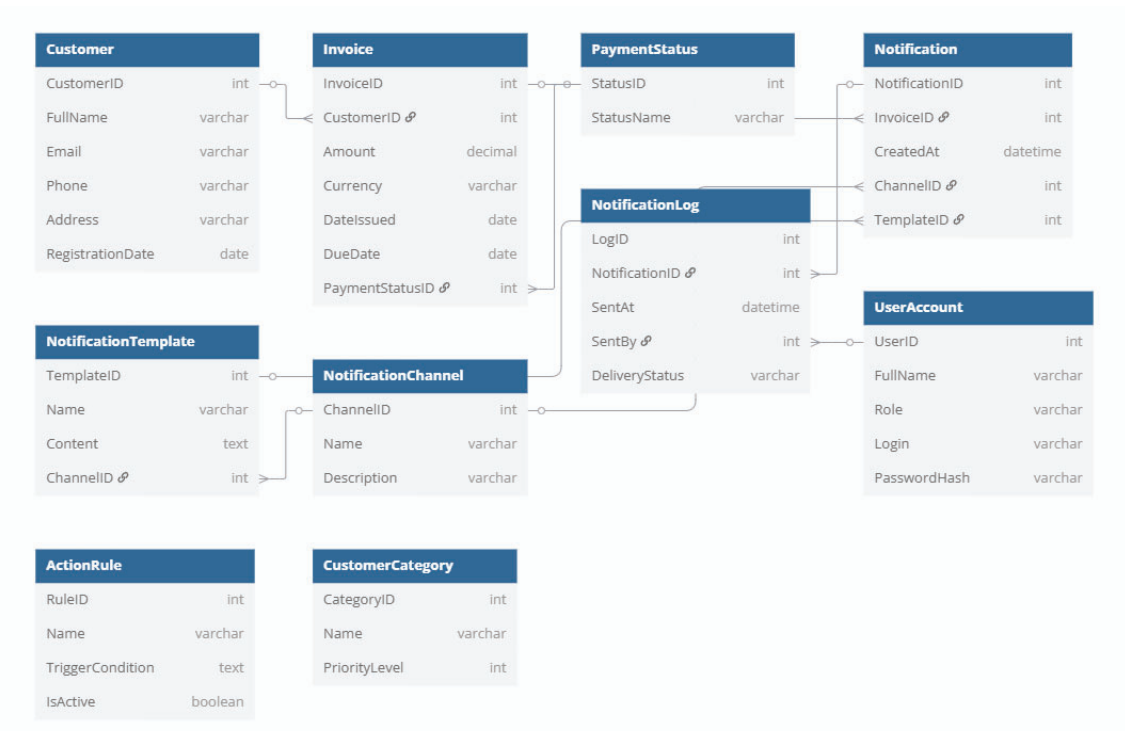


Рисунок 5.1 – Схема логічної моделі даних модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» інформаційної системи інтернет-провайдеру

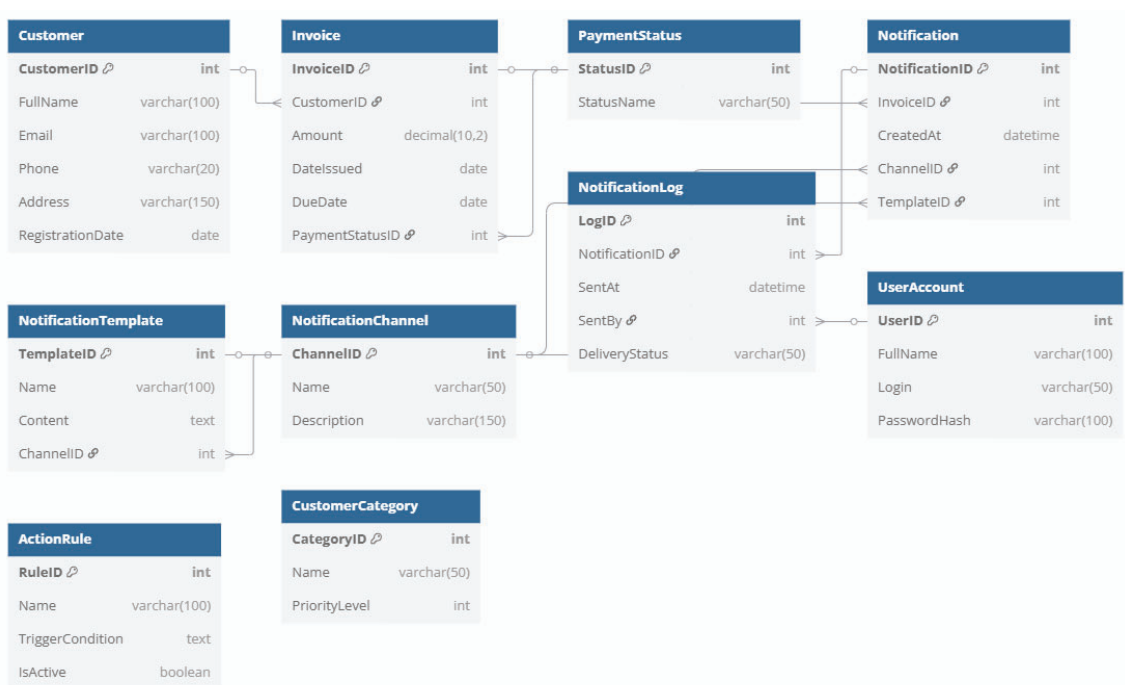


Рисунок 5.2 – Схема фізичної моделі даних модуля «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» інформаційної системи інтернет-провайдеру

6 РОЗРОБКА І ОБҐРУНТУВАННЯ ВИБОРУ ЕЛЕМЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧУЮЧЕННЯ МОДУЛЯ

У межах дипломної роботи було розроблено модуль автоматизації нагадувань про оплату рахунків, який функціонує як доповнення до білінгової системи інтернет-провайдера. Розробка охопила повний цикл створення, налаштування, тестування та впровадження модуля з урахуванням вимог реального виробничого середовища. Основне призначення програмного забезпечення - виявлення прострочених рахунків, формування відповідних повідомлень за заздалегідь заданими шаблонами та їх надсилання через визначені канали комунікації.

Програмна архітектура модулю побудована на основі принципу мікросервісів. Це забезпечує незалежність модулю від інших компонентів системи, гнучкість розгортання, масштабованість і простоту супроводу. Комунікація з основною базою даних здійснюється через REST API. Обраною мовою реалізації став JavaScript із використанням середовища виконання Node.js, що дозволяє ефективно обробляти асинхронні події, необхідні для регулярних перевірок і запуску нагадувань у визначені інтервали часу.

Для реалізації прикладної логіки було використано такі ключові технології:

- express.js - фреймворк для побудови веб-серверу та маршрутизації запитів;
- PostgreSQL - потужна СУБД, яка використовується для зберігання інформації про клієнтів, рахунки, шаблони повідомлень, канали, статуси оплат і логи надсилань;
- node-cron - бібліотека для організації запуску повторюваних задач (перевірка рахунків) за розкладом;

- nodemailer - засіб для реалізації сервісу електронної пошти, що дозволяє надсилати нагадування на email-адреси користувачів;
- dotenv - для зберігання та конфігурації середовищних змінних (доступи до бази, SMTP-параметри);
- bcrypt - для безпечного зберігання паролів адміністраторів у вигляді хешів;
- pg - клієнт PostgreSQL для Node.js.

Основна логіка модуля полягає у циклічній перевірці таблиці Invoice у базі даних на предмет наявності прострочених рахунків. Якщо для конкретного запису спрацьовує умова, вказана у таблиці ActionRule, система формує повідомлення відповідно до NotificationTemplate і надсилає його через NotificationChannel. У таблиці NotificationLog фіксуються такі параметри: дата та час надсилання, статус (успішно/невдало), канал передачі, а також email одержувача й номер рахунку. У разі ручного запуску нагадування адміністратором, лог зберігає також ID користувача (SentBy).

Адміністративна частина реалізована як веб-інтерфейс на основі HTML, CSS та JavaScript. Усі компоненти розділені на окремі сторінки (index.html, login.html, logs.html, templates.html). Це дозволяє персоналу компанії керувати базою рахунків, створювати й редагувати шаблони, переглядати історію сповіщень та авторизуватись для виконання критичних дій. Інтерфейс розроблено у мінімалістичному стилі з урахуванням ергономіки та доступності.

Додатково реалізовано механізм авторизації з поділом прав доступу на рівні ролей. Для зберігання облікових записів використовується таблиця UserAccount, де кожен користувач має логін, пароль (хеш), а також роль (адміністратор або менеджер). При кожному вході до системи дані перевіряються, і залежно від ролі користувач отримує доступ до відповідного функціоналу.

З точки зору надійності, модуль має вбудовані засоби обробки помилок - зокрема, перевірку на недоступність бази даних, невірні SMTP-параметри

або відсутність шаблонів. Логи помилок виводяться у консоль і можуть бути розширені для зберігання у спеціальному журналі (log.txt або таблиці помилок).

Перевагою побудованої системи є її адаптивність. Додавання нового каналу комунікації (наприклад, SMS чи push) вимагає лише доповнення таблиць NotificationChannel та незначного розширення сервісної логіки у файлах services/mailler.js чи services/sms.js. Структура коду відповідає принципам чистої архітектури: логіка, інтерфейс і зовнішні залежності чітко відокремлені, що спрощує підтримку та розширення проєкту.

Таким чином, у межах дипломної роботи було реалізовано повноцінний модуль для автоматичного нагадування клієнтам провайдера про оплату рахунків. Завдяки обраним технологіям та архітектурним підходам система є масштабованою, гнучкою у налаштуванні, придатною для реального виробничого впровадження й готовою до подальшого розширення у межах інформаційної системи підприємства.

7 ТЕСТУВАННЯ ТА ОЦІНКА НАДІЙНОСТІ ФУНКЦІОНУВАННЯ ПРОГРАМНИХ І ТЕХНІЧНИХ РІШЕНЬ МОДУЛЯ «АВТОМАТИЗАЦІЯ ПРОЦЕСІВ СИСТЕМИ НАГАДУВАНЬ ПРО ОПЛАТУ РАХУНКІВ КЛІЄНТАМ»

Розглянуто перевірку функціональності розробленого модуля автоматизації процесу надсилання нагадувань про оплату рахунків клієнтам інтернет-провайдера. Після завершення розробки було виконано повне тестування ключових елементів, зокрема інтерфейсів, логіки взаємодії з базою даних, сервісу email-повідомлень. В процесі тестування особлива увага приділялась як відповідності реалізації заявленим функціональним вимогам, так і стійкості системи до типових помилок, навмисних або випадкових порушень вводу.

У структурі проекту реалізованого модуля автоматизації нагадувань про оплату рахунків інтернет-провайдера застосовано логічне розділення на директорії та функціональні компоненти, що відповідають принципам чистої архітектури.

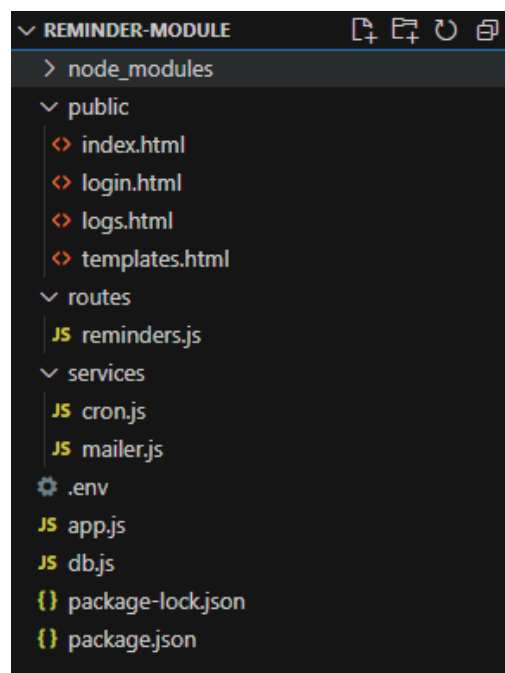


Рисунок 7.1 – Структура проекту

У директорії `public` знаходяться всі статичні ресурси - сторінки HTML, які відображають інтерфейси користувача. Зокрема, `index.html` є основною панеллю керування рахунками, `login.html` відповідає за форму авторизації користувача, `logs.html` відображає журнал надісланих нагадувань, а `templates.html` дозволяє адмініструвати шаблони повідомлень. Така сегментація забезпечує простоту навігації та можливість розширення інтерфейсів без зміни серверної логіки.

У папці `routes` знаходиться файл `reminders.js`, який містить маршрути для роботи з рахунками, логами та авторизацією. Усі запити API до `/api/reminders/...` обробляються саме тут. Це дозволяє централізовано керувати серверною логікою, не змішуючи її з HTML-частиною або службовими скриптами.

Директорія `services` вміщує окремі функціональні сервіси: `cron.js` - файл, відповідальний за автоматичну щоденну перевірку прострочених рахунків та запуск email-розсилки через CRON-задачу, а `mailer.js` - модуль для відправлення електронних листів за допомогою бібліотеки `nodemailer`.

У корені проєкту знаходиться конфігураційний файл `.env`, що містить налаштування для підключення до бази даних PostgreSQL і параметри поштового облікового запису. Це дозволяє зберігати конфіденційні дані поза межами коду та змінювати їх без перезапису логіки застосунку.

Файл `app.js` є точкою входу до програми. Саме тут відбувається ініціалізація Express-сервера, підключення маршрутів, `middleware` для авторизації та запуск сервера на вказаному порту. Файл `db.js` містить конфігурацію для з'єднання з базою PostgreSQL, реалізовану через бібліотеку `pg`.

Таке логічне структурування дозволяє легко орієнтуватись у коді, масштабувати застосунок у разі додавання нових каналів нагадувань (SMS, push), а також забезпечує зручність у супроводі й перевірці працездатності кожного окремого модуля.

На початковому етапі перевірки працездатності розробленого модуля було здійснено тестування механізму створення рахунків через головний інтерфейс користувача. Цей етап мав на меті перевірити коректність обробки введених користувачем даних, їх збереження у базі даних, а також відображення інформації на стороні клієнта.

Інтерфейс додавання рахунку реалізовано у вигляді простої веб-форми, що включає чотири поля: ідентифікатор клієнта (CustomerID), суму до сплати (Amount), дату завершення терміну оплати (DueDate) та статус платежу (PaymentStatusID). Для перевірки було введено як валідні, так і граничні значення - наприклад, максимально допустиму суму, некоректну дату або пусте поле - з метою виявлення можливих вразливостей у валідації форми.

Після натискання кнопки "Додати рахунок" відбувається відправка даних на сервер за допомогою HTTP-запиту типу POST до відповідного ендпоінту API. На сервері, в модулі маршрутизації, інформація перевіряється та вставляється до таблиці Invoice у базі даних PostgreSQL. У разі успішного збереження, відповідь від сервера підтверджує успішність операції, після чого на клієнтській стороні автоматично оновлюється список рахунків.

Список рахунків виводиться у вигляді HTML-елементів, кожен з яких містить основну інформацію про рахунок: номер, суму, дату та статус. Крім того, біля кожного запису розміщено кнопку видалення, яка дає змогу негайно протестувати операцію DELETE. Натискання на цю кнопку викликає HTTP-запит на видалення відповідного запису з бази. При успішному видаленні запис миттєво зникає зі списку без необхідності оновлення сторінки.

На рисунках 7.2, 7.3, 7.4 зображен процес додавання нового рахунку.

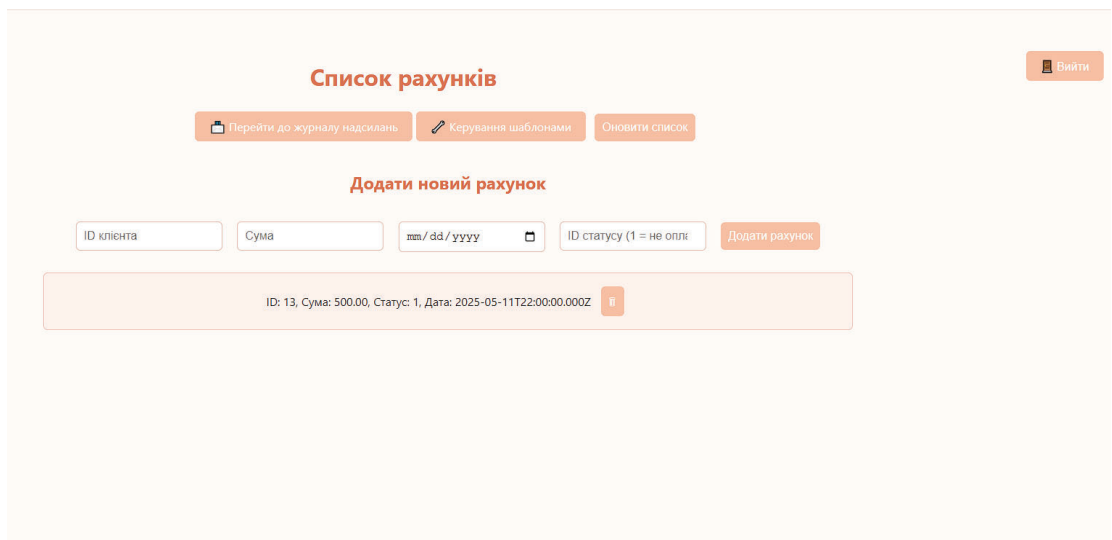


Рисунок 7.2 - Інтерфейс модуля для перегляду та створення рахунків

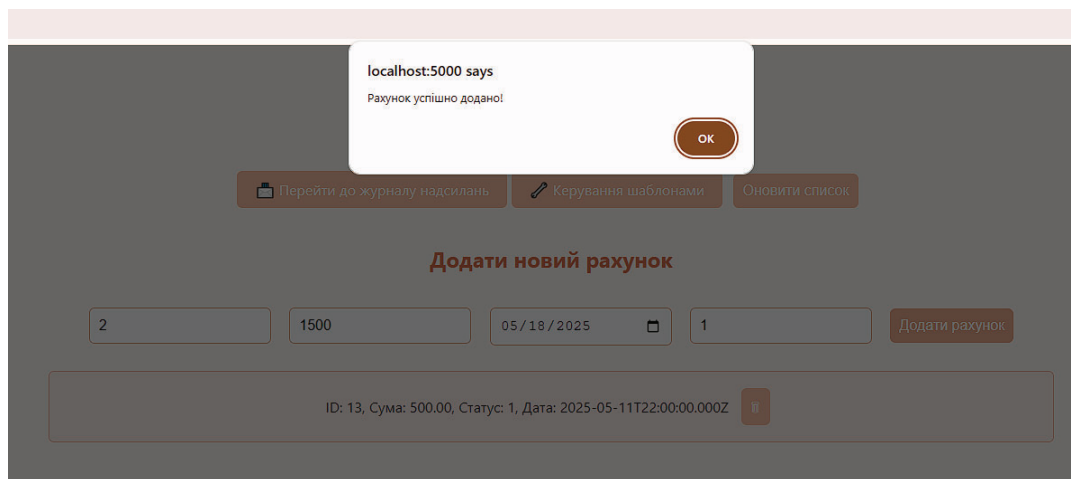


Рисунок 7.3 - Повідомлення про успішне додавання рахунку

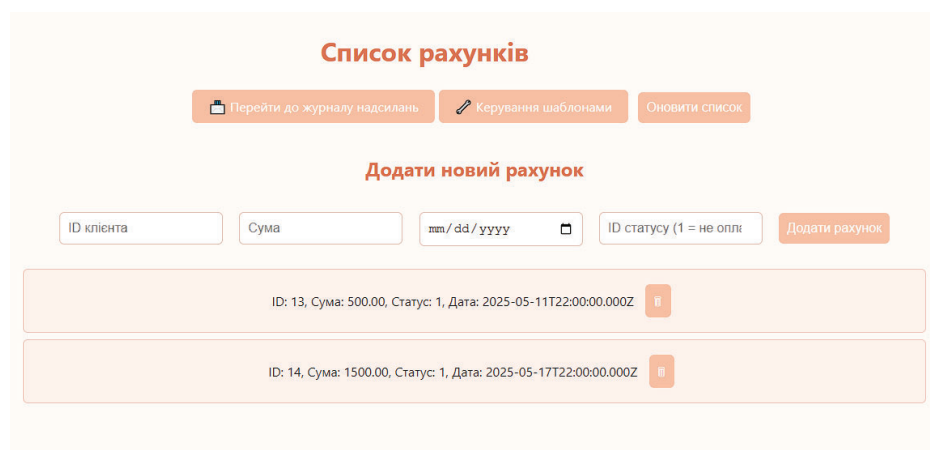


Рисунок 7.4 - Відображення списку рахунків після додавання нового запису

Для перевірки працездатності та логіки автоматичного механізму нагадування було здійснено ручне ініціювання CRON-задачі, яка за своєю суттю імітує щоденну перевірку стану рахунків користувачів. Цей сценарій тестування дозволив перевірити основну функціональність модуля без потреби чекати на реальний запуск задачі в обраний час (зазвичай щодня о 09:00).

Перед тестуванням у базу даних було попередньо додано кілька тестових рахунків із заздалегідь простроченою датою (DueDate) та статусом "не оплачено" (PaymentStatusID = 1). Це дозволило створити ситуацію, яка відповідає умовам автоматичного запуску повідомлення. Далі вручну було викликано метод, що відповідає за CRON-логіку, зокрема функцію cron.schedule(...), яка виконує запит до таблиці Invoice для отримання всіх рахунків, що підпадають під умови надсилання нагадувань.

Система успішно виявила відповідні записи. Для кожного простроченого рахунку було сформовано текст повідомлення на основі шаблону з таблиці NotificationTemplate. При формуванні враховувались змінні, які підставлялись до шаблону - наприклад, ім'я клієнта, сума до сплати та дата завершення терміну.

Надсилання електронного листа здійснювалося за допомогою бібліотеки nodemailer, яка конфігурована з обліковим записом пошти. Під час тесту в якості адресата використовувалась реальна тестова електронна пошта, що дозволило проконтролювати сам факт доставки повідомлення та зміст листа.

Після кожного успішного надсилання до таблиці NotificationLog в базі даних автоматично додавався новий запис із наступною інформацією:

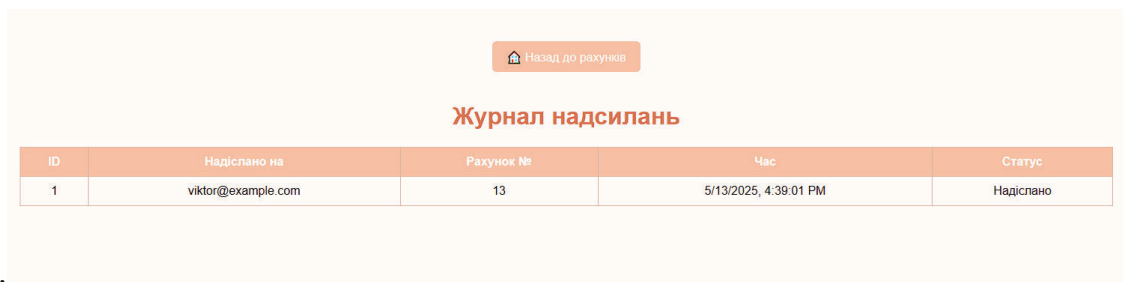
- ідентифікатор надісланого повідомлення (NotificationID);
- електронна адреса одержувача (Email);
- дата та час надсилання (SentAt);
- статус операції (DeliveryStatus);
- номер рахунку, до якого належало повідомлення (InvoiceID);

- відправник (у випадку автоматичного запуску - системний запис).

У процесі перевірки також оцінювався зміст журналу надсилай через веб-інтерфейс. Була протестована коректність відображення нових записів у таблиці, що підтвердило правильну інтеграцію серверної логіки з клієнтським інтерфейсом.

Результати показали, що повний цикл - від виявлення простроченого рахунку, формування повідомлення, його доставки до користувача і реєстрації у журналі - відбувається без збоїв, а модуль виконує поставлені функції автоматично, стабільно та згідно з очікуваною логікою.

На рисунку 7.5 представлено веб-інтерфейс журналу надсилай нагадувань клієнтам, реалізований у вигляді таблиці. Цей розділ відповідає за вивід інформації про всі автоматично надіслані повідомлення, що генеруються модулем на основі виявлених прострочених рахунків.



ID	Надіслано на	Рахунок №	Час	Статус
1	viktor@example.com	13	5/13/2025, 4:39:01 PM	Надіслано

Рисунок 7.5 - Таблиця обліку успішно надісланих нагадувань у модулі «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» інформаційної системи інтернет-провайдеру

На рисунку 7.6 зображено результат запити до таблиці NotificationLog, виконаного у середовищі pgAdmin.

Query Query History

```

1 SELECT * FROM public.notificationlog
2 ORDER BY logid ASC

```

Data Output Messages Notifications

	logid [PK] integer	notificationid integer	sentat timestamp without time zone	sentby integer	deliverystatus character varying (50)	invoiceid integer	email character varying (100)
1	1	[null]	2025-05-13 16:39:01.386171	[null]	Надіслано	13	viktor@example.com

Рисунок 7.6 - Вміст таблиці NotificationLog після виконання запиту в pgAdmin

У верхній частині розташовано кнопку "Назад до рахунків", яка реалізує навігацію до головного функціонального екрану модуля - сторінки з керуванням рахунками (index.html). Це покращує зручність користування та дозволяє адміністраторам легко перемикатися між розділами системи.

Нижче - заголовок "Журнал надсилань", після якого йде таблиця з наступними колонками:

- ID - унікальний ідентифікатор запису в журналі. Його формує база даних автоматично при кожному факті відправки повідомлення;
- надіслано на - адреса електронної пошти клієнта, на яку було відправлено повідомлення. Це дозволяє швидко ідентифікувати, кому саме було направлено нагадування;
- рахунок № - номер рахунку з таблиці Invoice, на який стосується повідомлення. Він дозволяє перейти до відповідного рахунку та перевірити деталі вручну (у майбутньому можлива інтеграція прямого перегляду PDF);
- час - дата та час фактичного надсилання повідомлення. У даному випадку використовується локалізований формат часу, що зчитується з поля

SentAt таблиці NotificationLog. Значення 5/13/2025, 4:30:01 PM свідчить про актуальність події;

– статус - фіксує стан повідомлення. У цьому прикладі - "Надіслано", що означає, що email було відправлено без помилок.

Цей журнал є критично важливою частиною системи, оскільки дозволяє:

- відслідковувати факт надсилання;
- верифікувати результати автоматичних дій системи;
- збирати аналітику за періодами;
- проводити ручну перевірку клієнтів у разі скарг або технічних збоїв.

Журнал оновлюється автоматично при кожному запуску CRON-задачі, яка щодня виявляє прострочені рахунки та надсилає нагадування. Таким чином, він є живим відображенням роботи автоматизованого модуля, що забезпечує прозорість і контроль усіх дій системи.

Інтерфейс шаблонів дозволяє переглядати існуючі шаблони повідомлень, а також редагувати їхній текст. Збереження змін у базу даних відбувається миттєво.

На рисунку 7.7 представлений інтерфейс сторінки керування шаблонами повідомлень (файл templates.html), що є частиною веб-модуля автоматизації нагадувань про оплату рахунків. Даний функціональний блок призначений для адміністраторів або менеджерів провайдера і дозволяє централізовано створювати, переглядати та видаляти шаблони сповіщень, які надсилаються клієнтам.

Назад до рахунків

Керування шаблонами

Назва шаблону

Текст повідомлення

ID каналу (необов'язково)

Додати шаблон

ID	Назва	Текст	Канал	Дія
2	1	Тест	-	В

Рисунок 7.7 - Інтерфейс керування шаблонами нагадувань у веб-системі

У верхній частині сторінки знаходиться навігаційна кнопка «Назад до рахунків», яка реалізує логіку повернення до головної сторінки управління рахунками (index.html).

Центральний блок сторінки - це форма створення шаблону. Вона складається з трьох полів:

- назва шаблону (Template Name) - коротка текстова мітка для ідентифікації шаблону;
- текст повідомлення (Content) - основний вміст повідомлення, який буде надісланий клієнту. Тут адміністратор може ввести звернення, текст нагадування, суму, крайній термін тощо;
- ID каналу - поле для зазначення способу надсилання. Це дозволяє прив'язати шаблон до конкретного каналу доставки, що передбачено гнучкою архітектурою системи.

Нижче розташована кнопка «Додати шаблон», яка викликає запит POST до API та надсилає дані у таблицю NotificationTemplate в базі даних. Після створення шаблон миттєво відображається в таблиці під формою.

У таблиці відображаються всі збережені шаблони з такими полями:

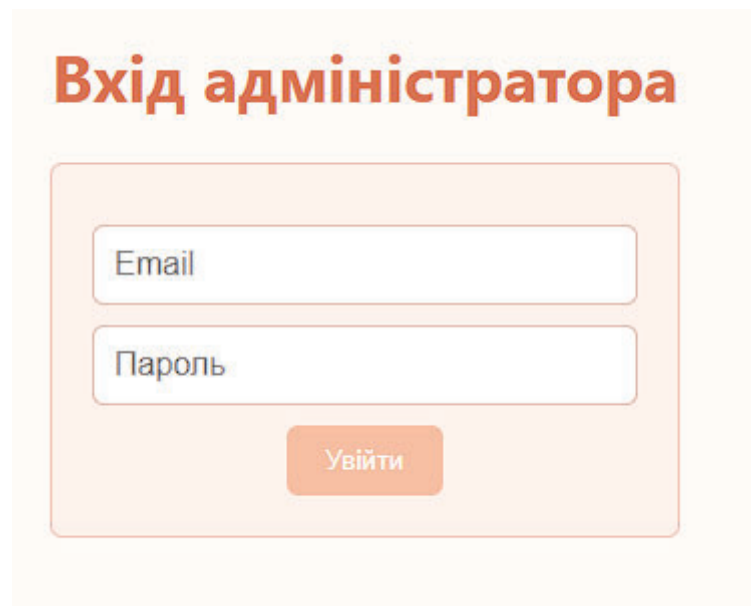
- ID - унікальний ідентифікатор шаблону;
- назва - введене користувачем ім'я шаблону;

- текст - вміст повідомлення;
- канал - тип каналу доставки;
- дія - кнопка, яка дозволяє видалити шаблон за відповідним `templateID` через API-запит `DELETE`.

Цей модуль суттєво спрощує управління контентом повідомлень без необхідності змінювати програмний код. Він дозволяє адміністраторам оперативно оновлювати тексти нагадувань у разі зміни політики, умов або стилю комунікації. У поєднанні з механізмом шаблонізації повідомлень (в майбутньому можливе використання змінних `{{fullname}}`, `{{amount}}`) функціонал є гнучким і масштабованим.

Усі сторінки захищені авторизацією: при спробі доступу до `index.html` без входу через `login.html` користувач буде автоматично перенаправлений. Було перевірено сценарій некоректного входу з неправильним логіном або паролем - у такому випадку система блокує доступ і виводить повідомлення. Після успішного входу відкривається головна сторінка, а при натисканні кнопки «Вийти» сеанс завершується, і користувач повертається на сторінку авторизації.

На рисунку 7.8 представлено вікно входу адміністратора - це стартовий екран для авторизованого доступу до функціоналу модуля нагадувань про оплату рахунків. Форма побудована з урахуванням принципів зручності та мінімалізму, забезпечує базовий захист за допомогою логіна та пароля.



The image shows a web form for administrator login. At the top, the title 'Вхід адміністратора' is displayed in a large, bold, orange font. Below the title, there is a light orange rounded rectangle containing the form. Inside this rectangle, there are two white input fields with orange borders. The first field is labeled 'Email' and the second is labeled 'Пароль'. Below these fields is an orange button with the text 'Увійти' in white.

Рисунок 7.8 - Вікно авторизації адміністратора у веб-системі нагадувань

У верхній частині розташовано заголовок «Вхід адміністратора», який однозначно вказує на те, що доступ до системи обмежений і потребує автентифікації.

Форма складається з двох полів вводу:

- email - поле для введення ідентифікатора користувача (логіна). Верифікація виконується за унікальним значенням з колонки login у таблиці UserAccount.

- пароль - захищене поле, в яке вводиться пароль. Пароль зберігається в базі у вигляді хешу (через bcrypt), що підвищує безпеку даних.

Кнопка «Увійти» ініціює запит до бекенду за маршрутом /api/reminders/login, де відбувається перевірка введених даних. У разі успішного входу користувач перенаправляється на головну сторінку модуля, де може переглядати рахунки, керувати шаблонами та переглядати журнал надсилань.

Цей екран є обов'язковою умовою для доступу до адміністраторського інтерфейсу та забезпечує базову функцію авторизації, необхідну для відокремлення внутрішніх функцій системи від загального користувача. Він

також може слугувати стартовою точкою для розширення - наприклад, додавання ролей, двофакторної автентифікації або журналу входів.

Журнал надсилань коректно відображає історію розсилок: email-адресу отримувача, дату та час надсилання, номер рахунку, статус повідомлення. При видаленні всіх записів система не порушує структури, а при повторному запуску CRON-завдання записи знову з'являються.

Результати перевірки підтвердили, що реалізована система є стабільною, працює згідно з поставленими вимогами та демонструє високий рівень надійності при роботі з реальними і тестовими даними.

8 ОПИС ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ МОДУЛЯ «АВТОМАТИЗАЦІЯ ПРОЦЕСІВ СИСТЕМИ НАГАДУВАНЬ ПРО ОПЛАТУ РАХУНКІВ КЛІЄНТАМ»

Розроблений у межах дипломної роботи модуль автоматизованого надсилання нагадувань про оплату призначено для вбудовування у внутрішнє інформаційне середовище компанії-провайдера телекомунікаційних послуг. Його основна мета - підвищення своєчасності оплати рахунків з боку клієнтів, зменшення фінансових ризиків компанії та підвищення ефективності обслуговування абонентів без участі оператора чи менеджера.

Перед безпосереднім розгортанням модуль має бути попередньо протестований у контрольованому середовищі (тестовий сервер або середовище staging), максимально наближеному до умов реального використання. Це дозволяє перевірити його сумісність із чинною базою даних клієнтів, рахунків, історією оплат, а також впевнитися у коректній роботі алгоритмів формування та надсилання повідомлень. Тестування охоплює такі компоненти, як:

- сценарії створення та зчитування рахунків;
- правильність генерації повідомлень згідно з шаблонами;
- роботу запланованих задач (cron);
- перевірку авторизації доступу до адміністративної панелі.

Для коректної роботи модуля необхідно налаштувати підключення до електронної пошти (SMTP-серверу). З метою безпеки рекомендується використовувати паролі додатків або OAuth2-токени, щоб уникнути збереження реального паролю в конфігураційних файлах. Якщо використовується пошта Google (Gmail), слід обов'язково активувати двофакторну автентифікацію та створити окремий пароль для сервісу.

Модуль реалізований як мікросервіс, побудований на основі технології Node.js та бази даних PostgreSQL. Його розгортання можливе як на

внутрішньому сервері компанії (наприклад, у межах корпоративної локальної мережі), так і у хмарному середовищі (наприклад, AWS, Heroku, DigitalOcean). Важливо забезпечити постійний доступ до порту, на якому працює сервер Express.js, а також встановити захищене з'єднання з базою даних.

Запуск запланованих задач (перевірки прострочених рахунків) виконується за допомогою бібліотеки `node-cron`. Для цього необхідно, щоб процес сервера залишався активним у фоновому режимі, або використовувався менеджер процесів на зразок PM2 чи системного планувальника задач (`cron` у Linux). Результати перевірок і надсилок логуються у таблиці `NotificationLog`, що дозволяє здійснювати подальший моніторинг та аудит роботи модуля.

Окрім електронної пошти, модуль може бути розширений для підтримки додаткових каналів комунікації. Наприклад:

- SMS - шляхом підключення до API сторонніх сервісів, таких як Twilio, TurboSMS, Kyivstar API тощо;
- push-нотифікації - за допомогою Firebase Cloud Messaging або подібних платформ;
- телефонні дзвінки - через VoIP-шлюзи або інтеграцію з Asterisk.

Ці канали потребують окремої авторизації, врахування політики конфіденційності та відповідності законодавству про обробку персональних даних.

Для стабільної та безпечної роботи модуля необхідно:

- забезпечити обмежений доступ до адміністративної панелі. Усі користувачі мають проходити автентифікацію з використанням логіну та паролю, бажано з реалізацією хешування паролів (наприклад, `bcrypt`);
- регулярно оновлювати шаблони повідомлень, щоб відповідати поточній політиці компанії або акційним пропозиціям;

- проводити періодичне резервне копіювання таблиць Invoice, NotificationLog, Customer та NotificationTemplate, що забезпечить захист даних у разі збоїв;

- упровадити логування помилок та створити механізм автоматичних сповіщень адміністраторам у разі виявлення технічних проблем.

У разі позитивного результату пілотного запуску, передбачено поетапне масштабування системи - зокрема, шляхом:

- додавання підтримки мультимовних шаблонів;
- впровадження динамічного налаштування часу відправки для різних сегментів клієнтів;
- розробки мобільної версії панелі керування;
- інтеграції з CRM-системами.

Таким чином, створений модуль дозволяє автоматизувати важливу частину клієнтського обслуговування, зменшити навантаження на персонал, скоротити кількість прострочених платежів та створити позитивний імідж провайдера як сучасної компанії, що дбає про зручність своїх користувачів.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розроблено функціональний модуль «Автоматизація процесів системи нагадувань про оплату рахунків клієнтам» ІС інтернет-провайдеру. Актуальність створеного рішення полягає в необхідності підвищення фінансової дисципліни клієнтів, а також зменшення навантаження на персонал, який займається ручною обробкою та надсиланням повідомлень про заборгованість.

На етапі аналізу предметної області було досліджено бізнес-процеси інтернет-провайдеру, визначено ключові функціональні вимоги до системи, а також розроблено структурну та логічну модель бази даних. Було сформовано архітектуру модуля, яка включає зручний web-інтерфейс для адміністратора, REST API для обробки запитів, а також CRON-сценарій для щоденної перевірки прострочених платежів та автоматичної генерації email-нагадувань. Завдяки реалізації модуля адміністратор отримує можливість додавати нові рахунки через веб-інтерфейс, формувати шаблони повідомлень, переглядати журнал надісланих нагадувань, а також здійснювати контроль за станом оплат. Інтерфейс підтримує авторизацію за логіном і паролем, що забезпечує базовий рівень безпеки доступу до модуля.

У ході практичного тестування було перевірено взаємодію між компонентами системи: обробку форм, збереження даних у базі, відображення динамічного списку рахунків та логів, а також відправку електронних листів на вказані email-адреси клієнтів. Окремо відзначено коректну роботу журналу надсилань, де фіксується час, адресат та статус повідомлення.

У результаті виконання роботи створено програмне рішення, яке відповідає поставленій меті та повністю виконує визначені функціональні вимоги. Модуль може бути інтегрований у внутрішні процеси компанії-провайдеру як окрема підсистема або частина більшої інформаційної системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до організації виконання та захисту кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти спеціальності 122 Комп'ютерні науки за освітньою програмою «Інформаційні технології управління» для студентів усіх форм навчання / Упоряд.: К.Е. Петров, А.В. Міхнова, М.С. Кудрявцева, М.В. Євланов, Т.І. Борисенко.– Електронне видання. – Харків: ХНУРЕ, 2023. – 68 с.
2. Salesforce Docs. Документація для розробників [Електронний ресурс]. - Режим доступу: <https://developer.salesforce.com/docs>.
3. ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. - Женева: International Organization for Standardization, 2011. - 34 с.
4. Закон України «Про захист персональних даних» : Закон України від 01.06.2010 № 2297-VI // Відомості Верховної Ради України. – 2010. – № 34. – Ст. 481. – [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2297-17>
5. Tilkov S., Vinoski S. Node.js: Using JavaScript to build high-performance network programs // IEEE Internet Computing. – 2010. – Vol. 14, No. 6. – P. 80–83.
6. SMS Club. Публічний договір (оферта) [Електронний ресурс] // SMS-Fly. – Режим доступу: https://sms-fly.ua/oferta/?utm_source=chatgpt.com
7. Ringostat. Регламент з роботи в CRM: навіщо потрібен та як створити [Електронний ресурс]. - Режим доступу: <https://blog.ringostat.com/uk/rehlament-z-roboty-v-crm-navishcho-potriben-ta-yak-stvoryty>
8. Stripe. Automated billing systems explained [Електронний ресурс]. - Режим доступу: <https://stripe.com/resources/more/automated-billing-systems-101>

9. DreamFactory. Service-Oriented Architecture and the Modular Approach to Functionality [Електронний ресурс]. - Режим доступу: <https://blog.dreamfactory.com/service-oriented-architecture-and-the-modular-approach-to-functionality/>

10. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 2017-07-01. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

11. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Чинний від 2016-07-01. – Вид. офіц. Київ : УкрНДНЦ, 2016. 16 с.