

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Веб-орієнтована система оператора мобільного
зв'язку»**

Виконав студент групи ПЗ-21-1

ГРИЦЮК Владислав Іванович

Керівник роботи:

ЛОБАНЧИКОВА Надія Миколаївна

Рецензент:

САВІЦЬКИЙ Роман Святославович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ГРИЦЮК Владислав Іванович**

Керівник роботи: **ЛОБАНЧИКОВА Надія Миколаївна**

Тема роботи: **«Веб-орієнтована система оператора мобільного зв'язку»**

затверджена Наказом закладу вищої освіти від **«14» лютого 2025р., №61/с**

Вихідні дані для роботи: Об'єктом дослідження є процеси та технології побудови веб-орієнтованих систем. Предметом дослідження є моделі, методи та алгоритми побудови веб-орієнтованих систем оператора мобільного зв'язку.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Надія ЛОБАНЧИКОВА	26.02.2025	26.02.2025
2	Надія ЛОБАНЧИКОВА	14.03.2025	14.03.2025
3	Надія ЛОБАНЧИКОВА	16.04.2025	16.04.2025

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебдодатку та пояснювальної записки. Пояснювальна записка кваліфікаційної роботи містить - 60 сторінок, 29 ілюстрацій та 3 таблиці, 4 додатки, що розміщено на 8 аркушах та списку використаних джерел, що включає 31 найменування.

Метою роботи є розробка веб-орієнтованої системи оператора мобільного зв'язку для підвищення зручності обслуговування абонентів та прискорення процесу взаємодії між клієнтом та оператором.

Об'єктом дослідження є процеси та технології побудови веб-орієнтованих систем. Предметом дослідження є моделі, методи, засоби та алгоритми побудови веб-орієнтованих систем оператора мобільного зв'язку.

Прототип веб-орієнтованої системи оператора мобільного зв'язку дає змогу отримати інформацію щодо балансу особового рахунку; підключити або відключити послуги мобільного зв'язку; змінювати тарифний план тощо. Для побудови інтерфейсу та реалізації логіки додатку використано клієнт-серверну архітектуру. У роботі використано методи об'єктно-орієнтованого проєктування, структурування даних, а також технології програмування з використанням технологій Spring Boot та PostgreSQL для реалізації серверної частини та технологію React для клієнтської частини.

КЛЮЧОВІ СЛОВА: ВЕБСИСТЕМА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, МОБІЛЬНИЙ ЗВ'ЯЗОК.

					ІПЗ.КР.Б-121-25-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Веб-орієнтована система оператора мобільного зв'язку	Літ.	Арк.	Аркуші
Розроб.		Грицюк В.І.					3	60
Керівник		Лобанчикова Н.М.						
Рецензент		Савіцький Р.С.						
Зав. каф.		Вакалюк Т.А.						
Пояснювальна записка					ФІКТ, гр. ІПЗ-21-1			

ABSTRACT

The final bachelor's work consists of a web application and an explanatory note. The explanatory note of the qualification work contains 60 pages, 29 illustrations and 3 tables, 4 appendices, which is placed on 8 sheets and a list of references that includes 31 items.

The purpose of the work is to develop a web-based system for a mobile operator to improve the convenience of customer service and accelerate the process of interaction between the client and the operator.

The object of research is the processes and technologies for building web-oriented systems. The subject of the research is models, methods, tools and algorithms for building web-based systems of a mobile operator.

The prototype of a web-based system of a mobile operator allows obtaining information about the balance of a personal account; connecting or disconnecting mobile services; changing the tariff plan, etc. A client-server architecture is used to build the interface and implement the application logic. The work uses object-oriented design methods, data structuring, and programming technologies using Spring Boot and PostgreSQL technologies to implement the server side and React technology for the client side.

KEYWORDS: WEB SYSTEM, SOFTWARE, MOBILE COMMUNICATION.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ПОБУДОВИ ВЕБ-ОРІЄНТОВАНИХ СИСТЕМ	10
1.1. Постановка задачі.....	10
1.2. Аналіз веб-орієнтованих систем операторів мобільного зв'язку в Україні..	11
1.3. Обґрунтування вибору інструментальних засобів	17
Висновки до першого розділу	20
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ ОПЕРАТОРА МОБІЛЬНОГО ЗВ'ЯЗКУ	21
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи	21
2.2. Розробка функціональних алгоритмів роботи системи	27
2.3. Програмна реалізація функціональних алгоритмів роботи системи	31
Висновки до другого розділу	34
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ОСОБЛИВОСТІ РОЗГОРТАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ.....	35
3.1. Опис інтерфейсу веб-орієнтованої системи оператора мобільного зв'язку .	35
3.2. Порядок розгортання клієнт-серверної системи.....	40
3.3. Тестування веб-орієнтованої системи оператора мобільного зв'язку.....	42
Висновки до третього розділу	45
ВИСНОВКИ	46
ДОДАТКИ.....	52

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – прикладний програмний інтерфейс

CRUD – основні операції, що використовуються для управління даними

БД – база даних

ПЗ – програмне забезпечення

СУБД – система управління базами даних

GUI (Graphic User Interface) – графічний інтерфейс користувача

MVVM (Model-View-ViewModel) – шаблон проєктування, що застосовується під час проєктування архітектури застосунків

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми. Розвиток національної економіки неможливо уявити без функціонування конкурентоспроможної телекомунікаційної та інформаційно-комунікаційної галузі. Стрімкий розвиток електронних та інформаційних комунікацій, а саме сфери послуг мобільного зв'язку сьогодні є необхідною умовою функціонування усіх сфер суспільного життя та досягнення розвитку постіндустріального суспільства. Мінусом українських операторів мобільного зв'язку є недостатньо гнучка тарифна та маркетингова політика: майже в усіх сучасних тарифах є обов'язкові пакети послуг на 4 тижні або місяць. Зрозуміло, що операторам більш цікаві абоненти, які мають стандартні пакети. Однак аналіз потреб абонентів показав зацікавленість абонентів у більш гнучких тарифних пакетах, зменшенні їх тривалості та тарифікації. Проведений аналіз доступних пакетів операторів мобільного зв'язку, зокрема Київстар, Водафон, Лайф, показав, що українські мобільні оператори впровадили спеціальні тарифні пропозиції, розраховані на студентів та пенсіонерів. Так, наприклад, Lifecell запустивши акційний тариф «Можливий» зі зниженою щомісячною вартістю в 100 гривень, яка залишатиметься актуальною до кінця 2025 року. Vodafone, також, пропонує тариф «Turbo» за аналогічною ціною – 100 гривень на місяць. Київстар розробив тарифний план «LOVE UA Комфорт», що коштує 140 гривень щомісяця. Але всі вони розраховані на 30-денний термін. Крім того, іноді, оператори нав'язують автоматичне підключення додаткових послуг, які зазвичай є платними.

Автоматизована система розрахунків (білінг) – це система обліку, розрахунку та виставляння рахунків за послуги. Це важливий елемент програмного забезпечення оператора електронних комунікаційних послуг. Білінг збирає інформацію про спожиті послуги, будь це дзвінки, SMS, використання Інтернету, розраховує вартість наданих послуг за тарифними планами, які можуть налаштувати адміністратори порталу оператора, дозволяє контролювати витрати користувачів та аналізувати їх потреби. Як правило, білінг має підсистему

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

сповіщень, що повідомляє користувачам про оплату, зміну параметрів послуг і тарифних планів, тощо. Білінг обліковує процеси автоматично згідно з наперед визначеними умовами. Автоматизація має й побічні ефекти, що часто призводить до некоректної тарифікації й, отже, непередбаченого виникнення заборгованості абонента. Крім того, білінгову систему використовують при наданні послуг в рамках сервісів Інтернету речей, причому тарифікація IoT-сервісів має бути як найточнішою.

Метою роботи є розробка веб-орієнтованої системи оператора мобільного зв'язку для комфортних та простих умов обслуговування абонентів та прискорення процесу взаємодії між клієнтом та оператором.

Для реалізації поставленої мети потрібно виконати наступні завдання:

1. Провести огляд функціональності веб-орієнтованих систем операторів мобільного зв'язку.
2. Здійснити вибір засобів проектування.
3. Провести моделювання веб-орієнтованої системи.
4. Розробити прототип веб-орієнтованої системи оператора мобільного зв'язку.

Прототип веб-орієнтованої системи оператора мобільного зв'язку повинен отримувати інформацію щодо балансу особового рахунку; підключити або відключити послуги мобільного зв'язку; змінювати тарифний план та подавати заявки на участь у акціях.

Об'єктом дослідження є процеси та технології побудови веб-орієнтованих систем.

Предметом дослідження є моделі, методи, засоби та алгоритми побудови веб-орієнтованих систем оператора мобільного зв'язку.

Методи дослідження. У роботі використано методи об'єктно-орієнтованого проектування, структурування даних, а також технології програмування з використанням технологій Spring Boot та БД (PostgreSQL), React. Для побудови інтерфейсу та реалізації логіки додатку використано клієнт-серверну архітектуру.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

За темою випускної кваліфікаційної роботи бакалавра було опубліковано тези:

Грицюк В.І., Лобанчикова Н.М. Проект веборієнтованої системи оператора мобільного зв'язку. Тези XV Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології», м. Житомир, 28-29 березня 2025 р. Житомир: Житомирська політехніка, 2025. С. 29-30.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ПОБУДОВИ ВЕБ-ОРІЄНТОВАНИХ СИСТЕМ

1.1. Постановка задачі

Для тарифікації телеком-сервісу кожному провайдеру необхідна білінгова система. Це спеціальний софт для розрахунку, який збирає первинні дані про постачання телеком-послуг і структурує їх для подальшої обробки. Білінгова система працює за принципом бази даних, до неї надходить уся необхідна інформація. Це може бути тривалість дзвінка, кількість інтернет-трафіку, що споживається, і так далі. У систему потрапляє дуже багато інформації, після чого вона структурується й обробляється.

Метою роботи є дослідження особливостей моделювання та аналізу заради створення гнучкого програмного комплексу для комфортних та простих умов обслуговування абонентів та прискорення процесу взаємодії між клієнтом та оператором.

Веб-орієнтована система оператора мобільного зв'язку – це підсистема, яка дає змогу розширити можливості білінгу та вивести клієнтів на самообслуговування. Користувач повинен мати змогу самостійно замовляти або відмовлятися від послуг мобільного зв'язку; змінювати тарифний план або подавати заявки на участь у акціях (у тому числі налаштовувати індивідуальний тариф); контролювати залишки коштів по кожному особовому рахунку і поповнювати рахунок, тощо.

Основними етапами розробки платформи є:

1. Розробка дизайну вебсайту:

створення загальної концепції дизайну додатку;

створення UI компонентів;

розробка дизайну сторінок.

2. Побудова архітектури клієнтської частини додатку.

3. Побудова архітектури серверної частини додатку.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

4. Розробка системи авторизації та ролей користувачів.
5. Розробка основного функціоналу:
 - створення логіки авторизації користувачів;
 - створення віддаленої бази даних для зберігання інформації;
 - імплементация бекенду на клієнтській стороні;
 - організація спілкування серверної частини з клієнтською.
6. Тестування додатку:
 - функціональне тестування додатку;
 - тестування графічного інтерфейсу.

1.2. Аналіз веб-орієнтованих систем операторів мобільного зв'язку в Україні

Розробка системи оператора мобільного зв'язку потребує аналізу конкурентів на ринку. В Україні працюють три найбільші мобільні оператори [20]: ПрАТ «Київстар» [19], Vodafone (ПрАТ «ВФ Україна») [11], lifecell (ТОВ «лайфселл») [24]. Всі вони мають застосунки для самообслуговування користувачів.

Kyivstar.Tech – українська ІТ-компанія, що спеціалізується на розробленні високоякісних технологічних продуктів і рішень для бізнесу – розробила та розвиває застосунок «Мій Київстар» [19]. Найбільша цінність застосунку полягає в тому, щоб користувачі отримали доступ до всіх телеком-послуг та сервісів, які надає «Київстар», не виходячи з дому. На сьогодні є лише застосунок для Android, iOS (рис. 1.1). Вебверсія з 2023 року закрыта, але є окрема для бізнесу.

Платформа від «Київстар» є провідною на ринку телеком-послуг [26]. Застосунок має гарний дизайн, який є приємним користувачу й дозволяє не лише зручно користуватись послугами мобільного зв'язку, але й відкриває користувачам світ розваг та дає можливість брати участь у різноманітних акціях.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

На відміну від аналогічних додатків інших операторів [15], у додатку Мій Київстар є можливість слухати радіо, дивитися телеканали на Київстар ТВ та переглядати стріми ігрових турнірів без тарифікації трафіку.

Серед інших ключових особливостей – можливість додати до 10 номерів телефонів, керувати ними і здійснювати всі ті ж операції, що і з власним номером; замовити eSIM в один клік, чи підключити Домашній Інтернет.

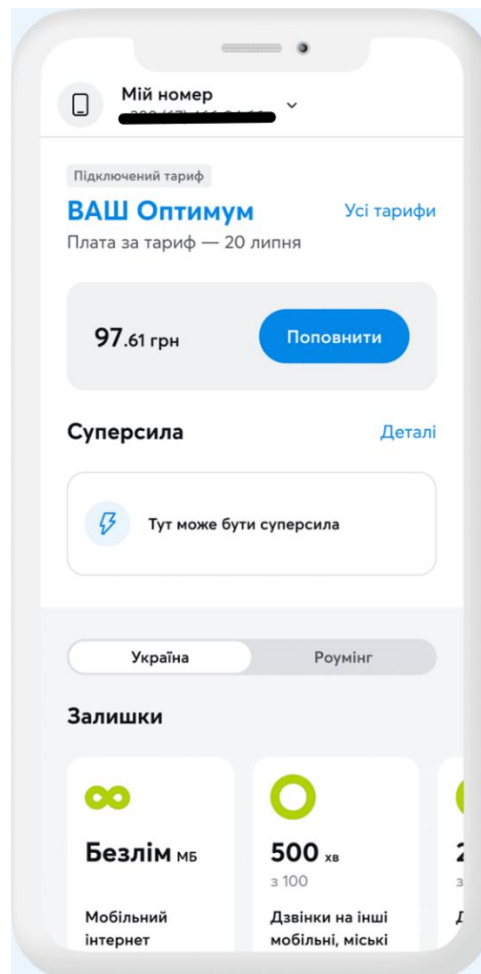


Рис.1.1. Мій Київстар

Додаток – це особистий помічник для абонента, ним можуть скористатися всі клієнти. Натомість вебплатформа доступна лише для бізнесу. Тут можна керувати номерами, формувати звіти, налаштовувати групові пропозиції, підключати послуги для співробітників. Фахівці компанії збирають відгуки, аналізують поведінку користувачів і в основу оновлень покладають зібрані дані, таким чином,

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

застосунок адаптується під кожного. Вигляд програми залежить від сегменту й стилю користування.

Один із пріоритетів – дати користувачеві можливість вирішити будь-яке питання без звернення до оператора. Для цього розробляють розумного чат-бота в застосунку. Він зможе швидко допомогти й замінити дзвінок у кол-центр.

«Мій Київстар» першим в Україні реалізував можливість перейти на контракт через ідентифікацію в «Дії» та BankID. Доступ до вебплатформи Мій Київстар мають лише координатори компаній, які користуються зв'язком Київстар. Щоб отримати права доступу, необхідно зареєструватись як координатор.

Застосунок My Vodafone доступний як мобільний застосунок (рис. 1.2) і як вебверсія (рис. 1.3), які мають схожий вигляд і функціонал (для приватного сегменту) [11].

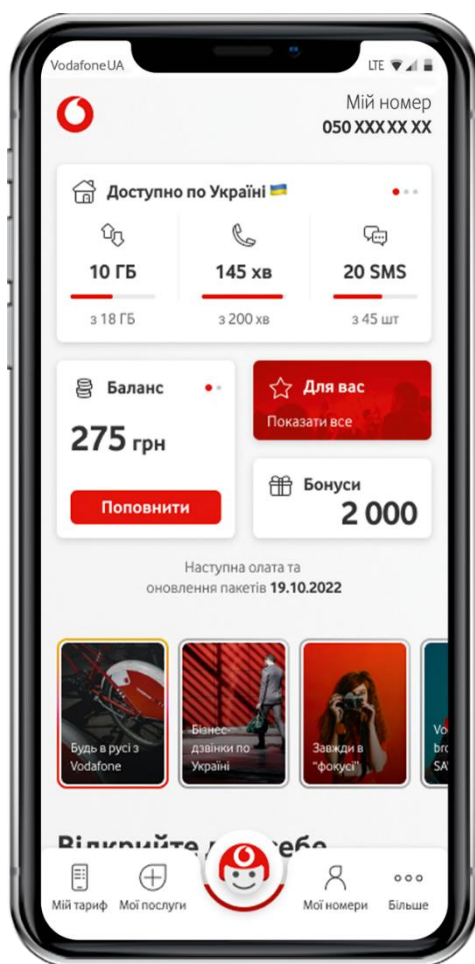


Рис.1.2. My Vodafone (мобільний застосунок)

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Розробник заявляє про такі функції: перевіряти залишок ГБ, хвилин та SMS; поповнювати рахунок вчасно та без комісії; піклуватись про своїх близьких; обмінювати бонуси на корисні послуги; обирати зручний для себе тариф; переглядати персональні пропозиції; звертатись до розумного помічника ТОБі; можна встановити PIN-код при запуску додатка, щоб зробити вхід більш безпечним.

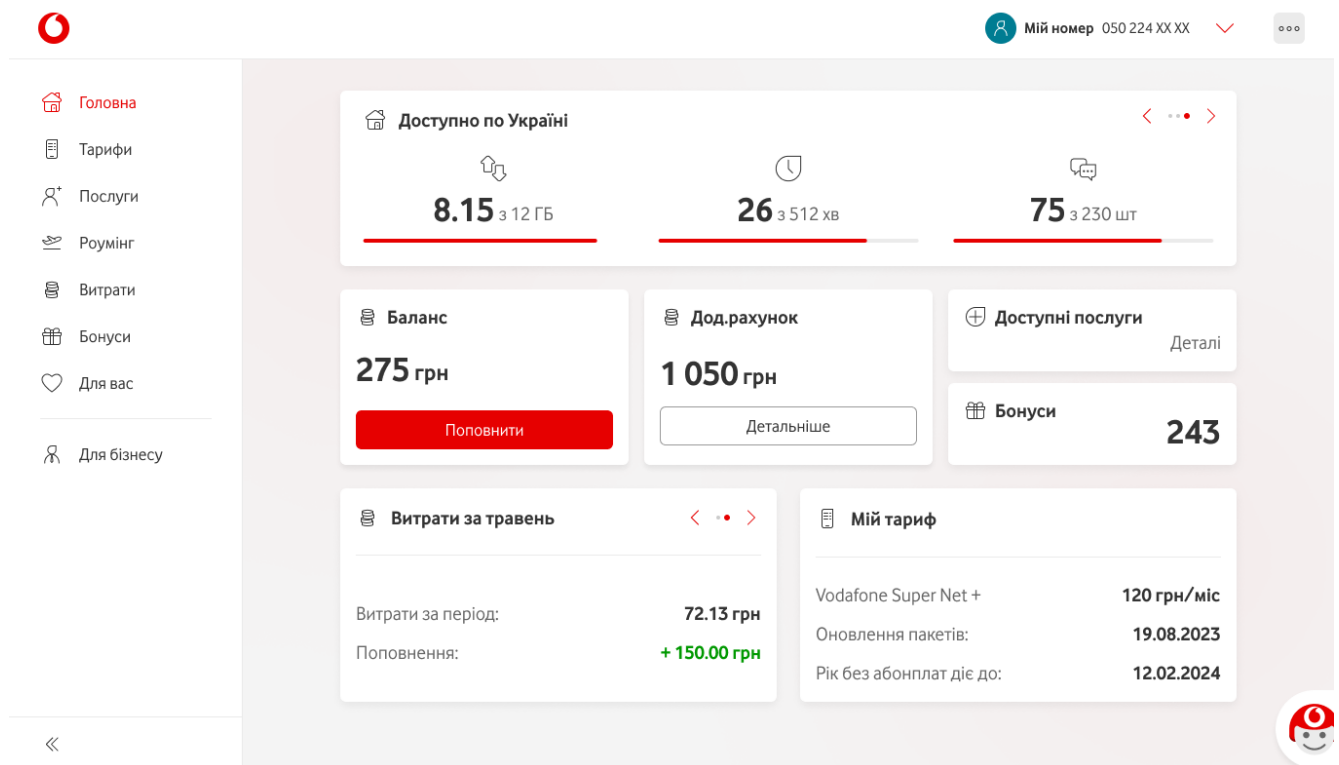


Рис. 1.3. My Vodafone (desktop)

Доступ до вебплатформи мають і організації (корпоративні клієнти) [11]. У цьому випадку передбачені чотири рівні доступу: адміністратор, коли абонент керує всіма номерами організації; адміністратор особового рахунку передбачає керування номерами особового рахунку; адміністратор номеру – керує тільки своїм номером; користувач – переглядає інформацію за своїм номером і керує мультиакаунтом для особистих номерів, але не може змінювати умови тарифу і підключати послуги.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

Бізнес-клієнти можуть робити заявки на підключення кількох номерів, змінювати тарифний план і параметри послуг одночасно для групи номерів, що особливо актуально для сервісів IoT.

My lifecell доступний як мобільний застосунок (рис. 1.4) і як вебверсія (рис. 1.5), що сильно різняться, а спільним є лише витягування даних за номером [24].

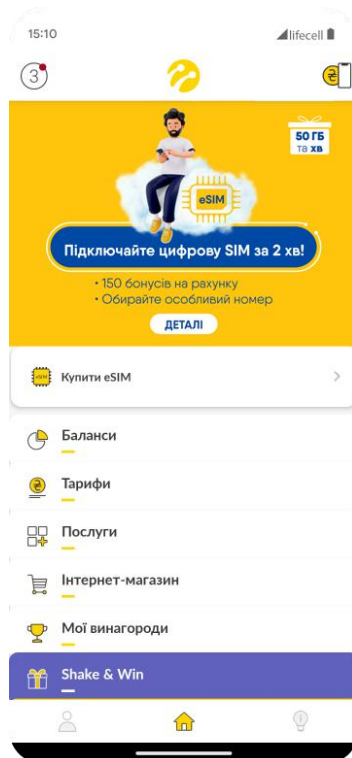


Рис. 1.4. My lifecell (мобільний застосунок)

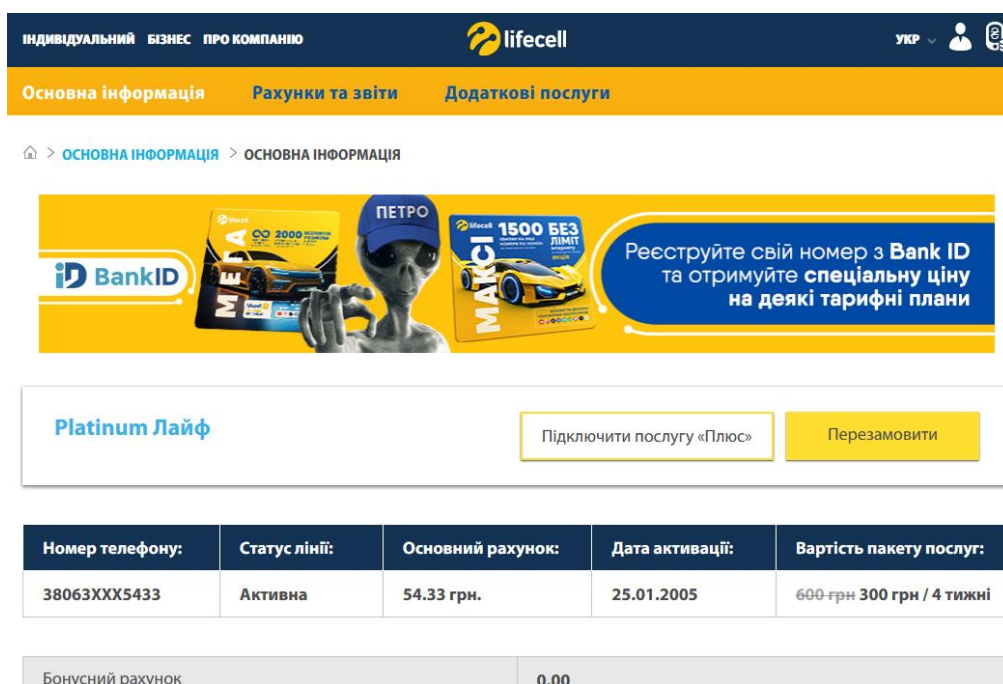


Рис. 1.5. My lifecell (desktop)

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк. 15
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Мультиакаунт реалізовано в усіх проаналізованих операторів, lifecell реалізує це найцікавіше. У мобільному застосунку від lifecell можна керувати номерами «Мобільної турботи», можна підключити до 10 номерів. Варто зауважити, що ви маєте вибрати вид доступу: повний, що здійснюється через єдиний пароль або обмежений, за якого інший абонент отримає SMS, відповівши на яке, він надасть дозвіл на управління та доступ до інформації свого номера.

Виходячи з проведеного аналізу програмних продуктів мобільних операторів України було виявлено ряд переваг та недоліків, які сформуємо у порівняльну таблицю (див. табл. 1.1).

Таблиця 1.1

Порівняння веб-орієнтованих систем операторів України

Характеристика	Мій Київстар	My Vodafone	My lifecell
Вебверсія для приватних користувачів	-	+	+
Інтеграція з сайтом оператора	+	+	+/-
Мультиакаунт (підключення кількох номерів для керування)	+	+	+
Код доступу для заходу в додаток	+	+	-
Зручність використання в цілому	+	+	+/-

В результаті порівняльного аналізу визначено, що вебзастосунки для всіх користувачів мають тільки Vodafone й lifecell, у той час як вебплатформа від Київстар наявна тільки для юридичних осіб. Приватні користувачі всіх операторів можуть користуватися мультиакаунтом для перегляду інформації про різні особисті чи сімейні номери.

1.3. Обґрунтування вибору інструментальних засобів

Веб-орієнтований програмний комплекс оператора мобільного зв'язку буде розроблено з використанням клієнт-серверної архітектури, що забезпечить ефективне розподілення навантаження між клієнтською та серверною частинами системи [3,4]. Такий підхід дозволить досягти високої продуктивності, гнучкості у розширенні функціональності, а також забезпечити безпеку та стабільність під час обробки великої кількості запитів та користувачів. Основними технологіями, що будуть використані при реалізації цього програмного комплексу, є Spring Boot, React та PostgreSQL. Кожна з цих технологій відіграє ключову роль у створенні ефективної, масштабованої та зручної у користуванні системи [7].

Для створення серверної частини буде застосовуватися фреймворк Spring Boot у поєднанні з реляційною базою даних PostgreSQL. Spring Boot є потужним засобом для створення надійних вебзастосунків з використанням мікросервісної архітектури. Його основним перевагою є можливість швидкого запуску проєктів, автоматичне конфігурування, а також широка підтримка стандартів REST API, що забезпечує зручну взаємодію між клієнтом та сервером. Завдяки Spring Boot можна ефективно реалізувати бізнес-логіку системи [12], настроїти авторизацію та аутентифікацію користувачів, інтегрувати зовнішні сервіси, забезпечити обробку великих обсягів запитів тощо [13].

PostgreSQL є однією з найпотужніших сучасних систем управління базами даних (СКБД), яка дозволяє надійно зберігати та обробляти дані. Вона забезпечує високу продуктивність, підтримку транзакцій, складні запити на мові SQL, механізми розмежування доступу до даних, шифрування, а також має широкі можливості розширення (наприклад, за допомогою написання власних функцій). PostgreSQL добре підходить для систем із високим навантаженням, таких як платформи мобільного зв'язку, де потрібна обробка великої кількості користувачів, їх запитів, історії взаємодії з послугами та іншої супутньої інформації [17,18].

Поєднання Spring Boot та PostgreSQL створює стабільну, масштабовану та безпечну серверну частину програмного комплексу [5]. Така архітектура дозволяє

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

легко адаптувати систему до нових вимог, розширювати її функціонал без повної перебудови, а також ефективно обробляти запити у реальному часі.

Для створення клієнтської частини буде використано React – сучасну JavaScript-бібліотеку для побудови інтерфейсів користувача. React відзначається високою продуктивністю, завдяки застосуванню віртуального DOM, та декларативним підходом до розробки, що дозволяє описувати бажаний стан інтерфейсу, а не послідовність дій для його побудови. Це значно полегшує розробку, тестування та підтримку інтерфейсу [27].

React дозволяє створювати компоненти, що повторно використовуються, а також ефективно працювати з динамічними даними, які змінюються залежно від дій користувача або стану системи. Наприклад, у вебінтерфейсі оператора мобільного зв'язку можуть відображатися особисті дані користувача, активні послуги, поточний баланс, історія поповнень, статистика використання мобільного трафіку, SMS, дзвінків тощо. Реалізація цих функцій за допомогою React дозволяє досягти високої швидкодії та зручності у користуванні [28].

Крім того, React добре інтегрується з REST API, створеним на Spring Boot, що забезпечує ефективну взаємодію між фронтендом і бекендом. Це дозволяє забезпечити узгоджене відображення даних, миттєве реагування на дії користувача, а також мінімізувати затримки при оновленні інформації на сторінці.

Однією з основних переваг клієнт-серверної архітектури є розподіл відповідальності між клієнтом та сервером [16]. Це забезпечує ефективніше використання ресурсів, дозволяє масштабувати систему, підтримувати високу продуктивність та забезпечувати безперебійну роботу навіть при великій кількості одночасних користувачів. Іншими перевагами є: гнучкість і масштабованість, безпека та стійкість до збоїв, інтерактивність та швидкодія, простота оновлення та супроводу [29].

Гнучкість і масштабованість забезпечується завдяки використанню Spring Boot та PostgreSQL. Завдяки їм систему можна адаптувати до різних рівнів навантаження, додавати нові сервіси без повного переписування коду. Безпека та

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

стійкість до збоїв забезпечується PostgreSQL, яка забезпечує захист даних за допомогою механізмів шифрування, резервного копіювання та журналювання транзакцій [30]. Крім того Spring Boot має потужні засоби аутентифікації та авторизації, що дозволяє обмежити доступ до конфіденційної інформації. Інтерактивність та швидкодія забезпечується React, яка дозволяє створити інтерфейс, що швидко реагує на запити користувача, з мінімальними затримками, завдяки оптимізації рендерингу та роботи з даними. Простота оновлення та супроводу забезпечується мікросервісною архітектурою, яку підтримує Spring Boot та дозволяє оновлювати окремі компоненти системи незалежно від решти, що значно спрощує процес супроводу та масштабування.

Хоча конкретні мобільні оператори рідко публічно розкривають повний технологічний стек своїх внутрішніх систем, відомо, що Spring Boot, React та PostgreSQL активно використовуються в IT-компаніях і телекомунікаційній галузі [22]. Зокрема, вони часто застосовуються у розробці CRUD-додатків (Create, Read, Update, Delete), що дозволяють керувати даними користувачів, послугами, тарифними планами, історією транзакцій тощо [6].

Такі технології можуть бути інтегровані у внутрішні CRM-системи, платформи самообслуговування для абонентів, панелі адміністрування та інші інструменти, що забезпечують щоденну роботу операторів мобільного зв'язку [23]. Їхнє використання дозволяє компаніям бути більш гнучкими, адаптуватися до потреб ринку, швидко впроваджувати нові функції та підтримувати високу якість обслуговування клієнтів.

Отже, обрана архітектура та набір технологій (Spring Boot, React, PostgreSQL) є сучасним, ефективним та перевіреним рішенням для створення веб-орієнтованого програмного комплексу оператора мобільного зв'язку. Такий підхід дозволить досягти високого рівня продуктивності, безпеки, масштабованості та зручності у використанні як для кінцевих користувачів, так і для внутрішніх операторів системи. Розробка з використанням мікросервісної клієнт-серверної моделі

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

забезпечить гнучкість, що є критично важливою в умовах стрімкого розвитку технологій та зростаючих вимог ринку телекомунікацій.

Висновки до першого розділу

Проведено аналіз предметної області для визначення основних вимог до платформи в системах оператора мобільного зв'язку.

В процесі дослідження аналогічних і схожих за концепцією програмних продуктів були проаналізовані та порівняні різні платформи з різними принципами роботи, основним функціоналом яких є перегляд інформації про стан рахунку й керування ним.

Крім цього, були сформовані основні вимоги для технологічного стеку, що включає Spring Boot і PostgreSQL у серверній частині, React у клієнтському інтерфейсі. Запропоновані рішення забезпечують високу продуктивність та безпеку для вебдодатку, масштабованість та гнучкість.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ ОПЕРАТОРА МОБІЛЬНОГО ЗВ'ЯЗКУ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Проведений аналіз аналогічних вебдодатків мобільних операторів України та потреб користувачів, створив підстави для визначення варіантів використання веб-орієнтованої системи оператора мобільного зв'язку. Для розробки моделей будемо використовувати уніфіковану мову моделювання UML [25]. На рис. 2.1 наведено діаграму варіантів використання системи, що пропонується. У системі передбачено два ключові актори – користувач та власник додатка.

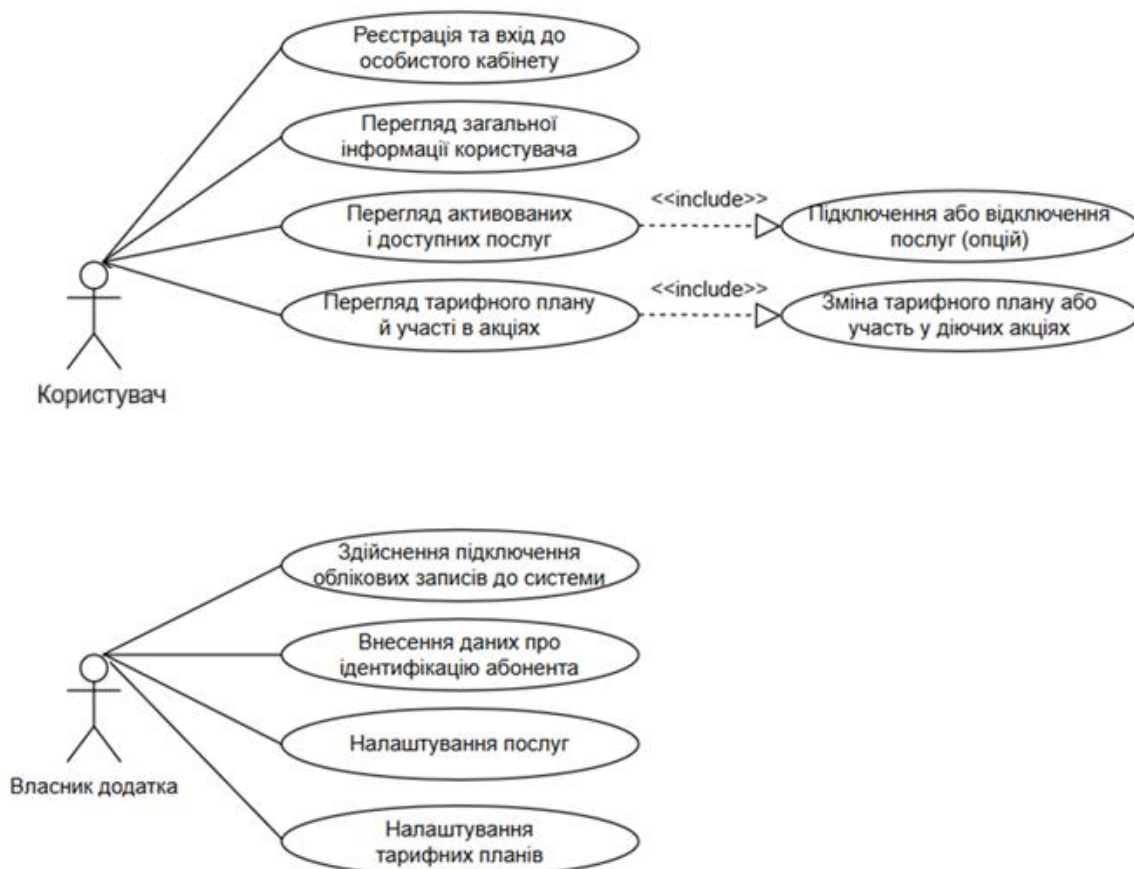


Рис. 2.1. Діаграма варіантів використання системи

Короткий опис акторів представлено в табл. 2.1. В системі визначено два типи користувача, а саме внутрішній користувач – власник додатка. Цю роль може

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

виконувати адміністратор системи, інженер-програміст та інші представники ІТ-відділу, які відповідають за функціонування даної системи.

Функціонал додатку для цієї ролі включає (див. рис. 2.1): здійснення підключення облікових записів до системи; внесення даних про ідентифікацію абонента; налаштування послуг для пропонування абонентам; налаштування тарифних планів для пропонування абонентам.

Таблиця 2.1

Опис ролей користувачів (акторів)

Актор	Короткий опис
Користувач	Може самостійно замовляти або відмовлятися від послуг мобільного зв'язку; змінювати тарифний план або подавати заявки на участь у акціях; проводити комплексний аналіз поточних витрат на послуги мобільного зв'язку та витрати за минулі розрахункові періоди; отримувати повну та структуровану інформацію про надані послуги мобільного зв'язку та витрати на них; контролювати залишки коштів по кожному особовому рахунку.
Власник додатка	Переглядає загальну статистику сайту, підключає абонента до системи через сервер, налаштовує параметри тарифних планів/пакетів послуг, що пропонуються абонентам. Не має доступу до користувацької частини сайту.

Також в системі передбачено роль «Користувач», що призначена безпосередньо для абонентів. В системі передбачено:

- процедуру авторизації;
- перегляд загальної інформації для користувача;
- перегляд активованих і доступних послуг, керування ними;
- перегляд тарифного плану та активної участі в акціях, керування ними.

Опис варіантів використання системи наведено в табл. 2.2.

Виявлення варіантів використання

Актор	Назва варіанту використання	Формулювання
Користувач	Реєстрація та вхід до особистого кабінету	Цей варіант дозволяє вводити свої облікові дані та отримати доступ до системи.
	Перегляд загальної інформації користувача	Цей варіант дозволяє бачити особисті дані абонента, в тому числі баланс особового рахунка.
	Перегляд активованих і доступних послуг/ Підключення або відключення послуг	Цей варіант дозволяє переглядати, замовляти або відмовлятися від послуг мобільного зв'язку.
	Перегляд тарифного плану й участі в акціях / Зміна тарифного плану або участь у діючих акціях	Цей варіант дозволяє переглядати умови обслуговування за тарифним планом, змінювати тарифний план або подавати заявки на участь у акціях.
Власник додатка	Здійснення підключення облікових записів до системи	Цей варіант дозволяє підключати користувачів або групу користувачів до системи для реалізації реєстрації.
	Внесення даних про ідентифікацію абонента	Цей варіант дозволяє вносити особисті дані абонента за його запитом у ручному режимі.
	Налаштування послуг	Цей варіант дозволяє встановлювати додаткові послуги та їхні параметри (вартість підключення, абонентна плата, пакет трафіку тощо).
	Налаштування тарифних планів	Цей варіант дозволяє створювати тарифні плани та встановлювати їхні параметри (тарифи на спожитий трафік, вартість переходу, абонентна плата, можливий пакет послуг, період оплати).

Функціональність розроблюваного програмного комплексу передбачає надання постійного доступу співробітникам оператора мобільного зв'язку (GSM/UMTS) до актуальної інформації щодо стану розрахунків окремих абонентів. Основою для формування вхідних даних виступає інформація з телекомунікаційної системи, що відображає обсяги споживаного трафіку.

Передбачається, що до системи матимуть доступ до 500 спеціалізованих робочих місць, а загальна кількість потенційних клієнтів, дані яких оброблятимуться, становитиме до 20 мільйонів абонентів. Загальний обсяг даних, що зберігатиметься у базі даних, оцінюється на рівні приблизно 5 мільярдів записів.

Система повинна бути доступною безперервно – 24 години на добу, 7 днів на тиждень, з високими показниками надійності та відмовостійкості.

У режимі on-line система повинна забезпечувати виконання таких запитів:

- подача заявок на зміну чинного тарифного плану або на підключення до акційних пропозицій, що діють на момент звернення;
- редагування переліку послуг мобільного зв'язку, зокрема підключення або відключення додаткових сервісів;
- перегляд детальної та структурованої інформації про отримані послуги мобільного зв'язку, включно з розшифруванням витрат, залишком коштів на абонентському номері та станом особового рахунку;
- аналіз поточних витрат на телекомунікаційні послуги, а також доступ до історії витрат за попередні розрахункові періоди;

Розробка системи повинна враховувати вимоги до високої продуктивності, масштабованості та швидкого реагування на дії користувача, з урахуванням великого обсягу даних і навантаження на систему.

Нефункціональні характеристики системи охоплюють низку важливих аспектів, що визначають зручність, надійність, продуктивність та придатність до експлуатації. Інтерфейс користувача повинен бути інтуїтивно зрозумілим і простим у використанні. Очікується, що звичайні користувачі зможуть опанувати роботу із системою протягом 2–3 годин, тоді як досвідченим користувачам на це знадобиться

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

не більше однієї години. Інформаційна взаємодія повинна бути комфортною, з можливістю отримання підказок та повідомлень, що полегшують навігацію. Середній час відгуку системи на типові запити не повинен перевищувати 15 секунд, а при обробці більш складних запитів – 30 секунд. Це забезпечить швидку реакцію на дії користувачів і ефективне обслуговування в режимі реального часу.

З точки зору надійності система повинна функціонувати стабільно та безперебійно, допускаючи лише незначні тимчасові перерви на технічне обслуговування, загальна тривалість яких не повинна перевищувати 5% від загального часу роботи. Програмний код має бути реалізований на високому рівні якості, з мінімальним рівнем помилок, що не перевищує одного дефекту на кожні десять тисяч рядків коду. Система також повинна забезпечувати одночасну роботу щонайменше ста активних користувачів, які мають доступ до централізованої бази даних. Продуктивність при цьому має залишатися стабільною і не погіршуватись із ростом навантаження.

Важливою є також експлуатаційна придатність, що передбачає можливість поступового розширення системи. Архітектура має бути масштабованою, тобто допускати збільшення кількості користувачів та обсягів даних без значних змін у структурі або втрати продуктивності. Розширення функціоналу не повинно впливати на стабільність роботи, що дозволить адаптувати систему до потреб бізнесу в довгостроковій перспективі.

На рис. 2.2 наведено діаграму класів з визначенням основних сутностей та їх зв'язків у системі. Основні сутності системи виступають: User – користувач; SubData – особові дані; RatePlan – тарифний план; Service – послуга; ServPack – пакет послуг.

Між об'єктами «User» і «RatePlan»; «User» і «Service»; «Service» і «ServPack» використовується зв'язок агрегації, адже видалення об'єкта-контейнера не призводить до видалення об'єктів, які він зберігав. Агрегація – це спеціалізований різновид асоціації, що описує зв'язки один-до-багатьох, багато-до-багатьох, частина-ціле між декількома об'єктами. Зазвичай під час агрегації один об'єкт

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

містить інші, тобто виступає контейнером або колекцією. Тут контейнер не керує життєвим циклом компонентів і компоненти цілком можуть існувати окремо від контейнера.

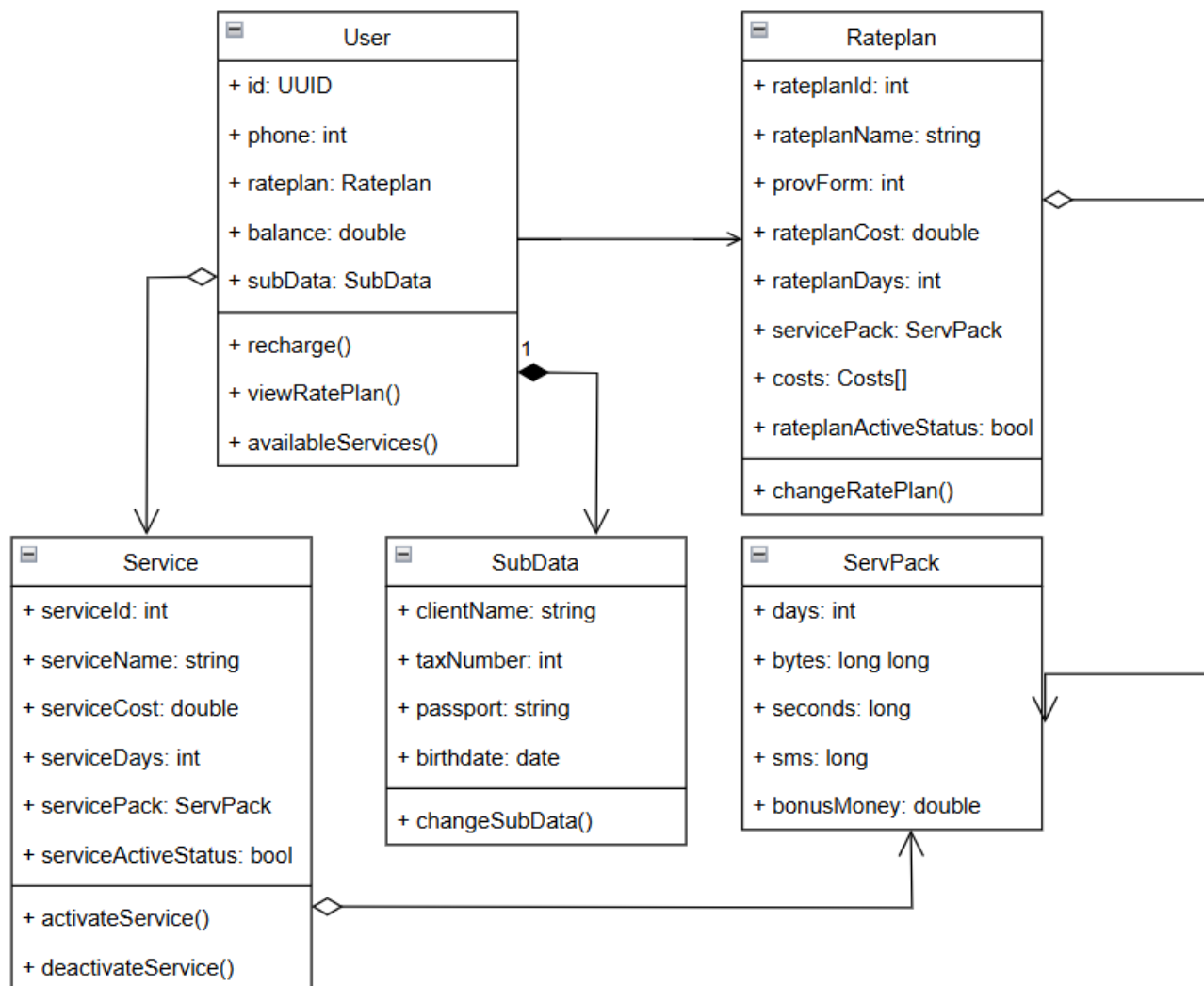


Рис.2.2. Діаграма класів

Між об'єктами «User» і «SubData» на діаграмі класів зображено композицію, це більш суворий варіант агрегації, коли компонент може існувати лише як частина контейнера. Коли буде видалено об'єкт-контейнер, то потрібно видалити й об'єкти, які він зберігав.

Асоціація між «User» і «RatePlan» передбачає, що абонент одночасно перебуває рівно на одному тарифному плані, а з одним тарифним планом пов'язується множина абонентів. Параметри тарифного плану включають назву

(rateplanName), форму обслуговування (provForm), абонентну плату за певний період (rateplanCost), кількість днів циклу (rateplanDays), вартість одиниці трафіку того чи іншого виду електронної комунікаційної послуги (costs), можливість підключення (rateplanActiveStatus). Тарифний план може містити пакет послуг (servicePack у типі ServPack), що надається у разі зняття абонентної плати і включає надання на певний період визначеної кількості інтернет-трафіку, голосового трафіку чи SMS, іноді може надавати обсяг бонусних коштів. Пакет послуг може не надаватися, тоді застосовується стандартна тарифікація в тарифному плані (costs).

2.2. Розробка функціональних алгоритмів роботи системи

Діаграми активності, разом з діаграмами варіантів використання та діаграмами станів, вважаються діаграмами поведінки, оскільки вони описують те, що має відбуватися в системі, що моделюється [21]. Діаграма активності (діаграма діяльності) допомагає моделювати послідовності дій у бізнес-процесах [948]. Вона показує, як методи або дії взаємодіють між собою. Діаграми активності схожі на блок-схеми. Вони представлені у вигляді графу, де дії – це вузли, а зв'язки між ними – переходи [1].

Кожний стан на діаграмі активності відповідає виконанню деякої елементарної операції, а перехід в наступний стан виконується тільки після завершення цієї операції [8]. Таким чином, діаграму активності можна вважати приватним випадком діаграми станів. Діаграма активності, що представлена на рис. 2.3 показує процес авторизації та реєстрації в системі, зокрема від введення логіну користувача та пароля, а також реєстраційної форми, до успішного входу в систему. Під час авторизації користувач може обрати один із варіантів «Вхід до системи» чи «Реєстрація».

Якщо акаунт є, то користувач вводить дані облікового запису, якщо ні, то заповнює реєстраційну форму. Користувачу буде виведено сторінку з формою для входу, де він має ввести свої облікові дані (в електронних комунікаційних інтернет-

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

системах обслуговування логін, як правило, відповідає абонентському номеру чи особовому рахунку).

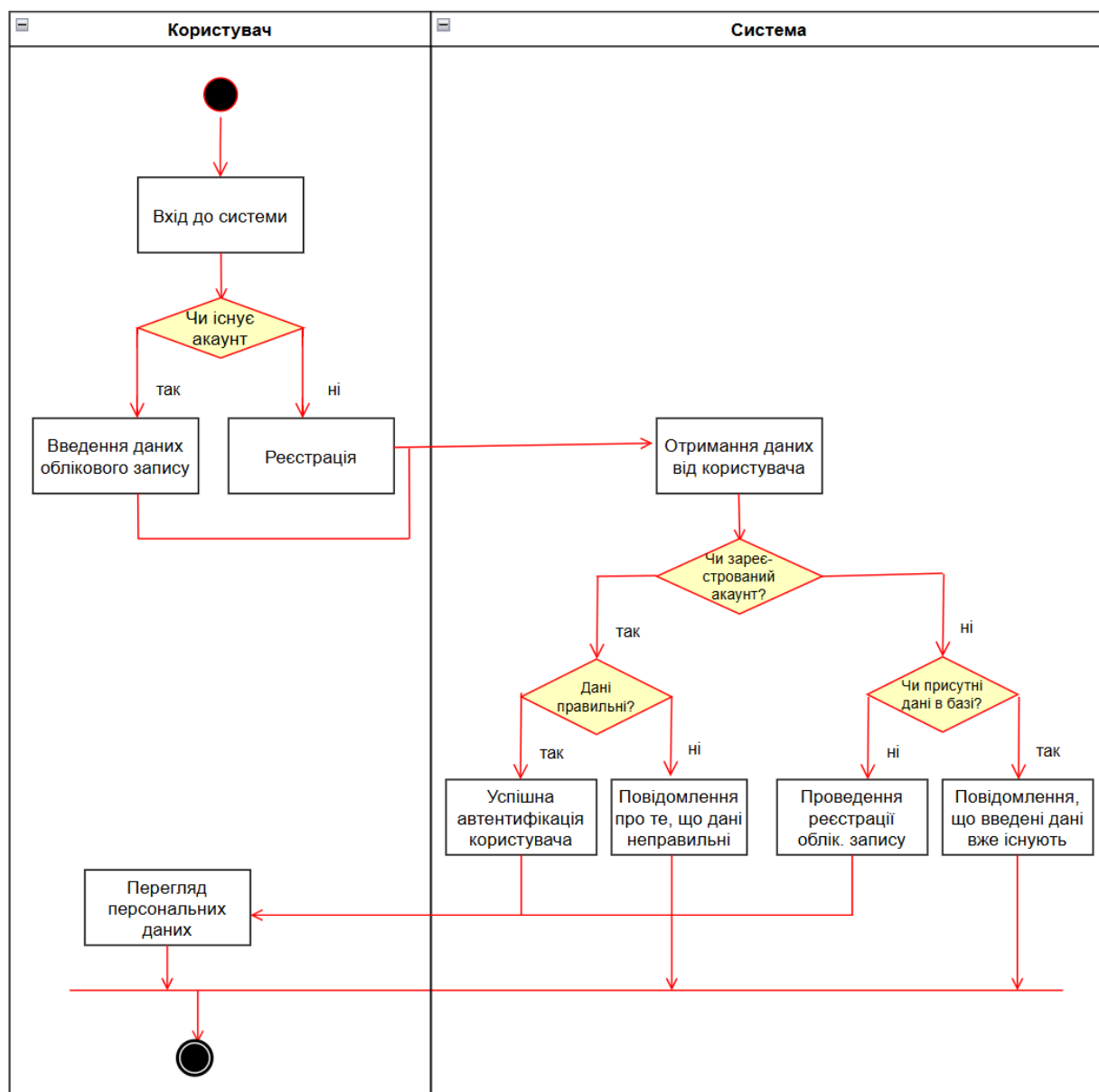


Рис. 2.3. Діаграма активності для процедури авторизації

Система перевіряє отримані дані від користувача. Якщо акаунт зареєстровано і дані введено правильно, то користувачу відкриваються персональні дані для перегляду, тобто він отримує доступ до функціоналу системи за своїм логіном. Якщо ж, дані введено неправильно але акаунт зареєстровано, то виводиться інформаційне повідомлення про неправильність введення даних, користувач має

спробувати повторно ввести свій логін та пароль. Якщо акаунт користувача відсутній, то проводиться перевірка присутності даних в базі даних. Якщо дані відсутні, то відбувається реєстрація облікового запису, інакше – видається повідомлення про те, що користувач з такими даними вже зареєстрований.

Якщо було обрано варіант «Реєстрація», то відобразиться реєстраційна форма, де користувач має заповнити всі обов’язкові поля так, щоб було пройдено валідацію. Після валідації даних на клієнті, вони відправляються на сервер і там також відбувається перевірка на правильність даних, наприклад на відсутність дублювання даних тощо.

На рис. 2.4 зображено діаграму активності, за якою відбувається процес підключення послуг (опцій).

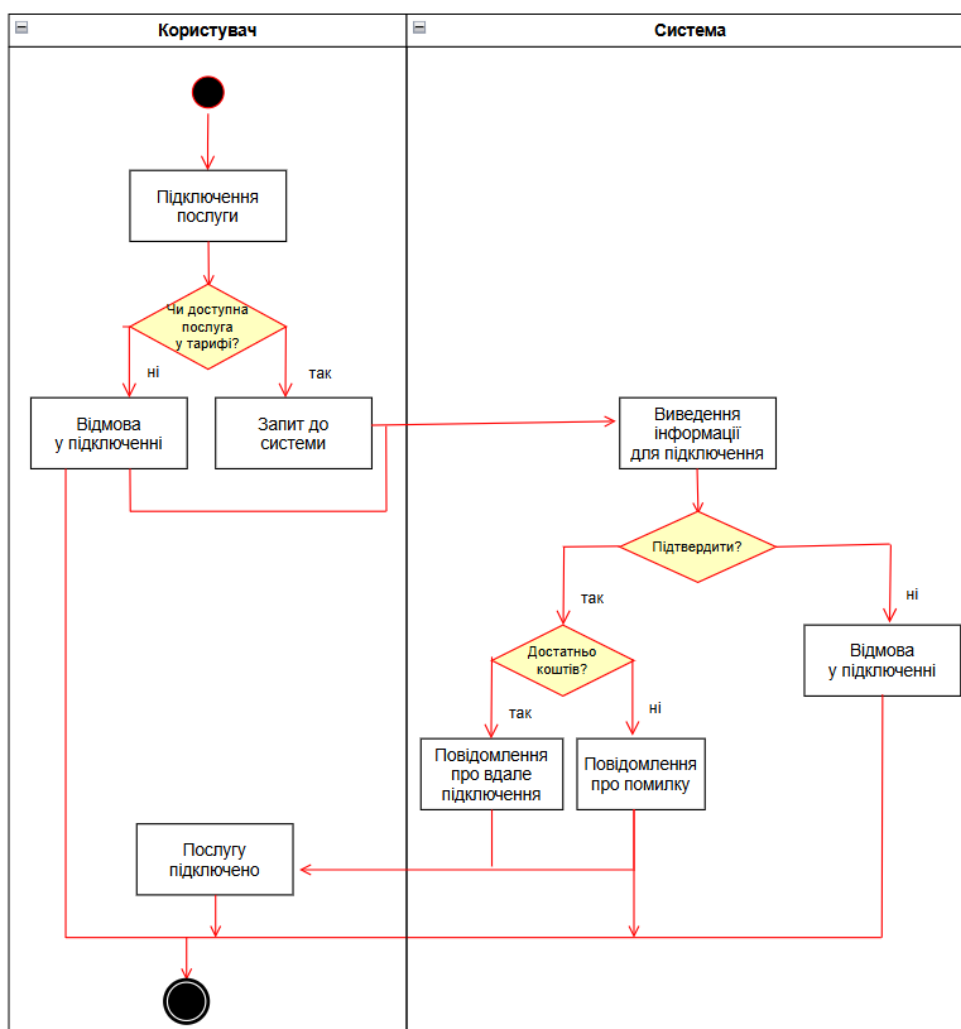


Рис. 2.4. Діаграма активності для підключення послуги

Діаграма активності демонструє можливість взаємодії з системою у процесі підключення послуги [10]. Окремі послуги можуть бути доступними не на всіх тарифних планах та (або) лише за умов сплати за пакет послуг, тож відбувається перевірка на сумісність із поточним тарифним планом абонента. Система відображає відомості про бажану послугу й повідомляє користувачу про її назву, принцип роботи й вартість підключення [26]. Після підтвердження користувачем, передбачається надсилання електронного комунікаційного повідомлення про успішне замовлення. При підключенні платної послуги необхідно мати достатній позитивний залишок балансу, в іншому випадку система повідомляє про помилку. Майже аналогічно відбувається процес зміни тарифного плану.

Діаграми послідовності, кооперації та взаємодії компонентів є важливими інструментами для документування та аналізу поведінки системи [2]. Вони можуть використовуватися для того, щоб показати, як система реагує на певні події, як компоненти співпрацюють для виконання певного завдання та як компоненти системи взаємодіють один з одним [14]. На рис. 2.5 представлено діаграму послідовності дій при вході користувача в систему.

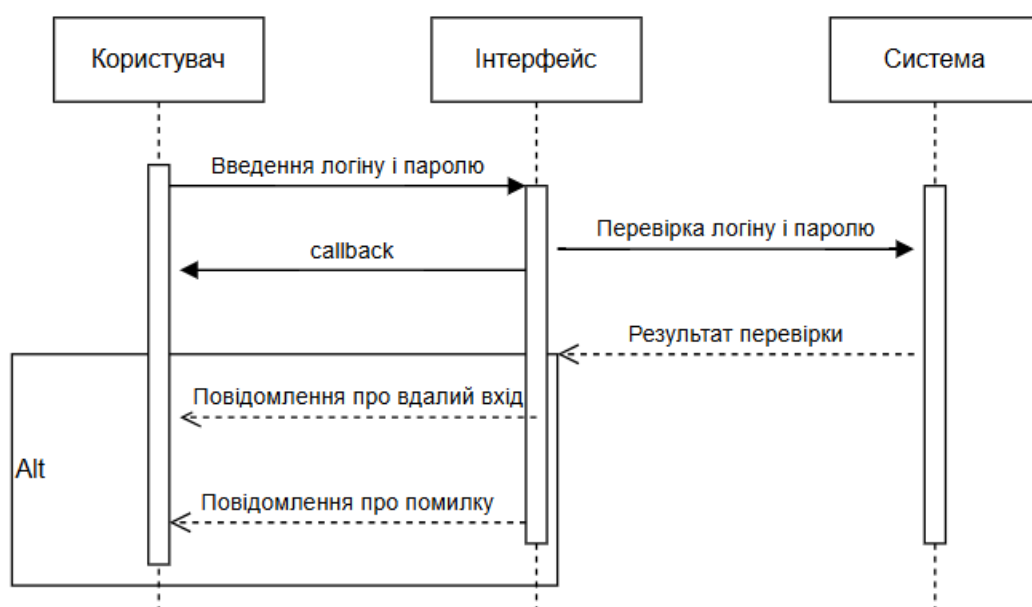


Рис. 2.5. Діаграма послідовності дій при вході користувача в систему

Клієнтська сторона представлена браузером або застосунком на мобільній операційній системі, через який здійснюється доступ до системи. Браузер взаємодіє

з серверною частиною через HTTPS-запити. Серверна сторона включає три основні компоненти: Nginx Server, вебзастосунок, базу даних. Взаємодія між компонентами відбувається у внутрішній мережі Docker.

2.3. Програмна реалізація функціональних алгоритмів роботи системи

При проєктуванні системи було використано принципи розділення програмного забезпечення на різні рівні [10]. Усі головні компоненти системи чітко розділені та інкапсульовані, утворюючи кілька шарів з односторонніми залежностями [31].

Domain. Це ядро, фундамент системи, навколо нього будується весь проєкт. Цей шар не має жодних залежностей від зовнішніх і містить лише сутності, перерахування, константні дані.

Application. Разом із доменним рівнем (Domain) прикладний рівень (Application) утворює ядро рішення, яке повинно мати можливість працювати та забезпечувати бізнес-логіку незалежно від зовнішніх рівнів і залежати виключно від Domain.

Infrastructure. Це рівень, на якому має бути реалізована вся логіка зв'язку із зовнішніми системами, як-от зв'язок із стороннім API. Він залежить лише від Application та реалізує його інтерфейси.

Persistence. Містить логіку для зв'язку із зовнішніми системами, але його конкретна мета – зв'язуватися з базами даних. Цей рівень залежить від Application та Domain.

API. Це рівень взаємодії із зовнішнім світом, який дозволяє клієнтам отримувати видимі результати після запиту даних (програмний інтерфейс). Він також залежить лише від Application, але також може використовувати елементи з Persistence.

Web. Найвищий рівень відповідає за реалізацію клієнтської частини, яка може включати в себе фронтенд (HTML, CSS, JavaScript) для створення користувацького вебінтерфейсу. Користувачі можуть використовувати такий

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

компонент для взаємодії з системою за допомогою HTTPS запитів (всі дані в системі мають бути захищені).

Код реалізації події на підключення/відключення має вигляд:

```
const onSaveServiceEnabled = (event) => {
  event.preventDefault();
  setCanSave(true);
  console.log(formData);
  dispatch(editService(formData));
  toast("Послугу підключено");
}

const onSaveServiceDisabled = (event) => {
  event.preventDefault();
  setCanSave(true);
  console.log(formData);
  dispatch(editService(formData));
  toast("Послугу відключено");
}

return (
  <section className='service-single-section px-4'>
    <div className='service-single-title'>
      <h2 className='my-2 fs-20'>{tempService[0].serviceTitle}</h2>
    </div>
    <form className='note-form'>
      <p>Вартість послуги <b>{tempService[0].servicePrice}</b>
грн</b>/{tempService[0].servicePeriod} дн.</p><br />
      <p>{tempService[0].serviceContent}</p>
      { tempService[0].serviceEnabled !== 0 ? <button type = "button"
onClick={onSaveServiceDisabled} className = "btn btn-default" disabled =
{!canSave}>Відключити</button> : <button type = "button"
onClick={onSaveServiceEnabled} className = "btn btn-default" disabled =
{!canSave}>Підключити</button> }
      <ToastContainer />
    </form>
  </section>
)
```

Програмна реалізація форми додавання послуг має наступний вигляд:

```
<div>
  <section className='note-form-section'>
    <h2 className='my-4 fs-16'>Додати послугу</h2>
    <form className='note-form'>
      <div className='form-element'>
        <label htmlFor='serviceTitle' className='form-label'>Назва
послуги:</label>
        <input type = "text" id = "serviceTitle" name = "serviceTitle"
placeholder='Введіть назву послуги' onChange={onFormDataChange} className = "form-
```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

```

control" value = {formData.serviceTitle} />
    <span className='form-error-text'>{titleError ? "Введіть назву
послуги!" : ""}</span>
</div>

    <div className='form-element'>
        <label htmlFor='servicePrice' className='form-label'>Вартість
(грн):</label>
        <input type = "text" id = "servicePrice" name = "servicePrice"
placeholder='Введіть вартість' onChange={onFormDataChange} className = "form-
control" value = {formData.servicePrice} />
        <span className='form-error-text'>{priceError ? "Введіть
вартість!" : ""}</span>
    </div>

    <div className='form-element'>
        <label htmlFor='servicePeriod' className='form-label'>Цикл
(дні):</label>
        <input type = "text" id = "servicePeriod" name = "servicePeriod"
placeholder='Введіть число днів' onChange={onFormDataChange} className = "form-
control" value = {formData.servicePeriod} />
        <span className='form-error-text'>{periodError ? "Введіть число
днів!" : ""}</span>
    </div>

    <div className='form-element'>
        <label htmlFor='serviceContent' className='form-
label'>Опис:</label>
        <textarea id = "serviceContent" name = "serviceContent"
placeholder='Введіть наповнення пакета' onChange={onFormDataChange} className =
"form-control" rows = "10" value = {formData.serviceContent}></textarea>
        <span className='form-error-text'>{contentError ? "Введіть текст!"
: ""}</span>
    </div>

    <button type = "button" onClick={{(onSaveServiceClicked)}} className =
"btn btn-default" disabled = {!canSave}>Надіслати</button>
    <ToastContainer />
</form>
</section>
</div>

```

Код реалізації процедури перегляду послуги, що включає інформацію про поточний пакет, зокрема вартість та кількість днів:

```

<section className='service-single-section px-4'>
    <div className='service-single-title'>
        <h2 className='my-2 fs-20'>{tempService[0].serviceTitle}</h2>
    </div>

    <div className='py-4'>

```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        <p>Вартість послуги <b>{tempService[0].servicePrice}
грн</b>/{tempService[0].servicePeriod} дн.</p><br />
        <p>{tempService[0].serviceContent}</p>
    </div>
</section>

```

Функція підправлення повідомлення реалізована наступним чином:

```

const onSaveNoteClicked = () => {
  if(!titleError && !contentError){
    dispatch(addNewNote(formData));
    toast("Повідомлення відправлено службі підтримки");
    setFormData({noteTitle: "", noteContent: ""});
  }
}

```

Здійснено програмну реалізацію частини модулів веб-орієнтованої системи відповідно до спроектованого функціоналу.

Висновки до другого розділу

На основі аналізу предметної області та реалізації вимог до веб-орієнтованої системи оператора мобільного зв'язку були розроблені UML-діаграми, зокрема діаграма варіантів використання для двох груп користувачів, а саме користувача (абонента) та власника додатку (адміністратора). Визначено функціональні та нефункціональні вимоги до системи.

Побудовано діаграму класів з визначенням основних сутностей та їх зв'язків у системі. Здійснено розробку функціональних діаграм, що розкривають сутність проекту.

На основі отриманих результатів було спроектовано структуру платформи. Проектування включало в себе моделювання структури проекту та взаємодії між складовими частинами.

Здійснено програмну реалізацію ключових модулів веб-орієнтованої системи.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ОСОБЛИВОСТІ РОЗГОРТАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ

3.1. Опис інтерфейсу веб-орієнтованої системи оператора мобільного зв'язку

У результаті проведеного проєктування у системі передбачено два ключових актора: користувач (абонент) та власник додатку (адміністратор), (див. рис.2.1). Проведемо опис інтерфейснудля кожного користувача.

Так, вікно для перегляду доступних тарифів для користувача представлене на рис.3.1.

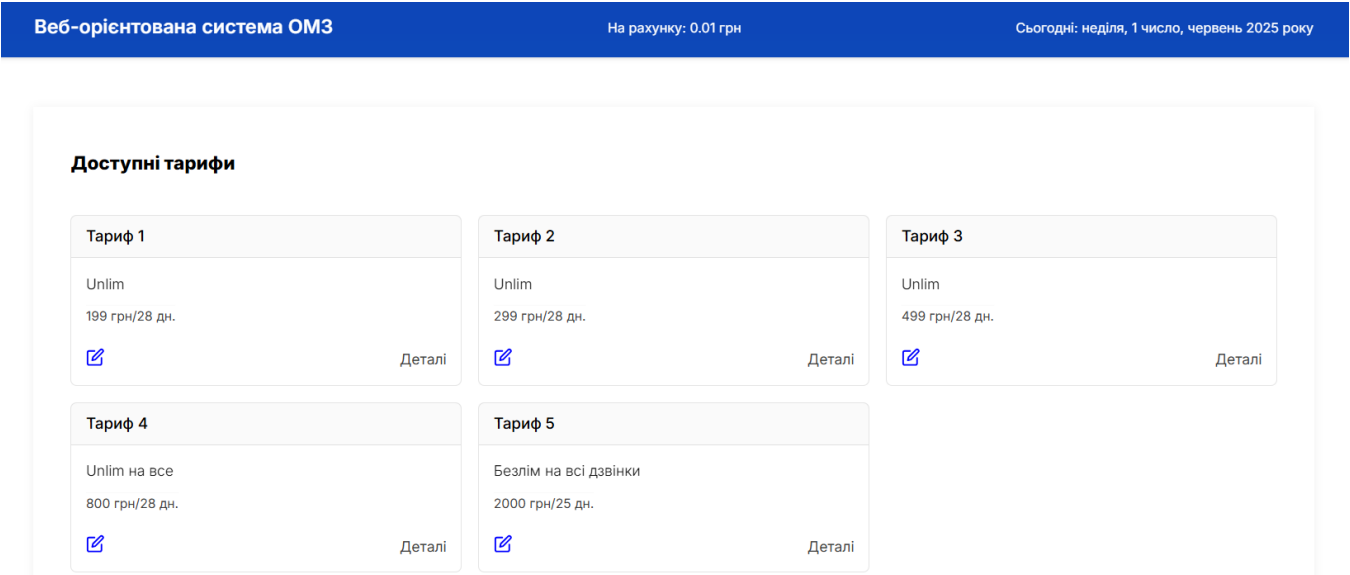


Рис. 3.1. Вигляд вікна перегляду доступних тарифів

На даному вікні висвічуються вгорі інформація про стан рахунку та поточна дата праворуч. Нижче наводиться перелік доступних тарифів, їх назва, вартість та термін дії. По кожному тарифу можна отримати детальну інформацію, натиснувши «Деталі». Вигляд вікна відображення тарифу представлено на рис. 3.2. В межах даного вікна висвітлюється інформацію про вартість пакету, його тривалість та анотована інформація про тариф.

Вікно перегляду доступних послуг для користувача представлено на рис.3.3.

Тариф 4

Вартість пакету послуг **800 грн/28 дн.**

Зміст тарифу:

Unlim на все

Рис. 3.2. Вигляд вікна перегляд тарифу

На даному вікні (рис.3.3) зберігається інформація про стан рахунку користувача, поточну дату, а нижче представляється інформація про доступні послуги. Слід відмітити, що вікно кожної послуги містить назву послуги, вартість, тривалість, перехід на деталі послуги, а також відбувається інформування щодо підключених послуг даного користувача. Вигляд вікна «Деталі» послуги наведено на рис. 3.4.

Веб-орієнтована система ОМЗ

На рахунку: 0.01 грн

Сьогодні: неділя, 1 число, червень 2025 року

Послуги		
Анлім на все літо Анлім на все літо 100 грн/100 дн. Деталі	Анлім на день Безлім дзвінки 10 грн/1 дн. Деталі	Безкоштовно БЕЗКОШТОВНО 0 грн/0 дн. Деталі
Назва послуги 1 Опис 0 грн/100 дн. Підключено Деталі	Назва послуги 2 1 123 грн/21 дн. Підключено Деталі	Анлім назавжди 1 100 грн/39999 дн. Підключено Деталі

Рис.3.3. Вигляд вікна переліку доступних послуг

Анлім на все літо

Вартість послуги **100 грн/100 дн.**

Анлім на все літо

Підключити

Рис. 3.4. Вигляд вікна перегляду інформації про послугу

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

Для організації зворотного зв'язку з мобільним оператором реалізовано функцію відправлення повідомлення. Вигляд вікна для відправлення повідомлення представлено на рис. 3.5.

Веб-орієнтована система ОМЗ

На рахунок: 0.01 грн

Повідомлення відправлено службі підтримки

Зв'язатися з нами

Ім'я:

Введіть ім'я...

Повідомлення:

Введіть текст ...

Надіслати

Рис.3.5. Вигляд вікна для відправлення повідомлення

Інтерфейс вікна доволі простий та доступний широкому колу користувачів. Для відправлення повідомлення потрібно зазначити ім'я, написати саме повідомлення та натиснути «Надіслати». У верхньому правому вікні висвітлюється повідомлення про успішну відправку.

Інтерфейс власника додатку (адміністратора) системи представлено на рисунку 3.6.

Веб-орієнтована система ОМЗ

На рахунок: 0.01 грн

Сьогодні: неділя, 1 число, червень 2025 року

Адмінпанель: тарифи

+ Додати тариф

Тариф 1	Тариф 2	Тариф 3
Unlim 199 грн/28 дн.	Unlim 299 грн/28 дн.	Unlim 499 грн/28 дн.
Деталі	Деталі	Деталі

Тариф 4	Тариф 5
Unlim на все 800 грн/28 дн.	Безлім 2000 грн/25 дн.
Деталі	Деталі

Рис. 3.6.Вигляд вікна для налаштування тарифів

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

Як видно з рисунку 3.6. в панелі адміністратора представлено перелік тарифів, де зазначено назву тарифу, коротку інформаційну довідку про тариф, можливість видалення та редагування тарифу, рис.3.7. також є можливість редагування деталей тарифу, назви тарифу.

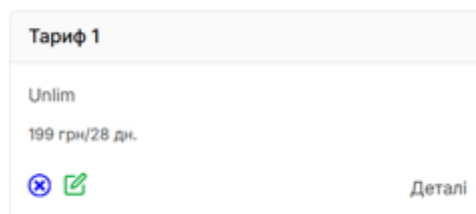


Рис.3.7. Вигляді вікна тарифу

Також на даній панелі (див. рис.3.6) є можливість додати тарифний пакет. Вигляді вікна для внесення даних по тарифу представлено на рис. 3.8.

Додати тарифний план

Назва ТП:

Денний пакет

Вартість (грн):

50

Цикл (дні):

1

Наповнення:

безлім на всі по Україні та безлім інтернет

Надіслати

Рис. 3.8 Вікно додавання тарифу

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

Як бачимо адміністратор може ввести назву тарифного плану, вартість, тривалість (цикл), опис тарифного плану (наповнення) та підтвердити свої дії. Про підтвердження проведеної операції висвітяться інформаційне повідомлення, рис. 3.9.

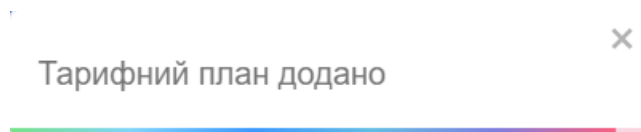


Рис. 3.9. Інформаційне повідомлення про додавання тарифу

На рисунку 3.10 представлено вікно адміністратора для роботи з панеллю послуг. Тут теж представлено перелік послуг з можливістю їх коригування та видалення, наведена коротка інформація про послугу та перехід у вікно детального опису тарифу, а також є можливість додавання послуг. Вікно введення даних при додаванні послуги наведено на рис. 3.11.

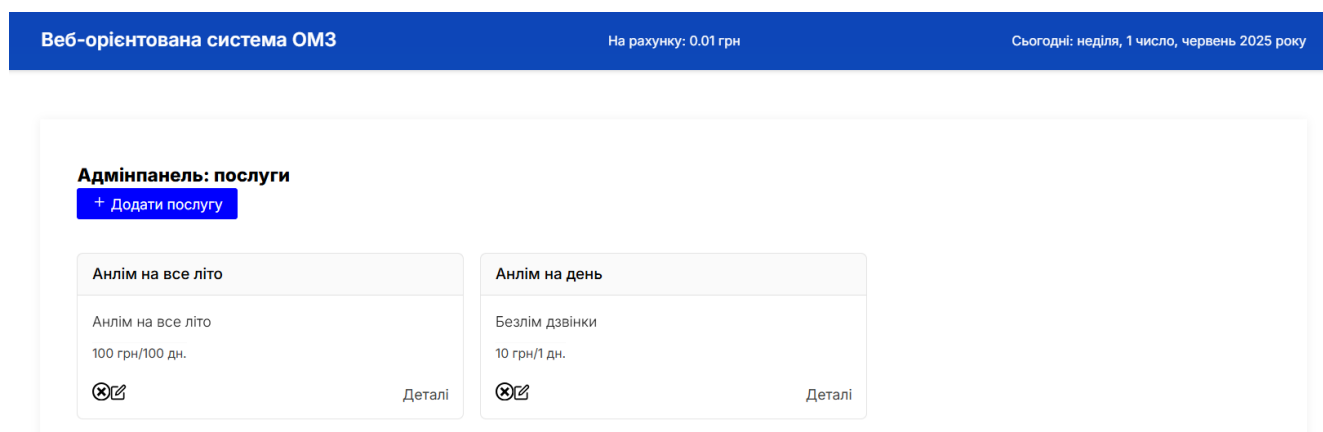


Рис. 3.10. Вигляд вікна адміністратора по роботі з послугами

Додати послугу

Назва послуги:

Анлім

Вартість (грн):

99

Цикл (дні):

99

Опис:

99

Рис. 3.11. Додавання послуги

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

В межах вікна можна ввести назву послуги, її вартість, тривалість та короткий опис. Після заповнення полів, потрібно підтвердити введену інформацію.

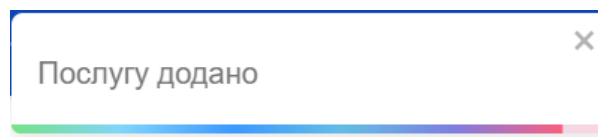


Рис. 3.12 – Послугу додано

Вигляд інформаційного повідомлення про додану послугу представлено на рис. 3.12.

На наступному етапі проведено опис порядку та особливостей розгортання веб-орієнтованої системи.

3.2. Порядок розгортання клієнт-серверної системи

Розгортання вебзастосунку є комплексом операцій, спрямованих на перенесення програмного продукту з локального середовища розробки у публічне середовище для кінцевих користувачів. Цей процес повинен забезпечити стабільність, безпеку та ефективність функціонування системи в умовах реального використання. Оскільки сучасні вебзастосунки зазвичай мають складну архітектуру, що складається з клієнтської (frontend) та серверної (backend) частин, важливо узгодити їхню взаємодію ще на етапі планування.

Передусім необхідно розробити структуру програми згідно з вимогами предметної області. Наприклад, якщо йдеться про облік користувачів, тарифних планів та послуг, потрібно створити відповідні моделі даних і структури бази даних. На цьому етапі визначаються ключові функціональні модулі системи, такі як система реєстрації та автентифікації, особистий кабінет користувача, адміністрування послуг та аналітика. Важливою умовою є забезпечення належної взаємодії між інтерфейсом користувача та серверною логікою, яка виконує обробку запитів, валідацію введених даних, збереження в базу та передачу відповіді у зручному для користувача форматі (наприклад, JSON).

Розроблений вебзастосунок побудовано на базі технологій Node.js, що дозволяє запускати серверні програми на JavaScript. Це забезпечує узгодженість

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

між клієнтською та серверною логікою та полегшує розробку. Перед розгортанням необхідно встановити необхідні модулі для обох частин системи — для цього у кореневих директоріях клієнтської та серверної частини виконується команда `npm install`, яка завантажує всі залежності, зазначені у файлах `package.json`. Додатково можуть встановлюватися інструменти для збірки застосунку, наприклад Webpack або Vite для фронтенду.

Наступним етапом є розміщення вебзастосунку на хостингу. Для забезпечення безперервного доступу користувачів до сервісу, вебсайт розміщується на віддаленому сервері або у хмарному середовищі. Популярними варіантами є Render, Heroku, Vercel, Netlify або AWS. Після вибору платформи потрібно налаштувати серверне середовище: встановити змінні середовища, такі як шляхи до бази даних, секретні ключі або токени доступу. Також здійснюється налаштування маршрутизації, логування, обробки помилок тощо.

Зважаючи на те, що система оперує персональними даними користувачів, включно з фінансовою інформацією, безпека є критично важливою складовою. Необхідно впровадити захист від типових кіберзагроз, таких як SQL-ін'єкції (особливо при використанні традиційних СУБД), міжсайтові скриптові атаки (XSS), підробка міжсайтових запитів (CSRF), а також атаки типу "відмова в обслуговуванні" (DDoS). В обов'язковому порядку має бути впроваджено шифрування трафіку за допомогою протоколу HTTPS, для чого встановлюється SSL-сертифікат.

Ще одним важливим аспектом є резервне копіювання. Регулярне створення резервних копій усієї системи, а також бази даних, дає змогу зберегти цілісність інформації навіть у разі апаратних збоїв, помилок користувача чи втручання зловмисників. Для цього можуть застосовуватись автоматизовані сервіси резервного копіювання, що дозволяють налаштувати періодичність збереження, шифрування даних та зручне відновлення.

На завершальному етапі перед публікацією варто провести комплексне тестування продуктивності та навантаження, щоб переконатися, що система

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

витримує очікувану кількість запитів та обробляє їх у встановлені терміни. Це дозволяє виявити потенційні "вузькі місця" та оптимізувати використання ресурсів. Існують спеціальні сервіси для таких перевірок, наприклад Apache JMeter або k6.

У результаті програмний продукт розгортається у хмарному середовищі з можливістю масштабування, резервного копіювання та цілодобового моніторингу. Такі сервіси, як Render, пропонують автоматизований деплоймент безпосередньо з репозиторіїв на GitHub або GitLab, що значно спрощує процес оновлення застосунку. Це дає змогу підтримувати стабільну роботу сервісу, швидко реагувати на зміни та ефективно масштабуватися відповідно до зростання навантаження.

3.3. Тестування веб-орієнтованої системи оператора мобільного зв'язку

Проведемо тестування реалізованої веб-орієнтованої системи оператора мобільного зв'язку. При описі інтерфейсу було описано основні дії користувачів системи. В даному пункті проведемо демонстрацію окремих функцій. Після входу в систему, користувачу (абоненту) висвітлюється вікно доступних тарифів, де він може бачити їх переліком у вигляді невеликих віконеч, рис. 3.13.

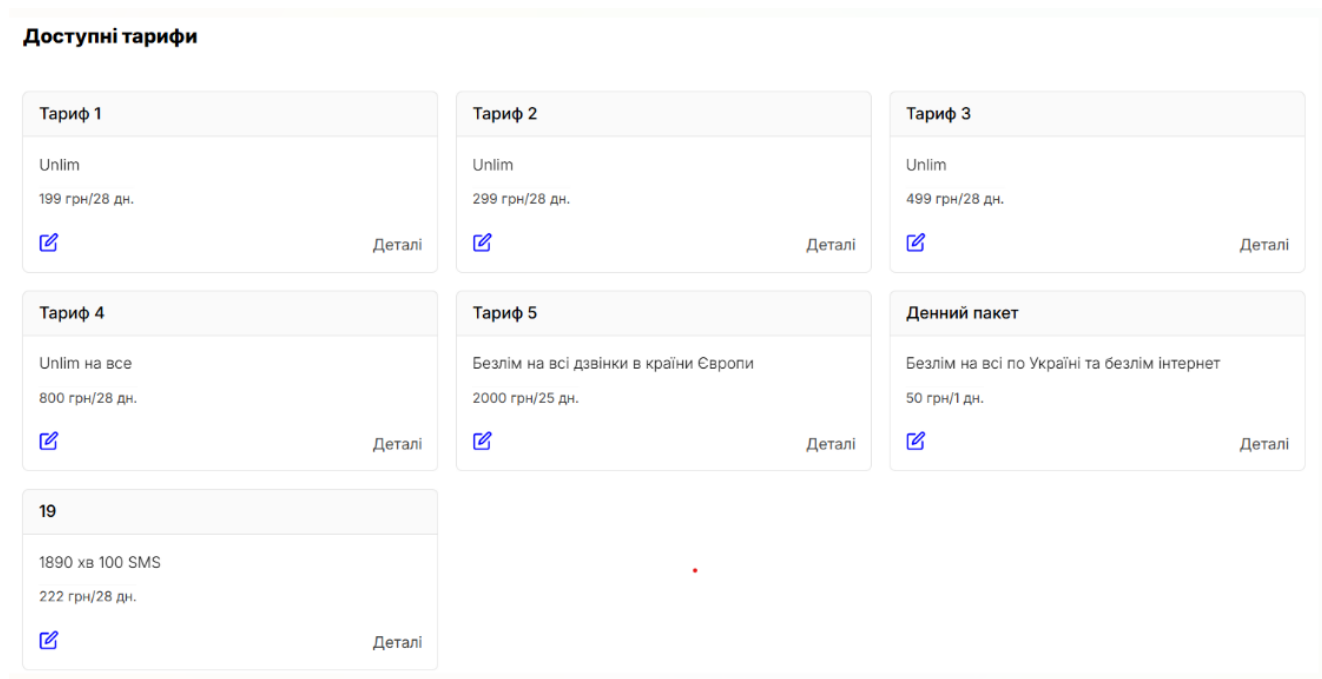


Рис.3.13. Вікно перегляду доступних тарифів

Як видно з рис.3.13 абоненту висвітлюється коротка інформація про тарифи, а саме назва тарифу, його опис, тривалість та вартість, рис.3.14.

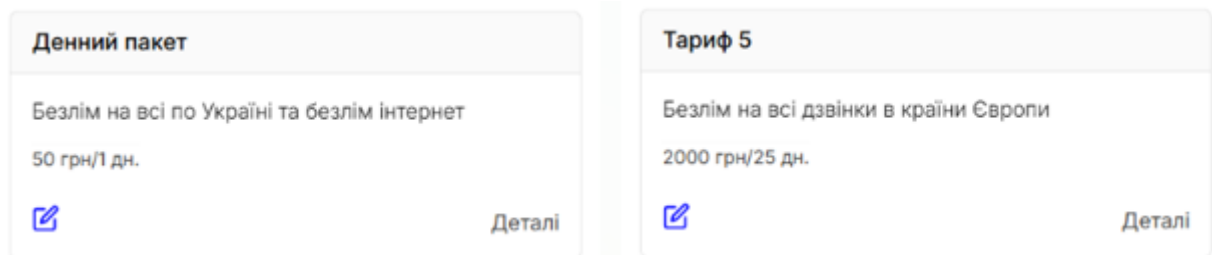


Рис.3.14 Інформація про тариф

Натиснувши на «Деталі», що розміщено в кожному тарифі, буде відображатися детальна інформація про обраний тариф. Вигляд інформаційних вікон по кожному тарифу представлено на рис. 3.15.

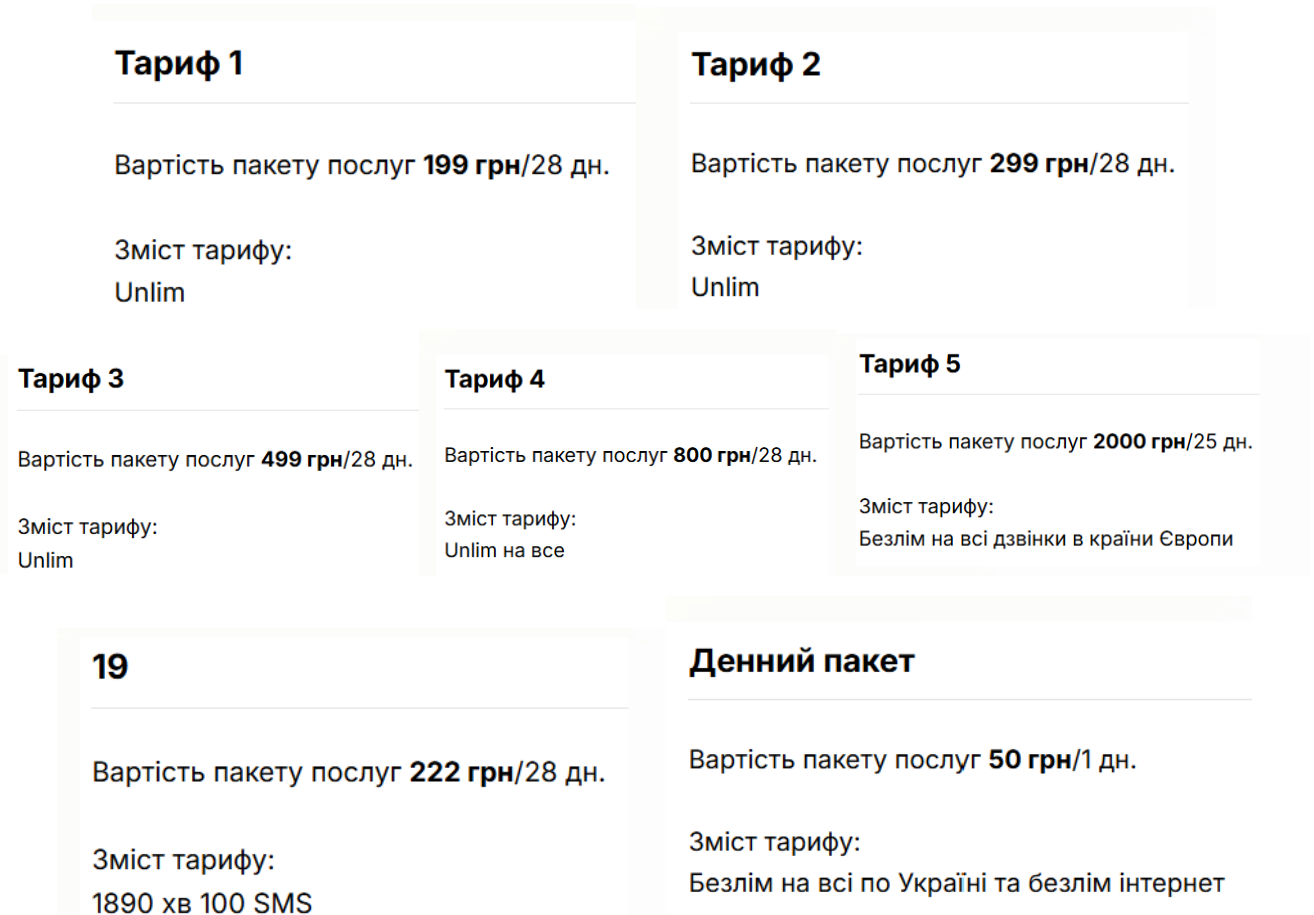


Рис. 3.15. Інформаційні вікна тарифів

Також для абонентів доступна інформація про послуги, де можна прочитати інформацію про послугу, перевірити які з них підключені, здійснити підключення та відключення послуги, рис. 3.16 та рис.3.17.

Рис. 3.16. Підключення послуги

Рис. 3.17. Відключення послуги

Проведемо тестування функції редагування параметрів послуги адміністратором системи. Для прикладу візьмемо послугу «Анлім на все літо», вибираємо її на адмінпанелі, натискаємо «Коригувати». В результаті таких дій з'являється вікно «Редагувати параметри послуги», що представлена на рисунку 3.18.

Редагувати параметри послуги

Назва послуги:

Анлім на все літо

Вартість (грн):

100

Цикл (дні):

100

Опис:

Анлім на все літо

Рис. 3.18. Вікно редагування послуги адміністратором

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

Проводимо коригування інформації відповідно потреб та підтверджуємо свої дії. На наступному кроці з'явиться повідомлення про зміну послуги, що представлено на рис. 3.19.

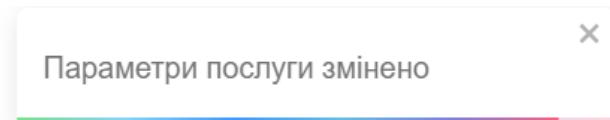


Рис. 3.19. Інформаційне повідомлення про зміну параметрів послуги

У ході проведених дій було здійснено перевірку основних функцій веб-системи оператора мобільного зв'язку. Зокрема, протестовано можливість створення та редагування тарифів та послуг, відображення інформаційних повідомлень. Аналіз використання інтерфейсних елементів та меню підтвердив правильність реалізації функціоналу та його зручність для користувачів системи.

Висновки до третього розділу

Було здійснено детальний аналіз інтерфейсної частини веб-орієнтованої системи оператора мобільного зв'язку. Розглянуто структуру меню та панелей інструментів, а також описано особливості взаємодії користувача із системою.

Окремо було описано процес розгортання клієнт-серверної веб-орієнтованої системи, зокрема послідовність етапів та специфіку налаштування як клієнтської, так і серверної частин системи.

В рамках дослідження функціональних можливостей веб-орієнтованої системи проведено її тестування, яке дозволило оцінити якість реалізації ключових функцій. Інтерфейс програми вирізняється зручністю використання завдяки логічному розміщенню елементів управління та наявності візуального зворотного зв'язку, що полегшує процес освоєння інструменту новими користувачами.

У підсумку, функціональні можливості розробленої веб-орієнтованої системи повністю відповідають поставленим цілям та завданням. Результати тестування свідчать про точність та стабільність роботи реалізованих функцій.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

ВИСНОВКИ

В результаті проведених досліджень були отримані наступні ключові результати:

1. Аналіз предметної області дав змогу окреслити базові потреби майбутніх користувачів та сформулювати основні функціональні та нефункціональні вимоги до веб-орієнтованої системи. Проведене дослідження допомогло визначити, яким чином система має реагувати на запити абонентів та адміністраторів, забезпечуючи гнучке управління рахунком, тарифами та сервісами.

2. Дослідження існуючих рішень та порівняння аналогічних систем дозволили відібрати оптимальні концепції для реалізації, а також врахувати сильні сторони конкурентів. Це стало основою для створення програмного продукту, орієнтованого на простоту використання та функціональну повноту.

3. Вибір технологічного стеку, зокрема використання Spring Boot і PostgreSQL для серверної частини, а також React для фронтенду, забезпечив не лише високу продуктивність, а й можливість масштабування та підтримки в майбутньому. Вибрані технології відповідають сучасним вимогам до безпеки, надійності та гнучкості.

4. Розробка UML-діаграм, включаючи діаграми варіантів використання для двох груп користувачів та діаграму класів, дозволила формалізувати логіку системи, чітко описати ролі, функціональні взаємодії та основні сутності проєкту.

5. Архітектурне проєктування системи охоплювало моделювання структури, встановлення зв'язків між її компонентами, а також визначення ключових модулів для реалізації. Це забезпечило логічну цілісність застосування та полегшило етап програмної реалізації.

6. Програмна реалізація включала створення клієнтської та серверної частин, що повноцінно взаємодіють між собою. Особливу увагу було приділено інтерфейсній частині, де реалізовано інтуїтивно зрозумілу структуру меню, панелей інструментів та візуального зворотного зв'язку.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

7. Розгортання клієнт-серверного застосування описано покроково з урахуванням особливостей налаштування середовища для обох частин. Це дозволяє швидко вводити систему в експлуатацію та підтримувати її у робочому стані.

8. Тестування вебзастосування підтвердило коректність реалізованих функцій. Інтерфейс користувача виявився зручним і зрозумілим, що робить систему придатною для широкого кола користувачів, включаючи абонентів без спеціальної технічної підготовки.

У підсумку реалізована система відповідає визначеним вимогам і може бути використана як ефективний інструмент для організації сервісів мобільного оператора, що забезпечує зручне управління рахунком, доступ до інформації та адміністративні функції.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A basic study and review of UML versions and UML tools. 2022. URL: <https://medium.com/@okworchukwuebube/a-basic-study-and-review-of-uml-ersions-and-uml-tools-af2c85e606>.
2. Component Diagram Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lucidchart.com/pages/uml-component-diagram>.
3. Freeman, A. Pro ASP. NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. *APress*. 2022. 1253 p.
4. Gilberto D., Jesslyn A., Afrianto Y. Analysis and Design of Web-based Information System for Coffeeshop Management using Design Thinking Methodology: Case of Kopi KurangLebih. *Journal of information systems and informatics*, 2023. Vol. 5. № 1. P. 217-231. DOI: <https://doi.org/10.51519/journalisi.v5i1.455>.
5. Hoffman, A. Web Application Security: Exploitation and Countermeasures for Modern Web Applications. *O'Reilly Media*. 2020. 327 p.
6. Karcic F. A Developer's Guide to CQRS Using .NET Core and MediatR [Електронний ресурс] / Faris Karcic. – 2020. – Режим доступу до ресурсу: <https://www.rubicon-world.com/blog/2020/06/a-developers-guide-to-cqrs-using-net-core-and-mediatr/>.
7. Patel A. Use MediatR in ASP.NET or ASP.NET Core / Ashish Patel. 2021. URL: <https://medium.com/dotnet-hub/use-mediatr-in-asp-net-or-asp-net-core-cqrs-and-mediator-in-dotnet-how-to-use-mediatr-cqrs-aspnetcore-5076e2f2880c>.
8. StarUML. URL: <https://www.educba.com/staruml/>.
9. UML Activity Diagram Tutorial URL: <https://www.lucidchart.com/pages/uml-activity-diagram>.
10. Visual Paradigm. URL: <https://bpms.com.ua/uk/visual-paradigm>.
11. Vodafone Україна – Мобільний зв'язок, Інтернет, Тарифи та Послуги. Вебсайт Vodafone. URL: <https://www.vodafone.ua>.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

12. Vynohradova, O. V., Yevtushenko, N. O., Drokina, N. I., Irtlach, M. O., & Korniiichuk, V. Ye. Shliakhy pidvyshchennia konkurentospromozhnosti na rynku telekomunikatsiinykh posluh Ukrainy [Ways to increase competitiveness in the telecommunications services market of Ukraine]. Ekonomika. Menedzhment. Biznes – Economy. Management. Business, 2020. vol. 1 (31), pp. 130–140.

13. Yulianto H. D., Fauzi R. Design of Web-based Online Sales Information System. IOP Conf. Series: Materials Science and Engineering, 879 (2020) 012007. IOP Publishing. DOI: <http://dx.doi.org/10.1088/1757-899X/879/1/012007>.

14. Діаграма розгортання. URL: https://vuzlit.com/1009782/diagrama_rozgortannya.

15. Квітка С. А. Доступ до мережі Інтернет через мобільні пристрої: світовий досвід та перспективи розвитку в Україні. Аспекти публічного управління. 2019. Вип. 7. № 9–10. С. 5–18. DOI: <https://doi.org/10.15421/15194425>.

16. Мартін Р. Чиста архітектура: Мистецтво створення програмного забезпечення / Роберт Мартін. – Харків: Фабула, 2019. – 368 с. – (Видання друге).

17. Методичні рекомендації для виконання курсових робіт з навчальної дисципліни «Моделювання та аналіз програмного забезпечення» URL: https://learn.ztu.edu.ua/pluginfile.php/8196/mod_resource/content/6/metod_kurs_20141.pdf.

18. Методичні рекомендації до лабораторного практикуму з дисципліни «Моделювання та аналіз програмного забезпечення» URL: https://learn.ztu.edu.ua/pluginfile.php/34807/mod_resource/content/0/Методичні%20вказівки%20до%20лаб_робіт.pdf.

19. Мобільний зв'язок від Київстар. Веб сайт Київстар. URL: <https://kyivstar.ua/>

20. Мобільний зв'язок в Україні. Вікіпедія. URL: <https://surl.li/imixrr>. (дата звернення 16.03.2025).

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		
						49

21. Моралес Д. Повне розуміння діаграми компонентів UML за допомогою легкого методу [Електронний ресурс] / Джейд Моралес. – 2023. – Режим доступу до ресурсу: <https://www.mindonmap.com/uk/blog/uml-component-diagram/>.

22. Національна комісія, що здійснює державне регулювання у сферах електронних комунікацій, радіочастотного спектра та надання послуг поштового зв'язку. Веб сайт НКРЕК. URL: <https://nkrzi.gov.ua/index.php?r=site/index&pg=1&language=uk26>.

23. Нечитайло Б. Роль персоналізованих цінових пропозицій у підвищенні конкурентоспроможності телекомунікаційних компаній. Економіка та суспільство. 2024. № 66.. С.1-7. DOI: <https://doi.org/10.32782/2524-0072/2024-66-30>

24. Оператор lifecell Україна – мобільний зв'язок. *lifecell.ua*. URL: <https://www.lifecell.ua/uk/>

25. Побудова моделей поведінки проектного ПЗ за допомогою діаграми діяльності Activity diagram. URL: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/laboratorna-robota-78-diahrama-dijalnosti.pdf>.

26. Подзігун С.М., Білошкурська Н.В., Омеляненко В.А., Корнєєва О.О. Аналіз конкурентоспроможності мобільних операторів України на основі бенчмаркінгу: оцінка ринку та формування інноваційного бізнес-середовища. *Здобутки економіки: перспективи та інновації*. 2024. №11. С.1-27. DOI: <https://doi.org/10.5281/zenodo.13937048>.

27. Стадник Ю. Технології створення програмних та інтелектуальних систем. Львів: Львівський національний університет імені Івана Франка, 2021. 46 с.

28. Структура сайту: основні види та правила їх розробки. URL: <https://webtune.com.ua/statti/web-rozrobka/struktura-sajtu/> (дата звернення: 16.05.2025).

29. Ткаченко О. і Грибок О. Розробка веборієнтованих систем: онтологічний підхід, *Цифрова платформа: інформаційні технології в*

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк. 50
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		

соціокультурній сфері. 2023. т. 6, вип. 1, с. 231–246. DOI: 10.31866/2617-796X.6.1.2023.283993.

30. Ткаченко, О., Ткаченко, К., Ткаченко, О., & Вознюк, В. J-Finder – веб-орієнтована система підтримки прийняття рішень при пошуку роботи. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2024. 1(25), С. 355–378. DOI: <https://doi.org/10.28925/2663-4023.2024.25.355378>.

31. Швець О. Занурення в патерни проектування / Олександр Швець., 2022.396 с.

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

ДОДАТКИ

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

Реалізація показу списку послуг

```
const ServicesList = ({services}) => {
  const dispatch = useDispatch();

  if(!services || services.length === 0){
    return (<div className='not-found'>Немає доступних послуг</div>)
  }

  return (
    <div className='services'>
      <h5 className='fs-18 fw-8 text-uppercase notes-
title'>Послуги</h5>
      <div className='services-list grid'>
        {
          services.map(service => {
            return (
              <div className='services-item' key =
{service.serviceId}>
                <div className='services-item-title'>
                  {service.serviceTitle.substring(0,
100)}
                </div>
                <div className='services-item-body'>
                  {service.serviceContent.substring(0,
1000)}
                </div>
                <div className='services-item-
date'>{service.servicePrice} грн/{service.servicePeriod} дн.</div>
                <div className='services-item-btns flex
align-center justify-between'>
                  <div>
                    {service.serviceEnabled !== 0 ?
```

"Підключено" : ""}

</div>

<Link to =

{`/enable/\${service.serviceId}`} className='read-more-btn fs-14'>Деталі</Link>

</div>

</div>

)

}}

}

</div>

</div>

)

}

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Витягнення даних про послугу при створенні чи редагуванні

```
const serviceSlice = createSlice({
  name: 'services',
  initialState,
  reducers: {
    addNewService(state, action){
      const {serviceTitle, servicePrice, servicePeriod,
serviceContent} = action.payload;
      let serviceId = uuid();
      let serviceEnabled = 0;
      let newPost = {serviceId, serviceTitle, servicePrice,
servicePeriod, serviceContent, serviceEnabled};
      newPost.noteDate = new Date().toISOString();
      state.services.push(newPost);
      storeInLocalStorage('services', state.services);
    },

    removeService(state, action){
      const tempId = action.payload;
      const tempServices = state.services.filter(service =>
service.serviceId !== tempId);
      state.services = tempServices;
      storeInLocalStorage('services', tempServices);
    },

    editService(state, action){
      const {serviceId, serviceTitle, serviceContent, serviceEnabled}
= action.payload;
      const tempServices = state.services.map(service => {
        if(service.serviceId === serviceId){
          service.serviceTitle = serviceTitle;
          service.serviceContent = serviceContent;
        }
      });
      state.services = tempServices;
      storeInLocalStorage('services', tempServices);
    }
  }
});
```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

```

        service.serviceEnabled = serviceEnabled;
        service.noteDate = new Date().toISOString();
    }
    return service;
});

state.service = tempServices;
storeInLocalStorage('services', tempServices);
},

increaseCount(state, action){
    state.count = state.count + 1;
}
}
});

```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

Відображення тарифу на основі вартості та періоду пакету

```

const SingleTariffPage = () => {
  const { id } = useParams();
  const tariffs = useSelector(getAllTariffs);
  let tempTariff = tariffs.filter(tariff => tariff.tariffId === id);
  return (
    <section className='tariff-single-section px-4'>
      <div className='tariff-single-title'>
        <h2 className='my-2 fs-20'>{tempTariff[0].tariffTitle}</h2>
      </div>

      <div className='py-4'>
        <p>Вартість пакету послуг <b>{tempTariff[0].tariffPrice}
грн</b>/{tempTariff[0].tariffPeriod} дн.</p><br />
        <p>Зміст тарифу:<br />{tempTariff[0].tariffContent}</p>
      </div>
    </section>
  )
}

```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

Форма надсилання повідомлення

```

const ContactForm = () => {
  const dispatch = useDispatch();
  const [formData, setFormData] = useState({
    noteTitle: "",
    noteContent: ""
  });
  const [titleError, setTitleError] = useState(false);
  const [contentError, setContentError] = useState(false);
  const [canSave, setCanSave] = useState(false);

  const onFormDataChange = (event) => {
    event.preventDefault();
    if(event.target.name === 'noteTitle'){
      if(event.target.value.length === 0){
        setTitleError(true);
      } else {
        setTitleError(false);
        setCanSave(true);
      }
    }

    if(event.target.name === 'noteContent'){
      if(event.target.value.length === 0){
        setContentError(true);
      } else {
        setContentError(false);
        setCanSave(true);
      }
    }

    setFormData(prevData => {

```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

```

    return {
      ...prevData,
      [event.target.name]: event.target.value
    }
  })
}

const onSaveNoteClicked = () => {
  if(!titleError && !contentError){
    dispatch(addNewNote(formData));
    toast("Повідомлення відправлено службі підтримки");
    setFormData({noteTitle: "", noteContent: ""});
  }
}

return (
  <div>
    <section className='note-form-section'>
      <h2 className='my-4 fs-16'>Зв'язатися з нами</h2>
      <form className='note-form'>
        <div className='form-element'>
          <label htmlFor='noteTitle' className='form-label'>Ім'я:</label>
          <input type = "text" id = "noteTitle" name = "noteTitle"
placeholder='Введіть ім'я...' onChange={onFormDataChange} className =
"form-control" value = {formData.noteTitle} />
          <span className='form-error-text'>{titleError ? "Введіть ім'я!"
: ""}</span>
        </div>

        <div className='form-element'>
          <label htmlFor='noteContent' className='form-
label'>Повідомлення:</label>
          <textarea id = "noteContent" name = "noteContent"
placeholder='Введіть текст ...' onChange={onFormDataChange} className =

```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

```

"form-control" rows = "10" value = {formData.noteContent}></textarea>
    <span className='form-error-text'>{contentError ? "Введіть
текст!" : ""}</span>
    </div>

    <button type = "button" onClick={onSaveNoteClicked} className =
"btn btn-default" disabled = {!canSave}>Надіслати</button>
    <ToastContainer />
  </form>
</section>
</div>
)
}

```

		Грицюк В.І.			ІПЗ.КР.Б-121-25-ПЗ	Арк.
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		60