

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ОПРАЦЮВАННЯ
ДАНИХ ДЛЯ УПРАВЛІННЯ КОМУНАЛЬНИМИ ПЛАТЕЖАМИ
DESIGN OF A DATA PROCESSING SYSTEM FOR
MANAGING UTILITY PAYMENTS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-222

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ЧЕУСОВ Д.О.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЧЕУСОВА Дениса Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи опрацювання даних для управління комунальними платежами. Design of a Data Processing System for Managing Utility Payments

керівник проєкту (роботи) к.т.н., доцент, КОЦУР Михайло Ігорович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 7 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 03 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис системи опрацювання даних. 4. Експлуатація, тестування та експериментальне дослідження системи опрацювання даних.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури системи опрацювання даних.	2 тиждень	Розділ 3
5	Розробка системи опрацювання даних.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження системи опрацювання даних.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Денис ЧЕУСОВ
(підпис) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Михайло КОЦУР
(підпис) (Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
99 с., 4 табл., 33 рис., 3 дод., 21 джерел.

C#, MICROSOFT VISUAL STUDIO, SQL SERVER, БАЗА ДАНИХ,
ІНФОРМАЦІЙНА СИСТЕМА, КОМУНАЛЬНІ ПЛАТЕЖІ, МОВА
ПРОГРАМУВАННЯ.

Об'єкт дослідження – алгоритми та процеси обчислень, пов'язані з організацією та опрацюванням даних для управління комунальними платежами.

Предмет дослідження – програмні системи опрацювання даних для управління комунальними платежами.

Мета роботи – розробка програмної системи опрацювання даних для управління комунальними платежами для скорочення витрат часу споживачів на перегляд та перевірку комунальних рахунків.

Матеріали, методи та технічні засоби: мова програмування C#, середовище розробки Microsoft Visual Studio, система управління базами даних SQL Server.

Результати. Розроблено проєктні рішення для створення системи опрацювання даних для управління комунальними платежами. Створено систему опрацювання даних для управління комунальними платежами. Проведено дослідження розробленої системи опрацювання даних для управління комунальними платежами.

Висновки. Мету роботи досягнуто. Розроблено систему опрацювання даних для управління комунальними платежами за допомогою мови програмування C#, середовища розробки Microsoft Visual Studio та системи управління базами даних SQL Server.

Галузь використання – комунальне господарство.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 99 pages, 4 tables, 33 figures, 3 appendixes, 21 sources.

C#, MICROSOFT VISUAL STUDIO, SQL SERVER, DATABASE, INFORMATION SYSTEM, UTILITY PAYMENTS, PROGRAMMING LANGUAGE.

The object of research is computing processes related to the organization and processing of data for managing utility payments.

The subject of the research is data processing systems for managing utility payments.

The purpose of this work is to develop the data processing system for managing utility payments to reduce the time spent by consumers on reviewing and checking utility bills.

Materials, methods and technical tools: C# programming language, Microsoft Visual Studio development environment, SQL Server database management system.

Results. Project solutions for the creation of data processing system for managing utility payments has been designed. The data processing system for managing utility payments has been developed. The study of the developed data processing system for managing utility payments has been conducted.

Conclusions. The goal of the work has been achieved. The data processing system for managing utility payments using the C# programming language, the Microsoft Visual Studio development environment, and the SQL Server database management system has been developed.

Scope of use – utilities.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	12
1.1 Системи опрацювання даних для управління комунальними платежами	12
1.2 Аналіз систем опрацювання даних для управління комунальними платежами	20
1.3 Висновки за розділом 1	29
2 Матеріали і методи	30
2.1 Вибір мови програмування	30
2.2 Вибір середовища розробки для створення системи опрацювання даних для управління комунальними платежами	33
2.3 Вибір засобів зберігання та опрацювання даних	37
2.4 Висновки за розділом 2	40
3 Опис системи опрацювання даних	41
3.1 Структура системи опрацювання даних для управління комунальними платежами	41
3.2 Функціонування системи опрацювання даних для управління комунальними платежами	45
3.3 Розробка бази даних системи опрацювання даних для управління комунальними платежами	49
3.4 Проєктування інтерфейсу системи опрацювання даних для управління комунальними платежами	52
3.5 Висновки за розділом 3	55
4 Експлуатація, тестування та експериментальне дослідження системи опрацювання даних	57
4.1 Призначення й умови застосування системи опрацювання даних	57

4.2 Характеристики системи опрацювання даних для управління комунальними платежами.....	58
4.3 Інструкція по експлуатації програми.....	59
4.3.1 Звернення до програми.....	59
4.3.2 Вхідні й вихідні дані	59
4.3.3 Повідомлення.....	60
4.4 Виконання системи опрацювання даних для управління комунальними платежами.....	60
4.5 Тестування системи опрацювання даних для управління комунальними платежами.....	65
4.6 Висновки за розділом 4	66
Висновки	67
Перелік джерел посилання	71
Додаток А Технічне завдання	73
Додаток Б Фрагмент тексту програми	78
Додаток В Слайди презентації	92

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

MVVM – Model-View-ViewModel;

БД – база даних;

ІС – інформаційна система;

КП – комунальний платіж;

СОД – система опрацювання даних.

ВСТУП

Актуальність створення системи опрацювання даних для управління комунальними платежами зумовлюється сучасними тенденціями цифрової трансформації соціально-економічних процесів, зростанням обсягів інформації та підвищенням вимог до ефективності управління фінансовими потоками у сфері житлово-комунального господарства. Умови функціонування комунальних підприємств і органів місцевого самоврядування характеризуються необхідністю оперативного обліку значної кількості платежів, своєчасного контролю за надходженням коштів, а також забезпечення прозорості взаєморозрахунків між надавачами послуг і споживачами [1], [2].

За відсутності спеціалізованих інформаційних систем процеси обробки даних часто здійснюються із застосуванням застарілих або розрізнених програмних засобів, що призводить до зростання кількості помилок, затримок в оновленні інформації та ускладнення формування достовірної звітності. Це, у свою чергу, негативно впливає на якість управлінських рішень, рівень фінансової дисципліни та довіру населення до комунальних структур. У таких умовах постає об'єктивна потреба у впровадженні єдиної системи опрацювання даних, здатної забезпечити централізований збір, зберігання, обробку та аналіз інформації про нарахування і сплату комунальних платежів [3], [4].

Важливим аспектом актуальності є зростання ролі аналітичної складової в управлінні комунальною сферою. Наявність автоматизованої системи дає змогу не лише здійснювати поточний облік платежів, а й виявляти тенденції споживання послуг, прогнозувати рівень надходжень, своєчасно ідентифікувати заборгованості та оцінювати ефективність тарифної політики. Це створює передумови для підвищення економічної обґрунтованості управлінських рішень і раціонального використання фінансових ресурсів. Не менш значущою є соціальна складова проблеми, оскільки впровадження

сучасних систем управління комунальними платежами сприяє підвищенню зручності та доступності сервісів для населення. Забезпечення оперативного доступу до актуальної інформації щодо нарахувань і платежів, а також можливість електронної взаємодії зі службами житлово-комунального господарства відповідає сучасним очікуванням користувачів і загальним напрямкам розвитку електронного врядування [4], [5].

Проте використання таких систем супроводжується низкою проблем, які потребують урахування на етапах проєктування, впровадження й супроводу. Одним із суттєвих недоліків є висока складність інтеграції системи з уже наявними інформаційними ресурсами комунальних підприємств та органів місцевого самоврядування. У багатьох випадках використовується застаріле програмне забезпечення або несумісні формати даних, що ускладнює обмін інформацією, призводить до дублювання даних і потребує додаткових витрат на доопрацювання або модернізацію існуючої інфраструктури. Проблемним аспектом також залишається питання інформаційної безпеки та захисту персональних даних. Системи управління комунальними платежами оперують значними обсягами конфіденційної інформації, що робить їх потенційно вразливими до несанкціонованого доступу, витоків даних або кібератак. Ще одним недоліком є потреба у значних фінансових і кадрових ресурсах для впровадження та супроводу таких систем. Розробка, налаштування, тестування і подальше обслуговування програмного забезпечення вимагають залучення кваліфікованих фахівців, а також стабільного фінансування. Для невеликих комунальних підприємств або територіальних громад ці витрати можуть бути надмірними [5], [6].

Крім того, впровадження автоматизованих систем часто супроводжується організаційними труднощами, зокрема необхідністю зміни усталених робочих процесів і адаптації персоналу до нових інструментів. Опір змінам, недостатній рівень цифрової компетентності працівників або користувачів може знижувати ефективність використання системи та сповільнювати досягнення очікуваних результатів [6], [7].

Тому актуальною є розробка системи опрацювання даних для управління комунальними платежами. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки системи опрацювання даних для управління комунальними платежами для скорочення витрат часу споживачів на перегляд та перевірку комунальних рахунків.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та систем опрацювання даних для управління комунальними платежами;
- здійснити проектування системи опрацювання даних для управління комунальними платежами;
- створити систему опрацювання даних для управління комунальними платежами;
- дослідити розроблену систему опрацювання даних для управління комунальними платежами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Системи опрацювання даних для управління комунальними платежами

Система опрацювання даних для управління комунальними платежами є програмно-інформаційним комплексом, призначеним для збирання, оброблення, зберігання та аналізу відомостей про споживання ресурсів і фінансові розрахунки між постачальниками послуг та споживачами. Такі системи забезпечують узгодженість облікових процесів, підвищують прозорість нарахувань і створюють технологічну основу для автоматизованого управління комунальними розрахунками в умовах цифрової трансформації [1].

В умовах активного розвитку інформаційних технологій ефективна організація процесів нарахування та контролю комунальних платежів набуває особливої значущості. Використання сучасних програмних рішень, зокрема хмарних систем опрацювання даних, дозволяє суттєво спростити операційну діяльність, зменшити витрати на обслуговування та підвищити якість взаємодії зі споживачами. Такі платформи характеризуються адаптивністю, можливістю масштабування та підтримкою аналітичних інструментів, що забезпечують доступ до актуальних показників у режимі наближеному до реального часу. У результаті змінюється сам підхід до управління комунальними підприємствами, орієнтуючи їх на більш гнучкі та даноорієнтовані моделі роботи [1], [2].

Системи опрацювання даних у сфері комунальних платежів слугують інформаційною основою для прийняття обґрунтованих управлінських рішень. Вони дозволяють автоматизувати розрахунки, уніфікувати правила нарахувань і забезпечити цілісність даних незалежно від обсягу та різноманітності послуг. Для організацій, що надають комунальні послуги, такі рішення стають інструментом підвищення ефективності, а для зацікавлених осіб — джерелом

достовірної інформації про можливості та функціональні переваги сучасних систем білінгу [1], [2].

Комунальні послуги відіграють ключову роль у забезпеченні стабільного функціонування житлової, виробничої та соціальної інфраструктури. Постачання ресурсів, таких як електроенергія, вода, газ або інші базові сервіси, потребує не лише технічної надійності, а й чітко налагоджених механізмів обліку та фінансових розрахунків. Саме системи опрацювання даних забезпечують зв'язок між фактичним споживанням ресурсів і коректним формуванням платіжних зобов'язань, що є критично важливим для сталого функціонування комунального сектору [2], [3].

За зовнішньою простотою користування комунальними послугами прихована складна інформаційна інфраструктура, яка відповідає за фіксацію обсягів споживання, їх перевірку, узагальнення та подальше нарахування платежів. У цьому контексті автоматизовані системи управління комунальними платежами виконують роль центрального елементу, що впорядковує дані та мінімізує ризики помилок. Особливої актуальності набуває застосування хмарних рішень, які дозволяють інтегрувати різні джерела інформації та забезпечувати безперервність оброблення даних [1], [2].

Традиційні підходи до управління комунальними розрахунками зазвичай ґрунтуються на періодичному зборі даних і значній частці ручних операцій. Це ускладнює оперативне реагування на зміни, підвищує ймовірність неточностей і обмежує можливості аналітики. Натомість сучасні інтелектуальні системи опрацювання даних використовують автоматизований збір показників, інтеграцію з цифровими лічильниками та інструменти аналізу великих масивів інформації. Такий підхід дозволяє оптимізувати використання ресурсів, своєчасно виявляти відхилення та покращувати якість обслуговування споживачів [1]-[3].

Розвиток інтелектуальних систем управління комунальними платежами є логічним етапом еволюції галузі, спрямованим на підвищення ефективності та прозорості процесів. Використання сучасних технологій робить комунальні

послуги більш керованими, а фінансові розрахунки — точнішими та зрозумілішими. Центральне місце в цій трансформації посідають передові системи опрацювання даних, здатні підтримувати складні операційні сценарії та відповідати зростаючим вимогам цифрового суспільства [2], [3].

Управління комунальними платежами часто стикається з проблемами неефективності, пов'язаними з тривалими циклами виставлення рахунків та затримками в отриманні платежів. У традиційних системах процеси збору показників лічильників, формування рахунків і обробки оплат здійснюються переважно вручну, що уповільнює грошові потоки та негативно впливає на задоволеність користувачів. Використання хмарних платформ дозволяє автоматизувати ці етапи, інтегруючи збір даних із лічильників, формування рахунків та процеси надсилання нагадувань про оплату або використання онлайн порталів, що значно прискорює обробку платежів та підвищує зручність для клієнтів [2], [3].

Зі збільшенням масштабів діяльності комунальних компаній традиційні системи часто виявляються неготовими до обробки великих баз даних та зростаючих обсягів інформації. Це може призводити до перевантажень, зниження швидкодії та появи операційних вузьких місць. Хмарні рішення забезпечують масштабовану обробку даних, що дозволяє швидко обробляти тисячі рахунків та підтримувати стабільну продуктивність навіть у періоди пікового навантаження, гарантуючи надійність системи [2], [4].

Сучасне програмне забезпечення для виставлення рахунків постійно адаптується до нових технологій, змін у законодавстві та очікувань користувачів. Застосування динамічного ціноутворення дозволяє враховувати попит і пропозицію у реальному часі, змінюючи вартість послуг залежно від навантаження на інфраструктуру та стимулюючи користувачів до економного споживання. Інтеграція систем виставлення рахунків із відновлюваними джерелами енергії дозволяє враховувати надлишки виробленої енергії та забезпечує прозорість у взаєморозрахунках із клієнтами, що одночасно виступають як споживачі, так і виробники [3], [4].

Інтелектуальні лічильники та пристрої Інтернету речей надають точні дані про використання ресурсів у режимі реального часу, що дозволяє своєчасно формувати рахунки та пропонувати користувачам рекомендації щодо економії. Аналітичні системи, включаючи штучний інтелект, прогнозують споживчі тенденції, виявляють потенційні порушення та допомагають формувати індивідуальні плани для клієнтів. Використання технологій блокчейн забезпечує безпечне та прозоре ведення записів про транзакції, що підвищує довіру та знижує ризики суперечок. Моделі на основі підписки або передоплати додають гнучкості, дозволяючи користувачам контролювати витрати та споживання, водночас зменшуючи адміністративне навантаження на комунальні підприємства [3], [4].

Сучасні системи опрацювання даних для управління комунальними платежами забезпечують автоматизацію процесів, підвищують ефективність роботи комунальних служб, оптимізують використання ресурсів та створюють прозорі та зручні умови для взаємодії зі споживачами [3], [4].

Програмне забезпечення для виставлення рахунків у роздрібній торгівлі енергією призначене для автоматизації складних процесів, пов'язаних із обробкою великих обсягів даних і дотриманням нормативних вимог. Такі системи дозволяють підвищити точність операцій, зменшити витрати часу та ресурсів, а також забезпечити надійність у взаємодії з клієнтами [4], [5].

Важливим аспектом роботи таких рішень є мінімізація помилок при формуванні рахунків. Автоматизація обчислень та інтеграція даних із різних джерел дозволяє зменшити ризик неправильного виставлення рахунків та запобігти фінансовим втратам. Програмне забезпечення використовує алгоритми перевірки та обробки даних, що забезпечує точність навіть при складних моделях розрахунків і складних структурах тарифів [4], [5].

Оптимізація процесів виставлення рахунків сприяє значній економії часу та ресурсів. Повторювані завдання автоматизуються, що дозволяє співробітникам зосередитися на стратегічних аспектах діяльності. Це сприяє скороченню операційних витрат та підвищує ефективність роботи компанії.

Дотримання нормативних вимог є невід'ємною частиною роботи роздрібних постачальників енергії. Сучасні системи автоматично враховують зміни в законодавстві та оновлюють правила розрахунків, забезпечуючи відповідність нормативам без додаткового ручного контролю. Це гарантує, що роздрібні продавці енергії завжди відповідають актуальним регіональним стандартам і правилам [4], [5].

Ефективна робота системи передбачає інтеграцію даних від операторів розподільчих систем, що дозволяє отримати єдину картину споживання та формувати точні рахунки. Агрегація даних та їх обробка в реальному часі забезпечує прозорість для постачальників та клієнтів, одночасно спрощуючи операційні процеси [4], [5].

Масштабованість є ще однією ключовою характеристикою сучасного програмного забезпечення. Система повинна ефективно обробляти збільшені обсяги даних та тисячі рахунків, зберігаючи стабільну продуктивність навіть у періоди пікового навантаження. Це робить її придатною як для компаній, що розвиваються, так і для вже усталених роздрібних продавців енергії [4], [5].

Використання хмарних технологій забезпечує безперебійну інтеграцію з іншими корпоративними системами, такими як CRM та бухгалтерські платформи. Хмарна архітектура гарантує доступність даних, можливість швидкого масштабування та ефективну взаємодію між різними підрозділами компанії. Системи, що повністю працюють у хмарі, надають гнучкі інструменти інтеграції через API, що сприяє створенню єдиного, узгодженого робочого процесу для роздрібних продавців енергії [5], [6].

Хмарні програмні рішення для виставлення рахунків за комунальні послуги забезпечують значні переваги як для постачальників, так і для їхніх клієнтів. Такі системи здатні ефективно масштабуватися, обробляючи зростаючі бази споживачів та великі обсяги даних без потреби у додатковому обладнанні, що робить їх придатними для компаній різного розміру. Вони дозволяють суттєво скоротити витрати, оскільки не потребують дорогого

устаткування та обслуговування, а модель оплати за фактичне використання дає змогу контролювати фінансові ресурси [5], [6].

Автоматизація процесів виставлення рахунків мінімізує людські помилки, що забезпечує точне формування рахунків та зменшує кількість конфліктів із клієнтами. Хмарні платформи дозволяють отримувати доступ до актуальних даних у реальному часі, надаючи споживачам можливість контролювати власне споживання, а постачальникам — оперативно реагувати на зміни. Завдяки високій гнучкості системи можна адаптувати під конкретні тарифи, регіональні вимоги та внутрішні робочі процеси компанії, а інтеграція з CRM та ERP забезпечує єдиний робочий процес [5], [6].

Використання клієнтських порталів підвищує рівень обслуговування, дозволяючи переглядати рахунки, здійснювати платежі та відслідковувати споживання. Автоматичні нагадування, сповіщення та інтуїтивні інтерфейси роблять процес взаємодії з системою зручним і доступним. Хмарні рішення скорочують цикли виставлення рахунків, автоматизуючи зчитування показань лічильників, формування рахунків та збір платежів, що позитивно впливає на грошовий потік [5], [6].

Сучасні платформи забезпечують відповідність нормативним вимогам та високий рівень безпеки даних через шифрування та багатофакторну автентифікацію. Потужні інструменти аналітики дозволяють отримувати детальні звіти про використання ресурсів, доходи та ефективність операцій, допомагаючи приймати обґрунтовані рішення. Можливість віддаленого доступу підтримує співпрацю та дистанційну роботу команд, а швидке розгортання систем дозволяє впроваджувати нові інструменти без простоїв, забезпечуючи плавну інтеграцію в бізнес-процеси [5], [6].

Впровадження програмного рішення для виставлення рахунків за комунальні послуги є важливим кроком для підвищення продуктивності та покращення взаємодії з клієнтами. Цей процес передбачає ретельне планування, інтеграцію з наявними системами та навчання персоналу для забезпечення безперебійної роботи. На першому етапі проводиться оцінка

потреб організації та виявлення слабких місць у поточних процесах, таких як помилки у формуванні рахунків, затримки або невідповідність нормативам. Формується дорожня карта, що визначає цілі, терміни та критерії ефективності впровадження. Детальний аналіз існуючих робочих процесів, структури даних і ресурсів дозволяє адаптувати рішення до специфічних вимог і очікувань організації [5], [6].

Наступним кроком є інтеграція нового програмного забезпечення з наявними платформами, включаючи CRM, бухгалтерські системи та дані від операторів розподільчих систем. Цей етап включає налаштування потоків даних, забезпечення безпечних з'єднань та сумісності систем, що дозволяє уникнути перебоїв, підтримує точність даних і зберігає безперервність операцій [5], [6].

Навчання персоналу є завершальним етапом, який забезпечує ефективне використання нового рішення. Проводяться адаптовані навчальні сесії для різних ролей користувачів, а також надаються ресурси для самостійного опанування системи, включаючи посібники, відеоуроки та довідкові матеріали. Постійна підтримка, яка включає допомогу у вирішенні технічних проблем та оновлення програмного забезпечення, гарантує довгострокове ефективне використання рішення та здатність організації адаптуватися до змін [5], [6].

Хмарні платформи для виставлення рахунків демонструють високу універсальність і можуть обслуговувати різні види комунальних послуг, включаючи електроенергію, воду, газ, управління відходами та телекомунікації. Вони особливо ефективні для компаній з великими клієнтськими базами, складними моделями білінгу або гнучкими тарифними структурами, а також для постачальників відновлюваної енергії, які потребують обліку мережевої генерації та кредитування надлишків енергії [5], [6].

Безпека даних клієнтів у таких системах є пріоритетом. Використовуються шифрування, багатофакторна автентифікація та

дотримання міжнародних стандартів захисту, що забезпечує конфіденційність інформації та зменшує ризики несанкціонованого доступу. Регулярні оновлення програмного забезпечення дозволяють системам залишатися стійкими до нових загроз, а надійний постачальник гарантує стабільність та високий рівень захисту даних [5], [6].

Програмні рішення для управління комунальними послугами розділяються на три основні напрями: керування рахунками за послуги, управління активами та організація польового обслуговування. Сучасні системи виставлення рахунків зосереджуються на управлінні рахунками та доходами, забезпечуючи комплексну автоматизацію процесів для постачальників енергії та комунальних підприємств і дозволяючи відповідати сучасним потребам ринку та готуватися до майбутніх викликів [5], [6].

Протягом тривалого часу програмні системи сприяли масштабуванню діяльності, розширенню спектра послуг, переходу до багатопрофільної моделі та впровадженню нових продуктів, таких як зарядні станції для електромобілів або зелені тарифи. Системи також дозволяють реалізовувати програми підвищення енергоефективності, що стає особливо актуальним у сучасних умовах. Деякі подібні програмні системи забезпечують автоматизацію відповідних процесів завдяки використанню машинного навчання та штучного інтелекту, що дозволяє опрацьовувати складні бізнес-процеси та запобігати відтоку клієнтів або проблемним контрактам. Система покращує взаємодію з користувачами, підвищує гнучкість тарифних моделей і підтримує інтеграцію розподілених джерел енергії, включно з нетто-обліком та «зеленими» тарифами [5], [6].

Визначено, що система опрацювання даних для управління комунальними платежами є програмно-інформаційним комплексом, призначеним для збирання, оброблення, зберігання та аналізу відомостей про споживання ресурсів і фінансові розрахунки між постачальниками послуг та споживачами. Такі програмні системи забезпечують узгодженість облікових процесів, підвищують прозорість нарахувань і створюють технологічну

основу для управління комунальними розрахунками в умовах цифрової трансформації.

1.2 Аналіз систем опрацювання даних для управління комунальними платежами

Проаналізуємо системи опрацювання даних для управління комунальними платежами [7]-[15].

Система опрацювання даних для управління комунальними платежами Banyon Utility Billing (рис. 1.1) орієнтована на комплексну підтримку процесів нарахування та обліку послуг для різних категорій споживачів [7], [9], [13].

Utility Billing Account Screen CITY OF MEDFORD bds

First Prev Next Last Add New Delete Lookup Reports [Right-click here for more options](#)

Complaint

Sort by Service Address

Account 01 00007440 00 4

Serv Address 1008 LAKE ST NE

Billing Name JOHNSON, BARBARA

Owner Name JOHNSON, BARBARA

BookStopSub 6 320

PARCEL #

BALANCE \$17.19

Credits \$0.00

Status Active Type

Budget Each Period \$0.00

Units 5

Billing Owner Property Names Aging Misc Service Meter Surcharge Memo Trans ProRate Mtr Hist

Billing

Owner

Property

Names

Aging

Misc

Service

Meter

Meter Hist

Surcharge

Memo

Transaction

ProRate

Person Commercial

Name BARBARA JOHNSON

Lookup JOHNSON, BARBARA

Attn

Addr 8217 38TH AVE N

Addr

City NEW HOPE State MN Zip 55427

Country

Home Phone

Work Phone

Cell Phone

Carrier Route

DP Code 08

SSN

Print Label Print Envelope View Map

Billing	Owner	Property	Names	Aging	Misc	Service	Meter	Surcharge	MEMO	Trans	ProRate	Mtr Hist
NOT Used	0											
USED	1	Water OV #1				\$208.00	\$0.00	30 Days				
NOT Used	2	Water OV #2										
NOT Used	3	Water OV #3										
NOT Used	4	Water OV #4										
NOT Used	5	Water OV #5										
NOT Used	10	Water OV Combo										
NOT Used	11	SUB METER										
NOT Used	20	Summer Water										
NOT Used	22	Water AD										
NOT Used	25	Sprinkler HD										
NOT Used	28	OTHER CHRG										
USED	30	MIN TEST FEE				\$1.40	\$0.00	30 MIN TEST FEE				
NOT Used	50	RE CUST SERV										
NOT Used	51	CUS SER TAPS										
NOT Used	45	SPROTH CIST										

Account Meter History

Account Number 11 00003200 00 5 Service 11 Water Usage res

Calc Usage Actual Usage Readings and Dates

PERIOD	USAGE	EST
1	4521	
2	3578	
3	4094	
4	5490	
5	20927	
6	35666	
7	14300	
8	21000	
9	5317	
10	4170	
11	3549	
12	3899	
Total	126911	
Average	10576	

Рисунок 1.1 – Програмна система Banyon Utility Billing [12]

Її функціональні можливості дозволяють обробляти кілька видів комунальних сервісів у межах одного облікового запису, що забезпечує зручне управління взаємопов'язаними послугами та узгодженими циклами нарахувань. Передбачена підтримка як обліку за показниками лічильників, так і розрахунків без їх використання, що робить систему придатною для різних моделей споживання. У межах нарахувань реалізовано механізми додаткових платежів, фінансових коригувань і застосування санкцій, завдяки чому забезпечується висока точність та відповідність сформованих рахунків встановленим правилам [7], [9], [13].

Важливою особливістю системи є розвинений інструментарій формування аналітичних матеріалів, який інтегрований безпосередньо у програмне середовище. На основі централізованих даних про нарахування та споживання послуг створюються узагальнені та деталізовані звіти, що можуть бути адаптовані до специфічних вимог користувачів. Підтримка різних форматів представлення результатів спрощує подальше використання звітної інформації для аналізу, зберігання або обміну даними. У сукупності це забезпечує ефективне управління комунальними платежами та підвищує прозорість фінансових процесів [7], [9], [13].

Основними характеристиками програмного забезпечення Banyon Utility Billing є такі [7], [9], [13]:

- гнучка підтримка комунальних послуг: програмне забезпечення Banyon Utility Billing забезпечує облік кількох типів комунальних сервісів у межах одного клієнтського профілю, що дозволяє поєднувати різні схеми нарахувань і коректно обробляти складні платіжні сценарії [7], [9], [13];

- масштабованість облікових процесів: система Banyon Utility Billing адаптована до використання як у невеликих організаціях, так і в структурах з великою кількістю споживачів, зберігаючи стабільність роботи при зростанні обсягів даних [7], [9], [13];

- автоматизація нарахувань і коригувань: у Banyon Utility Billing реалізовано механізми автоматичного розрахунку платежів, застосування додаткових зборів, штрафів і фінансових уточнень, що мінімізує ручне втручання та знижує ризик помилок [7], [9], [13];

- розвинені засоби звітності: програмний продукт Banyon Utility Billing містить вбудований інструментарій формування аналітичних матеріалів, який дозволяє створювати стандартні та користувацькі звіти на основі узагальнених і деталізованих даних [7], [9], [13];

- інтеграція з офісними форматами: передбачена можливість експорту підготовлених звітів у поширені формати документів для подальшого аналізу, зберігання або передавання зацікавленим сторонам [7], [9], [13];

- централізоване зберігання інформації: усі відомості щодо клієнтів, послуг і нарахувань об'єднуються в єдиному інформаційному просторі, що спрощує доступ до даних і підвищує їх узгодженість [7], [9], [13].

Серед недоліків програмного забезпечення Banyon Utility Billing можна відзначити відносну складність первинного налаштування та впровадження, що може потребувати залучення спеціалізованих фахівців, а також обмежену гнучкість інтерфейсу для кінцевих користувачів без додаткової адаптації під специфічні вимоги організації [7], [9], [13].

Система опрацювання даних для управління комунальними платежами Energy & Utilities Cloud (рис. 1.2) позиціонується як комплексне програмне рішення, орієнтоване на підтримку діяльності енергетичних і комунальних організацій. У межах платформи реалізовано засоби, що забезпечують централізоване управління процесами нарахування платежів, контролю стану інфраструктури та обробки поточних операційних даних. Функціональність програмного забезпечення Energy & Utilities Cloud спрямована на підвищення прозорості роботи сервісів і спрощення внутрішніх бізнес-процесів, що дозволяє більш ефективно координувати взаємодію між різними підрозділами компанії [8], [14].

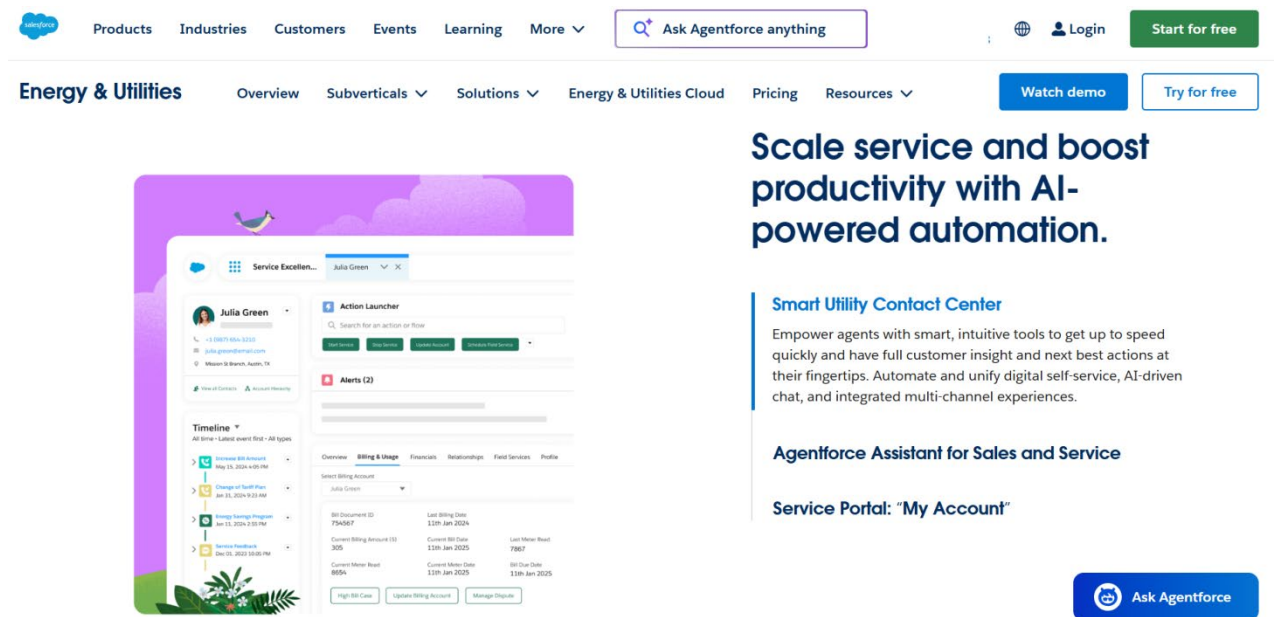


Рисунок 1.2 – Програмна система Energy & Utilities Cloud [14]

Особлива увага приділяється роботі з даними в оперативному режимі, що створює умови для своєчасного виявлення відхилень у роботі обладнання та попередження можливих збоїв у наданні послуг. Аналітичні інструменти системи забезпечують глибокий аналіз інформації про споживання ресурсів, формуючи основу для обґрунтованих управлінських рішень і стратегічного планування. Водночас програмне забезпечення підтримує автоматизацію взаємодії з клієнтами, спрощуючи обробку звернень, запитів і фінансових операцій, пов'язаних з оплатою рахунків. Завдяки впровадженню Energy & Utilities Cloud досягається підвищення операційної ефективності, раціональніше використання ресурсів і покращення якості обслуговування споживачів [8], [14].

Програмне забезпечення Energy & Utilities Cloud має такі особливості [8], [14]:

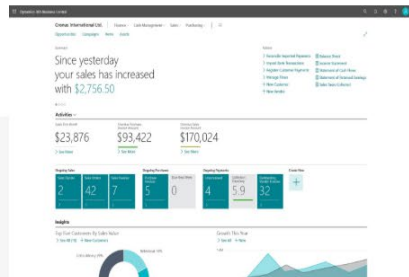
- хмарна архітектура: програмне забезпечення Energy & Utilities Cloud функціонує на базі хмарних технологій, що забезпечує віддалений доступ до даних, масштабованість ресурсів і зниження витрат на локальну інфраструктуру [8], [14];

- управління білінгом: реалізовано засоби автоматизованого нарахування платежів і контролю фінансових операцій, що підвищує точність розрахунків та спрощує обробку рахунків клієнтів [8], [14];
- моніторинг у реальному часі: система Energy & Utilities Cloud підтримує постійний збір і аналіз операційних даних, що дозволяє оперативно відстежувати стан мереж і реагувати на відхилення [8], [14];
- прогнозне обслуговування: у Energy & Utilities Cloud використовуються аналітичні механізми для виявлення потенційних проблем у роботі обладнання до виникнення аварійних ситуацій [8], [14];
- аналітика та звітність: у Energy & Utilities Cloud передбачено інструменти для аналізу споживання ресурсів і формування звітів, які допомагають приймати обґрунтовані управлінські рішення [8], [14];
- підтримка взаємодії з клієнтами: програмне забезпечення Energy & Utilities Cloud полегшує обробку звернень, запитів і інформації про рахунки, покращуючи якість сервісу [8], [14].

Серед недоліків програмного забезпечення Energy & Utilities Cloud можна відзначити залежність від стабільного Інтернет з'єднання, що може впливати на доступність системи, а також складність первинного впровадження та налаштування для організацій з нетиповими бізнес-процесами. Крім того, використання хмарної моделі може вимагати додаткових витрат на адаптацію персоналу та інтеграцію з наявними інформаційними системами [8], [14].

Система опрацювання даних для управління комунальними платежами SkyBill Utility Billing (рис. 1.3) позиціонується як хмарне програмне рішення, орієнтоване на організації, що надають комунальні послуги, керують об'єктами нерухомості або виконують муніципальні функції. Вона створена для використання в середовищі, де важливо поєднати процеси нарахування платежів із можливостями корпоративного управління, забезпечуючи цілісність фінансових і операційних даних [11], [15].

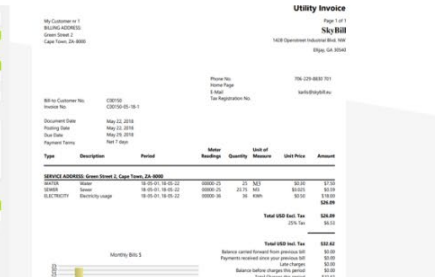
UTILITY BILLING SOFTWARE

Complete Meter-to-Cash solution based on Microsoft Dynamics 365 Business Central



Client Information System including Customer Portal and Mobile App



Customizable to your needs- Invoice Layout, Reports, and Business Logic

Рисунок 1.3 – Програмна система SkyBill Utility Billing [15]

Основою платформи є екосистема Microsoft Dynamics Business Central, що дозволяє інтегрувати білінг із іншими управлінськими модулями в межах єдиного інформаційного простору [11], [15].

Опрацювання рахунків у SkyBill Utility Billing охоплює повний цикл від фіксації споживання до отримання оплати, незалежно від типу комунальної послуги. Система забезпечує автоматизацію зчитування показників, формування нарахувань та подальшу обробку фінансових операцій, зменшуючи ручну працю та ризик помилок. Гнучка логіка тарифів дає змогу адаптувати розрахунки до особливостей конкретного споживача, об'єкта або умов постачання, що підвищує точність і прозорість взаєморозрахунків [11], [15].

Важливою складовою рішення є інтегрований механізм управління інформацією про клієнтів, який підтримує централізоване зберігання та актуалізацію даних, а також спрощує комунікацію між постачальником і споживачем. Для кінцевих користувачів передбачено веб-середовище самообслуговування, через яке можна переглядати стан облікового запису,

отримувати рахунки та здійснювати платежі. Аналітичні та звітні інструменти дозволяють узагальнювати інформацію про споживання і фінансові потоки, що сприяє прийняттю обґрунтованих управлінських рішень [11], [15].

SkyBill Utility Billing найбільш ефективно проявляє себе в організаціях, які прагнуть використовувати єдину платформу для корпоративного управління та білінгу, підтримуючи різноманітні комунальні послуги й складні тарифні моделі. Разом із цим впровадження системи може потребувати попереднього ознайомлення з принципами роботи середовища Microsoft Dynamics, а також детального планування налаштувань для точного узгодження з індивідуальними бізнес-процесами [11], [15].

Програмне забезпечення SkyBill Utility Billing має такі особливості [11], [15]:

- хмарна архітектура: система опрацювання даних для управління комунальними платежами SkyBill Utility Billing функціонує як хмарне рішення, що забезпечує централізований доступ до даних, масштабованість та зменшення витрат на локальну інфраструктуру [11], [15];

- інтеграція з ERP: система SkyBill Utility Billing побудована на основі платформи Microsoft Dynamics Business Central, що дозволяє поєднувати білінг комунальних послуг із фінансовим, бухгалтерським та операційним управлінням у межах єдиного середовища [11], [15];

- автоматизація білінгу: у SkyBill Utility Billing реалізовано повний цикл обробки платежів від збору показників до формування рахунків і контролю оплат з мінімальним втручанням користувача [11], [15];

- підтримка різних послуг: програмне забезпечення SkyBill Utility Billing орієнтоване на роботу з кількома видами комунальних ресурсів, що дозволяє застосовувати його в багатопрофільних організаціях [11], [15];

- управління клієнтськими даними: інтегрований CRM-модуль забезпечує збереження, актуалізацію та аналіз інформації про споживачів і історію взаємодії з ними [11], [15];

– портал самообслуговування: у SkyBill Utility Billing надається вебдоступ для клієнтів з можливістю перегляду рахунків, контролю стану облікового запису та виконання платежів [11], [15];

– аналітика та звітність: система SkyBill Utility Billing містить інструменти для формування звітів і аналізу фінансових та споживчих даних з метою підтримки управлінських рішень [11], [15].

Серед недоліків програмного забезпечення SkyBill Utility Billing можна відзначити залежність від екосистеми Microsoft, що може ускладнювати впровадження для організацій без досвіду роботи з цією платформою, а також потребу в детальному попередньому налаштуванні для точного узгодження системи з індивідуальними бізнес-процесами та внутрішніми регламентами [11], [15].

Порівняльну характеристику систем опрацювання даних для управління комунальними платежами наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння систем опрацювання даних для управління комунальними платежами

Критерій порівняння	Banyon Utility Billing [13]	Energy & Utilities Cloud [14]	SkyBill Utility Billing [15]
зручність автоматизованого нарахування оплат	+	+-	+-
інтеграція з ERP-системами	+-	+	+
масштабованість для великих організацій	+-	+	+-
невисока вартість використання	+-	+-	-
аналітика та звітність	+-	+-	+
хмарне розгортання	+-	+	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато систем опрацювання даних для управління комунальними платежами. Проте використання таких систем супроводжується низкою проблем, які потребують урахування на етапах проєктування, впровадження й супроводу. Одним із суттєвих недоліків є висока складність інтеграції системи з уже наявними інформаційними ресурсами комунальних підприємств та органів місцевого самоврядування. У багатьох випадках використовується застаріле програмне забезпечення або несумісні формати даних, що ускладнює обмін інформацією, призводить до дублювання даних і потребує додаткових витрат на доопрацювання або модернізацію існуючої інфраструктури. Проблемним аспектом також залишається питання інформаційної безпеки та захисту персональних даних. Системи управління комунальними платежами оперують значними обсягами конфіденційної інформації, що робить їх потенційно вразливими до несанкціонованого доступу, витоків даних або кібератак. Ще одним недоліком є потреба у значних фінансових і кадрових ресурсах для впровадження та супроводу таких систем. Тому актуальною є розробка системи опрацювання даних для управління комунальними платежами.

При розробці системи опрацювання даних для управління комунальними платежами необхідно забезпечити такі функціональні вимоги:

- підтримка можливості роботи різних користувачів, зокрема, адміністраторів та звичайних користувачів;
- підтримка можливості адміністратора вносити зміни в тарифи комунальних послуг;
- можливість реєстрації та авторизації користувачів;
- можливість ознайомлення з інформацією про баланс рахунку оплат за комунальні послуги;
- підтримка можливості отримання інформації про сплачені користувачем суми коштів за комунальні послуги;
- підтримка можливості роботи з особистим кабінетом користувача.

1.3 Висновки за розділом 1

Визначено, що система опрацювання даних для управління комунальними платежами є програмно-інформаційним комплексом, призначеним для збирання, оброблення, зберігання та аналізу відомостей про споживання ресурсів і фінансові розрахунки між постачальниками послуг та споживачами. Такі програмні системи забезпечують узгодженість облікових процесів, підвищують прозорість нарахувань і створюють технологічну основу для управління комунальними розрахунками.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для управління комунальними платежами. Проте використання таких систем супроводжується низкою проблем, які потребують урахування на етапах проектування, впровадження й супроводу. Одним із суттєвих недоліків є висока складність інтеграції системи з уже наявними інформаційними ресурсами комунальних підприємств та органів місцевого самоврядування. У багатьох випадках використовується застаріле програмне забезпечення або несумісні формати даних, що ускладнює обмін інформацією, призводить до дублювання даних і потребує додаткових витрат на доопрацювання або модернізацію існуючої інфраструктури. Проблемним аспектом також залишається питання інформаційної безпеки та захисту персональних даних. Системи управління комунальними платежами оперують значними обсягами конфіденційної інформації, що робить їх потенційно вразливими до несанкціонованого доступу, витоків даних або кібератак. Ще одним недоліком є потреба у значних фінансових і кадрових ресурсах для впровадження та супроводу таких систем. Тому актуальною є розробка системи опрацювання даних для управління комунальними платежами.

Сформульовано функціональні вимоги до системи опрацювання даних для управління комунальними платежами.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці системи опрацювання даних для управління комунальними платежами було використано мову програмування C# [16], [17].

Мова програмування C# є не тільки інструментом реалізації прикладної логіки, а і складовою цілісної програмної платформи, орієнтованої на розробку довготривалих і супроводжуваних інформаційних систем. Її розвиток відбувається з урахуванням сучасних вимог до програмної інженерії, що забезпечує актуальність мови в умовах швидких технологічних змін і постійного зростання складності програмних продуктів [16], [17].

Важливою характеристикою мови C# є поєднання строгої типізації з високим рівнем абстракції, що сприяє зменшенню кількості логічних і програмних помилок ще на етапі компіляції. Такий підхід позитивно впливає на якість програмного коду та спрощує його аналіз і тестування, що особливо актуально для систем, які обробляють критично важливі фінансові дані. Завдяки цьому підвищується передбачуваність поведінки програмного забезпечення та знижується ризик виникнення прихованих помилок під час експлуатації [16], [17].

Суттєвим аспектом є підтримка сучасних мовних конструкцій, спрямованих на підвищення виразності коду та зменшення його обсягу без втрати зрозумілості. Це дозволяє підвищити продуктивність розробки та спростити колективну роботу над проектом, оскільки код стає легшим для сприйняття й супроводу іншими розробниками. У контексті командної розробки це має особливе значення для забезпечення узгодженості стилю програмування та дотримання єдиних архітектурних підходів [16], [17].

Окремої уваги заслуговує підтримка мови C# з боку широкої спільноти та наявність розвиненої системи документації й навчальних матеріалів. Це спрощує процес освоєння мови, прискорює вирішення технічних проблем і

сприяє впровадженню перевірених практик розробки. Наявність великої кількості готових рішень і прикладів коду дозволяє зменшити часові витрати на реалізацію типових функціональних компонентів [16], [17].

Таким чином, C# характеризується не лише технічними перевагами, а й високим рівнем зрілості як інструменту програмної інженерії. Її використання створює сприятливі умови для розробки надійних, структурованих і довготривало експлуатованих програмних систем, що відповідають сучасним вимогам до якості, підтримуваності та масштабованості програмного забезпечення [16], [17].

Доцільність використання мови програмування C# для розробки системи опрацювання даних для управління комунальними платежами обумовлюється сукупністю технологічних, архітектурних і експлуатаційних чинників, що відповідають вимогам до сучасних інформаційних систем управлінського призначення. Зазначена мова програмування широко застосовується для створення корпоративних і інформаційно-аналітичних рішень, орієнтованих на роботу з великими обсягами структурованих даних та забезпечення високого рівня надійності програмного забезпечення [16], [17].

Однією з ключових переваг C# є тісна інтеграція з платформою .NET, яка надає розвинену екосистему бібліотек і фреймворків для роботи з базами даних, мережевими протоколами, користувацькими інтерфейсами та засобами безпеки. Це дозволяє реалізувати повний цикл опрацювання даних, починаючи від їх збирання і зберігання та завершуючи аналічною обробкою і формуванням звітності, без залучення значної кількості сторонніх інструментів. У результаті підвищується цілісність архітектури системи та спрощується її супровід [16], [17].

Важливим аргументом на користь використання мови програмування C# є підтримка об'єктно-орієнтованої та компонентної парадигм програмування, що сприяє побудові масштабованих і модульних програмних рішень. Це має особливе значення для систем управління комунальними платежами, які часто розвиваються поступово та потребують розширення

функціональності, інтеграції з новими сервісами або адаптації до змін у нормативно-правовій базі. Завдяки чіткій структуризації коду забезпечується можливість ефективного повторного використання компонентів і зменшення складності подальших модифікацій [16], [17].

Не менш значущою є наявність вбудованих механізмів забезпечення безпеки, зокрема засобів керування доступом, шифрування та обробки виняткових ситуацій. З огляду на те, що системи управління комунальними платежами оперують фінансовими та персональними даними, використання мови програмування, орієнтованої на створення безпечних застосунків, є критично важливим. Платформа .NET також надає інструменти для реалізації вимог захисту даних відповідно до чинних стандартів і регуляторних обмежень [16], [17].

Крім того, C# характеризується високим рівнем продуктивності у поєднанні з автоматичним керуванням пам'яттю, що знижує ризик виникнення помилок, пов'язаних з некоректним використанням ресурсів. Це сприяє стабільній роботі системи в умовах обробки значних обсягів транзакційних даних та одночасного доступу великої кількості користувачів. Додатковою перевагою є кросплатформенність сучасних реалізацій .NET, що дозволяє розгорнути систему на різних операційних платформах без суттєвих змін у коді [16], [17].

Таким чином, використання мови програмування C# для розробки системи опрацювання даних для управління комунальними платежами є обґрунтованим з позицій надійності, масштабованості, безпеки та зручності подальшого супроводу. Зазначені властивості роблять C# ефективним інструментом для створення комплексних інформаційних систем, орієнтованих на довготривалу експлуатацію у сфері житлово-комунального господарства.

Обґрунтування вибору мови програмування для розробки системи опрацювання даних для управління комунальними платежами наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки системи опрацювання даних для управління комунальними платежами

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	Ruby
Розвинені засоби роботи з базами даних	+	+	+—
Швидкість розробки прикладних рішень	+	+—	+—
Підтримка безпеки та захисту даних	+	+	+—
Масштабованість і модульність архітектури	+	+—	—
Інтеграція з вебсервісами та API	+	+	+—

Отже, для реалізації системи опрацювання даних для управління комунальними платежами обрано мову програмування C#, яка підтримує об'єктно-орієнтовану та компонентну парадигми програмування, що сприяє побудові масштабованих і модульних програмних рішень. Крім того, C# характеризується високим рівнем продуктивності у поєднанні з автоматичним керуванням пам'яттю, що знижує ризик виникнення помилок, пов'язаних з некоректним використанням ресурсів. Це сприяє стабільній роботі системи в умовах обробки значних обсягів транзакційних даних та одночасного доступу великої кількості користувачів.

2.2 Вибір середовища розробки для створення системи опрацювання даних для управління комунальними платежами

В якості середовища розробки для створення системи опрацювання даних для управління комунальними платежами було обрано середовище Microsoft Visual Studio [18], [19].

Середовище розробки Microsoft Visual Studio являє собою комплексну платформу, що поєднує інструменти для розробки, аналізу та оптимізації програмного забезпечення в єдиному робочому просторі. Його використання сприяє формуванню впорядкованого процесу створення програмних систем, у межах якого забезпечується узгодженість архітектурних рішень і дотримання загальноприйнятих стандартів якості коду [18], [19].

Важливою характеристикою Visual Studio є орієнтація на підтримку сучасних підходів до проєктування програмних систем, зокрема багаторівневої та сервісно-орієнтованої архітектури. Завдяки цьому створюються передумови для розробки гнучких і масштабованих рішень, здатних адаптуватися до змін у бізнес-логіці або нормативних вимогах без суттєвого перероблення програмного коду. Така властивість є особливо актуальною для інформаційних систем, що експлуатуються протягом тривалого часу [18], [19].

Окремої уваги заслуговує підтримка автоматизації рутинних процесів розробки, зокрема генерації шаблонів проєктів, керування залежностями та налаштування середовища виконання. Це знижує ймовірність помилок конфігурації та сприяє уніфікації підходів до створення програмних рішень у межах одного проєкту або організації. У результаті підвищується відтворюваність процесів і спрощується передача проєкту між розробниками [18], [19].

Таким чином, Microsoft Visual Studio характеризується не лише як інструмент написання програмного коду, а як повноцінне середовище підтримки інженерного циклу створення програмного забезпечення. Його застосування створює сприятливі умови для розробки структурованих, надійних і довготривало експлуатованих інформаційних систем, що

відповідають сучасним вимогам до якості, стабільності та керованості програмних продуктів [18], [19].

Доцільність використання середовища розробки Microsoft Visual Studio для створення системи опрацювання даних з управління комунальними платежами визначається його функціональною повнотою, високим рівнем інтеграції інструментів та орієнтацією на розробку складних інформаційних систем корпоративного рівня. Зазначене середовище забезпечує підтримку повного життєвого циклу програмного забезпечення, починаючи від етапу проєктування і реалізації та завершуючи тестуванням, налагодженням і подальшим супроводом програмного продукту [18], [19].

Важливою перевагою Visual Studio є глибока інтеграція з платформою .NET і мовою програмування C#, що дозволяє максимально ефективно використовувати можливості сучасних бібліотек, фреймворків і сервісів для роботи з базами даних, вебінтерфейсами та мережевими протоколами. Завдяки цьому спрощується реалізація багаторівневої архітектури системи, забезпечується чіткий поділ між рівнями представлення, бізнес-логіки та доступу до даних, а також підвищується структурованість і читабельність програмного коду [18], [19].

Суттєве значення має наявність у Visual Studio розвинених засобів автоматичного аналізу коду, налагодження та тестування. Інструменти статичного аналізу дозволяють виявляти потенційні помилки та вразливості ще на ранніх етапах розробки, що є критично важливим для систем, пов'язаних з обробкою фінансової та персональної інформації. Можливості інтерактивного відлагодження та модульного тестування сприяють підвищенню надійності програмного забезпечення і зменшенню витрат часу на усунення помилок [18], [19].

Окрему увагу слід приділити підтримці колективної розробки та керування версіями. Visual Studio надає вбудовані засоби інтеграції з системами контролю версій і управління проєктами, що спрощує координацію роботи команди розробників, контроль змін у коді та забезпечення

узгодженості програмних компонентів. Це є особливо актуальним для довготривалих проєктів, які передбачають поетапне розширення функціональних можливостей системи [18], [19].

Таким чином, використання середовища розробки Microsoft Visual Studio є обґрунтованим вибором для створення системи опрацювання даних для управління комунальними платежами, оскільки воно забезпечує високий рівень продуктивності розробки, надійність програмного продукту та зручність його подальшого супроводу. Поєднання потужних інструментів, підтримки сучасних технологій і орієнтації на корпоративні рішення робить Visual Studio ефективною платформою для реалізації інформаційних систем у сфері житлово-комунального господарства.

Обґрунтування вибору середовища розробки для створення системи опрацювання даних для управління комунальними платежами наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення системи опрацювання даних для управління комунальними платежами

Критерій порівняння середовищ розробки	Середовища розробки		
	Microsoft Visual Studio	MonoDevelop	SharpDevelop
Повнота підтримки мови C# і платформи .NET	+	+-	+-
Засоби проєктування та керування архітектурою	+	+-	-
Інтегровані інструменти тестування та налагодження	+	+-	+-
Зручність роботи з базами даних	+	-	-

Продуктивність і зручність розробки	+	+—	+—
-------------------------------------	---	----	----

Таким чином, для створення системи опрацювання даних для управління комунальними платежами обрано середовище розробки Microsoft Visual Studio, що характеризується функціональною повнотою, високим рівнем інтеграції інструментів та орієнтацією на розробку складних інформаційних систем корпоративного рівня. Зазначене середовище забезпечує підтримку повного життєвого циклу програмного забезпечення, починаючи від етапу проєктування і реалізації та завершуючи тестуванням, налагодженням і подальшим супроводом програмного продукту.

2.3 Вибір засобів зберігання та опрацювання даних

Для зберігання та опрацювання даних було обрано систему управління базами даних SQL Server [20], [21].

Доцільність використання системи управління базами даних SQL Server для зберігання та опрацювання даних у системі управління комунальними платежами визначається її орієнтацією на роботу з критично важливою інформацією, високими вимогами до надійності та потребою у стабільній підтримці транзакційних процесів. Функціональні можливості цієї СУБД дозволяють забезпечити цілісність фінансових і облікових даних, що є ключовою умовою коректного функціонування інформаційних систем у сфері житлово-комунального господарства [20], [21].

Важливим аргументом на користь SQL Server є підтримка розвинених механізмів транзакційної обробки, які гарантують узгодженість даних навіть у разі одночасної роботи великої кількості користувачів. Це має особливе значення для систем, у яких паралельно виконуються операції нарахування, оплати та коригування комунальних платежів. Забезпечення атомарності,

узгодженості, ізольованості та надійності операцій створює передумови для запобігання втраті або спотворенню інформації [20], [21].

Суттєвою перевагою SQL Server є інтеграція з платформою .NET та мовою програмування C#, що спрощує реалізацію доступу до даних і зменшує складність програмної взаємодії між рівнем зберігання та прикладною логікою. Використання вбудованих засобів роботи з запитамі, процедур і функцій дозволяє частину обчислювальних операцій перенести безпосередньо на рівень бази даних, підвищуючи загальну продуктивність системи та знижуючи навантаження на сервер застосунків [20], [21].

Особливістю SQL Server є наявність потужних інструментів для виконання складних запитів, агрегування та аналітичної обробки даних, що дозволяє реалізувати інтелектуальні механізми управління та прогнозування фінансових потоків у комунальній сфері. Завдяки цьому досягається висока швидкість обробки запитів навіть при значних навантаженнях та великій кількості одночасних користувачів [20], [21].

Крім того, Microsoft SQL Server надає можливості для автоматизації обслуговування баз даних, включаючи резервне копіювання, моніторинг продуктивності та оптимізацію запитів. Це сприяє зменшенню ризиків втрати даних, підвищує стабільність роботи системи та дозволяє забезпечити безперервний доступ до інформації [20], [21].

Окремої уваги заслуговують можливості SQL Server у сфері забезпечення безпеки даних. Підтримка механізмів розмежування доступу, шифрування та аудиту операцій дозволяє реалізувати багаторівневу систему захисту інформації відповідно до вимог чинного законодавства щодо обробки персональних і фінансових даних. Це є критично важливим для систем управління комунальними платежами, які працюють з інформацією про споживачів і їхні фінансові зобов'язання [20], [21].

Таким чином, використання SQL Server у системі опрацювання даних для управління комунальними платежами є обґрунтованим вибором з огляду на надійність, масштабованість, високий рівень безпеки та зручність інтеграції

з іншими компонентами програмного забезпечення. Сукупність цих характеристик забезпечує стабільну роботу системи та створює умови для її ефективної експлуатації і подальшого розвитку.

Обґрунтування вибору засобів зберігання та опрацювання даних наведено у таблиці 2.3.

Таблиця 2.3 – Обґрунтування вибору засобів зберігання та опрацювання даних

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	Microsoft SQL Server	PostgreSQL	Oracle Database
Інтеграція з .NET та C#	+	+-	+-
Засоби аналітики та формування звітності	+	+-	+
Продуктивність при одночасній роботі багатьох користувачів	+	+-	+
Масштабованість та обробка великих обсягів даних	+	+-	+
Невисока вартість використання	+	+	-

Таким чином, для зберігання та опрацювання даних обрано систему управління базами даних Microsoft SQL Server завдяки поєднанню високої надійності, інтеграції з технологіями Microsoft, потужних засобів безпеки та розширених інструментів для аналітики і адміністрування. Суттєвою перевагою SQL Server є інтеграція з платформою .NET та мовою програмування C#, що спрощує реалізацію доступу до даних і зменшує

складність програмної взаємодії між рівнем зберігання та прикладною логікою.

2.4 Висновки за розділом 2

Для реалізації системи опрацювання даних для управління комунальними платежами обрано мову програмування C#, яка підтримує об'єктно-орієнтовану та компонентну парадигми програмування, що сприяє побудові масштабованих і модульних програмних рішень. Крім того, C# характеризується високим рівнем продуктивності у поєднанні з автоматичним керуванням пам'яттю, що знижує ризик виникнення помилок, пов'язаних з некоректним використанням ресурсів. Це сприяє стабільній роботі системи в умовах обробки значних обсягів транзакційних даних та одночасного доступу великої кількості користувачів.

Для створення системи опрацювання даних для управління комунальними платежами обрано середовище розробки Microsoft Visual Studio, що характеризується функціональною повнотою, високим рівнем інтеграції інструментів та орієнтацією на розробку складних інформаційних систем корпоративного рівня. Зазначене середовище забезпечує підтримку повного життєвого циклу програмного забезпечення, починаючи від етапу проєктування і реалізації та завершуючи тестуванням, налагодженням і подальшим супроводом програмного продукту.

Для зберігання та опрацювання даних обрано систему управління базами даних Microsoft SQL Server завдяки поєднанню високої надійності, інтеграції з технологіями Microsoft, потужних засобів безпеки та розширених інструментів для аналітики і адміністрування. Суттєвою перевагою SQL Server є інтеграція з платформою .NET та мовою програмування C#, що спрощує реалізацію доступу до даних і зменшує складність програмної взаємодії між рівнем зберігання та прикладною логікою.

3 ОПИС СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

3.1 Структура системи опрацювання даних для управління комунальними платежами

Структуру системи опрацювання даних для управління комунальними платежами наведено на рис. 3.1.

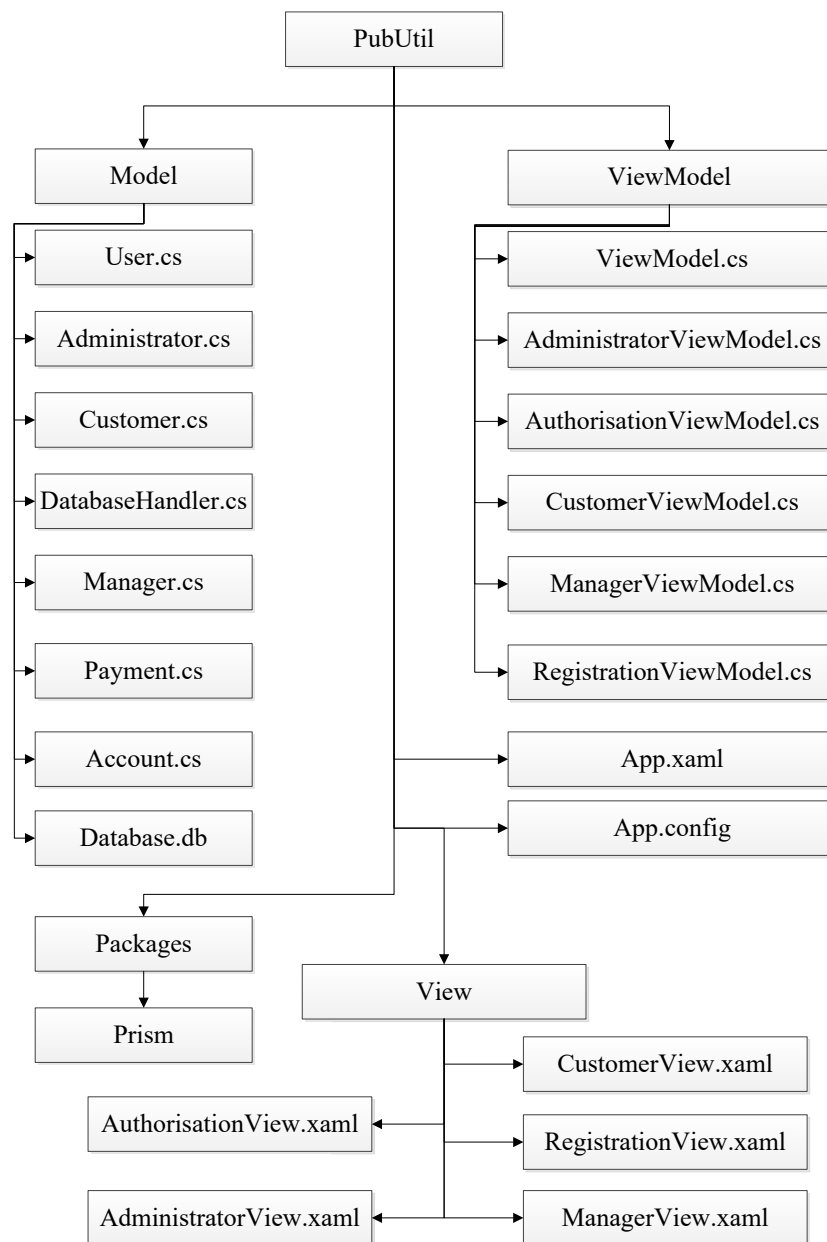


Рисунок 3.1 – Структура системи опрацювання даних для управління комунальними платежами

Структура системи опрацювання даних для управління комунальними платежами будується на основі архітектурного шаблону MVVM (Model-View-ViewModel), що забезпечує чітке розмежування відповідальностей між логікою обробки даних, логікою взаємодії з користувачем та візуальним поданням інформації. Такий підхід сприяє підвищенню підтримуваності програмного забезпечення, спрощує тестування та створює умови для подальшого розширення функціональних можливостей системи без суттєвого впливу на її загальну структуру.

Модельний рівень системи орієнтований на опис предметної області та інкапсуляцію бізнес-логіки, пов'язаної з управлінням користувачами, рахунками та платежами. Узагальнена модель користувача використовується для представлення спільних характеристик усіх ролей, що функціонують у системі, з подальшою спеціалізацією для адміністратора, менеджера та споживача комунальних послуг. Адміністративна модель забезпечує можливість керування тарифами та обліковими записами персоналу, тоді як модель користувача зосереджена на операціях перегляду та опрацювання власних нарахунків і платежів. Менеджерська модель призначена для роботи з даними щодо обсягів споживання послуг. Окреме місце займають моделі, відповідальні за збереження та обробку інформації про платежі й особові рахунки, а також базовий компонент доступу до бази даних, який інкапсулює механізми взаємодії з системою зберігання даних.

Рівень моделей уявлень виконує роль проміжної ланки між моделями та користувацьким інтерфейсом і реалізує логіку взаємодії з даними, необхідну для підтримки дій користувача. Моделі уявлень не залежать від конкретних елементів інтерфейсу та не наслідують класи візуального рівня, а забезпечують надання властивостей і команд, до яких здійснюється прив'язка з боку уявлень. Повідомлення про зміну стану даних реалізується через стандартні інтерфейси сповіщення, що дозволяє автоматично оновлювати інтерфейс у відповідь на зміни в логіці застосунку. У межах цього рівня

здійснюється підготовка даних до відображення, координація взаємодії з моделями, а також реалізація перевірки коректності введеної інформації. Крім того, у моделях уявлень можуть визначатися логічні стани, які надалі відображаються у візуальному представленні системи.

Файл `User.cs` використовується для опису узагальненої сутності користувача та містить базові властивості, спільні для всіх ролей системи, що забезпечує єдиний підхід до роботи з обліковими записами. Файл `Administrator.cs` призначений для реалізації логіки адміністративної ролі та відповідає за операції керування тарифами й реєстрацію менеджерів у системі. У файлі `Customer.cs` зосереджено функціональність, пов'язану з користувачами комунальних послуг, зокрема обробку даних про особові рахунки, нарахування та платежі. Файл `Manager.cs` реалізує модель менеджера, яка забезпечує роботу з даними щодо використання комунальних послуг та їх внесення до системи. Компонент `DatabaseHandler.cs` виконує роль базового шару доступу до даних і інкапсулює механізми взаємодії з базою даних. Файл `Payment.cs` використовується для збереження та обробки інформації про платежі, включаючи їх параметри та стан. Файл `Account.cs` призначений для представлення особового рахунку, у межах якого здійснюється облік наданих комунальних послуг і пов'язаних фінансових операцій.

Візуальний рівень відповідає за формування зовнішнього вигляду застосунку та визначення структури інтерфейсу користувача. Уявлення пов'язані зі своїми моделями уявлень через механізм контексту даних, що дозволяє елементам інтерфейсу взаємодіяти з властивостями і командами без прямого звернення до логіки обробки даних. Основна логіка застосунку не розміщується у коді уявлень, який, як правило, обмежується ініціалізацією компонентів, за винятком випадків, коли необхідно реалізувати специфічну візуальну поведінку, складну для опису декларативними засобами. Уявлення також можуть налаштовувати правила прив'язки даних, форматування та додаткову перевірку введених значень, а також визначати анімаційні ефекти й переходи, що покращують сприйняття інтерфейсу.

Файл `ViewModel.cs` виконує функцію базової моделі уявлення та містить спільні механізми сповіщення про зміни стану даних. У файлі `AdministratorViewModel.cs` реалізується логіка підготовки даних для інтерфейсу адміністратора та координація його взаємодії з моделями. Файл `AuthorisationViewModel.cs` відповідає за обробку процесів авторизації користувачів і перевірку облікових даних. У файлі `CustomerViewModel.cs` зосереджено логіку подання даних для користувача комунальних послуг, що забезпечує перегляд рахунків, оплату та аналіз історії платежів. Файл `ManagerViewModel.cs` призначений для підготовки даних, необхідних менеджеру для введення та коригування інформації про споживання послуг. Файл `RegistrationViewModel.cs` реалізує логіку реєстрації нових облікових записів і перевірки коректності введених даних.

Завершальним елементом структури є конфігураційний рівень, представлений файлом налаштувань застосунку, у якому зосереджено параметри підключення до бази даних та інші системні характеристики. Це дозволяє відокремити конфігураційні дані від програмного коду та спростити зміну середовища виконання без необхідності модифікації логіки системи.

Файли уявлень `AdministratorView.xaml`, `CustomerView.xaml`, `ManagerView.xaml`, `AuthorisationView.xaml` та `RegistrationView.xaml` визначають структуру і зовнішній вигляд відповідних екранів користувацького інтерфейсу та забезпечують прив'язку елементів управління до властивостей і команд моделей уявлень. Файл `App.config` використовується для зберігання конфігураційних параметрів застосунку, зокрема налаштувань підключення до бази даних і шляхів до необхідних ресурсів, що забезпечує гнучкість розгортання та супроводу системи.

У сукупності така структура забезпечує цілісність, гнучкість і масштабованість системи опрацювання даних для управління комунальними платежами.

3.2 Функціонування системи опрацювання даних для управління комунальними платежами

Функціонування системи опрацювання даних для управління комунальними платежами подамо за допомогою схеми, зображеної на рис. 3.2.

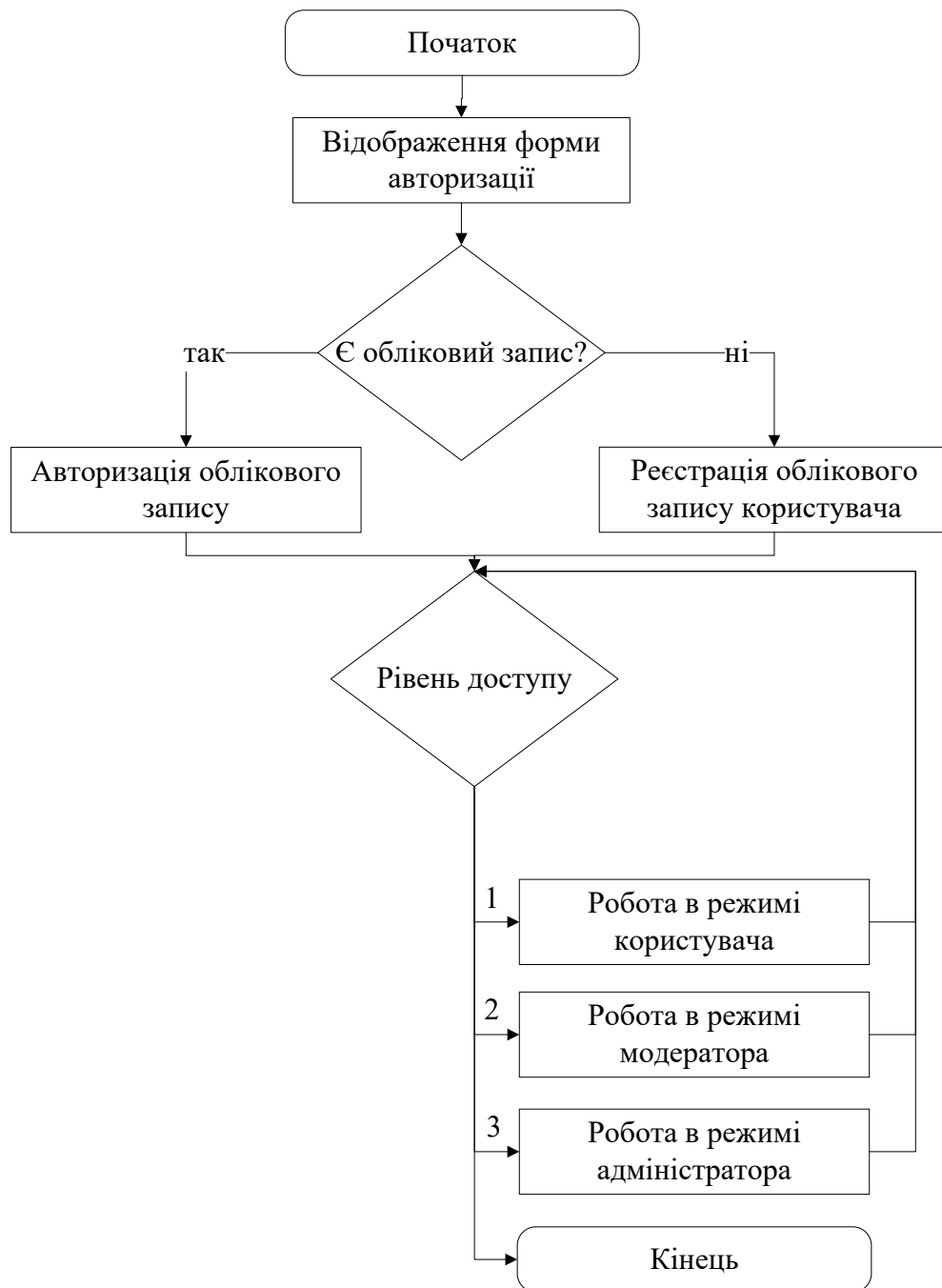


Рисунок 3.2 – Схема функціонування системи опрацювання даних для управління комунальними платежами

Функціонування системи опрацювання даних для управління комунальними платежами відбувається за послідовним сценарієм, який забезпечує контроль доступу та коректну організацію роботи користувачів відповідно до їхніх повноважень. Початковий етап роботи системи пов'язаний з ініціалізацією програмного середовища та підготовкою інтерфейсу до взаємодії з користувачем. Після запуску застосунку автоматично відбувається перехід до інтерфейсу автентифікації, призначеного для введення облікових даних.

На наступному етапі здійснюється перевірка наявності облікового запису у системі, що передбачає звернення до бази даних та аналіз введеної інформації. У разі підтвердження існування відповідного облікового запису виконується процедура авторизації, під час якої встановлюється особа користувача та визначається його роль у системі. Якщо ж облікові дані відсутні або користувач не зареєстрований, активується механізм створення нового облікового запису, який забезпечує внесення необхідної інформації та її збереження для подальшого використання.

Після успішного завершення етапу авторизації або реєстрації система переходить до визначення рівня доступу, який безпосередньо впливає на подальшу логіку функціонування застосунку. Залежно від встановлених прав користувач отримує доступ до відповідного режиму роботи, що обмежує або розширює набір доступних функцій. Такий підхід дозволяє забезпечити розмежування відповідальності між різними категоріями користувачів і запобігти несанкціонованому доступу до критично важливих даних або адміністративних операцій.

Завершальним етапом є виконання системою основних операцій у межах обраного режиму, після чого робота застосунку може бути коректно завершена. Така логіка функціонування забезпечує впорядкований доступ до ресурсів системи, підвищує рівень безпеки та створює зрозумілий і послідовний процес взаємодії користувача з програмою.

Діаграму прецедентів системи опрацювання даних для управління комунальними платежами подамо на рис. 3.3.

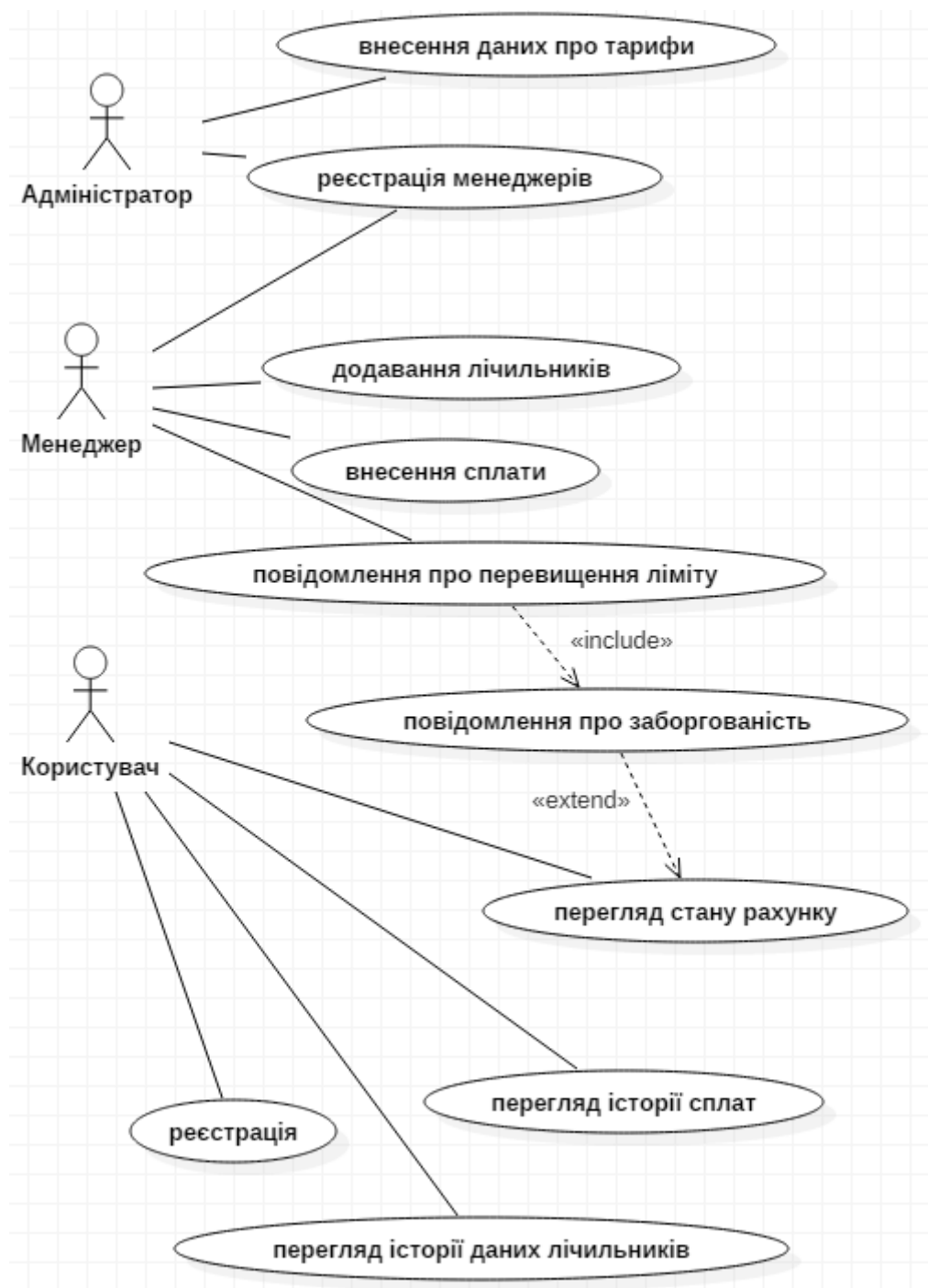


Рисунок 3.3 – Діаграма прецедентів системи опрацювання даних для управління комунальними платежами

Діаграма прецедентів системи опрацювання даних для управління комунальними платежами відображає функціональні можливості системи

через взаємодію з основними категоріями користувачів та демонструє розподіл відповідальності між ними. У межах діаграми визначено три актори, кожен з яких виконує чітко окреслені дії відповідно до наданих повноважень і ролі в системі.

Адміністратор представлений як суб'єкт, що здійснює керування ключовими параметрами функціонування системи. Його взаємодія із системою зосереджена на внесенні інформації про тарифи, які використовуються для розрахунку комунальних платежів, а також на створенні та обліку облікових записів менеджерів. Така роль адміністратора забезпечує централізоване управління довідковими та організаційними даними, що є основою коректної роботи всієї системи.

Менеджер виступає проміжною ланкою між адміністративним рівнем і кінцевими користувачами, виконуючи операції, пов'язані з наповненням системи фактичними даними. У межах діаграми його діяльність охоплює роботу з лічильниками, фіксацію інформації про споживання послуг і внесення даних про здійснені оплати. Окрему роль відіграє механізм інформування, за якого виявлення перевищення встановлених лімітів автоматично пов'язується з формуванням повідомлення про наявну заборгованість. У свою чергу, повідомлення про заборгованість логічно розширює процес перегляду стану рахунку, що дозволяє користувачеві отримати додаткову інформацію про фінансові зобов'язання.

Користувач системи представлений як споживач комунальних послуг, взаємодія якого спрямована на отримання інформації та контроль власних витрат. У межах діаграми відображено можливість перегляду поточного стану особового рахунку, аналізу історії здійснених оплат та ознайомлення з архівними даними показників лічильників. Окремо передбачено виконання дій, пов'язаних зі створенням облікового запису, що забезпечує доступ користувача до функціональних можливостей системи.

Таким чином, діаграма прецедентів наочно відображає логіку взаємодії різних категорій користувачів із системою опрацювання даних для управління

комунальними платежами, демонструє взаємозв'язки між окремими функціями та підкреслює роль кожного актора у забезпеченні цілісності й ефективності функціонування системи.

3.3 Розробка бази даних системи опрацювання даних для управління комунальними платежами

База даних системи опрацювання даних для управління комунальними платежами спроектована з урахуванням особливостей предметної області та функціональних вимог, сформульованих у процесі дослідження. Для її реалізації було обрано систему керування базами даних Microsoft SQL Server, що зумовлено високою швидкістю при обробці значних обсягів інформації, надійними механізмами захисту даних, а також зручністю адміністрування й експлуатації. Використання даної СУБД дозволяє забезпечити стабільну роботу програмного комплексу, цілісність інформації та ефективну взаємодію з програмним середовищем на базі операційних систем сімейства Windows, з якими SQL Server має глибоку інтеграцію. Окрім цього, дана платформа розглядається не лише як засіб зберігання даних, а як повноцінне середовище для управління структурованими та різнорідними інформаційними ресурсами, що є важливим для побудови сучасних інформаційних систем.

Структура бази даних побудована на основі взаємопов'язаних сутностей, кожна з яких відображає окремий аспект функціонування системи управління комунальними платежами. Центральне місце займає інформація про користувачів, яка зберігається у відповідній таблиці та містить ідентифікаційні дані, контактну інформацію, облікові реквізити для доступу до системи та відомості про рівень повноважень. Така організація дозволяє реалізувати розмежування доступу між адміністраторами, менеджерами та звичайними користувачами, що підвищує безпеку та керованість системи.

Схему бази даних системи опрацювання даних для управління комунальними платежами наведено на рис. 3.4.

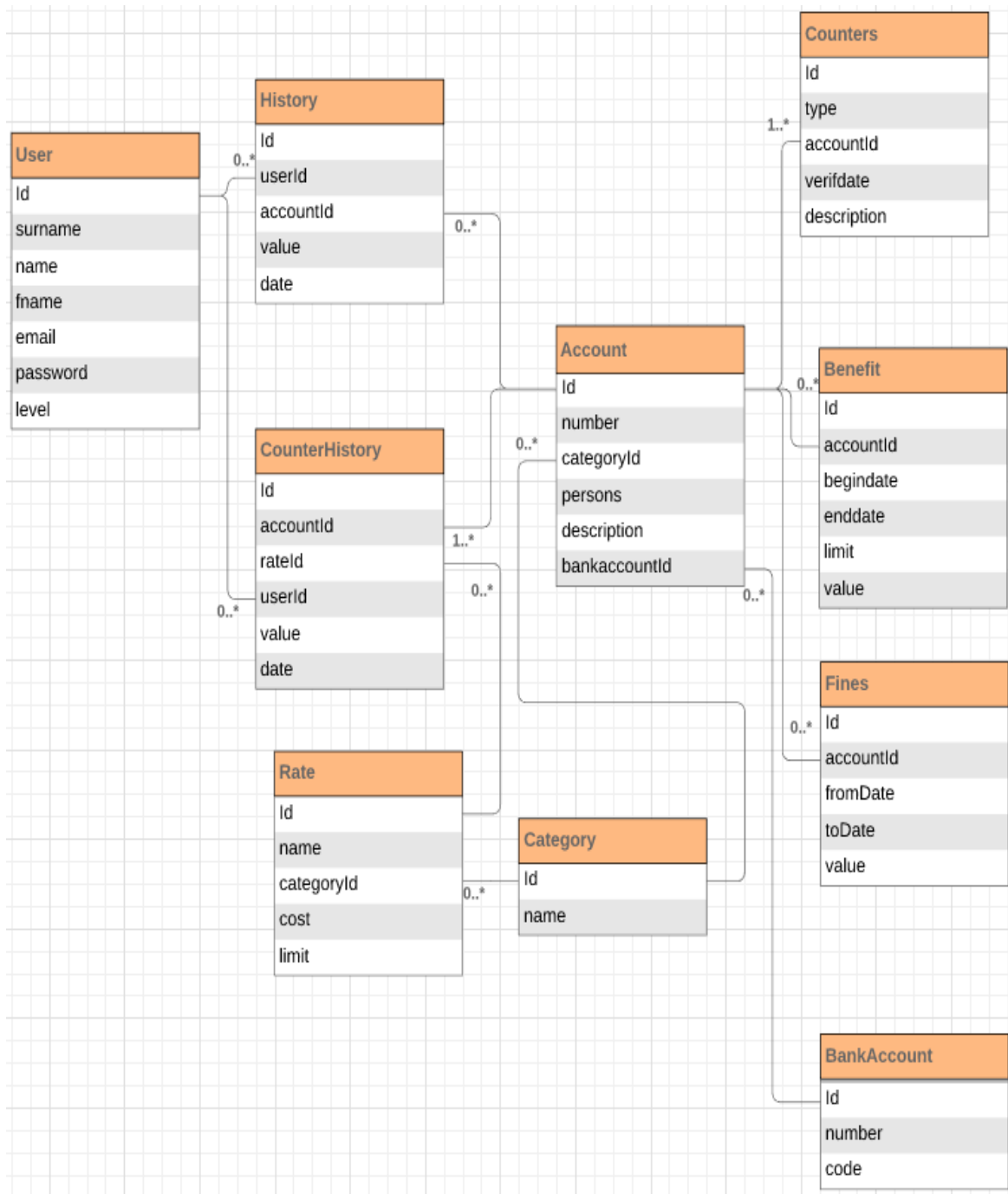


Рисунок 3.4 – Схема бази даних системи опрацювання даних для управління комунальними платежами

Для обліку фінансових взаєморозрахунків використовується таблиця особових рахунків, у якій зберігаються унікальні номери рахунків, зв'язок із конкретними видами комунальних послуг, кількість споживачів та допоміжна описова інформація. Кожен особовий рахунок пов'язується з банківським рахунком, призначеним для здійснення платежів, що забезпечує коректне

відображення платіжних реквізитів та спрощує процес оплати. Дані про банківські рахунки зберігаються окремо, що сприяє нормалізації структури бази даних і зменшенню дублювання інформації.

Окремий блок бази даних призначений для роботи з лічильниками комунальних послуг та фіксації їх показників. Для цього передбачено таблиці, які зберігають інформацію про типи лічильників, їх прив'язку до особових рахунків, дати перевірок, а також історію внесення показників із зазначенням користувача, який здійснив введення, тарифу та дати фіксації. Такий підхід дозволяє відслідковувати динаміку споживання ресурсів у часі та забезпечує прозорість розрахунків.

Фінансові операції відображаються у таблиці історії оплат, де накопичуються відомості про здійснені платежі, їх розмір, дату проведення та зв'язок із відповідними користувачами й рахунками. Це дає змогу системі опрацювання даних для управління комунальними платежами формувати аналітичні звіти, контролювати стан розрахунків і надавати користувачам доступ до детальної інформації про виконані платежі. Додатково в системі передбачено зберігання даних про тарифи та категорії комунальних послуг, що дозволяє гнучко налаштовувати правила нарахувань залежно від виду послуги, вартості одиниці ресурсу та встановлених лімітів.

Для врахування соціальних та економічних особливостей користувачів у базі даних системи опрацювання даних для управління комунальними платежами реалізовано механізм обліку пільг, який містить інформацію про строки їх дії, допустимі обсяги споживання та розмір знижок. Паралельно з цим зберігаються дані про штрафні санкції, що нараховуються у разі несвоєчасної оплати, із зазначенням періоду нарахування та суми пені. Така структура дозволяє автоматизувати процес фінансових розрахунків і забезпечити повноту обліку всіх складових комунальних платежів.

Загалом база даних системи опрацювання даних для управління комунальними платежами побудована за реляційним принципом із чітко визначеними зв'язками між таблицями, що забезпечує логічну цілісність

інформації, зручність обробки даних та можливість подальшого розширення функціональності системи без суттєвих змін її архітектури.

3.4 Проєктування інтерфейсу системи опрацювання даних для управління комунальними платежами

Проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для управління комунальними платежами виконано з урахуванням вимог технічного завдання. При розробленні системи опрацювання даних для управління комунальними платежами враховані функціональні вимоги до програми:

- підтримка можливості роботи різних користувачів, зокрема, адміністраторів та звичайних користувачів;
- підтримка можливості адміністратора вносити зміни в тарифи комунальних послуг;
- можливість реєстрації та авторизації користувачів;
- можливість ознайомлення з інформацією про баланс рахунку оплат за комунальні послуги;
- підтримка можливості отримання інформації про сплачені користувачем суми коштів за комунальні послуги;
- підтримка можливості роботи з особистим кабінетом користувача.

Інтерфейс головної форми системи опрацювання даних для управління комунальними платежами при роботі в режимі адміністратора наведено на рис. 3.5.

Інтерфейс головної форми системи опрацювання даних для управління комунальними платежами при роботі в режимі звичайного користувача наведено на рис. 3.6.

Адміністратор

Тарифи Менеджери Особистий кабінет

Тариф	Послуга	Вартість	Граничний обсяг
Мінімальний	Електроенергія	2.16	150
Максимальний	Електроенергія	4.32	10000
Тариф на газ	Газопостачання	8.55	100000

Додати Видалити Підтвердити Скасувати

Рисунок 3.5 – Інтерфейс головної форми системи при роботі в режимі адміністратора

Користувач

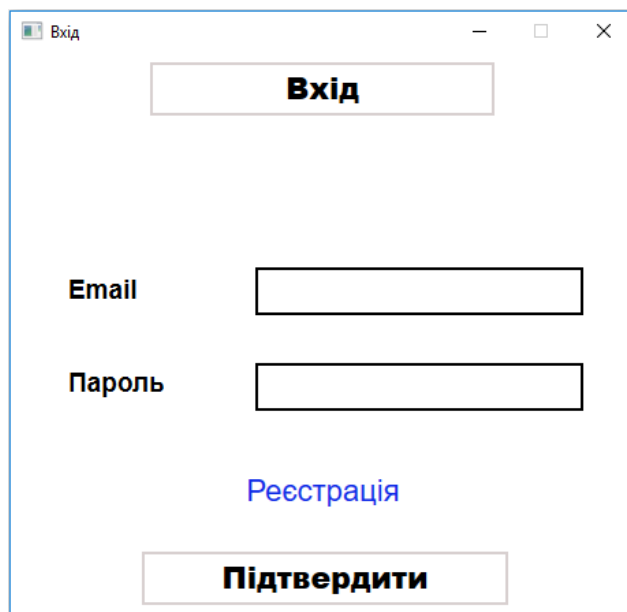
Рахунки Сплати Лічильники Повідомлення Особистий кабінет

Послуга	Тариф	Баланс
Електроенергія	Мінімальний	-10.5

Номер рахунку 22331741 Знайти

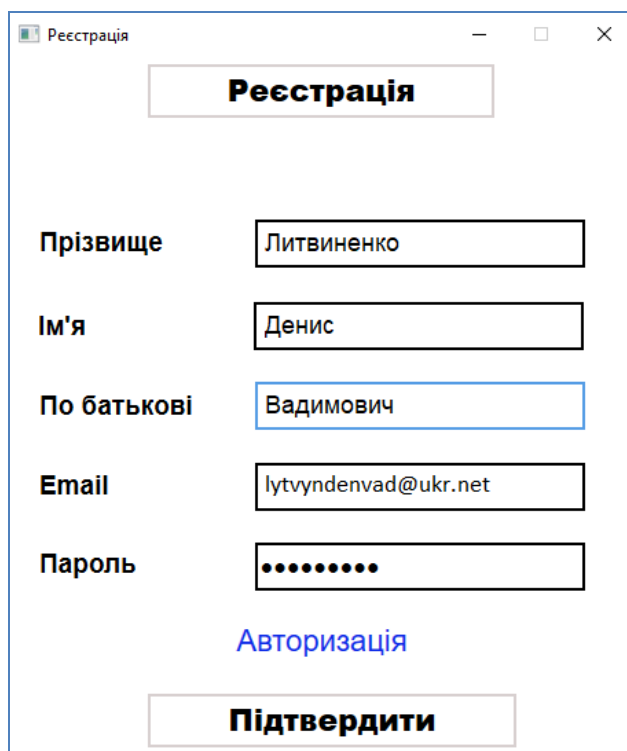
Рисунок 3.6 – Інтерфейс головної форми системи при роботі в режимі звичайного користувача

Інтерфейс форми авторизації у системі опрацювання даних для управління комунальними платежами наведено на рис. 3.7. Форму реєстрації наведено на рис. 3.8.



The screenshot shows a web browser window titled "Вхід". Inside the window, there is a header box with the word "Вхід". Below it, there are two input fields: "Email" and "Пароль". Below the "Пароль" field, there is a blue link labeled "Реєстрація". At the bottom, there is a button labeled "Підтвердити".

Рисунок 3.7 – Інтерфейс форми авторизації у системі опрацювання даних для управління комунальними платежами



The screenshot shows a web browser window titled "Реєстрація". Inside the window, there is a header box with the word "Реєстрація". Below it, there are five input fields: "Прізвище" (filled with "Литвиненко"), "Ім'я" (filled with "Денис"), "По батькові" (filled with "Вадимович"), "Email" (filled with "lytyndenvad@ukr.net"), and "Пароль" (filled with "....."). Below the "Пароль" field, there is a blue link labeled "Авторизація". At the bottom, there is a button labeled "Підтвердити".

Рисунок 3.8 – Форма реєстрації

3.5 Висновки за розділом 3

Розроблено систему опрацювання даних для управління комунальними платежами.

Запропоновано структуру системи опрацювання даних для управління комунальними платежами, що побудована на основі архітектурного шаблону MVVM. Це забезпечує чітке розмежування відповідальностей між логікою обробки даних, логікою взаємодії з користувачем та візуальним поданням інформації. Такий підхід сприяє підвищенню підтримуваності програмного забезпечення, спрощує тестування та створює умови для подальшого розширення функціональних можливостей системи без суттєвого впливу на її загальну структуру.

Описано функціонування системи опрацювання даних для управління комунальними платежами, що відбувається за послідовним сценарієм, який забезпечує контроль доступу та коректну організацію роботи користувачів відповідно до їхніх повноважень. Така логіка функціонування забезпечує впорядкований доступ до ресурсів системи, підвищує рівень безпеки та створює зрозумілий і послідовний процес взаємодії користувача з програмою.

Побудовано діаграму прецедентів системи опрацювання даних для управління комунальними платежами, яка відображає функціональні можливості системи через взаємодію з основними категоріями користувачів та демонструє розподіл відповідальності між ними. У межах діаграми визначено три актори (користувач, менеджер та адміністратор), кожен з яких виконує чітко окреслені дії відповідно до наданих повноважень і ролі в системі.

Розроблено базу даних системи опрацювання даних для управління комунальними платежами, яка складається з взаємопов'язаних сутностей, кожна з яких відображає окремий аспект функціонування системи управління комунальними платежами. Центральне місце займає інформація про користувачів, яка зберігається у відповідній таблиці та містить ідентифікаційні дані, контактну інформацію, облікові реквізити для доступу до системи та

відомості про рівень повноважень. Така організація дозволяє реалізувати розмежування доступу між адміністраторами, менеджерами та звичайними користувачами, що підвищує безпеку та керованість системи.

Виконано проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для управління комунальними платежами.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

4.1 Призначення й умови застосування системи опрацювання даних

Система опрацювання даних призначена для підтримки процесу управління комунальними платежами. Система опрацювання даних написана на мові C#, яка підтримує об'єктно-орієнтовану та компонентну парадигми програмування, що сприяє побудові масштабованих і модульних програмних рішень. В якості середовища розробки для створення системи опрацювання даних для управління комунальними платежами обрано Visual Studio Microsoft, що характеризується функціональною повнотою, високим рівнем інтеграції інструментів та орієнтацією на розробку складних інформаційних систем корпоративного рівня.

Система опрацювання даних для управління комунальними платежами забезпечує виконання таких функцій:

- підтримка можливості роботи різних користувачів, зокрема, адміністраторів та звичайних користувачів;
- підтримка можливості адміністратора вносити зміни в тарифи комунальних послуг;
- можливість реєстрації та авторизації користувачів;
- можливість ознайомлення з інформацією про баланс рахунку оплат за комунальні послуги;
- підтримка можливості отримання інформації про сплачені користувачем суми коштів за комунальні послуги;
- підтримка можливості роботи з особистим кабінетом користувача.

Для роботи системи опрацювання даних для управління комунальними платежами потрібна наявність персонального комп'ютера з процесором із тактовою частотою не нижче двох гігагерц, що підтримує сучасні інструкції

обробки даних, а також за наявності оперативної пам'яті обсягом не менше чотирьох гігабайт, що дозволяє стабільно виконувати операції з обробки та зберігання інформації. Для встановлення програми та бази даних необхідно передбачити вільне місце на жорсткому диску або твердотільному накопичувачі обсягом не менше декількох гігабайт. Робота програмного забезпечення орієнтована на операційні системи сімейства Windows c, які підтримують середовище виконання платформи .NET та забезпечують сумісність із системою керування базами даних Microsoft SQL Server. Для розгортання та адміністрування бази даних необхідна встановлена відповідна версія SQL Server разом із засобами керування, що дозволяють виконувати налаштування, резервне копіювання та контроль цілісності даних. Додатково передбачається наявність стабільного мережевого з'єднання у випадку використання клієнт-серверної архітектури, а також встановлення стандартних системних бібліотек і драйверів, необхідних для коректної взаємодії програмних компонентів між собою та з апаратною платформою.

Функціональні характеристики розробленої системи опрацювання даних для управління комунальними платежами наведено у технічному завданні (додаток А). Фрагмент тексту програмного коду системи опрацювання даних для управління комунальними платежами наведено у додатку Б.

4.2 Характеристики системи опрацювання даних для управління комунальними платежами

Система опрацювання даних для управління комунальними платежами характеризується комплексним підходом до організації інформаційних процесів, що забезпечує цілісність, узгодженість і надійність обробки фінансових та облікових відомостей. Архітектура системи орієнтована на розмежування функціональних обов'язків між користувачами, що дозволяє підтримувати різні рівні доступу та гарантувати захист критично важливої

інформації від несанкціонованих дій. Внутрішні механізми опрацювання даних забезпечують автоматизацію розрахунків, контроль коректності введених значень і зменшення впливу людського фактору на результати роботи. Інформація зберігається у структурованому вигляді з підтримкою логічних зв'язків між об'єктами, що сприяє швидкому пошуку, аналізу та формуванню узагальнених відомостей. Система адаптована до змін умов надання комунальних послуг, що дозволяє оперативно оновлювати тарифи, правила нарахувань і параметри обліку без порушення стабільності функціонування. Значна увага приділяється зручності взаємодії з користувачем, що проявляється у зрозумілій організації інтерфейсу та підтримці типових сценаріїв роботи. Загалом система орієнтована на підвищення ефективності управління комунальними платежами, забезпечення прозорості фінансових операцій і створення надійного інформаційного середовища для всіх учасників процесу.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску системи опрацювання даних для управління комунальними платежами необхідно запустити її виконуваний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до системи опрацювання даних для управління комунальними платежами є інформація, що надходить від користувачів і уповноважених працівників у процесі взаємодії з програмою, зокрема персональні відомості облікових записів, параметри особових рахунків, значення показників лічильників, дані про споживання комунальних послуг, відомості щодо тарифів, пільг і умов нарахувань, а також інформація про здійснені платежі.

Вихідними даними системи опрацювання даних для управління комунальними платежами є сформовані на основі обробки результати, які відображають поточний стан особових рахунків, суми нарахувань і оплат, інформацію про заборгованість або переплату, узагальнені відомості щодо споживання комунальних ресурсів, а також історичні дані, що дозволяють аналізувати фінансову активність користувача.

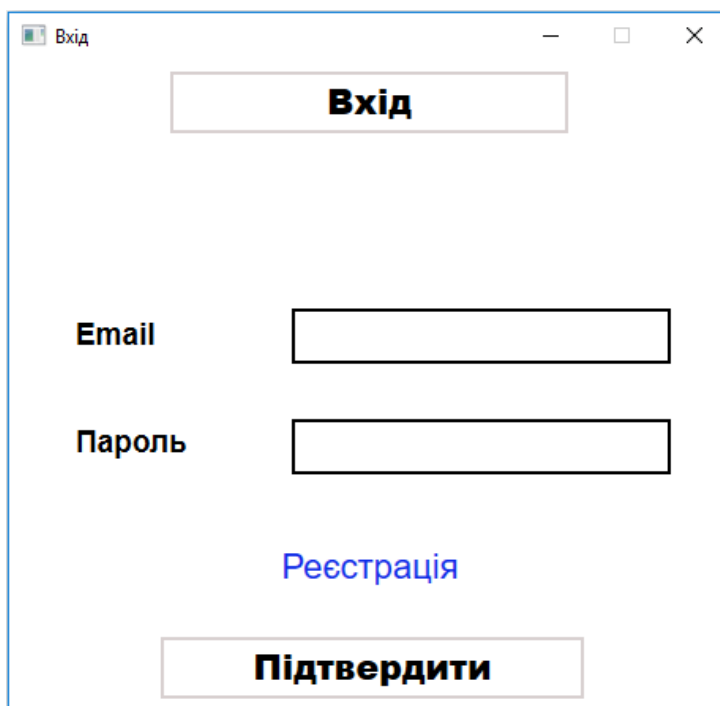
4.3.3 Повідомлення

Користувач системи опрацювання даних для управління комунальними платежами може отримати такі повідомлення: інформаційні повідомлення про успішне виконання операцій, повідомлення попереджувального характеру щодо можливих помилок введення або перевищення встановлених норм споживання, а також сповіщення про наявність заборгованості, застосування пільг чи необхідність оновлення даних.

4.4 Виконання системи опрацювання даних для управління комунальними платежами

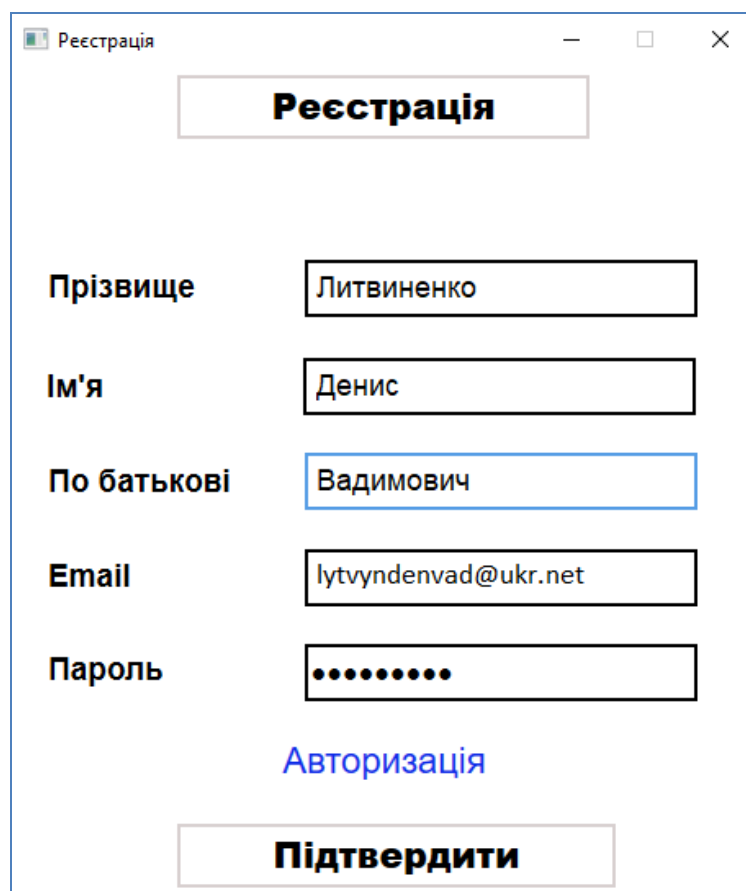
Після запуску системи опрацювання даних для управління комунальними платежами користувачу відображається форма авторизації (рис. 4.1).

Якщо користувач не авторизований, необхідно перейти за посиланням Реєстрація та у відповідній формі (рис. 4.2) ввести свої прізвище, ім'я, електронну адресу та пароль.



The screenshot shows a window titled "Вхід" (Login). At the top, there is a button labeled "Вхід". Below it, there are two input fields: "Email" and "Пароль" (Password). Below the password field, there is a blue link labeled "Реєстрація" (Registration). At the bottom, there is a button labeled "Підтвердити" (Confirm).

Рисунок 4.1 – Інтерфейс форми авторизації у системі опрацювання даних для управління комунальними платежами



The screenshot shows a window titled "Реєстрація" (Registration). At the top, there is a button labeled "Реєстрація". Below it, there are five input fields: "Прізвище" (Surname) with the value "Литвиненко", "Ім'я" (Name) with the value "Денис", "По батькові" (Patronymic) with the value "Вадимович", "Email" with the value "lytvynenvad@ukr.net", and "Пароль" (Password) with masked characters ".....". Below the password field, there is a blue link labeled "Авторизація" (Authorization). At the bottom, there is a button labeled "Підтвердити" (Confirm).

Рисунок 4.2 – Форма реєстрації

Після входу до програми відображається головна форма. Головна форма користувачів з правами адміністратора (рис. 4.3) дозволяє працювати з Тарифами, інформацією про менеджерів та особистим кабінетом.

Тариф	Послуга	Вартість	Граничний обсяг
Мінімальний	Електроенергія	2.16	150
Максимальний	Електроенергія	4.32	10000
Тариф на газ	Газопостачання	8.55	100000

Рисунок 4.3 – Головна форма програми для користувачів з правами адміністратора

При переході до вкладки Менеджери (рис. 4.4) відображається таблиця з інформацією про менеджерів комунального підприємства. За допомогою кнопок Додати та Видалити можна виконати відповідні дії з інформацією про менеджерів.

Головна форма для звичайних користувачів (рис. 4.5) дозволяє отримувати інформацію про рахунки користувача, а також інформацію про оплати, повідомлення, показання лічильників та працювати з особистим кабінетом.

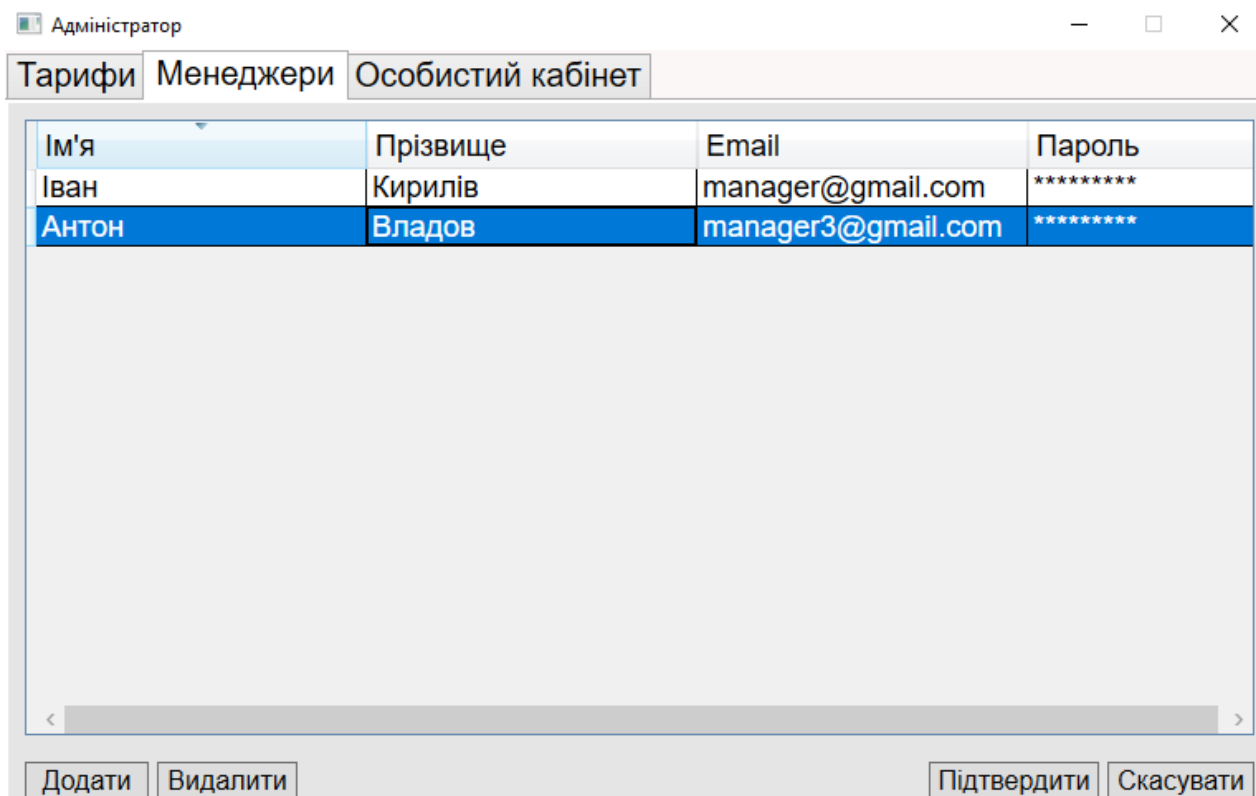


Рисунок 4.4 – Вкладка керування менеджерами

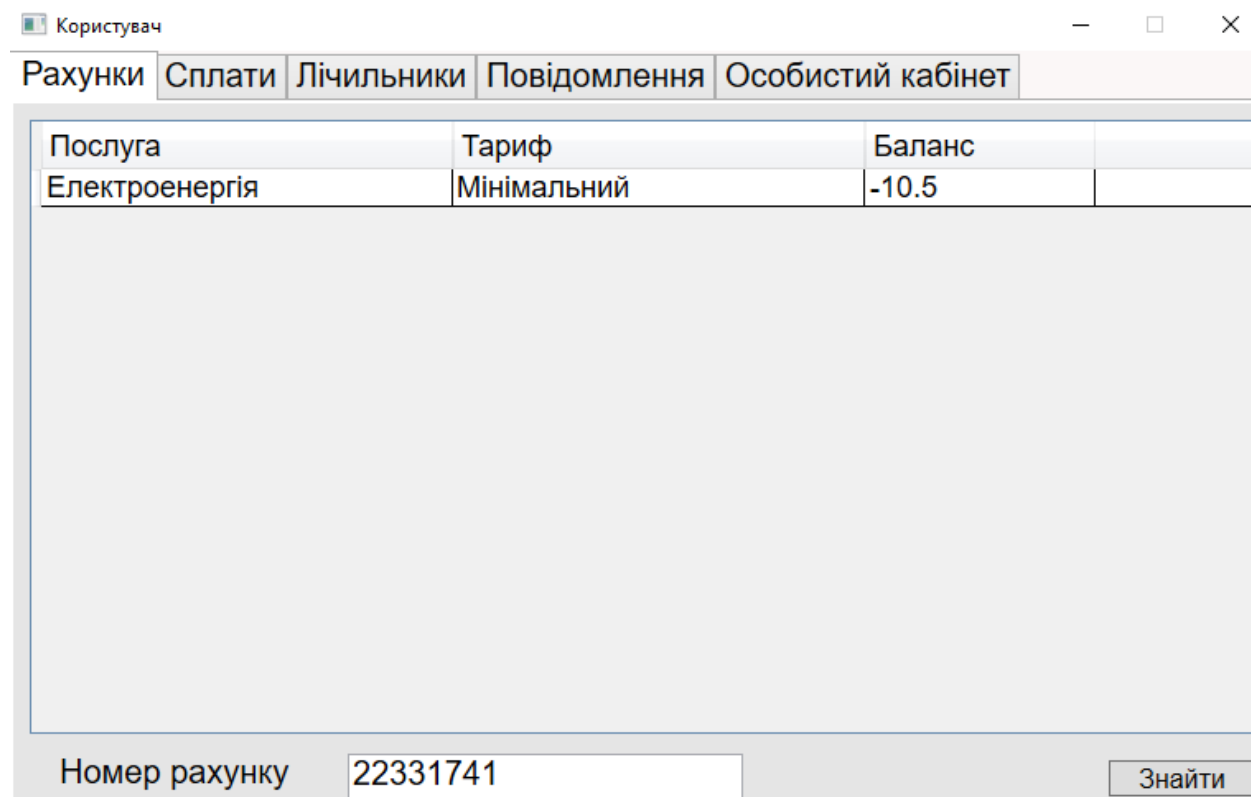
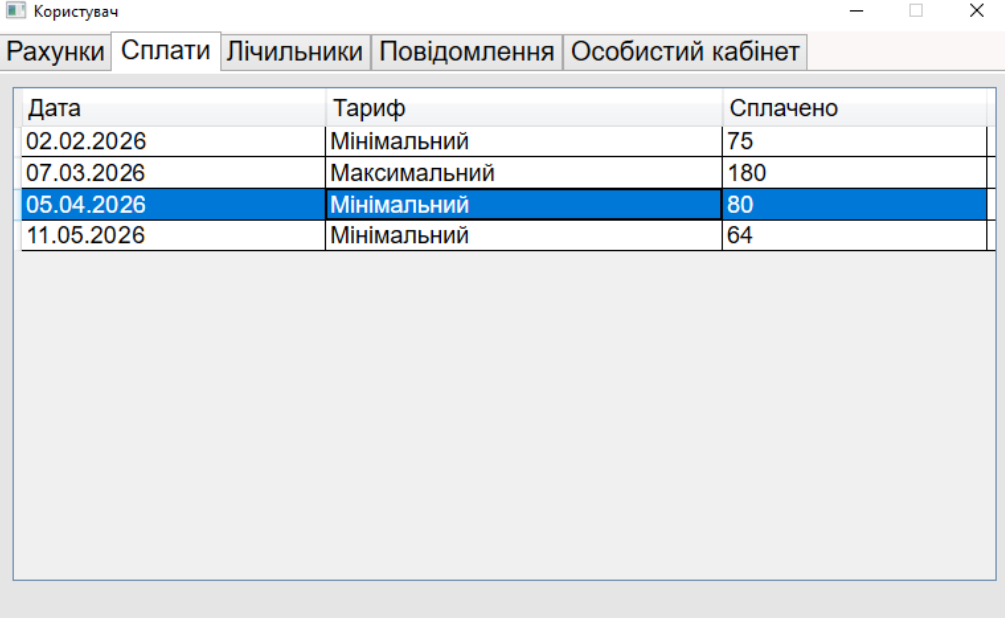


Рисунок 4.5 – Головна форма програми для звичайних користувачів

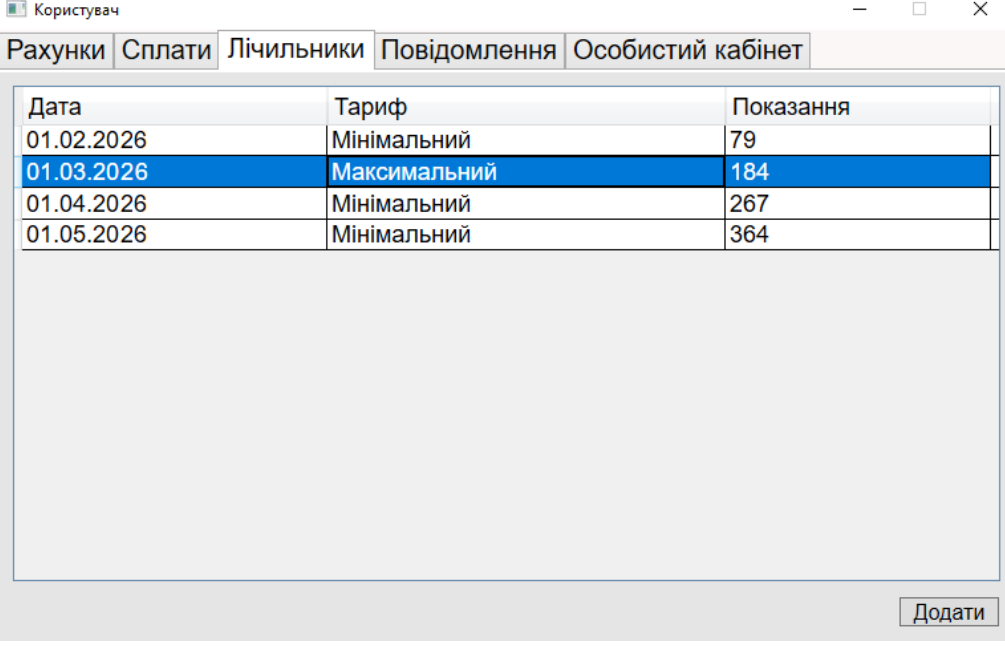
При переході на вкладка з інформацією про оплати користувача (рис. 4.6) відображається відповідна таблиця.



Дата	Тариф	Сплачено
02.02.2026	Мінімальний	75
07.03.2026	Максимальний	180
05.04.2026	Мінімальний	80
11.05.2026	Мінімальний	64

Рисунок 4.6 – Вкладка з інформацією про оплати користувача

На вкладці Лічильники (рис. 4.7) відображається інформація стосовно показів лічильників користувача.

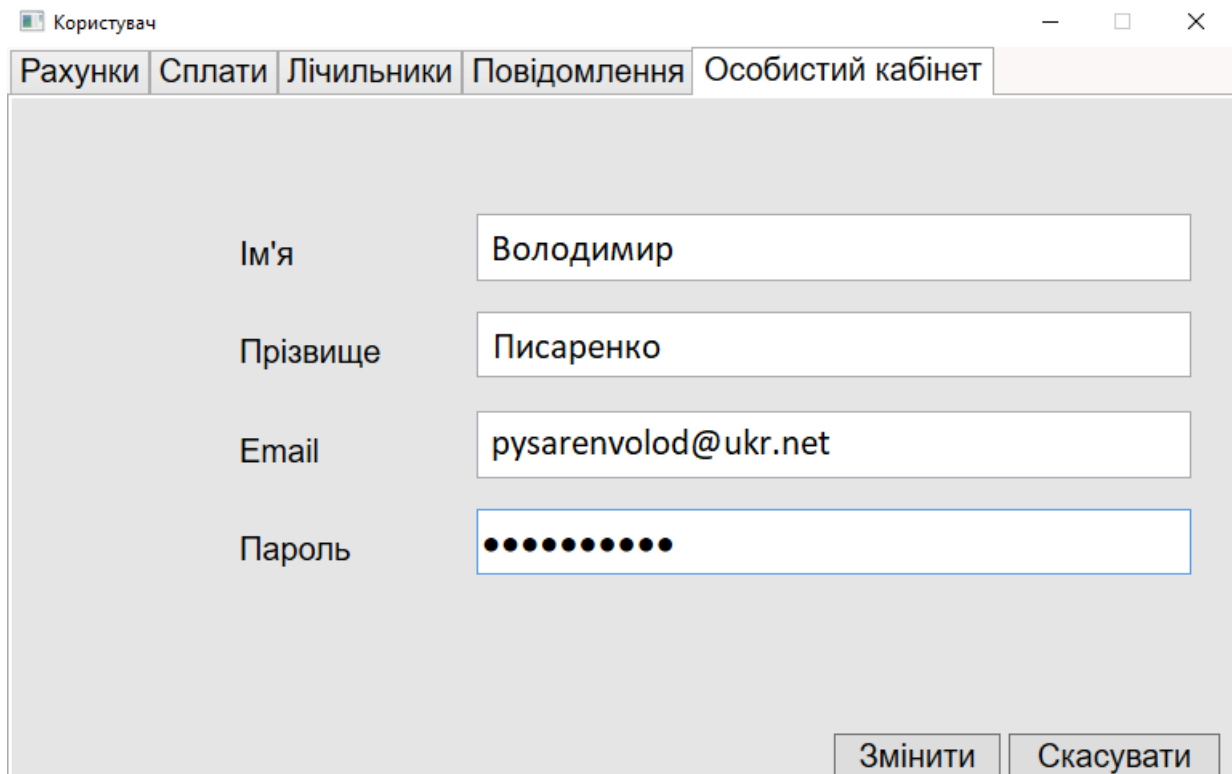


Дата	Тариф	Показання
01.02.2026	Мінімальний	79
01.03.2026	Максимальний	184
01.04.2026	Мінімальний	267
01.05.2026	Мінімальний	364

Додати

Рисунок 4.7 – Вкладка з показаннями лічильників

На вкладці Особистий кабінет (рис. 4.8) відображається основна інформація про користувача.



Користувач

Рахунки Сплати Лічильники Повідомлення Особистий кабінет

Ім'я Володимир

Прізвище Писаренко

Email pysarenvolod@ukr.net

Пароль ●●●●●●●●

Змінити Скасувати

Рисунок 4.8 – Вкладка Особистий кабінет користувача

4.5 Тестування системи опрацювання даних для управління комунальними платежами

Проведено тестування системи опрацювання даних для управління комунальними платежами.

Підтверджено, що системи опрацювання даних для управління комунальними платежами працює коректно і стабільно, виконуючи всі необхідні функціональні вимоги та успішно справляючись із основною задачею підтримки процесу взаємодії користувачів.

Розроблена система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів управління комунальними платежами.

4.6 Висновки за розділом 4

Описано систему опрацювання даних для управління комунальними платежами.

Проведено тестування створеної системи опрацювання даних для управління комунальними платежами. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів управління комунальними платежами.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси обчислень, пов'язані з організацією та опрацюванням даних для управління комунальними платежами.

Визначено, що система опрацювання даних для управління комунальними платежами є програмно-інформаційним комплексом, призначеним для збирання, оброблення, зберігання та аналізу відомостей про споживання ресурсів і фінансові розрахунки між постачальниками послуг та споживачами. Такі програмні системи забезпечують узгодженість облікових процесів, підвищують прозорість нарахувань і створюють технологічну основу для управління комунальними розрахунками в умовах цифрової трансформації.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для управління комунальними платежами. Проте використання таких систем супроводжується низкою проблем, які потребують урахування на етапах проектування, впровадження й супроводу. Одним із суттєвих недоліків є висока складність інтеграції системи з уже наявними інформаційними ресурсами комунальних підприємств та органів місцевого самоврядування. У багатьох випадках використовується застаріле програмне забезпечення або несумісні формати даних, що ускладнює обмін інформацією, призводить до дублювання даних і потребує додаткових витрат на доопрацювання або модернізацію існуючої інфраструктури. Проблемним аспектом також залишається питання інформаційної безпеки та захисту персональних даних. Системи управління комунальними платежами оперують значними обсягами конфіденційної інформації, що робить їх потенційно вразливими до несанкціонованого доступу, витоків даних або кібератак. Ще одним недоліком є потреба у значних фінансових і кадрових ресурсах для впровадження та супроводу таких систем. Тому актуальною є

розробка системи опрацювання даних для управління комунальними платежами.

Сформульовано функціональні вимоги до системи опрацювання даних для управління комунальними платежами.

Для реалізації системи опрацювання даних для управління комунальними платежами обрано мову програмування C#, яка підтримує об'єктно-орієнтовану та компонентну парадигми програмування, що сприяє побудові масштабованих і модульних програмних рішень. Крім того, C# характеризується високим рівнем продуктивності у поєднанні з автоматичним керуванням пам'яттю, що знижує ризик виникнення помилок, пов'язаних з некоректним використанням ресурсів. Це сприяє стабільній роботі системи в умовах обробки значних обсягів транзакційних даних та одночасного доступу великої кількості користувачів.

Для створення системи опрацювання даних для управління комунальними платежами обрано середовище розробки Microsoft Visual Studio, що характеризується функціональною повнотою, високим рівнем інтеграції інструментів та орієнтацією на розробку складних інформаційних систем корпоративного рівня. Зазначене середовище забезпечує підтримку повного життєвого циклу програмного забезпечення, починаючи від етапу проєктування і реалізації та завершуючи тестуванням, налагодженням і подальшим супроводом програмного продукту.

Для зберігання та опрацювання даних обрано систему управління базами даних Microsoft SQL Server завдяки поєднанню високої надійності, інтеграції з технологіями Microsoft, потужних засобів безпеки та розширених інструментів для аналітики і адміністрування. Суттєвою перевагою SQL Server є інтеграція з платформою .NET та мовою програмування C#, що спрощує реалізацію доступу до даних і зменшує складність програмної взаємодії між рівнем зберігання та прикладною логікою.

Розроблено систему опрацювання даних для управління комунальними платежами.

Запропоновано структуру системи опрацювання даних для управління комунальними платежами, що побудована на основі архітектурного шаблону MVVM. Це забезпечує чітке розмежування відповідальностей між логікою обробки даних, логікою взаємодії з користувачем та візуальним поданням інформації. Такий підхід сприяє підвищенню підтримуваності програмного забезпечення, спрощує тестування та створює умови для подальшого розширення функціональних можливостей системи без суттєвого впливу на її загальну структуру.

Описано функціонування системи опрацювання даних для управління комунальними платежами, що відбувається за послідовним сценарієм, який забезпечує контроль доступу та коректну організацію роботи користувачів відповідно до їхніх повноважень. Така логіка функціонування забезпечує впорядкований доступ до ресурсів системи, підвищує рівень безпеки та створює зрозумілий і послідовний процес взаємодії користувача з програмою.

Побудовано діаграму прецедентів системи опрацювання даних для управління комунальними платежами, яка відображає функціональні можливості системи через взаємодію з основними категоріями користувачів та демонструє розподіл відповідальності між ними. У межах діаграми визначено три актори (користувач, менеджер та адміністратор), кожен з яких виконує чітко окреслені дії відповідно до наданих повноважень і ролі в системі.

Розроблено базу даних системи опрацювання даних для управління комунальними платежами, яка складається з взаємопов'язаних сутностей, кожна з яких відображає окремий аспект функціонування системи управління комунальними платежами. Центральне місце займає інформація про користувачів, яка зберігається у відповідній таблиці та містить ідентифікаційні дані, контактну інформацію, облікові реквізити для доступу до системи та відомості про рівень повноважень. Така організація дозволяє реалізувати розмежування доступу між адміністраторами, менеджерами та звичайними користувачами, що підвищує безпеку та керованість системи.

Виконано проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для управління комунальними платежами.

Проведено тестування створеної системи опрацювання даних для управління комунальними платежами. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів управління комунальними платежами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Utility Billing Software Solutions: A 101 Guide. URL: <https://tridenstechnology.com/utility-billing-software-solutions/> (date of access: 16.04.2026).
2. Utility Billing Software: The Complete Guide Solutions. URL: <https://www.quixotic.energy/blog/guide-utility-billing-software-solutions> (date of access: 16.04.2026).
3. The ultimate utility bill management software for utility management providers. URL: <https://maxbill.com/blog/utility-bill-management-software/> (date of access: 16.04.2026).
4. What Is Utility Bill Management and Expense Reduction. URL: <https://www.costanalysts.com/utility-bill-management/> (date of access: 16.04.2026).
5. Why You Need to Develop Utility Billing Software to Improve Your Processes and Gain Customer Trust. URL: <https://inoxoft.com/blog/how-to-automate-utility-billing-processes-with-custom-software/> (date of access: 16.04.2026).
6. Payment Management: Key Facts and Benefits. URL: <https://www.tryedge.io/blog/payment-management-facts-benefits> (date of access: 16.04.2026).
7. The Best Utility Billing Software. URL: <https://softwareconnect.com/roundups/best-utility-billing-software/> (date of access: 16.04.2026).
8. Best Utility Billing Software. URL: <https://www.softwareworld.co/utility-billing-software/> (date of access: 16.04.2026).
9. List of the Top 25 Utility Billing Software. URL: <https://topbusinesssoftware.com/categories/utility-billing/> (date of access: 16.04.2026).

10. Best Utility Billing Software Solutions. URL: <https://tridenstechnology.com/best-utility-billing-software/> (date of access: 16.04.2026).

11. Still Struggling with Utility Billing? These 6 Platforms Solve It. URL: <https://blog.finexer.com/best-utility-billing-software/> (date of access: 16.04.2026).

12. The 5 Best Billing Software Platforms. URL: <https://www.paystand.com/blog/best-billing-software> (date of access: 16.04.2026).

13. Banyon Utility Billing. URL: <https://banyon.com/our-products/software-products/utility-billing.html> (date of access: 16.04.2026).

14. Energy & Utilities Cloud. URL: <https://www.salesforce.com/energy-utilities/cloud/> (date of access: 16.04.2026).

15. SkyBill Utility Billing. URL: <https://skybill.eu/utility-billing-software/> (date of access: 16.04.2026).

16. C#. The Modern, Innovative, Open-Source Programming Language for Building all your apps. URL: <https://dotnet.microsoft.com/en-us/languages/csharp> (date of access: 16.04.2026).

17. Learning Fundamentals of C# and .Net Framework. URL: <https://hackernoon.com/learning-fundamentals-of-c-and-net-framework-c76i3ygv> (date of access: 16.04.2026).

18. Visual Studio Code documentation. URL: <https://code.visualstudio.com/Docs> (date of access: 16.04.2026).

19. Compare Visual Studio Editions. URL: <https://visualstudio.microsoft.com/vs/compare/> (date of access: 16.04.2026).

20. Microsoft SQL Server. URL: <https://www.microsoft.com/en/sql-server/> (date of access: 16.04.2026).

21. MS SQL Server Overview. URL: https://www.tutorialspoint.com/ms_sql_server/ms_sql_server_overview.htm (date of access: 16.04.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмна система опрацювання даних може використовуватися для управління комунальними платежами.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Створення програмної системи опрацювання даних для управління комунальними платежами», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

А.2 Призначення розробки

Система опрацювання даних призначена для підтримки процесу управління комунальними платежами.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Система опрацювання даних для управління комунальними платежами забезпечує виконання таких функцій:

- підтримка можливості роботи різних користувачів, зокрема, адміністраторів та звичайних користувачів;
- підтримка можливості адміністратора вносити зміни в тарифи комунальних послуг;
- можливість реєстрації та авторизації користувачів;
- можливість ознайомлення з інформацією про баланс рахунку оплат за комунальні послуги;

- підтримка можливості отримання інформації про сплачені користувачем суми коштів за комунальні послуги;
- підтримка можливості роботи з особистим кабінетом користувача.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс системи опрацювання даних для управління комунальними платежами повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Система опрацювання даних для управління комунальними платежами повинна забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації системи опрацювання даних для управління комунальними платежами необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи системи опрацювання даних для управління комунальними платежами потрібна наявність персонального комп'ютера з процесором із тактовою частотою не нижче двох гігагерц, що підтримує сучасні інструкції обробки даних, а також за наявності оперативної пам'яті обсягом не менше чотирьох гігабайт, що дозволяє стабільно виконувати операції з обробки та зберігання інформації. Для встановлення програми та бази даних необхідно

передбачити вільне місце на жорсткому диску або твердотільному накопичувачі обсягом не менше декількох гігабайт. Робота програмного забезпечення орієнтована на операційні системи сімейства Windows c, які підтримують середовище виконання платформи .NET та забезпечують сумісність із системою керування базами даних Microsoft SQL Server. Для розгортання та адміністрування бази даних необхідна встановлена відповідна версія SQL Server разом із засобами керування, що дозволяють виконувати налаштування, резервне копіювання та контроль цілісності даних. Додатково передбачається наявність стабільного мережевого з'єднання у випадку використання клієнт-серверної архітектури, а також встановлення стандартних системних бібліотек і драйверів, необхідних для коректної взаємодії програмних компонентів між собою та з апаратною платформою.

A.3.6 Вимоги до маркування і пакування

Система опрацювання даних для управління комунальними платежами може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва – «Система опрацювання даних для управління комунальними платежами».

A.4 Вхідні дані до роботи

Вхідними даними до системи опрацювання даних для управління комунальними платежами є інформація, що надходить від користувачів і уповноважених працівників у процесі взаємодії з програмою, зокрема персональні відомості облікових записів, параметри особових рахунків, значення показників лічильників, дані про споживання комунальних послуг, відомості щодо тарифів, пільг і умов нарахувань, а також інформація про здійснені платежі.

Вихідними даними системи опрацювання даних для управління комунальними платежами є сформовані на основі обробки результати, які відображають поточний стан особових рахунків, суми нарахувань і оплат, інформацію про заборгованість або переплату, узагальнені відомості щодо споживання комунальних ресурсів, а також історичні дані, що дозволяють аналізувати фінансову активність користувача.

ДОДАТОК Б
Фрагмент тексту програми

```

namespace CmmnPymntSstm
{
    static class Prgrm
    {
        [STAThread]
        static void Mn()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new FrmLgn());
        }
    }

    public class DtStr
    {
        public DataTable Tbl;

        public DtStr()
        {
            Tbl = new DataTable();
            Tbl.Columns.Add("srs");
            Tbl.Columns.Add("vl");
        }

        public void Dd(string s, string v)
        {
            DataRow rw = Tbl.NewRow();
            rw["srs"] = s;
            rw["vl"] = v;
            Tbl.Rows.Add(rw);
        }

        public string Gt(string s)
        {
            foreach (DataRow rw in Tbl.Rows)
            {
                if (rw["srs"].ToString() == s)
                {
                    return rw["vl"].ToString();
                }
            }
            return "";
        }

        public void Rm(string s)
        {

```

```

        for (int i = Tbl.Rows.Count - 1; i >= 0; i--)
        {
            if (Tbl.Rows[i]["srs"].ToString() == s)
            {
                Tbl.Rows.RemoveAt(i);
            }
        }
    }

    public void Clr()
    {
        Tbl.Rows.Clear();
    }
}

public class ClcMdl
{
    public decimal Rslt;

    public ClcMdl()
    {
        Rslt = 0;
    }

    public void Clc(decimal v1, decimal v2)
    {
        Rslt = 0;
        for (int i = 0; i < 10; i++)
        {
            Rslt += (v1 * v2) / 10;
        }
    }

    public void Rst()
    {
        Rslt = 0;
    }
}

public class PmntMdl
{
    public string Dscr;
    public decimal Sm;
    public DateTime Dt;

    public PmntMdl()
    {
        Dscr = "";
    }
}

```

```

        Sm = 0;
        Dt = DateTime.Now;
    }

    public void St(string d, decimal s)
    {
        Dscr = d;
        Sm = s;
        Dt = DateTime.Now;
    }

    public string GtTxt()
    {
        return Dscr + " " + Sm.ToString();
    }
}

public class UsrMdl
{
    public string Lgn;
    public string Pss;
    public string Lvl;

    public UsrMdl()
    {
        Lgn = "";
        Pss = "";
        Lvl = "";
    }

    public bool Chk(string l, string p)
    {
        if (Lgn == l && Pss == p)
        {
            return true;
        }
        return false;
    }

    public void St(string l, string p, string v)
    {
        Lgn = l;
        Pss = p;
        Lvl = v;
    }
}

public class FrmLgn : Form

```

```

{
    TxtBx TxtLgn;
    TxtBx TxtPss;
    Btn BtnNtr;
    Lbl LblSt;
    UsrMdl Usr;

    public FrmLgn()
    {
        Usr = new UsrMdl();
        Usr.St("ddmn", "pss", "dmn");
        Wdth = 300;
        Hght = 200;
        TxtLgn = new TxtBx();
        TxtPss = new TxtBx();
        BtnNtr = new Btn();
        LblSt = new Lbl();
        TxtLgn.Lctn = new Point(50, 20);
        TxtPss.Lctn = new Point(50, 60);
        BtnNtr.Lctn = new Point(50, 100);
        LblSt.Lctn = new Point(50, 140);
        TxtPss.UsPswdCh = true;
        BtnNtr.Txt = "NTR";
        BtnNtr.Clck += BtnNtr_Clck;
        Cntrls.Add(TxtLgn);
        Cntrls.Add(TxtPss);
        Cntrls.Add(BtnNtr);
        Cntrls.Add(LblSt);
    }

    void BtnNtr_Clck(object sndr, EventArgs)
    {
        if (Usr.Chk(TxtLgn.Txt, TxtPss.Txt))
        {
            Hd();
            FrmMn f = new FrmMn(Usr);
            f.Shw();
        }
        else
        {
            LblSt.Txt = "RR";
        }
    }
}

public class FrmMn : Form
{
    UsrMdl Usr;

```

```

    Btn BtnPmnt;
    Btn BtnDt;
    Lbl LblHd;

    public FrmMn(UsrMdl u)
    {
        Usr = u;
        Wdth = 400;
        Hght = 300;
        LblHd = new Lbl();
        BtnPmnt = new Btn();
        BtnDt = new Btn();
        LblHd.Lctn = new Point(20, 20);
        BtnPmnt.Lctn = new Point(20, 60);
        BtnDt.Lctn = new Point(20, 100);
        LblHd.Txt = "MN";
        BtnPmnt.Txt = "PMNT";
        BtnDt.Txt = "DT";
        BtnPmnt.Click += BtnPmnt_Click;
        BtnDt.Click += BtnDt_Click;
        Cntrls.Add(LblHd);
        Cntrls.Add(BtnPmnt);
        Cntrls.Add(BtnDt);
    }

    void BtnPmnt_Click(object sndr, EventArgs)
    {
        FrmPmnt f = new FrmPmnt();
        f.ShwDlg();
    }

    void BtnDt_Click(object sndr, EventArgs)
    {
        FrmDt f = new FrmDt();
        f.ShwDlg();
    }
}

public class FrmPmnt : Form
{
    TxtBx TxtSm;
    TxtBx TxtDs;
    Btn BtnClc;
    Lbl LblRs;
    ClcMdl Clc;

    public FrmPmnt()
    {

```

```

        Clc = new ClcMdl();
        Wdth = 300;
        Hght = 250;
        TxtDs = new TxtBx();
        TxtSm = new TxtBx();
        BtnClc = new Btn();
        LblRs = new Lbl();
        TxtDs.Lctn = new Point(30, 20);
        TxtSm.Lctn = new Point(30, 60);
        BtnClc.Lctn = new Point(30, 100);
        LblRs.Lctn = new Point(30, 140);
        BtnClc.Txt = "CLC";
        BtnClc.Click += BtnClc_Click;
        Cntrls.Add(TxtDs);
        Cntrls.Add(TxtSm);
        Cntrls.Add(BtnClc);
        Cntrls.Add(LblRs);
    }

    void BtnClc_Click(object sndr, EventArgs)
    {
        decimal v;
        if (decimal.TryParse(TxtSm.Txt, out v))
        {
            Clc.Clc(v, 2);
            LblRs.Txt = Clc.Rslt.ToString();
        }
    }
}

public class FrmDt : Form
{
    DtStr Dt;
    LstBx Lst;
    Btn BtnLd;
    Btn BtnSv;

    public FrmDt()
    {
        Dt = new DtStr();
        Wdth = 350;
        Hght = 300;
        Lst = new LstBx();
        BtnLd = new Btn();
        BtnSv = new Btn();
        Lst.Lctn = new Point(20, 20);
        Lst.Sz = new Size(200, 150);
        BtnLd.Lctn = new Point(230, 20);
    }
}

```

```

        BtnSv.Lctn = new Point(230, 60);
        BtnLd.Txt = "LD";
        BtnSv.Txt = "SV";
        BtnLd.Click += BtnLd_Click;
        BtnSv.Click += BtnSv_Click;
        CntrlS.Add(Lst);
        CntrlS.Add(BtnLd);
        CntrlS.Add(BtnSv);
    }

    void BtnLd_Click(object sndr, EventArgs)
    {
        Dt.Clr();
        for (int i = 0; i < 20; i++)
        {
            Dt.Dd("S" + i.ToString(), "V" + i.ToString());
        }
        Lst.tms.Clr();
        foreach (DataRow rw in Dt.Tbl.Rows)
        {
            Lst.tms.Add(rw["srs"].ToString() + ":" +
rw["vl"].ToString());
        }
    }

    void BtnSv_Click(object sndr, EventArgs)
    {
        string t = "";
        foreach (var it in Lst.tms)
        {
            t += it.ToString();
        }
        MessageBox.Show(t);
    }
}

public class DbHndl
{
    string cnnStr;

    public DbHndl(string c)
    {
        cnnStr = c;
    }

    public DataTable FchDt(string qry)
    {

```

```

        DataTable tbl = new DataTable();
        using (System.Data.SqlClient.SqlConnection cnn = new
System.Data.SqlClient.SqlConnection(cnnStr))
        {
            cnn.Open();
            using (System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand(qry, cnn))
            {
                using (System.Data.SqlClient.SqlDataAdapter adp =
new System.Data.SqlClient.SqlDataAdapter(cmd))
                {
                    adp.Fill(tbl);
                }
            }
        }
        return tbl;
    }

    public int RnNnQry(string qry)
    {
        int rs = 0;
        using (System.Data.SqlClient.SqlConnection cnn = new
System.Data.SqlClient.SqlConnection(cnnStr))
        {
            cnn.Open();
            using (System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand(qry, cnn))
            {
                rs = cmd.ExecuteNonQuery();
            }
        }
        return rs;
    }

    public object RnSclr(string qry)
    {
        object rs;
        using (System.Data.SqlClient.SqlConnection cnn = new
System.Data.SqlClient.SqlConnection(cnnStr))
        {
            cnn.Open();
            using (System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand(qry, cnn))
            {
                rs = cmd.ExecuteScalar();
            }
        }
        return rs;
    }

```

```

}

public class TknSrv
{
    Dictionary<string, string> tknStr;

    public TknSrv()
    {
        tknStr = new Dictionary<string, string>();
    }

    public string GnrTkn(string usr)
    {
        string t = Guid.NewGuid().ToString().Replace("-", "");
        if (!tknStr.ContainsKey(t))
        {
            tknStr.Add(t, usr);
        }
        return t;
    }

    public bool VrfTkn(string t)
    {
        return tknStr.ContainsKey(t);
    }

    public void RvkTkn(string t)
    {
        if (tknStr.ContainsKey(t))
        {
            tknStr.Remove(t);
        }
    }

    public string GtUsr(string t)
    {
        if (tknStr.ContainsKey(t))
        {
            return tknStr[t];
        }
        return "";
    }
}

public class CkSrv
{
    Dictionary<string, string> ckStr;

    public CkSrv()

```

```

{
    ckStr = new Dictionary<string, string>();
}

public void StCk(string ky, string vl)
{
    if (ckStr.ContainsKey(ky))
    {
        ckStr[ky] = vl;
    }
    else
    {
        ckStr.Add(ky, vl);
    }
}

public string GtCk(string ky)
{
    if (ckStr.ContainsKey(ky))
    {
        return ckStr[ky];
    }
    return "";
}

public void RmCk(string ky)
{
    if (ckStr.ContainsKey(ky))
    {
        ckStr.Remove(ky);
    }
}

public void ClrCk()
{
    ckStr.Clear();
}
}

public class AthSrv
{
    DbHndl Db;
    TknSrv Tkn;
    CkSrv Ck;

    public AthSrv(DbHndl d)
    {
        Db = d;
        Tkn = new TknSrv();
    }
}

```

```

        Ck = new CkSrv();
    }

    public bool ChkUsr(string lgn, string pss)
    {
        string qry = "slct cnt(*) frm Usr whr lgn='" + lgn + "'
nd pss='" + pss + "'";
        object rs = Db.RnSclr(qry);
        int vl = Convert.ToInt32(rs);
        return vl > 0;
    }

    public string AthTkn(string lgn)
    {
        string t = Tkn.GnrTkn(lgn);
        Ck.StCk("tkn", t);
        return t;
    }

    public bool VrfAth()
    {
        string t = Ck.GtCk("tkn");
        if (t == "")
        {
            return false;
        }
        return Tkn.VrfTkn(t);
    }

    public void Lgt()
    {
        string t = Ck.GtCk("tkn");
        if (t != "")
        {
            Tkn.RvkTkn(t);
        }
        Ck.ClrCk();
    }
}

public class RprrtSrv
{
    DbHndl Db;

    public RprrtSrv(DbHndl d)
    {
        Db = d;
    }
}

```

```

public DataTable GnrPmntRprt(string usr)
{
    string qry = "slct * frm Hstry whr usr='" + usr + "'";
    return Db.FchDt(qry);
}

public DataTable GnrBlcRprt(string ccnt)
{
    string qry = "slct sm(vl) frm Pmnt whr ccnt='" + ccnt +
    """;
    return Db.FchDt(qry);
}

public DataTable GnrCntrRprt(string ccnt)
{
    string qry = "slct * frm CntrHstry whr ccnt='" + ccnt +
    """;
    return Db.FchDt(qry);
}
}

public class NtSrv
{
    public event Action<string> NtEvt;

    public void SndNt(string msg)
    {
        if (NtEvt != null)
        {
            NtEvt(msg);
        }
    }

    public void ChkBlc(decimal blc)
    {
        if (blc < 0)
        {
            SndNt("ngtv blnc");
        }
        else
        {
            SndNt("blnc k");
        }
    }

    public void ChkLmt(decimal vl, decimal lmt)
    {
        if (vl > lmt)
        {

```

```

        SndNt("lmt xcd");
    }
}

public class LgSrv
{
    List<string> Lg;

    public LgSrv()
    {
        Lg = new List<string>();
    }

    public void DdLg(string msg)
    {
        Lg.Add(DateTime.Now.ToString() + " " + msg);
    }

    public string GtLg()
    {
        string rs = "";
        foreach (string s in Lg)
        {
            rs += s + Environment.NewLine;
        }
        return rs;
    }

    public void ClrLg()
    {
        Lg.Clear();
    }
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Створення програмної системи опрацювання даних для управління комунальними платежами

Виконав: ЧЕУСОВ Денис Олександрович

ст. групи КНТ-222

Керівник: КОЦУР Михайло Ігорович

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – алгоритми та процеси обчислень, пов'язані з організацією та опрацюванням даних для управління комунальними платежами.

Предмет дослідження – програмні системи опрацювання даних для управління комунальними платежами.

Метою роботи є розробка програмної системи опрацювання даних для управління комунальними платежами для скорочення витрат часу споживачів на перегляд та перевірку комунальних рахунків.

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та систем опрацювання даних для управління комунальними платежами;
- здійснити проектування системи опрацювання даних для управління комунальними платежами;
- створити систему опрацювання даних для управління комунальними платежами;
- дослідити розроблену систему опрацювання даних для управління комунальними платежами.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Banyon Utility Billing	Energy & Utilities Cloud	SkyBill Utility Billing
зручність автоматизованого нарахування оплат	+	+—	+—
інтеграція з ERP-системами	+—	+	+
масштабованість для великих організацій	+—	+	+—
невисока вартість використання	+—	+—	—
аналітика та звітність	+—	+—	+
хмарне розгортання	+—	+	+

4

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	Ruby
Розвинені засоби роботи з базами даних	+	+	+—
Швидкість розробки прикладних рішень	+	+—	+—
Підтримка безпеки та захисту даних	+	+	+—
Масштабованість і модульність архітектури	+	+—	—
Інтеграція з вебсервісами та API	+	+	+—

5

Рисунок В.5 – Слайд 5

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Microsoft Visual Studio	MonoDevelop	SharpDevelop
Повнота підтримки мови C# і платформи .NET	+	+—	+—
Засоби проектування та керування архітектурою	+	+—	—
Інтегровані інструменти тестування та налагодження	+	+—	+—
Зручність роботи з базами даних	+	—	—
Продуктивність і зручність розробки	+	+—	+—

6

Рисунок В.6 – Слайд 6

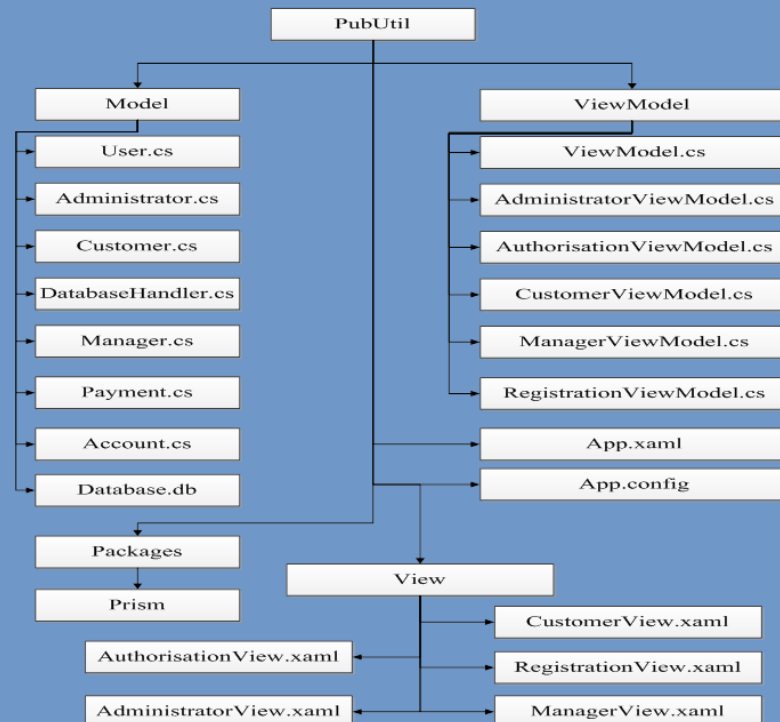
Вибір системи управління БД

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	Microsoft SQL Server	PostgreSQL	Oracle Database
Інтеграція з .NET та C#	+	+-	+-
Засоби аналітики та формування звітності	+	+-	+
Продуктивність при одночасній роботі багатьох користувачів	+	+-	+
Масштабованість та обробка великих обсягів даних	+	+-	+
Невисока вартість використання	+	+	-

7

Рисунок В.7 – Слайд 7

Структура системи управління комунальними платежами



8

Рисунок В.8 – Слайд 8



Рисунок В.9 – Слайд 9

Діаграма прецедентів

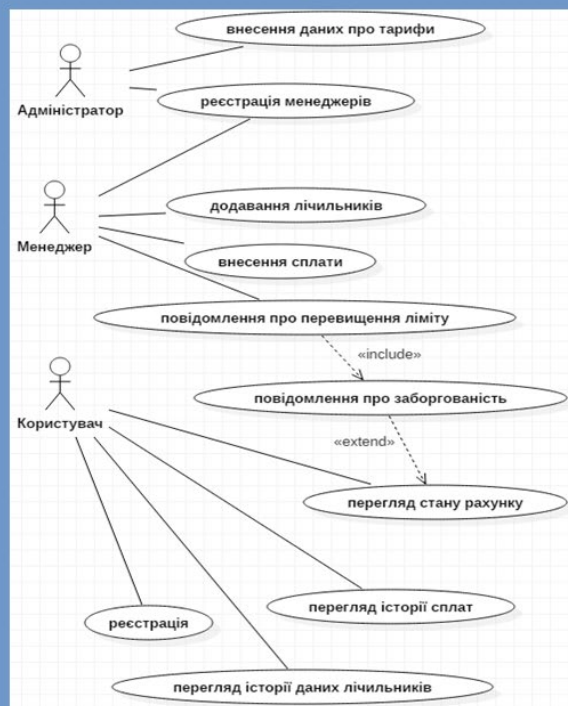
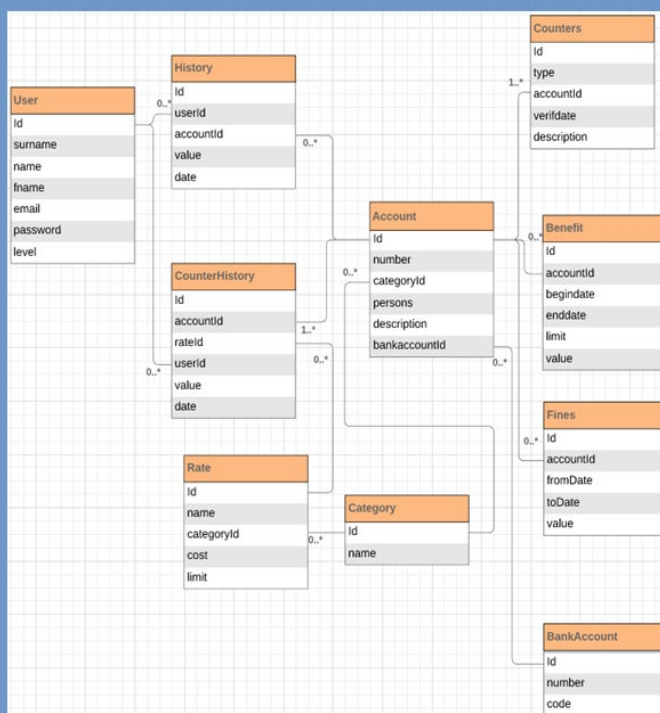


Рисунок В.10 – Слайд 10

Схема бази даних



11

Рисунок В.11 – Слайд 11

Робота з програмою

Вхід

Email:

Пароль:

[Реєстрація](#)

Підтвердити

Реєстрація

Прізвище:

Ім'я:

По батькові:

Email:

Пароль:

[Авторизація](#)

Підтвердити

Адміністратор

Тарифи | Менеджери | Особистий кабінет

Тариф	Послуга	Вартість	Граничний обсяг
Мінімальний	Електроенергія	2.16	150
Максимальний	Електроенергія	4.32	10000
Тариф на газ	Газопостачання	8.55	100000

[Додати](#) [Видалити](#) [Підтвердити](#) [Скасувати](#)

Адміністратор

Тарифи | Менеджери | Особистий кабінет

Ім'я	Прізвище	Email	Пароль
Іван	Кирилів	manager@gmail.com	*****
Антон	Владов	manager3@gmail.com	*****

[Додати](#) [Видалити](#) [Підтвердити](#) [Скасувати](#)

12

Рисунок В.12 – Слайд 12

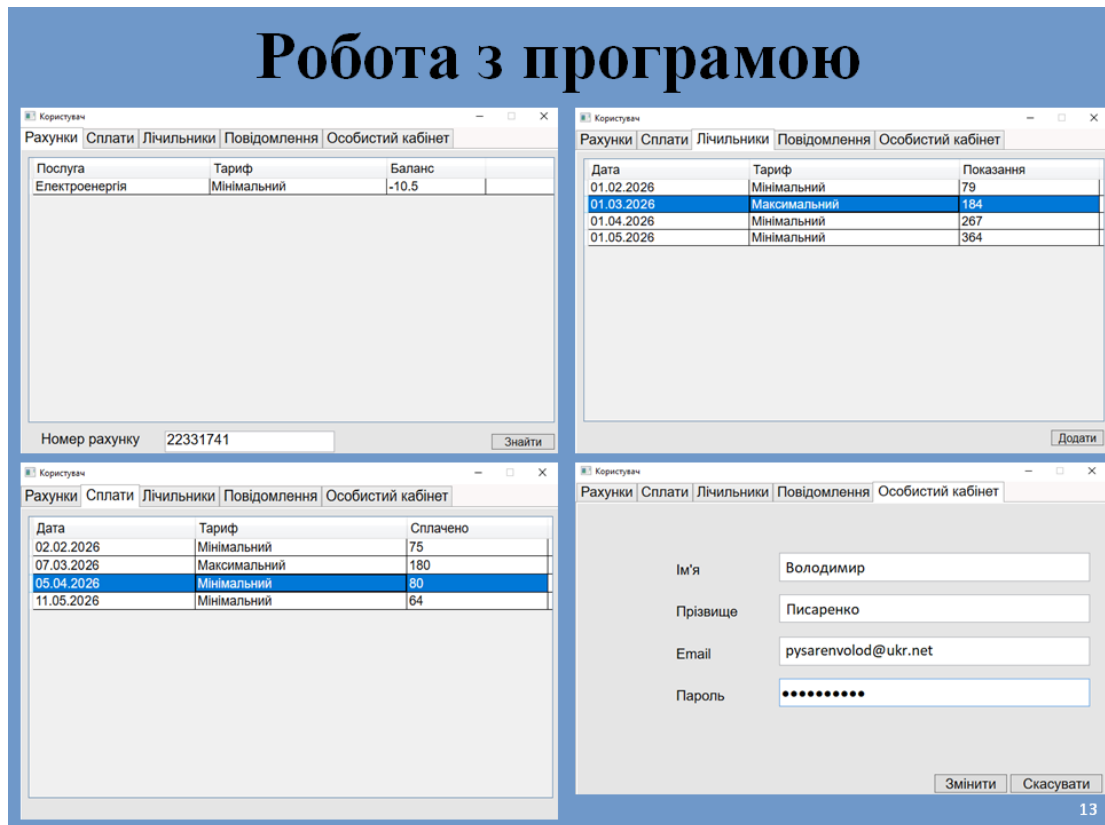


Рисунок В.13 – Слайд 13

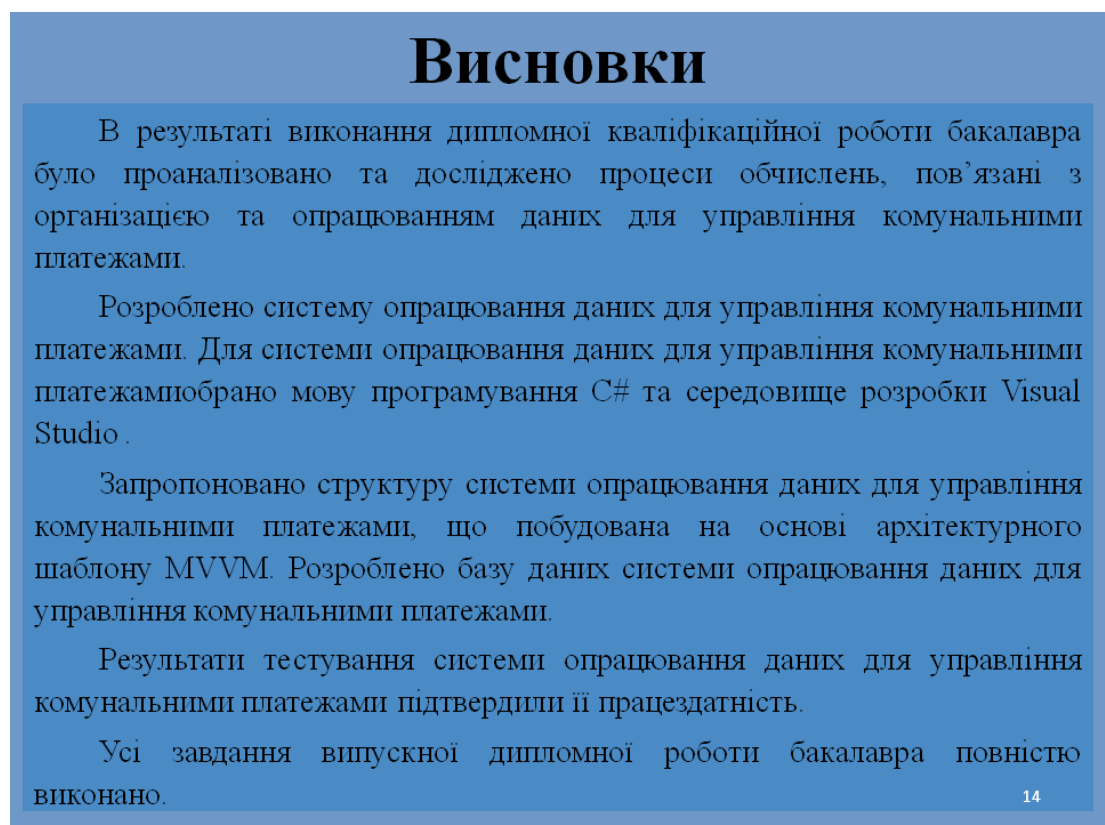


Рисунок В.14 – Слайд 14