

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій
Кафедра технологій управління
Спеціальність 122 «Комп'ютерні науки»
Освітньо-наукова програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

« Дослідження процесів управління проектом розробки інтернет-сервісу для телекомунікаційної компанії»

Студента 2-го курсу групи УП-21

Науковий керівник:

Арсентій ГРИГОР'ЄВ

К.т.н, доцент

(прізвище, ім'я)

(науковий ступінь, вчене звання)

Анна КОЛОМІЄЦЬ

(прізвище, ім'я)

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри
технологій
управління

Віктор МОРОЗОВ

(підпис)

(дата)

Київ-2025

ЗАТВЕРДЖУЮ
Завідувач кафедри
Професор Морозов В.В.

“ _____ ” _____ 2025 року

ЗАВДАННЯ НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент: Григор'єв Арсентій Георгійович

Група: УП-21

1. Тема кваліфікаційної роботи: *«Дослідження процесів управління проектом розробки інтернет-сервісу для телекомунікаційної компанії»*
Затверджена на засіданні кафедри технологій управління, протокол №5 від «26» листопада 2024 року.
2. Строк подання студентом готової роботи – «10» травня 2025 р.
3. Цільова установка та вихідні дані до роботи: Аналіз особливостей управління ІТ-проектами у сфері телекомунікацій, дослідження специфіки процесів розробки інтернет-сервісів, планування життєвого циклу, ресурсів, витрат, ризиків та комунікацій в проекті, побудова організаційної структури управління, розробка документації та підбір інструментів для реалізації проекту.
4. Зміст роботи:
 - Аналіз предметної галузі, ринку та конкурентів у сфері телекомунікацій;
 - Формулювання мети проекту, SMART-цілей та паспорту проекту;
 - Постановка задач дослідження, побудова дерева цілей і дерева проблем;

- Визначення зацікавлених сторін, створення матриці відповідальності (RACI);
- Побудова ієрархічної структури робіт (WBS), організаційної структури та формування проєктної команди;
- Розробка життєвого циклу проєкту та календарного плану;
- Планування бюджету, ресурсів, витрат та резервів;
- Аналіз ризиків та розробка плану їх мінімізації;
- Вибір методів і засобів розробки: інструментів дизайну, комунікації, управління (наприклад, Jira, Gantt, BPMN);
- Проектування та розробка клієнт-серверної архітектури інтернет-сервісу;
- Моделювання бази даних, побудова навігаційної та функціональної структури системи;
- Моніторинг та оцінка виконання проєкту (освоєний обсяг, якість, зміни);
- Формування висновків та рекомендацій.

5. Перелік графічного матеріалу (слайдів):

- Титульна сторінка, тема, мета, новизна;
- SMART-цілі, дерево цілей, дерево проблем;
- Життєвий цикл проєкту, ієрархічна структура робіт, організаційна структура;
- Визначення стейкхолдерів, матриця RACI, ризик-менеджмент;
- Календарний план, бюджет, графік виконання, ресурси;
- Архітектура ПЗ, структура бази даних, логіка сервісу;
- Навігаційна структура інтернет-сервісу, візуалізація інтерфейсів;
- Висновки, використані джерела, дякую за увагу.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	План виконання роботи
1	Обрання теми кваліфікаційної роботи	30.10.2024
2	Затвердження теми кваліфікаційної роботи та призначення наукового керівника	06.11.2024
3	Підбір і формування списку літератури та джерел	13.11.2024-04.12.24
4	Розробка календарного плану кваліфікаційної роботи та його затвердження	04.12.24-18.12.24
5	Написання 1 розділу роботи	18.12.24-15.01.25
6	Написання II розділу роботи	15.01.25-11.03.25
7	Написання III розділу роботи	11.03.25-22.04.25
8	Оформлення кваліфікаційної роботи	22.04.25-06.05.25
9	Попередній захист кваліфікаційної роботи	10.05.25
10	Захист кваліфікаційної роботи	26.05.25

Дата видачі завдання «26» листопада 2024 р.

Керівник роботи кандидат технічних наук, доцент

Анна КОЛОМІЄЦЬ

(підпис)

Завдання прийняв до виконання студент групи УП-21

Арсентій ГРИГОР'ЄВ

(підпис)

ЗМІСТ

АНОТАЦІЯ	6
ВИЗНАЧЕННЯ ТЕРМІНІВ ТА АБРЕВІАТУР	9
ВСТУП	10
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПРОЄКТУ	13
1.1 Галузевий контекст і підходи до управління ІТ-проєктами	13
1.2 Стратегічний аналіз проєкту	23
1.3 Технічне завдання та паспорт проєкту	31
1.3.1 Опис, мета, цілі проєкту	31
1.3.2 Дерево цілей проєкту	34
1.3.3 Економічне обґрунтування проєкту	37
1.4 Постановка задачі дослідження	41
РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ТА ТЕХНІЧНЕ ПРОЄКТУВАННЯ	43
2.1 Розробка концептуальної моделі інформаційної системи	44
2.2 Розробка математичної моделі інформаційної системи.....	47
2.2.1 Диференціальна синхронізація баз даних.....	52
2.3 Архітектура інтернет-сервісу	54
2.4 Проектування бази даних	69
РОЗДІЛ 3. ПЛАНУВАННЯ ТА УПРАВЛІННЯ ЕЛЕМЕНТАМИ ПРОЄКТУ	72
3.1 Життєвий цикл проєкту та визначення фаз	72
3.2 Побудова ієрархічної структури робіт (WBS)	75
3.3 Організаційна структура управління проєктом	76
3.4 Планування часу та ресурсів.....	85
3.5 Контроль та моніторинг виконання проєкту	93
РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ	96
4.1 Опис структури програмного забезпечення.....	96
4.2 Реалізація інтерфейсу ІТ-сервісу	100
4.3 Тестування програмного забезпечення	104
ВИСНОВКИ	108
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	110
ДОДАТКИ	116

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

«Дослідження процесів управління проєктом розробки інтернет-сервісу для телекомунікаційної компанії»

Студента: Григор'єв Арсентій Георгійович

Науковий керівник: Коломієць Анна Степанівна

Рік захисту – 2025

Темою роботи було обрано «Дослідження процесів управління проєктом розробки інтернет-сервісу для телекомунікаційної компанії», предметною областю якої є телекомунікаційна галузь.

Метою роботи є розробка рекомендацій та моделей ефективного управління проєктом створення інтернет-сервісу для телекомунікаційної компанії з урахуванням сучасних методологій управління та специфіки галузі.

Об'єкт дослідження — процес розробки інтернет-сервісу в межах телекомунікаційної компанії.

Предмет дослідження — процеси управління проєктом, зокрема управління змістом, вартістю, термінами, ризиками, якістю та комунікаціями в контексті розробки ІТ-продуктів.

У межах дослідження здійснено аналіз галузі та потреб користувачів, сформовано паспорт проєкту, визначено життєвий цикл, створено ієрархічну структуру робіт, організаційну структуру команди та модель управління ризиками. Також було розроблено календарний план, бюджет проєкту та систему взаємодії зі стейкхолдерами.

Наукова новизна роботи полягає у запропонованому інтегрованому підході до управління проєктом, що поєднує класичні та гнучкі методології та адаптується до умов телекомунікаційної галузі.

Практичне значення результатів полягає у можливості застосування розроблених моделей у діяльності телекомунікаційних компаній для оптимізації процесів планування, реалізації та контролю ІТ-проектів.

Кваліфікаційна робота складається зі вступу, чотирьох основних розділів, висновків та додатків.

У *першому розділі* виконано комплексне аналітичне дослідження предметної області телекомунікаційних систем, проведено огляд сучасних підходів до управління ІТ-проектами та особливостей цифрових продуктів у телеком-секторі. Здійснено SWOT-аналіз компанії, сформульовано проблемну область, обґрунтовано потребу в розробці інтернет-сервісу, здатного підвищити ефективність взаємодії з користувачами. У результаті було визначено наукову новизну запропонованого рішення та сформульовано перелік дослідницьких та прикладних завдань, що вирішуються у межах роботи.

У *другому розділі* відбулася технічна формалізація задачі розробки інтернет-системи. Побудовано концептуальну та математичну моделі програмного продукту, розроблено архітектурну схему системи. Деталізовано складові бізнес-логіки, побудовано діаграми класів, прецедентів, послідовності та діяльності, що дозволили формалізувати функціональні вимоги. Обґрунтовано вибір стеку технологій, мови програмування та принципів взаємодії сервісів.

У *третьому розділі* описано процес управління проектом. Побудовано життєвий цикл проекту відповідно до методології Water-Scrum-Fall, сформовано ієрархічну структуру робіт (WBS) та матрицю відповідальності (RACI). Визначено склад проектної команди та обґрунтовано організаційну структуру. Проведено календарне та ресурсне планування з побудовою діаграми Ганта, розраховано бюджет та економічні показники ефективності (NPV, IRR, ROI, термін окупності).

У четвертому розділі розглянуто аспекти реалізації програмного забезпечення, включно з описом кожного мікросервісу, використаних технологій і шаблонів програмування. Подано розроблені інтерфейси користувача та адміністратора з поясненням принципів UX/UI-дизайну. Здійснено тестування основних функцій системи шляхом ручного модульного тестування, побудови тест-кейсів.

Робота містить 117 сторінок без додатків, 31 рисунок, 15 таблиць, 45 джерел. Додатки складають 4 сторінки.

Ключові слова: *управління проєктом, інтернет-сервіс, телекомунікації, розробка ПЗ, управління ризиками, планування проєкту, WBS, інформаційна система, концептуально модель, матриця відповідальності, SWOT-аналіз, RSYNC.*

ВИЗНАЧЕННЯ ТЕРМІНІВ ТА АБРЕВІАТУР

Абревіатура	Визначення
API	Інтерфейс прикладного програмування, що забезпечує взаємодію між програмними модулями або сервісами.
JWT	Формат токена для передачі інформації між сторонами у вигляді зашифрованого JSON-об'єкта
REST API	Архітектурний стиль обміну даними між клієнтом і сервером з використанням стандартних HTTP-запитів
PWA	Прогресивний веб-застосунок, який поєднує можливості вебу та мобільних застосунків
BPMN	Стандарт моделювання бізнес-процесів у вигляді діаграм
WBS	Ієрархічна структура робіт, що використовується для планування та контролю проєктів
RACI	Модель розподілу відповідальностей у команді проєкту
SWOT	Аналітична методика, що оцінює сильні (Strengths), слабкі (Weaknesses) сторони, можливості (Opportunities) і загрози (Threats)
KPI	Ключовий індикатор ефективності виконання завдань або досягнення цілей
MFA	Багатофакторна автентифікація, яка потребує декілька рівнів підтвердження особи
RSYNC	Протокол для ефективної синхронізації файлів та даних між системами
NPS	Метрика лояльності клієнтів, що базується на відповіді на питання: «Чи порекомендуєте ви наш сервіс іншим?»
MD5	Криптографічна хеш-функція, що використовується для перевірки цілісності даних
Project Lifecycle	Життєвий цикл проєкту
Gantt Chart	Діаграма Ганта, інструмент для візуалізації строків виконання завдань та залежностей між ними у часі.
Stakeholder	Будь-яка фізична чи юридична особа, яка має інтерес до проєкту або впливає на нього
Sprint	Фіксована за часом ітерація у Scrum, зазвичай триває 1–4 тижні
Sprint Planning	Зустріч для планування завдань на спринт, де визначається, що буде зроблено

ВСТУП

У сучасних умовах цифрової трансформації телекомунікаційна галузь зазнає значних змін, спричинених стрімким розвитком інформаційних технологій та зростанням вимог споживачів до якості, швидкості й зручності комунікаційних послуг. Висока конкуренція на ринку змушує телекомунікаційні компанії постійно шукати нові підходи до покращення взаємодії з клієнтами, оптимізації внутрішніх процесів та впровадження сучасних інтернет-сервісів, здатних забезпечити стабільний розвиток та інноваційність бізнесу. Ефективне управління проектами в цій сфері стає критичним фактором успіху, адже воно дозволяє не лише реалізовувати технічні рішення, але й досягати стратегічних цілей підприємства.

Актуальність теми магістерської роботи зумовлена необхідністю вдосконалення процесів управління проектами розробки інтернет-сервісів для телекомунікаційних компаній з метою підвищення їх ефективності, надійності та швидкості впровадження. Враховуючи складність таких проектів, необхідно досліджувати як загальні, так і специфічні аспекти управління, включаючи планування, управління ресурсами, ризиками, комунікаціями, а також технічну реалізацію продукту. У межах даної роботи увагу зосереджено на комплексному дослідженні процесів управління проектом створення інтернет-сервісу, який може бути використаний як канал обслуговування клієнтів, інтерактивна платформа для послуг або внутрішній корпоративний інструмент для автоматизації операцій.

Зв'язок теми з науковими дослідженнями обґрунтовується її відповідністю до сучасних пріоритетів у сфері управління проектами, цифрових інновацій та телекомунікацій. Тематика роботи гармонійно вписується в напрями підготовки

фахівців за освітньо-науковою програмою "Управління проєктами", де особлива увага приділяється впровадженню ІТ-рішень у бізнес-процеси.

Метою даної магістерської роботи є розробка рекомендацій та моделей ефективного управління проєктом розробки інтернет-сервісу для телекомунікаційної компанії, що враховують сучасні методології, галузеві особливості та технічні обмеження.

Для досягнення мети поставлено такі завдання:

- провести аналіз предметної області, ринку та конкурентного середовища;
- сформулювати паспорт проєкту з описом його мети, функціоналу, зацікавлених сторін;
- здійснити SWOT-аналіз та дослідити очікування користувачів;
- визначити життєвий цикл проєкту та обґрунтувати вибір методології;
- створити організаційну структуру управління проєктом;
- побудувати ієрархічну структуру робіт (WBS) та матрицю відповідальності;
- розробити календарний план, бюджет, провести оцінку вартості;
- дослідити ризики проєкту та розробити план їх управління;
- визначити та впровадити інструменти контролю виконання проєкту;
- надати рекомендації для вдосконалення управлінських процесів у подібних проєктах.

Об'єктом дослідження є процес розробки інтернет-сервісу для телекомунікаційної компанії.

Предметом дослідження є процеси управління проєктом, зокрема управління вартістю, термінами, якістю, ризиками, комунікаціями, командною взаємодією та змінами в контексті реалізації ІТ-проєктів у сфері телекомунікацій.

Методологічну базу дослідження становлять загальнонаукові методи (аналіз, синтез, абстрагування), системний підхід, елементи моделювання, експертного оцінювання, а також інструменти сучасного проєктного

менеджменту, такі як BPMN-моделювання, Gantt-діаграми, матриці RACI, SWOT-аналіз, метод критичного шляху (CPM) тощо. Застосовано також методи збору та аналізу емпіричних даних: інтерв'ювання стейкхолдерів, аналіз документації та звітів попередніх проєктів.

Наукова новизна роботи полягає в розробці інтегрованого підходу до управління проєктом розробки інтернет-сервісу з урахуванням специфіки телекомунікаційної галузі, що поєднує елементи гнучкого та класичного управління (Agile та Waterfall), а також пропонує нові методи комунікації зі стейкхолдерами та управління ризиками у високодинамічному середовищі.

Практичне значення результатів дослідження полягає у можливості використання розроблених моделей і рекомендацій у практичній діяльності телекомунікаційних компаній під час планування та реалізації проєктів цифрової трансформації. Вони можуть бути адаптовані для впровадження в корпоративну проєктну документацію та системи управління проєктами, а також слугувати основою для навчальних курсів та тренінгів з IT-менеджменту.

Апробація результатів дослідження здійснювалася шляхом обговорення із фахівцями під час переддипломної практики на базі компанії *DevBrother*. Також в межах кваліфікаційної роботи було розроблено алгоритм диференціальної синхронізації бази даних, що було застосовано у межах проєкту.

Таким чином, тема магістерської роботи є актуальною та практично значущою, а отримані результати мають потенціал для подальшого використання в професійній діяльності у сфері управління IT-проєктами.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПРОЄКТУ

1.1 Галузевий контекст і підходи до управління ІТ-проєктами

Телекомунікаційна індустрія пройшла довгий шлях від перших днів телеграфу та стаціонарних телефонів [9]. Сьогодні вона перебуває у стані безперервної та глибокої трансформації, еволюціонуючи від старої моделі з монополіями та мережами з комутацією каналів до нової епохи, тісно пов'язаної з інтернетом і цифровими технологіями. Ця еволюція призвела до появи інфокомунікаційної галузі, де передача інформації та комунікації нерозривно пов'язані [6].

Інформаційно-комунікаційні технології, відносяться до технологій, які забезпечують доступ до інформації через телекомунікації, де телекомунікації визначаються як «передача інформації, як слів, звуків або зображень, зазвичай на великі відстані, у вигляді електромагнітних сигналів, як телеграфом, телефоном, радіо або телебаченням» (словник). Жодна інша галузь не зачіпає стільки пов'язаних з технологіями бізнесів, як телекомунікаційна, адже вона включає в себе сфери місцевого та міжміського телефонного зв'язку, в тому числі дротового та бездротового, мікрохвильового зв'язку, волоконної оптики, супутників, Інтернету, штучного інтелекту, а також різні види послуг з розповсюдження інформації, такі як радіо, системи кабельного телебачення, носячі пристрої, телематику транспортних засобів.

Таким чином, ми говоримо про швидкозростаючий сектор, що підтверджується збільшенням кількості мобільних підписок до 8.7 мільярда, з яких майже 2.3 мільярда становлять 5G-підключення. Швидке впровадження 5G-технологій, зростання мобільного широкосмугового доступу (90% усіх підписок) та стабільне зростання обсягів мобільного трафіку (+20% рік до року) вказують на високий попит на мобільні послуги та цифровий контент. Найактивніше

зростання спостерігається в Індії, Китаї та країнах Азіатсько-Тихоокеанського регіону, що свідчить про глобальну динаміку переходу до нових поколінь зв'язку[5].

Сьогодні індустрія телекомунікацій охоплює широкий спектр технологій, від традиційних дротових мереж до передових бездротових рішень. Поява інтернету, мобільних пристроїв та інтернету речей (IoT) змінила ландшафт, перетворивши комунікацію на всюдисущу силу, яка долає кордони і з'єднує людей у спосіб, який неможливо було уявити ще кілька десятиліть тому[4].

Дивлячись на кількість технологій, які визначають світ телекомунікацій, вже можна зрозуміти, як у ньому задіяні різні гравці. Починаючи від компаній, що виробляють обладнання, закінчуючи тими, що виробляють програмне забезпечення, і закінчуючи тими, що надають послуги.

Серед складнощів телекомунікаційного ландшафту ефективно управління проектами стає стрижнем, який забезпечує успішне розгортання нових технологій, розширення мереж і адаптацію до мінливого регуляторного середовища. Керівники проектів у телекомунікаційній галузі відіграють вирішальну роль у визначенні цілей проекту, визначенні ключових зацікавлених сторін та розробці комплексних планів проекту - це фундаментальні кроки на етапі ініціації.

Це створює основу для успішного життєвого циклу проекту. Важливо збалансувати обсяг проекту з технологічними інноваціями та реальними потребами [21]. Керівники проектів повинні враховувати такі фактори, як вимоги до інфраструктури, регуляторні обмеження та запити споживачів. З огляду на динамічний характер телекомунікаційного ландшафту, ефективно управління ризиками має першорядне значення. Виявлення потенційних викликів, розробка стратегій пом'якшення наслідків та адаптація до непередбачуваних обставин є критично важливими компонентами успішного управління проектами [27].

Забігаючи наперед, можна сказати, що телекомунікаційний ландшафт зазнає подальшої трансформації. Нові технології, такі як штучний інтелект, периферійні обчислення та передова аналітика даних, відіграватимуть ключову роль у формуванні майбутнього телекомунікацій.

Стійкість галузі та її здатність адаптуватися до цих змін визначатимуть її успіх у наступні роки. Коли ми рухаємося складними шляхами телекомунікаційного ландшафту, одне залишається зрозумілим: потреба в стратегічному, далекоглядному управлінні проектами ще ніколи не була такою важливою, як зараз. Від викликів, пов'язаних з інтеграцією нових технологій, до можливостей, які відкриває все більш взаємопов'язаний світ, телекомунікаційний ландшафт - це полотно, на якому інновації та ефективне управління проектами малюють картину нашого взаємопов'язаного майбутнього [7].

Управління проектами є однією з ключових функцій сучасного менеджменту, особливо в умовах інформаційного суспільства, де реалізація більшості змін відбувається в проектному форматі. Проект - тимчасове зусилля для досягнення однієї чи кількох визначених цілей.[1] Відповідно до стандарту РМВОК, проект – це тимчасова діяльність, спрямована на створення унікального продукту, послуги або результату. Тимчасовий характер проектів вказує на початок і завершення проектної роботи або її фази [2].

ІТ-проекти мають низку відмінних рис, зокрема:

- висока динамічність та змінність вимог;
- складність у плануванні через швидкий розвиток технологій;
- значна роль людського чинника;

У телеком сфері ІТ-проекти є основою для реалізації інновацій, автоматизації бізнес-процесів, модернізації інфраструктури та розвитку нових послуг. Життєвий цикл проекту має ітеративну або інкрементальну структуру, що

дозволяє адаптуватися до змін зовнішнього середовища та вимог замовника та включає декілька етапів: ініціація, планування, виконання, моніторинг і контроль, завершення.

Основні особливості ІТ-інфраструктури телекомунікаційної галузі:

1. високе навантаження (великий обсяг трафіку, кількість абонентів, складність реалізованого функціоналу) при обмеженому бюджеті;
2. висока вартість помилки/відмови системи;
3. безперервний цикл виробництва, надання послуг, обмежений бюджет на інновації;
4. швидкісне впровадження та розгортання нового функціоналу та амортизація існуючого при обмеженому бюджеті;
5. інтеграція з багатьма зовнішніми ІТ-системами, зачіпаються багато бізнес-процесів.

Тому вимоги до управління проектами наступні:

- Управління розробкою складних масштабованих систем, що складаються з багатьох компонентів;
- Підвищені вимоги до якості продукції;
- Забезпечення систематичної ітеративної розробки без кардинальних змін;
- Управління розробкою на основі декомпозиції функціональної архітектури і на цій основі локалізація і обмеження внесених змін;
- Управління діяльністю, пов'язаною з інтеграцією із зовнішніми системами.

Таким чином, вимоги до методологій управління проектами, визначені на основі аналізу ІТ-інфраструктури телекомунікаційної галузі, зводяться до досягнення чотирьох операційних показників:

- Вартість (зниження);

- Якість (підвищення);
- Доставка (точно в дедлайн);
- Гнучкість.

Серед основних типів проєктів у телекомунікаційному середовищі можна виділити:

- впровадження нових білінгових систем;
- розробка мобільних застосунків для користувачів;
- інтеграція CRM-систем;
- автоматизація обслуговування (чат-боти, self-service портали);
- міграція даних та перехід до хмарних рішень;
- кібербезпекові ініціативи.

Управління проєктами відіграє ключову роль у телекомунікаційній галузі, яка характеризується швидким технологічним розвитком, мінливими вимогами споживачів та динамічним регуляторним середовищем. Ефективні практики управління проєктами є надзвичайно важливими для успішної реалізації, розгортання та обслуговування складних телеком-проєктів.

Основні аспекти ролі проєкт менеджменту в телеком-галузі:

- Забезпечення успіху проєктів: Проєкт менеджери відповідають за планування, виконання та завершення телеком-проєктів, гарантуючи досягнення поставлених цілей у межах визначених термінів і бюджетів.
- Навігація в складному середовищі: Телеком-проєкти часто є складними та взаємопов'язаними, вимагаючи від проєкт менеджерів вміння орієнтуватися в технологічних, регуляторних та ринкових аспектах.

- Оркестрація ресурсів: Проджект менеджери координують людські, фінансові та технологічні ресурси, необхідні для виконання проєкту.
- Управління зацікавленими сторонами: Важливою функцією є взаємодія з різними зацікавленими сторонами, включаючи регуляторні органи, постачальників, клієнтів та внутрішні команди, для забезпечення узгодженості та підтримки проєкту.
- Баланс між інноваціями та практичністю: Проджект менеджери повинні знаходити баланс між впровадженням новітніх технологій та врахуванням реальних потреб різних користувачів та географічних контекстів.
- Адаптація до змін: У швидко мінливому телеком-ландшафті проджект менеджери повинні бути гнучкими та готовими адаптувати плани у відповідь на технологічні прориви та зміни ринкових умов.
- Управління ризиками: Проджект менеджери відповідають за ідентифікацію потенційних ризиків, розробку стратегій їхнього пом'якшення та реагування на непередбачені ситуації.
- Впровадження найкращих практик: Проджект менеджмент у телеком-галузі передбачає визначення та застосування ефективних методологій та інструментів, включаючи гнучкі підходи (Agile).
- Сприяння міжкультурній співпраці: У глобальних телеком-проєктах проджект менеджери повинні враховувати культурні нюанси та сприяти ефективній комунікації між командами з різних країн.
- Інтеграція нових технологій: Проджект менеджери відіграють ключову роль у впровадженні новітніх технологій, таких як мережі 5G та штучний інтелект, у телекомунікаційну інфраструктуру та послуги.

- **Забезпечення безперервного навчання:** У зв'язку з постійним розвитком технологій, проджект менеджери повинні підтримувати культуру безперервного навчання в командах для освоєння нових знань та навичок.

Таким чином, проджект менеджмент є критично важливим для успішного розвитку та функціонування телекомунікаційної галузі, забезпечуючи ефективно впровадження інновацій, розширення мереж та надання якісних послуг у глобальному масштабі. Проджект менеджери є ключовими фігурами, які оркеструють складні процеси та забезпечують зв'язок між технологіями, бізнесом та потребами споживачів[4].

1.1.1 Обґрунтування методології управління проєктом

Основною відмінною рисою підходів до управління проєктами є поділ на традиційне та гнучке управління проєктами, які принципово відрізняються за своїми структурами та процесами. Через постійне незадоволення традиційним планово-орієнтованим (важким) підходом, останніми роками впроваджуються різні гнучкі (легкі) методології [26], включаючи Scrum, Kanban, eXtreme Programming (XP) та інші [27, 28]. Термін «підхід до управління проєктами» - це «найвищий рівень абстракції, який використовується при описі того, як проєкт буде розроблений» [29], тоді як методологія управління проєктами є більш деталізованою і надає конкретні вказівки щодо того, як управляти проєктом [29]. У цій статті ми використовуємо термін підхід для розрізнення традиційного, гнучкого та гібридного управління проєктами, тоді як методологія відноситься до конкретних моделей, таких як Waterfall або Scrum. Терміни методи або практики стосуються окремих інструментів або прийомів, які використовуються в рамках методології, таких як індивідуальні методи планування або конкретні типи зустрічей. Однак в літературі немає єдиного використання або визначення цих термінів.

У традиційному підході до управління проєктами обсяг, час і вартість проєкту визначаються на ранніх фазах життєвого циклу. Будь-які зміни в обсязі ретельно контролюються. Однією з ключових характеристик традиційного управління проєктами є те, що воно прагне мінімізувати зміни під час виконання проєкту шляхом збору вимог, аналізу та Проєктування заздалегідь для досягнення більш якісних результатів.

Гнучке управління проєктами набуває все більшої популярності в останні роки, особливо після виходу Маніфесту гнучкої розробки програмного забезпечення на початку 21-го століття [11-13]. Гнучке управління проєктами намагається зробити виконання проєкту гнучким до змін у навколишньому середовищі та обсязі послуг.

Гнучкі вимоги мають тенденцію бути в першу чергу функціональними і досить неформальними [14]. Це досягається переважно за допомогою коротких, послідовних циклів планування та виконання [15, 16], з метою забезпечення автономності проєктних команд, частого та чесного зворотного зв'язку з клієнтами та іншими зацікавленими сторонами, а також гнучкості обсягу проєкту [17]. Для завершення проєктів, гнучке управління проєктами фокусується на інкрементних, ітеративних циклах розробки [18]. Підхід гнучкого управління проєктами є популярним у багатьох компаніях, особливо в ІТ-секторі. Гнучкі методології є більш новими і походять з розробки програмного забезпечення [19]. Основною перевагою всіх цих методологій є їхня гнучкість [20-22]. Гнучкий підхід ідеально підходить для короткострокових проєктних ініціатив, які потребують виявлення вимог та оцінки нових технологій [23]. Для невеликих автономних проєктів гнучкі методології є менш обтяжливими і більше відповідають зростаючим потребам індустрії програмного забезпечення у швидкій розробці та адаптації до постійних змін [14].

Незважаючи на ці переваги, останнім часом гнучкий підхід все частіше піддається критиці. Щоб пришвидшити процес розробки, гнучкий підхід

дозволяє командам розробників програмного забезпечення зосередитися на кінцевому продукті, а не дизайні та документації [24, 25]. Це може призвести до нехтування проектною документацією, оскільки розробка рішення може бути дуже трудомісткою, а проектна документація часто має нижчий пріоритет. Ще одним недоліком гнучкого підходу може бути певна неточність у плануванні часу та бюджету, оскільки загальний огляд проекту може стати заплутаним через постійну зміну пріоритетів завдань. Крім того, існує численні бар'єри та виклики, які перешкоджають реалізації переваг гнучких підходів в організаційному контексті. Особливо у великомасштабних трансформаційних проектах важко підтримувати основні принципи гнучкого підходу, такі як автономія команди.

Методології управління ІТ-проектами можна порівняти за такими критеріями: рівень гнучкості, вимоги до документації, швидкість реалізації, ступінь залученості замовника та масштабованість. У телеком-середовищі класичні методики можуть використовуватися на рівні інфраструктурних або нормативних проектів, тоді як Agile більше підходить для розробки ПЗ, створення мобільних застосунків та запуску нових цифрових сервісів. У даній роботі буде представлено гібридну методологію Water-Scrum-Fall для створення інтернет-сервісу для телекомунікаційної компанії.

Методологія Water-Scrum-Fall (Рис. 1.1) Веста [13] поєднує традиційну методологію Waterfall з гнучким Scrum. Вона базується на думці, що для проекту повинна існувати структурна основа, яка забезпечується за допомогою методології Waterfall, розробленої на сайті . В рамках цього традиційного процесного підходу інтегровані гнучкі фази. Складна частина проекту, розробка, виконується на основі методології Scrum [13].

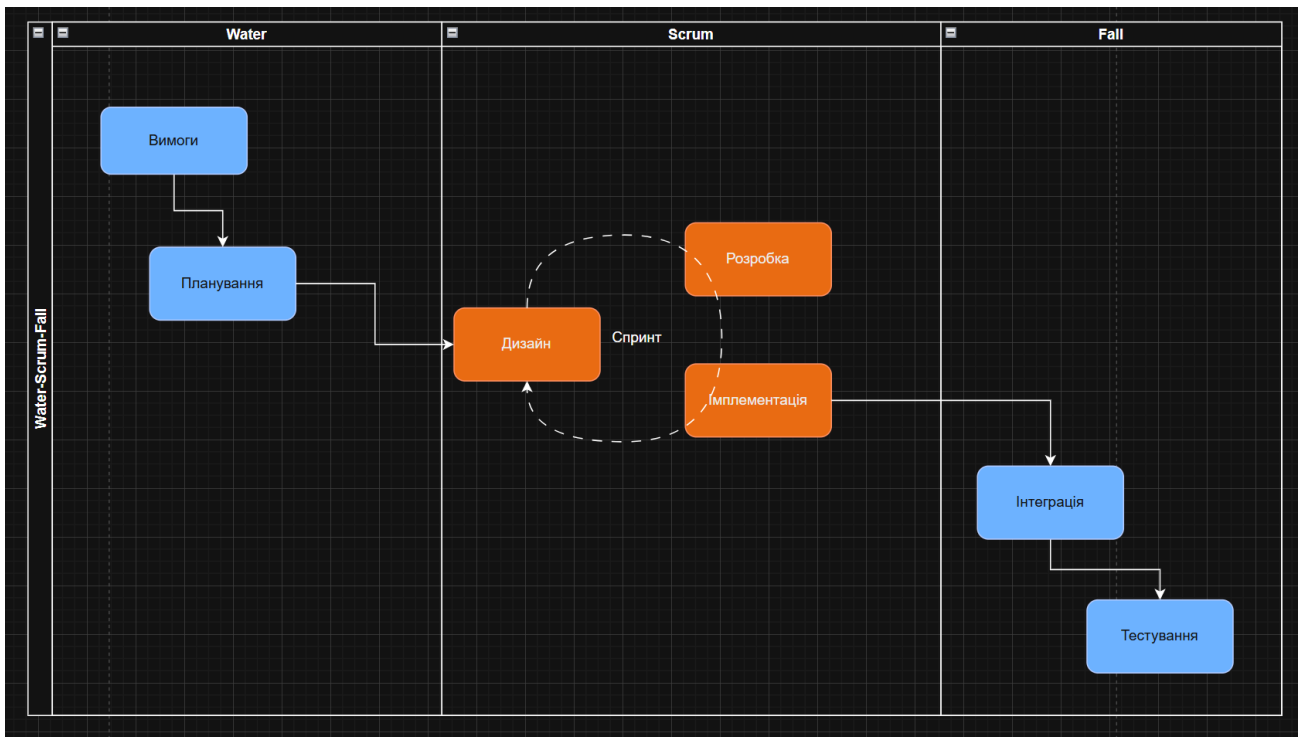


Рис. 1.1. – діаграма Water-Scrum-Fall

Waterfall (каскадна) методологія — це найвідоміша та найпростіша традиційна модель управління проектами, яка виконується послідовно: кожна фаза завершується перед початком наступної. У разі потреби, можливо повернутися до попередньої фази для внесення змін чи виправлень.

Scrum — одна з найпопулярніших гнучких (agile) методологій. Основні характеристики Scrum включають ітеративну розробку (спринти), самоуправління команди, гнучкість до змін вимог. Проект ділиться на короткі спринти тривалістю 2–4 тижні. На початку спринту команда обирає функціонал для реалізації, наприкінці — презентує результати замовнику та отримує зворотний зв'язок. Замовник активно залучений до процесу розробки. Список завдань спринту (backlog) фіксується на весь період спринту й не змінюється. Scrum дозволяє адаптуватися до змін ринку, команди або вимог без перегляду контракту.

Water-Scrum-Fall — це гібридний підхід, що поєднує етапи Waterfall і Scrum. Спочатку проект планується традиційно: визначаються вимоги, бюджет, час,

готується документація. Далі реалізація виконується ітеративно за принципами Scrum, що дозволяє зменшити ризики переробок та затримок. Після завершення розробки проєкт передається замовнику з використанням контрольних точок як у Waterfall, що допомагає оцінити якість і прийняти проєкт. Water-Scrum-Fall часто використовується як перехідний етап для компаній, які переходять від традиційного до гнучкого управління проєктами. Він вимагає введення нових ролей, таких як Scrum Master та Product Owner.

1.2 Стратегічний аналіз проєкту

У даному пункті проведемо стратегічний аналіз проєкту, який є основою для прийняття ефективних управлінських рішень на всіх етапах життєвого циклу проєкту. Зокрема, зробимо SWOT-аналіз, що дозволить виявити сильні та слабкі сторони проєкту, а також зовнішні можливості й загрози, що сприяє формуванню обґрунтованої стратегії розвитку, проаналізуємо ризики, щоб заздалегідь виявити потенційні проблеми, знизити їх негативний вплив і забезпечити безперервність виконання проєкту та визначимо зацікавлені сторони, адже їх підтримка, інтереси й очікування мають критичне значення для успішної реалізації ініціативи.

У сучасному динамічному середовищі телекомунікаційних та ІТ-послуг ключовим фактором успіху є здатність компанії адаптуватися до змін ринку та ефективно конкурувати на технологічному рівні. Аналіз конкурентного середовища дозволяє виявити сильні та слабкі сторони основних гравців ринку, визначити рівень інноваційності їхніх рішень, а також оцінити стратегії, що застосовуються для залучення та утримання клієнтів. Проведення порівняльного аналізу конкурентів є необхідною складовою стратегічного планування, що забезпечує ухвалення обґрунтованих управлінських рішень при впровадженні ІТ-продуктів у телеком-галузі.

Для порівняння візьмемо до прикладу інтернет-системи двох найбільших постачальників інтернет послуг в Україні: Київстар та Воля (Табл. 1.1).

Порівняння конкурентів

Компанія	Основні послуги	Переваги	Недоліки
Воля	<ul style="list-style-type: none"> • Інтернет • Телебачення • Телефонія 	<ul style="list-style-type: none"> • Широкий вибір тарифних планів • Зручний онлайн-кабінет • Мобільний додаток 	<ul style="list-style-type: none"> • Обмежена зона покриття в деяких регіонах • Висока вартість деяких послуг
Київстар	<ul style="list-style-type: none"> • Інтернет • Телебачення • Мобільний зв'язок • Телефонія 	<ul style="list-style-type: none"> • Велика зона покриття • Інтеграція з мобільними послугами • Підтримка 4G і 5G 	<ul style="list-style-type: none"> • Нестабільна швидкість інтернету в деяких районах • Недосконалий онлайн-кабінет

При розробці нашого інтернет-сервісу будуть враховані наступні аспекти:

1. Широкий вибір тарифних планів: Запропонуємо гнучкі тарифні плани, які зможуть задовольнити різні потреби клієнтів.
2. Зручний онлайн-кабінет: Створимо інтуїтивно зрозумілий і зручний інтерфейс для управління послугами.
3. Мобільний додаток: Розробимо повноцінний мобільний додаток для зручного доступу до послуг з будь-якого пристрою.
4. Розширена зона покриття: Врахуємо необхідність забезпечення широкої зони покриття.
5. Стабільність і якість обслуговування: Зосередимося на забезпеченні стабільної та високоякісної роботи інтернет-сервісу.

Наш інтернет-сервіс буде мати наступні переваги:

- Інтеграція з різними видами послуг: Поєднання інтернету, телебачення та телефонії в одному пакеті.

- Покращена система моніторингу сервісів: Використання сучасних інструментів для моніторингу та управління послугами.
- Оптимізація бізнес-процесів: Завдяки використанню новітніх технологій для автоматизації та оптимізації внутрішніх процесів компанії.
- Висока якість обслуговування клієнтів: Забезпечення стабільної роботи послуг та швидке реагування на запити клієнтів.

Дані переваги дозволять нам запропонувати конкурентоспроможний продукт, який буде відповідати сучасним вимогам ринку та задовольняти потреби клієнтів телекомунікаційної компанії.

Важливо підкреслити значущість комплексного підходу до оцінки його перспектив. SWOT-аналіз дозволяє всебічно розглянути внутрішні та зовнішні фактори, що впливають на проєкт, ідентифікувати його сильні та слабкі сторони, а також виявити можливості та загрози, які можуть виникнути в процесі реалізації. Це дозволить сформувану стратегію, яка враховує всі аспекти і сприяє успішному впровадженню та експлуатації інтернет-сервісу. В Таблиці 1.2 ми можемо бачити результати аналізу.

Таблиця 1.2.

SWOT аналіз

Strengths	Weaknesses
<ol style="list-style-type: none"> 1. <u>Зручність та доступність</u>: розробка сучасного та зручного сайту покращить доступність послуг для клієнтів. 2. <u>Конкурентоспроможність</u>: наявність функціонального та привабливого веб-сайту може підвищити конкурентоспроможність компанії на ринку. 	<ol style="list-style-type: none"> 1. <u>Високі початкові витрати</u>: витрати на розробку та підтримку сайту можуть бути значними. 2. <u>Необхідність у технічних знаннях</u>: потрібна наявність фахівців з необхідними знаннями та навичками для створення та підтримки сайту.

<p>3. <u>Збільшення охоплення аудиторії</u>: можливість залучити ширшу аудиторію через онлайн-платформу.</p> <p>4. <u>Автоматизація процесів</u>: Автоматизація продажу послуг та управління клієнтськими даними можуть знизити витрати на обслуговування.</p> <p>5. <u>Брендування</u>: можливість посилення іміджу та впізнаваності бренду через професійний сайт.</p>	<p>3. <u>Проблеми з безпекою</u>: ризики, пов'язані з кібербезпекою та захистом даних клієнтів.</p> <p>4. <u>Складність інтеграції</u>: можливі складнощі під час інтеграції сайту з існуючими системами та платформами компанії.</p> <p>5. <u>Залежність від Інтернет-з'єднання</u>: ефективність роботи сайту залежить від якості Інтернет-з'єднання користувачів.</p>
<p>Opportunities</p>	<p>Threats</p>
<p>1. <u>Зростання ринку онлайн-послуг</u>: збільшення попиту на онлайн-послуги та покупки через інтернет.</p> <p>2. <u>Розширення географії продажів</u>: можливість виходу на нові ринки та залучення клієнтів із різних регіонів.</p> <p>3. <u>Персоналізація послуг</u>: можливість пропозиції персоналізованих послуг та акцій для різних сегментів клієнтів.</p> <p>4. <u>Аналіз даних</u>: збір та аналіз даних щодо поведінки клієнтів для покращення маркетингових стратегій.</p> <p><u>Додаткові сервіси</u>: введення додаткових сервісів та функцій, таких як чат-підтримка та онлайн-консультації</p>	<p>1. <u>Конкуренція</u>: високий рівень конкуренції над ринком онлайн-послуг.</p> <p>2. <u>Технологічні зміни</u>: швидке старіння технологій та необхідність постійного оновлення та покращення сайту.</p> <p>3. <u>Кіберзагрози</u>: загрози з боку хакерів та вірусів, які можуть призвести до витоку даних та порушення роботи сайту.</p> <p>4. <u>Регуляторні зміни</u>: зміни в законодавстві щодо онлайн-торгівлі та захисту даних.</p> <p><u>Економічна нестабільність</u>: вплив економічної нестабільності на купівельну спроможність клієнтів та інвестиції у технології.</p>

Окрім життєспроможності проєкту, також важливо оцінювати можливі ризики, пов'язані з його запуском та їхній вплив на функціонування. Це допоможе запобігти більшості та/або якомога краще підготуватися до форс-

мажору. Для проекту створення інтернет-сервісу для телекомунікаційної компанії, ризики представлені в Таблиці 1.3.

Таблиця 1.3

Аналіз ризиків

Технічні ризики:	Імовірність	Вплив	Запобіжні заходи
1	2	3	4
Технічні проблеми або збої в роботі сайту	Середня	Високий	Регулярне тестування та моніторинг сайту, наявність резервних копій, використання перевірених технологій та хостингу
Кібератаки та витік даних	Висока	Високий	Впровадження сучасних систем захисту, регулярні оновлення безпеки, навчання співробітників з питань кібербезпеки.
Несумісність з існуючими системами	Середня	Середній	Ретельна інтеграція та тестування, вибір сумісних технологій, консультації з технічними фахівцями.
Фінансові ризики:			
Перевищення бюджету проекту	Середня	Високий	Чітке планування бюджету, резервування коштів у непередбачені витрати, контроль витрат за всіх етапах проекту.
Нестача фінансових ресурсів для підтримки та оновлення сайту	Середня	Високий	Залучення інвесторів, розробка довгострокового фінансового плану, моніторинг фінансового стану проекту.
Ризики, пов'язані з людським фактором:			
Нестача кваліфікованих кадрів	Середня	Високий	Навчання та розвиток поточних співробітників, залучення фахівців на аутсорсинг, створення привабливих умов для співробітників.

Продовження табл. 1.3.

1	2	3	4
Опір змінам із боку співробітників	Низька	Середній	Проведення тренінгів та семінарів, пояснення переваг нового сайту, залучення співробітників до процесу змін.
Маркетингові ризики:			
Низька відвідуваність сайту	Середня	Високий	Розробка та реалізація ефективної маркетингової стратегії, використання SEO та SMM, аналіз та покращення користувальницького досвіду.
Низька конверсія відвідувачів у клієнтів	Середня	Високий	Оптимізація користувальницького інтерфейсу та навігації, проведення А/В тестування, покращення контенту та пропозицій.
Юридичні ризики:			
Невідповідність законодавства	Середня	Високий	Юридичні консультації, дотримання всіх і правил, регулярне оновлення правової інформації.

1	2	3	4
Проблеми з авторськими правами та інтелектуальною власністю	Низька	Середній	Перевірка всіх матеріалів на відповідність авторським правам, дотримання законів про інтелектуальну власність.
Екологічні ризики:			
Екологічні вимоги до дата-центрів	Низька	Низький	Використання екологічно чистих рішень для дата-центрів, дотримання всіх екологічних норм і стандартів.

Далі необхідно визначити зацікавлені сторони проєкту. За визначенням РМВОК[4] зацікавлена сторона – будь-яка особа, група чи організація, яка може впливати на проєкт або відчувати його вплив. Рання ідентифікація, аналіз інтересів та розробка стратегії взаємодії дають змогу:

- узгодити очікування щодо бюджету, термінів і якості;
- заздалегідь передбачити конфлікти та ризики;
- побудувати прозору систему комунікацій, що підвищує довіру та знижує опір змінам.

Методика, застосована в роботі — чотириетапний цикл Identify → Assess → Plan → Engage (рекомендований PMI), доповнений матрицею Power/Interest (Табл.1.4, Табл. 1.5)

Матриця впливу

Категорія	Представники	Ключові інтереси та очікування	Вплив / влада
Власники бізнесу (Sponsor, C-level)	CEO, CFO, COO	Дотримання стратегічних KPI (NPS +15 %, OPEX – 20 %), швидкий time-to-market, позитивний NPV	Високий – ухвалюють рішення “Go/No-Go”, забезпечують фінансування
Замовник-представник (Product Owner)	Директор з цифрових продуктів	Реалізація потрібного функціоналу, пріоритезація backlog, ROI	Високий – визначає цінність і пріоритети
Кінцеві користувачі (Абоненти B2C/B2B)	Індивідуальні та SME-клієнти	Зручність, 24/7 доступ, прозорі тарифи, безпечні платежі	Середній – впливають через NPS, відтік, публічні відгуки
Внутрішні підрозділи	ІТ-департамент, Білінговий відділ, CRM-служба, Call-центр	Безпечна інтеграція, спрощення операцій, зменшення ручної роботи	Високий – постачають експертизу, інтерфейси та ресурси
Проектна команда	PM, BA/PO, Tech Lead, Devs, QA, DevOps, UX	Чіткі вимоги, стабільний процес, доступні ресурси, відсутність мікроменеджменту	Високий в зоні виконання – формують продуктові рішення
Постачальники та партнери	Хмарний провайдер (AWS), Постачальник білінгу, SMS/Email-шлюзи	SLA, бізнес-безперервність, своєчасні платежі	Середній – впливають на доступність сервісу
Регулятор та аудитори	НКЕКУ, CERT-UA, фінансовий аудит	Дотримання законів (ЗУ «Про електронні комунікації»), GDPR/ЗУ «Про захист даних»	Високий – можуть зупинити або оштрафувати проєкт

Продовження табл. 1.4.

Інвестори / Рада директорів	Приватні чи венчурні інвестори	Позитивна економіка: IRR > 20 %, Payback ≤ 2 р.	Середній/Високий – затверджують бюджет і вимоги прибутковості
Користувацька спільнота / медіа	Телеком-блоги, профільні видання	Інноваційність, публічна репутація оператора	Низький – але високий вплив на бренд

Таблиця 1.5

Матриця Power\Interest

	Високий інтерес	Низький інтерес
Висока влада	Вести тісну співпрацю (Manage Closely) <ul style="list-style-type: none"> • CEO, CFO, COO • Product Owner • IT/Білінг керівники 	Тримати задоволеними (Keep Satisfied) <ul style="list-style-type: none"> • Регулятор (НКЕКУ) • Інвестори, рада директорів
Низька влада	Тримати поінформованими (Keep Informed) <ul style="list-style-type: none"> • Команда підтримки • Кінцеві користувачі-бета-тестери 	Моніторити (Monitor) <ul style="list-style-type: none"> • Профільні медіа • Широка спільнота абонентів

Таким чином, чітке картування зацікавлених сторін, їхніх ролей, рівня впливу й очікувань дозволяє сформувати керовану систему комунікацій, мінімізувати конфлікти та забезпечити досягнення стратегічних і операційних цілей проєкту розробки інтернет-системи для телекомунікаційної компанії.

1.3 Технічне завдання та паспорт проєкту

1.3.1 Опис, мета, цілі проєкту

Опис проєкту: Проєкт спрямований на створення інтернет-системи обслуговування абонентів телекомунікаційної компанії, яка поєднує особистий кабінет, онлайн-платежі, підтримку, інтерактивну аналітику та адміністративний

модуль. Рішення охоплює весь цифровий ланцюг взаємодії клієнта з оператором – від підключення послуг і розрахунку вартості до отримання статистики споживання та управління тарифами. Система будується на базі гібридної методології Water-Scrum-Fall, що забезпечує структурованість планових етапів і гнучкість ітераційної розробки. Проєкт дозволить оптимізувати процеси управління клієнтськими запитами, розширити канали зворотного зв'язку, автоматизувати обробку звернень, а також підвищити рівень доступності цифрових послуг.

Мета: забезпечення цифрової трансформації телекомунікаційної компанії шляхом впровадження сучасного веб-сервісу, який відповідає вимогам ринку та очікуванням користувачів щодо якості, швидкості та зручності отримання послуг.

Цілі проєкту:

1. Створення функціонального дизайну та архітектури інтернет-сервісу з урахуванням специфіки галузі.
2. Підвищити рівень задоволеності клієнтів (NPS) не менше ніж на 15 % протягом першого року після запуску завдяки зручним онлайн-сервісам.
3. Скоротити навантаження на контакт-центр на 25 % шляхом переведення типових операцій у режим самообслуговування.
4. Забезпечення високої продуктивності та масштабованості системи для обслуговування широкої бази абонентів.
5. Забезпечити приріст 1 000+ нових абонентів у перші шість місяців за рахунок швидкого підключення та прозорого тарифоутворення.
6. Оптимізувати операційні витрати на обробку заявок і оплат мінімум на 20 % завдяки автоматизації back-office-процесів.

Продукт проєкту:

- Повнофункціональний інтернет-сервіс (SPA на React) з адаптивним UI, особистим кабінетом і кастомним «кошиком» послуг.
- Мобільна версія PWA з push-нотифікаціями про платежі, аварії та акції.
- Backend-сервіси (Node.js + Express) з REST-/GraphQL-API для інтеграції з CRM, білінгом, SMS/e-mail-шлюзами.
- Аналітичний модуль: карта покриття, модель «корисності покриття», панель KPI для менеджменту.
- Адмін-консоль для управління тарифами, контентом, користувачами та перегляду журналів подій.

Цільова аудиторія:

- Індивідуальні абоненти (B2C) віком 18–55 років, що цінують самообслуговування і швидкий сервіс.
- Малий та середній бізнес (SoHo/SME), який потребує прозорого управління пакетами послуг і детальної статистики споживання.
- Внутрішні підрозділи оператора — менеджери з продажу, техпідтримка, аналітичні відділи, яким необхідний центральний інструмент роботи з клієнтськими даними.

Потреби аудиторії:

- Миттєве підключення або зміна тарифу без візиту до офісу.
- Прозора історія оплат і можливість швидко закрити заборгованість.
- Оперативне рішення технічних проблем через онлайн-тікети та чат-бот.
- Доступ до персоналізованих пропозицій на підставі фактичного використання послуг.

Опис цінності для бізнесу:

- Фінансова вигода: економія OPEX за рахунок скорочення ручних операцій та підвищення конверсії продажів (up-/cross-sell) через персоналізовані акції.

- Конкурентна перевага: швидший тайм-то-маркет нових послуг (додавання тарифу у бек-офісі → відразу доступно на порталі).
- Дані як актив: централізовані логі та поведінкова аналітика забезпечують точний таргетинг і прогнозування відтоку абонентів.
- Гнучкість масштабування: хмарна інфраструктура дозволяє безболісно обслуговувати пікові навантаження (акційні кампанії, масові платежі).
- Підвищення лояльності: інтуїтивний UX і 24/7 доступ зменшують кількість скарг і збільшують Net Promoter Score, що безпосередньо впливає на ARPU.

Таким чином, запропонована інтернет-система формує для оператора комплексну цифрову екосистему, яка одночасно підвищує сервісну якість для абонента і забезпечує бізнес-цілі компанії у прибутковості, масштабованості та стратегічній стійкості.

1.3.2 Дерево цілей проєкту

З метою ефективного управління проєктом та досягнення очікуваних результатів доцільно сформувавати дерево цілей (Рис. 1.2.), що дозволяє логічно структурувати стратегічні, тактичні та оперативні завдання. Такий підхід сприяє кращому розумінню взаємозв'язків між цілями, їхньому формулюванню відповідно до принципу SMART, а також забезпечує основу для планування ресурсів, термінів та відповідальності.

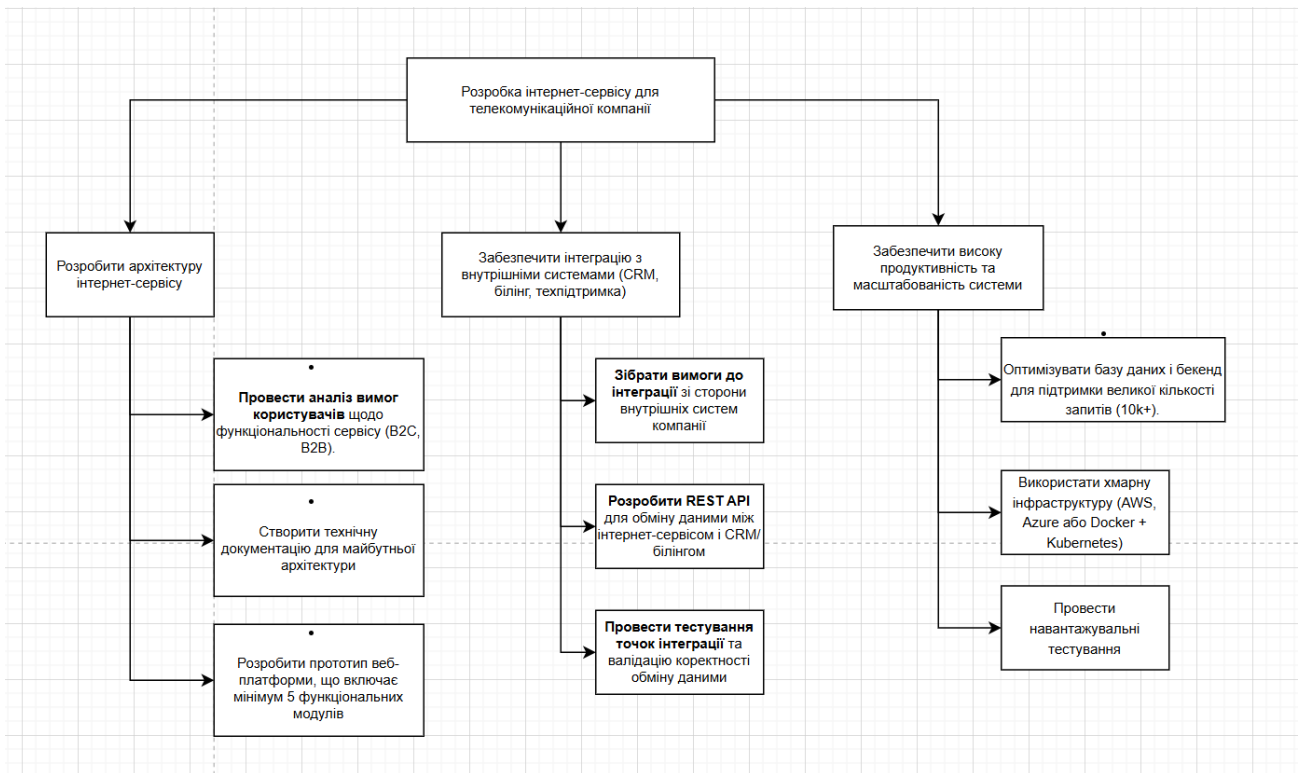


Рис. 1.2. – Дерево цілей проєкту

Опишемо кожену ціль за принципом SMART. Дана концепція є широко використовуваним інструментом управління, що забезпечує чіткість та спрямованість при визначенні цілей.

Розроблена для підвищення ефективності управління, вона передбачає, що кожна ціль має бути конкретною (Specific), вимірною (Measurable), досяжною (Achievable), релевантною (Relevant) та обмеженою в часі (Time-bound). Описуючи цілі за допомогою цього фреймворку, відбувається перехід від загальних формулювань до чітких і практичних завдань.

Це допомагає відрізнити зусилля від результатів, оскільки стає зрозуміло, що саме потрібно зробити та як буде оцінюватися успіх. (9)

Ціль 1. Розробити архітектуру інтернет-сервісу з урахуванням потреб телеком-ринку

- S (конкретність): Створити модульну архітектуру веб-платформи для надання послуг абонентам.
- M (вимірність): Не менше 5 ключових функціональних модулів: облік, підтримка, управління акаунтом, статистика, повідомлення.
- A (досяжність): Архітектура базується на сучасних фреймворках і відповідає вимогам компанії.
- R (релевантність): Потреба зумовлена недостатньою інтеграцією поточних каналів зв'язку з абонентами.
- T (терміновість): Прототип архітектури — до кінця другого місяця проєкту.

Ціль 2. Забезпечити інтеграцію з внутрішніми системами (CRM, білінг, техпідтримка)

- S: Реалізувати API-взаємодію між сервісом та внутрішніми системами компанії.
- M: Мінімум 3 успішно протестовані точки інтеграції.
- A: Підтверджено наявність відкритих API на стороні систем.
- R: Успішна інтеграція дозволить уникнути дублювання даних і підвищити ефективність обслуговування.
- T: Інтеграція завершена протягом 3 місяців з початку розробки.

Ціль 3. Забезпечити високу продуктивність та масштабованість системи

- S: Оптимізувати бекенд-частину сервісу для обробки не менше ніж 10 000 одночасних запитів.
- M: Провести 3 етапи навантажувального тестування (25%, 50%, 100%).

- А: Застосування контейнеризації та хмарної інфраструктури (наприклад, Docker + Kubernetes).
- R: Висока продуктивність — критично важливий фактор для телеком-сервісів.
- T: Перше навантажувальне тестування — на 6-му тижні розробки.

1.3.3 Економічне обґрунтування проєкту

Успішна реалізація будь-якого проєкту, особливо в умовах високої конкуренції та обмежених ресурсів телекомунікаційної галузі, вимагає не лише технічного обґрунтування, а й глибокої економічної оцінки. Економічне обґрунтування дозволяє оцінити доцільність інвестицій, визначити потенційну вигідність проєкту та забезпечити обґрунтованість прийняття управлінських рішень. Усі розрахунки були зроблені за допомогою онлайн калькулятора <https://www.calculatestuff.com/financial/npv-calculator>.

Початкові вихідні дані для розрахунку

- Загальні інвестиції: $IC = 1\,500\,000$
- Очікуваний річний дохід(вигода): $CF = 900\,000$
- Період розрахунку ефективності: 5 років
- Дисконтна ставка: $r = 10\% (0,10)$

Розрахунок чистої приведеної вартості (NPV)

Чиста приведена вартість (NPV) обчислюється за формулою:

$$NPV = \sum_{t=1}^n \frac{CF_t}{(1+r)^t} - IC, \quad (1.1)$$

де CF_t – грошовий потік за період t ;

r – ставка дисконтування;

IC – початкові інвестиції;

n – кількість періодів (років).

Розрахунки проведені в онлайн калькуляторі, результати зображені на рисунку 1.3:

$$NPV = \frac{1\,500\,000}{(1+0,10)^1} + \frac{1\,500\,000}{(1+0,10)^2} + \dots + \frac{1\,500\,000}{(1+0,10)^5} \approx 1\,911\,700 \text{ €} \quad (1.2)$$

Net Present Value (NPV) Calculator

The screenshot shows an online NPV calculator interface. At the top, 'Initial Investment' is set to \$1,500,000.00 and 'Discount Rate' is 10.000%. Under 'Cash Flow', five years are listed, each with a value of \$900,000. A blue 'Calculate' button is visible. Below the calculator, the result is displayed as '\$1,911,708.09 Net Present Value'.

Рис. 1.3. – Результати обчислення NPV

Оскільки $NPV > 0$, проєкт економічно ефективний

Внутрішня ставка дохідності(IRR)

Внутрішня ставка дохідності (IRR) визначається ставкою дисконтування, за якої $NPV = 0$:

$$0 = \sum_{t=1}^n \frac{CF_t}{(1+IRR)^t} - IC \quad (1.3)$$

За розрахунками онлайн калькулятора (Рис. 1.4), виходить, що $IRR \approx 52,8\%$

Internal Rate of Return (IRR) Calculator

Initial Investment: \$ 1500000.00

Cash Flow

Year 1: \$	900000	✖
Year 2: \$	900000	✖
Year 3: \$	900000	✖
Year 4: \$	900000	✖
Year 5: \$	900000	✖

Guess: Optional %

+ Add Year Calculate

52.796%
Internal Rate of Return

Рис. 1.4. – Результати обчислення IRR

Рентабельність інвестицій (ROI)

$$ROI = \frac{\text{Загальний прибуток за період} - IC}{IC} \times 100\% \quad (1.4)$$

де загальний прибуток за 5 років, результат обчислень в онлайн-калькуляторі зображено на рисунку 1.5:

$$CF_{\text{total}} = 900\,000 \times 5 = 4\,500\,000\text{€}$$

Return on Investment (ROI) Calculator

Amount Invested	\$	1500000.00
Amount Returned	\$	4500000.00
<input type="button" value="Calculate"/>		
ROI: 200%		
Share Results:	https://www.calculatestuff.com/financial/roi-calculator?amount_investe	<input type="button" value="Share"/> <input type="button" value="Like"/> <input type="button" value="Twitter"/>

Рис. 1.5. – Результати обчислення ROI

Підставимо значення:

$$ROI = \frac{4\,500\,000 - 1\,500\,000}{1\,500\,000} = 2.0 * 100\% = 200\%$$

Отже, кожен вкладений долар приносить два долари чистого прибутку. Це високий показник, який свідчить про економічну ефективність та інвестиційну привабливість проєкту

Термін окупності проєкту (Payback period):

$$Payback\ period = \frac{\text{Інвестиції}}{\text{Річний чистий грошовий потік}} = \frac{1\,500\,000}{900\,000} \approx 2\ \text{роки}$$

Це означає, що вкладені кошти повністю повернуться менш ніж за 2 роки, що вважається дуже хорошим показником для ІТ-проєктів у телеком-сфері.

Представимо усі розрахунки у вигляді таблиці, яка дозволяє наочно представити основні показники ефективності, порівняти їх з рекомендованими критеріями, а також зробити короткий висновок щодо кожного з них. Це важливо для подальшого прийняття управлінського рішення про доцільність інвестування в проєкт. (табл. 1.6)

Оцінка вартості проєкту

Показник економічної ефективності	Значення	Рекомендований критерій	Висновок
Загальні інвестиції	1 500 000	Не перевищує планового бюджету	Відповідає бюд
Очікуваний річний дохід	900 000	Достатній для окупності проєкту	Стійкий грошовий потік
NPV (внутрішня норма прибутковості)	1 911 700	NPV > 0	Висока економічна ефективність
IRR (Внутрішня ставка дохідності)	52,8%	>0	Інвестиційно привабливо
ROI (рентабельність інвестицій)	200%	Позитивне	Висока окупність вкладених коштів
Період окупності	2 року	≤ 3 роки	Швидке повернення інвестицій

Отже на основі визначених показників можна стверджувати, що проєкт є рентабельним та інвестиційно привабливим.

1.4 Постановка задачі дослідження

У даній роботі розв'язується комплексна науково-практична задача — обґрунтувати та розробити модель управління проєктом створення інтернет-системи самообслуговування клієнтів для телекомунікаційної компанії. Дослідження охоплює:

- аналіз галузевих особливостей та бізнес-процесів оператора зв'язку;

- вибір гібридної методології Water-Scrum-Fall для поєднання структурності та гнучкості;
- Проектування архітектури, моделі даних і алгоритмів (A*-підрахунок вартості підключення, модель «корисності покриття»);
- планування часу, вартості, ризиків і механізмів контролю.

Очікуваний результат — працездатна веб-система з особистим кабінетом абонента, підтримкою замовлень та аналітичними модулями, а також набір методичних рекомендацій щодо управління подібними ІТ-проектами у телеком-середовищі.

Для повної розробки необхідно виконати наступні завдання:

1. Бізнес-аналіз та SWOT – визначити проблеми, конкурентів, стейкхолдерів, сформулювати паспорт проекту
2. Збір вимог та формування backlog – Інтерв'ю з користувачами, специфікація use-case-ів, user-store mapping.
3. Проектування архітектури – вибір технологічного стека, Проектування REST-API, моделі бази даних
4. Розробка прототипів UI\UX – створення макетів, сценаріїв навігації, тестування з фокус-групами.
5. MVP-розробка – Авторизація, тарифи, оплати, звернення до підтримки, адмін-панель
6. Інтеграція з внутрішніми системами – CRM, білінг, SMS\імейл шлюзи.
7. Тестування – підготовка тест-кейсів, автоматизовані тести, усунення дефектів
8. Реліз – Docker-контейнери, Kubernetes, моніторинг Grafana/Prometheus
9. Контроль та звітність – Earned value, SPI\CPI, risk log, щотижневі репорти
10. Завершення та ретроспектива – Lessons Learned, фінальний акт, передача знань команді підтримки.

РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ТА ТЕХНІЧНЕ ПРОЕКТУВАННЯ

Для розробки інтернет-системи необхідно здійснити низку підготовчих етапів, які забезпечують її ефективність, масштабованість та відповідність очікуванням користувачів і бізнесу. Зокрема, критично важливими є:

- формалізація функціональних та нефункціональних вимог, що окреслюють поведінку системи, обмеження, продуктивність, доступність та безпеку;
- створення концептуальної моделі інформаційної системи — для відображення її логічної структури, взаємодії компонентів та життєвого циклу даних;
- розробка математичної моделі, що описує правила, залежності та аналітичні оцінки (наприклад, розрахунок вартості підключення або вплив навантаження);
- Проектування бази даних, як основи для збереження, обробки та аналізу даних, із дотриманням вимог нормалізації, цілісності та продуктивності.

Функціональними вимогами(FR) даного інтернет-сервісу є:

1. Реєстрація та аутентифікація абонентів (OAuth 2.0, MFA).
2. та зміна тарифних планів, історія оплат.
3. Подання заявок до служби підтримки, трекінг статусу.
4. Он-лайн розрахунок вартості підключення (алгоритм A*).
5. Адмін-панель для менеджменту тарифів, користувачів, контенту.
6. API-інтеграція з CRM та білінговою системою через REST/JSON.

Нефункціональними вимогами(NFR) інтернет-сервісу є:

1. Продуктивність: $\geq 10\ 000$ одночасних запитів, TTFB ≤ 200 мс.
2. Масштабованість — горизонтальне масштабування через Kubernetes.
3. Доступність SLA $\geq 99,5$ %.
4. Безпека — OWASP Top-10 compliance, шифрування TLS 1.3.

5. Зручність (UX) — час виконання базової операції ≤ 3 кроків.
6. Сумісність — крос-браузерна підтримка (Chrome, Edge, Safari).
7. Логування та аудит — централізований збір подій (ELK-stack).

2.1 Розробка концептуальної моделі інформаційної системи

Інформаційна модель інтернет-системи для телекомунікаційної компанії складається з різних модулів, які взаємодіють між собою для забезпечення ефективного надання послуг користувачам. Кожен модуль представляє певну функціональну частину системи, забезпечуючи виконання конкретних завдань і обмін інформацією з іншими модулями. Модулі системи включають клієнтський інтерфейс, серверну частину, модуль моніторингу сервісу та модуль баз даних. Схема концептуальної моделі зображено на рис. 2.1.

Основні компоненти системи:

1. Клієнтська частина(інтерфейс) - є головним інтерфейсом для взаємодії клієнтів із системою. Він забезпечує зручний доступ до інформації про послуги, дозволяє замовляти та управляти послугами.

Функції:

- Особистий кабінет
 - Авторизація
 - Платіжна система
 - Навігаційна панель
2. Серверна частина - Центральна частина системи, яка обробляє запити від клієнтів і виконує бізнес-логіку. Відповідає за обробку даних користувачів, управління замовленнями, забезпечення безпеки, інтеграція з іншими модулями, виконання бізнес-логіки.

Функції:

- АРІ-сервер
- Бізнес логіка

- Модуль управління заявками
3. Модуль моніторингу сервісу - Система відстеження та аналізу стану сервісів. Відповідає за відстеження статусу активних сервісів, аналіз навантаження на ресурси, генерація сповіщень про проблеми, створення тикетів на ремонт і обслуговування.

Функції:

- Моніторинг сервісів
 - Аналіз навантаження
 - Сповіщення
 - Тікет-система
4. Модуль баз даних - Центральне сховище даних системи. Зберігання даних про користувачів, замовлення, сервіси та логи, забезпечення швидкого доступу до даних, підтримка цілісності та безпеки даних

Функції:

- База користувачів
- База замовлень
- База сервісів
- Логи та аналітика

Надсистема

Надсистема включає телекомунікаційну компанію, яка забезпечує середовище та ресурси для функціонування інтернет-системи:

- **Телекомунікаційна компанія:** забезпечує інфраструктуру, підтримку та інтеграцію інтернет-системи з основними сервісами компанії.

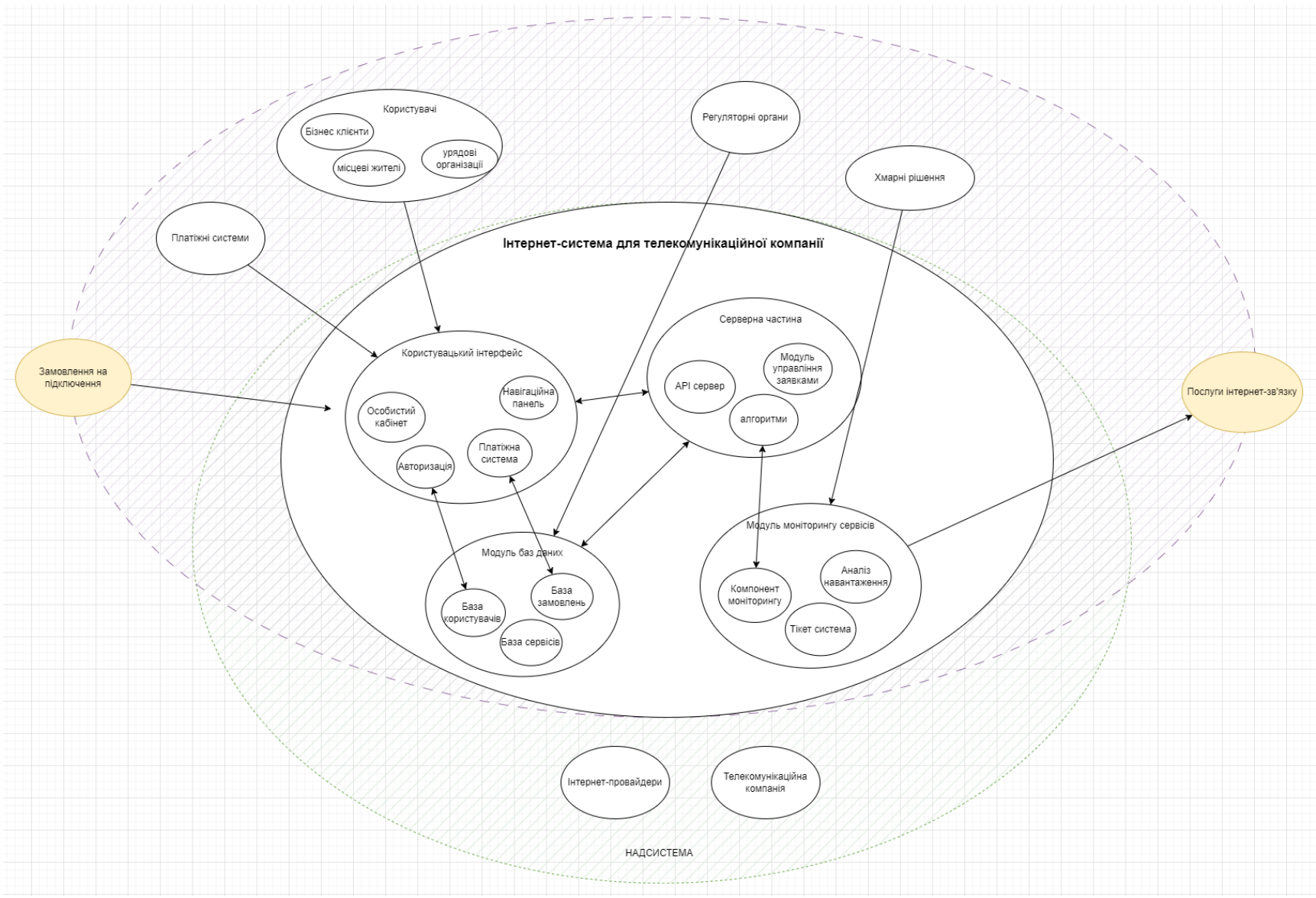


Рис. 2.1 – Концептуальна модель ІС

2.2 Розробка математичної моделі інформаційної системи

Математичне моделювання є критично важливим інструментом для нашого проєкту, оскільки дасть можливість розрахувати вартість затрачених ресурсів та мати прогнозовані витрати для різних маршрутів. Зокрема, впровадження таких моделей у веб-платформу дозволяє клієнтам отримувати точну інформацію про вартість та терміни підключення безпосередньо під час оформлення замовлення, що збільшує задоволеність клієнтів і зменшує кількість запитів до служби підтримки.

Однією з ключових математичних моделей, яка буде використовуватись у веб-платформі, є модель для розрахунку вартості прокладання оптоволоконного кабелю до місця проживання клієнта. Ця модель включає в себе алгоритм A^* , який використовується для визначення оптимального шляху прокладання кабелю, що мінімізує витрати. Крім того, модель враховує вартість кабелю за метр та додаткові витрати, пов'язані з етажністю клієнта. Це дозволяє компанії точно оцінити вартість підключення для кожного клієнта і уникнути непередбачених витрат.

Математична модель для розрахунку вартості прокладання оптоволоконного кабелю

Для розрахунку вартості прокладання оптоволоконного кабелю використовуватиметься наступна формула:

$$V_{min} = \left(\sum_{i=1}^N d_i \cdot C \right) + H \cdot E + F \quad (2.1)$$

де:

- V — загальна вартість прокладання кабелю,
- d_i — відстань між вузлами на оптимальному шляху (метри),
- C — вартість кабелю за метр,
- H — номер поверху клієнта,
- E — додаткова вартість за підключення на кожен поверх,
- F — фіксована базова вартість підключення.

Параметри формули:

Вартість кабелю за метр C : це встановлена ціна за метр кабелю, яка включає матеріальні витрати та вартість роботи з прокладання.

Номер поверху клієнта N : номер поверху, на якому знаходиться квартира клієнта. Враховується додаткова складність та витрати на прокладання кабелю на вищі поверхи.

Додаткова вартість за підключення на кожен поверх E : вартість, яка додається за кожен поверх, починаючи з другого. Це враховує додаткові матеріали та зусилля, необхідні для прокладання кабелю на висоту.

Фіксована базова вартість підключення F : це стартова вартість підключення, яка включає підготовчі роботи, обстеження об'єкту та інші початкові витрати, незалежно від довжини кабелю та етажності.

Алгоритм A^* (A -зі зірочкою) дозволяє визначити найкоротший та найекономічніший шлях для прокладання кабелю. Є найшвидшим та оптимальнішим алгоритмом серед інших собі подібних, на рис. 2.2 зображено порівняння даного методу у порівнянні з іншими:

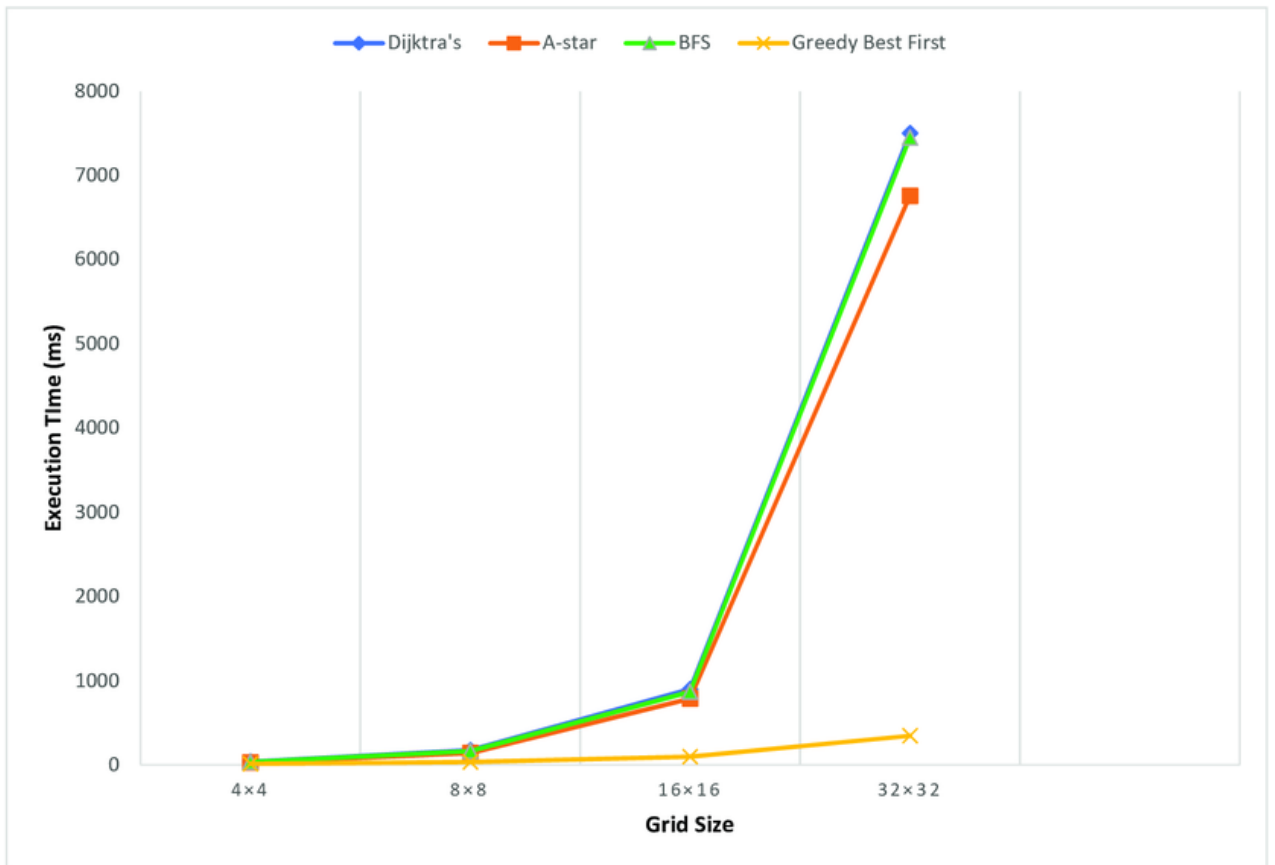


Рис. 2.2 – Порівняння методів пошуку шляху

Розпишемо порядок роботи алгоритму:

Ініціалізація: Початкова точка (станція підключення) додається до відкритого списку.

Оцінка вартості: Для кожної точки обчислюється вартість $f(n) = g(n) + h(n)$, де $g(n)$ — шляху від початкової точки до поточної, а $h(n)$ — евристична оцінка вартості шляху від поточної точки до кінцевої.

Вибір шляху: Вибирається точка з найнижчим значенням $f(n)$ та додається до закритого списку.

Оновлення сусідів: Для всіх сусідніх точок обчислюється вартість шляху i , якщо знайдено кращий шлях, оновлюються значення $g(n)$ та $f(n)$.

Повторення: Процес повторюється, поки кінцева точка не буде додана до закритого списку або поки відкритий список не стане порожнім.

Основні формули та параметру алгоритму:

Оцінка вартості шляху

$$f(n) = g(n) + h(n) \quad (2.2)$$

де

$f(n)$ - загальна оцінка вартості шляху через вузол n ,

$g(n)$ - фактична вартість шляху від початкового вузла до n ,

$h(n)$ - евристична оцінка вартості шляху від вузла n , до кінцевого вузла

Фактична вартість шляху

$$g(n) = \sum_{i=1}^k d_i \quad (2.3)$$

де

k - кількість вузлів від початкового вузла,

d_i - відстань між вузлами i та $i+1$.

Евристична оцінка

Евристична оцінка $h(n)$ може бути різною залежно від конкретної задачі, часто використовується евклідова або мангетенська відстань:

Евклідова відстань

$$h(n) = \sqrt{(x_n - x_{goal})^2 + (y_n - y_{goal})^2} \quad (2.4)$$

Мангетенська відстань

$$h(n) = |x_n - x_{goal}| + |y_n - y_{goal}| \quad (2.5)$$

де

Дана модель дозволяє телекомунікаційній компанії оптимізувати витрати на прокладання кабелю та надавати клієнтам точні розрахунки вартості підключення, що підвищує ефективність та конкурентоспроможність компанії.

Розробимо наступну модель, що буде частиною системи моніторингу для адміністратору інтернет сервісу - корисність покриття. Математичне моделювання корисності покриття є важливим інструментом для адміністратора телекомунікаційної компанії. Ця модель дозволяє оцінити ефективність використання інфраструктури в певному районі, враховуючи поточне покриття та потенціал для підключення нових клієнтів. Використання такої моделі дозволяє оптимізувати ресурси, планувати розширення мережі та приймати обґрунтовані рішення щодо інвестицій.

Опис моделі

Модель корисності покриття базується на оцінці ваги будинків у певному радіусі від об'єктів, де вже проведено оптоволоконний інтернет. Вага будинків зростає з віддаленістю від найближчого будинку з підключенням, що дозволяє визначити області з низьким покриттям та високим потенціалом для нових підключень. Модель розраховує коефіцієнт корисності для кожного району, що допомагає приймати рішення про подальший розвиток мережі.

Формула для розрахунку корисності покриття:

$$U = \frac{\sum_{i=1}^N w_i \cdot P_i}{\sum_{i=1}^N w_i}, \quad (2.6)$$

де

N - загальна кількість будинків у радіусі аналізу,

w_i - ваговий коефіцієнт для будинку i ,

P_i - показник підключення для будинку i (1 - підключено, 0 - не підключено)

Розрахунок вагових коефіцієнтів

Вагові коефіцієнти w_i визначаються залежно від відстані d_i до найближчого будинку з підключенням.

Формула вагового коефіцієнта:

$$w_i = e^{\alpha \cdot d_i} \quad (2.7)$$

, де

α - коефіцієнт масштабування, що визначає швидкість зростання ваги з відстанню,

d_i - відстань від будинку i до найближчого будинку з підключенням

Параметри моделі

- 1 Загальна кількість будинків (N): кількість будинків у вибраному радіусі аналізу.
- 2 Ваговий коефіцієнт (w_i): визначає важливість кожного будинку залежно від відстані до найближчого підключеного будинку.
- 3 Показник підключення (P_i): показує, чи підключено будинок до мережі (1) або ні (0).
- 4 Відстань до найближчого підключеного будинку (d_i): розраховується за допомогою географічних координат або іншої методики визначення відстані.
- 5 Коефіцієнт масштабування α : визначає, наскільки швидко зростає вага з віддаленістю.

Ця модель дозволяє адміністратору оцінити поточний стан покриття в будь-якому районі та виявити області з високим потенціалом для розширення мережі. Наприклад, якщо в певному районі коефіцієнт корисності UUU низький, це може вказувати на низьку ефективність покриття і потребу в підключенні нових клієнтів.

Команда може використовувати цю інформацію для прийняття рішень про інвестиції у розширення мережі, модернізацію обладнання або проведення маркетингових кампаній для залучення нових клієнтів

2.2.1 Диференціальна синхронізація баз даних

У рамках адміністрування інтернет-системи телекомунікаційної компанії важливо забезпечити оперативну синхронізацію даних між основним сервером і

інтерфейсом адміністратора. Одним із ключових сценаріїв є необхідність доступу адміністратора до актуальних даних про користувачів, підписки та звернення майже в режимі реального часу. Це дозволяє швидко виявляти критичні запити, аналізувати навантаження, а також знижує ймовірність прийняття рішень на основі застарілої інформації.

Ураховуючи можливі обмеження в пропускній здатності мережі, чутливість даних та потребу в економному споживанні ресурсів, доцільним є впровадження локального копіювання бази даних із використанням диференціального підходу на основі протоколу RSync (Remote Synchronization). На відміну від повного дублювання, RSync дозволяє передавати лише змінені частини файлів, що забезпечує економію часу, мережевого трафіку та обчислювальних ресурсів. Приклад роботи протоколу зображено на рис. 2.3.

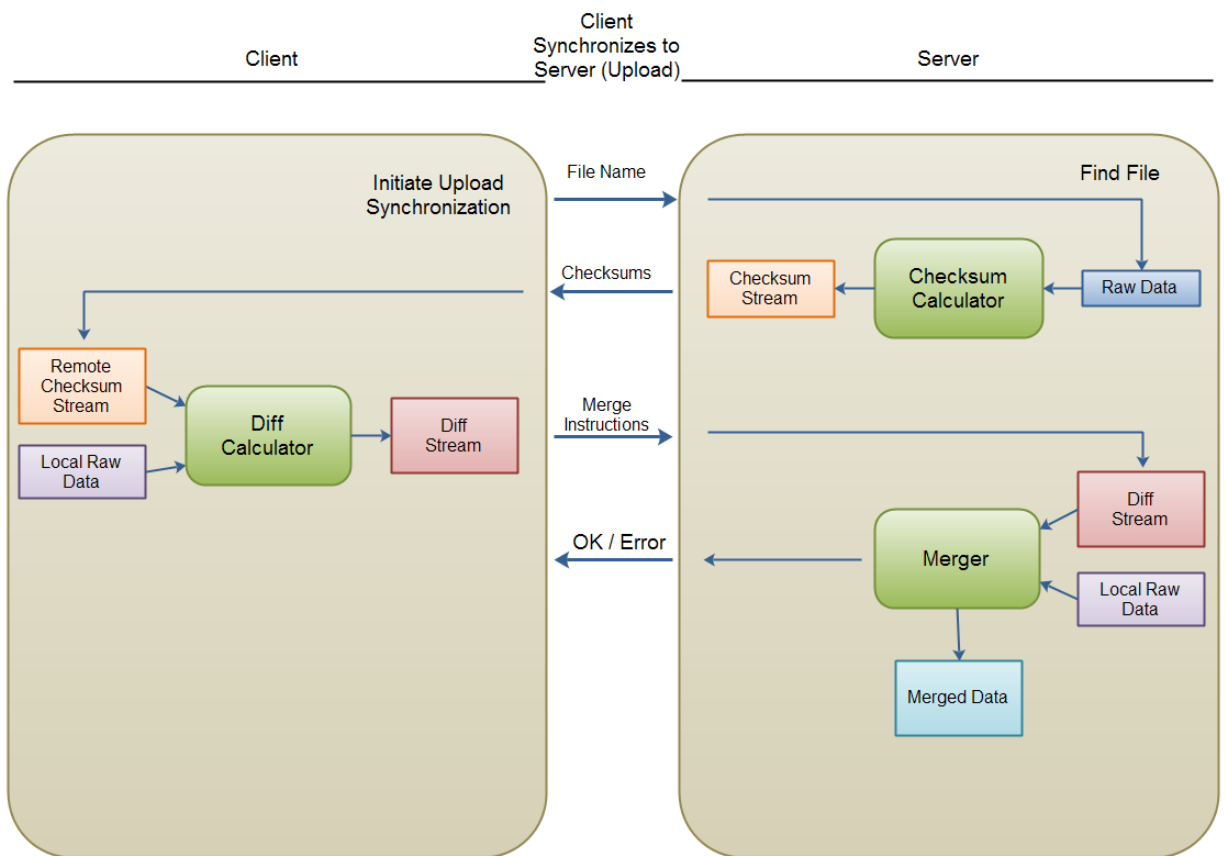


Рис. 2.3. – Схема роботи протоколу RSync

Принцип роботи RSync базується на поділі файлів на блоки фіксованого розміру та обчисленні контрольних сум для кожного блоку. Під час синхронізації використовується алгоритм rolling checksum, який дозволяє швидко виявляти змінені блоки навіть при незначному зміщенні даних. Для кожного блоку розміром SS байтів обчислюється контрольна сума за формулою:

$$R = a + b * 2^{16}, \quad (2.8)$$

де:

- $a = \sum_{i=1}^S B_i$ – сума всіх байтів у блоці,
- $b = \sum_{i=1}^S (S - i + 1) * B_i$ – зважена сума байтів.

Такий підхід дозволяє уникнути повного перерахунку контрольних сум при зсуві даних. Для підтвердження унікальності блоків додатково використовується криптографічна контрольна сума MD5:

$$MD5(B) = hash(B) \quad (2.9)$$

Процес синхронізації включає три основні етапи:

1. Сервер обчислює контрольні суми (Rolling + MD5) для кожного блоку бази даних.
2. Клієнт порівнює їх із локальними блоками.
3. Передаються лише ті блоки, які були змінені, а сервер або клієнт оновлює відповідні фрагменти.

Таким чином, застосування RSync дозволяє підтримувати актуальність адміністративної бази даних без надмірного навантаження на мережу. Цей підхід є особливо ефективним для систем, що обробляють великий обсяг транзакційних або динамічно змінюваних даних, де необхідна оперативна реакція на події.

2.3 Архітектура інтернет-сервісу

Архітектурний підхід є фундаментальним рішенням при розробці будь-якого програмного продукту, зокрема інформаційних систем для

телекомунікаційної галузі. На сучасному етапі найбільш поширеними підходами є монолітна, багаторівнева та мікросервісна архітектури.

Монолітна архітектура – це традиційний підхід, коли вся функціональність системи знаходиться у межах єдиного програмного компоненту. Такий підхід забезпечує простоту розробки, розгортання та тестування на ранніх етапах. Проте, моноліт важко масштабувати, будь-які зміни вимагають перезбірки та перезапуску всієї системи, що з часом знижує гнучкість і продуктивність. Монолітна архітектура зображена на рис. 2.4.

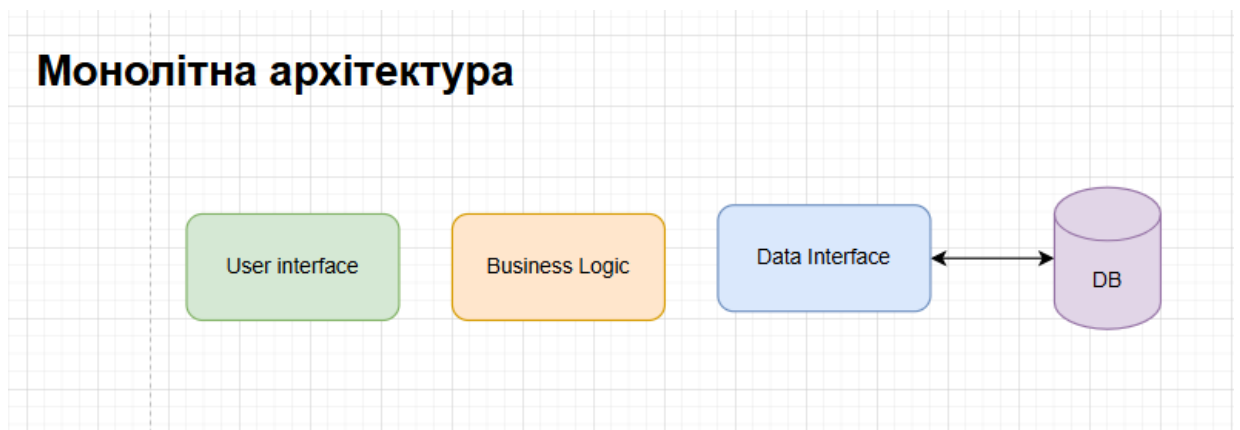


Рис. 2.4. – Монолітна архітектура

Багаторівнева (multi-tier) архітектура передбачає чіткий поділ відповідальностей системи на декілька логічних шарів (клієнтський, бізнес-логіка, база даних). Кожен шар відповідає за окремий аспект функціонування продукту і може розвиватися та підтримуватись незалежно. Це дозволяє краще управляти складністю проєкту, спрощує масштабування та обслуговування, але зберігає певну централізованість і залежність шарів один від одного. Багаторівнева архітектура зображена на рис. 2.5.

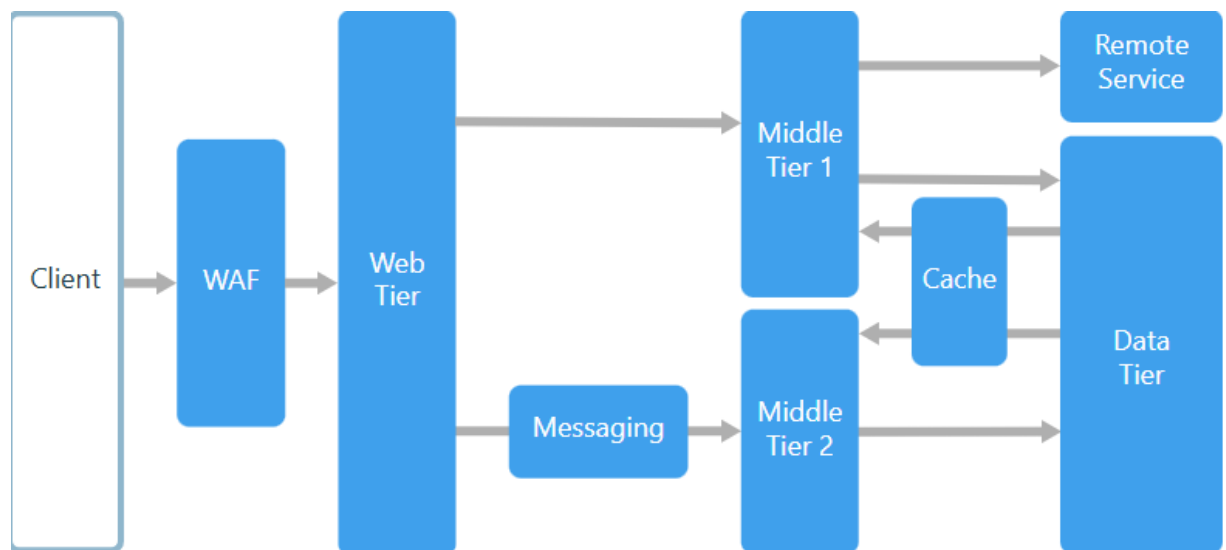


Рис. 2.5. – Багаторівнева архітектура

Мікросервісна архітектура базується на розбитті функціональності системи на незалежні сервіси, що взаємодіють між собою за допомогою стандартних інтерфейсів (наприклад, REST API).

Кожен мікросервіс відповідає за окремий бізнес-процес, легко масштабується, може бути написаний на різних технологіях та розгорнутий окремо. Проте управління великою кількістю мікросервісів вимагає високого рівня автоматизації, складної інфраструктури та має вищий поріг входу для команди. Приклад мікросервісної архітектури зображено на рис. 2.6.

Мікросервісна архітектура

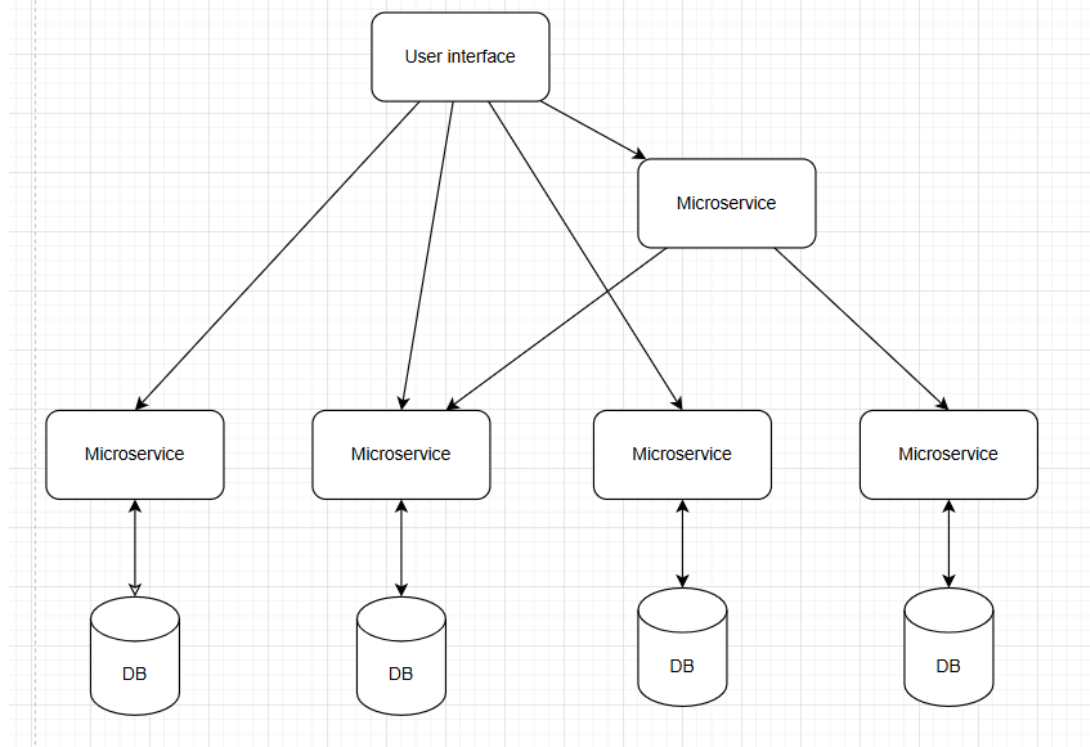


Рис. 2.6. – Мікросервісна архітектура

Для розробки інтернет-системи телекомунікаційної компанії у межах даного проєкту найбільш оптимальним рішенням є вибір саме *мікросервісної архітектури*, у зв'язку з тим, що розроблювана інтернет-система передбачає масштабування функціональності, незалежне керування модулями (аналітика, підтримка, білінг, тощо) та активну інтеграцію із зовнішніми сервісами. Це забезпечить більшу автономність модулів, пришвидшить оновлення окремих компонентів і дозволить ефективно реагувати на зміни вимог бізнесу.

Основні сервіси

Auth Service (Сервіс автентифікації): Призначення: Забезпечення безпечного входу користувачів у систему, керування сесіями, токенами, авторизацією доступу.

Функції:

- Реєстрація користувачів та адміністраторів

- Вхід із перевіркою облікових даних
- Генерація та перевірка JWT-токенів
- Вихід та завершення сесій
- Ротація токенів (Refresh Tokens)
- Верифікація доступу для інших мікросервісів

User Service (Сервіс профілю користувача): Призначення: Обробка даних користувача, керування обліковими записами, особистими налаштуваннями.

Функції:

- Збереження і редагування профілів
- Підключення послуг, адрес, контактних даних
- Управління налаштуваннями акаунту
- Пов'язування користувача з підписками

Subscription Service (Сервіс послуг): Призначення: Організація взаємодії користувача з тарифами, замовленнями послуг та змінами планів.

Функції:

- Отримання і активація тарифних планів
- Історія підписок та змін
- Призупинення / відновлення підписки
- Перевірка статусу послуги

Billing Service (Сервіс білінгу та платежів): Призначення: Відповідає за облік послуг, тарифікацію, інтеграцію з платіжними шлюзами.

Функції:

- Генерація рахунків та інвойсів
- Облік витрат за підписками
- Інтеграція з платіжними системами (Stripe, LiqPay тощо)
- Виявлення заборгованості або переплат
- Автоматичне списання коштів за підписками

Support Service (Сервіс технічної підтримки): Призначення: Забезпечення зворотного зв'язку між користувачами та службою підтримки.

Функції:

- Прийом і збереження звернень
- Призначення операторів підтримки
- Зміна статусів звернень (нове, у процесі, вирішено)
- Відповіді на звернення в особистому кабінеті

Analytics Service (Аналітичний сервіс). Призначення: Збір, обробка та візуалізація статистичних даних для адміністраторів.

Функції:

- Збір метрик з інших мікросервісів (використання сервісів, доходи, активність)
- Побудова аналітичних звітів і графіків
- Агрегація щоденних/місячних даних
- API для отримання дашбордів в адмін-панелі

Notification Service (Сервіс сповіщень). Призначення: Надсилання користувачам повідомлень електронною поштою, SMS, пушами.

Функції:

- Надсилання підтверджень про замовлення
- Нагадування про оплату або зміни у підписці
- Повідомлення про зміни в акаунті
- Підтримка шаблонів повідомлень

Admin Panel (Інтерфейс адміністратора). Призначення: Інструмент для керування платформою телекомунікаційної компанії.

Функції:

- Доступ до аналітики та статистики
- Керування тарифами, підписками, користувачами
- Перевірка та відповіді на заявки в підтримку

- Звітність та журнал активності

API Gateway (Шлюз API). Призначення: Централізований вхід у систему, маршрутизація запитів до мікросервісів.

Функції:

- Аутентифікація та авторизація запитів
- Балансування навантаження
- Захист від зовнішніх атак
- Логування та трасування запитів

Logging Service. Призначення: Відстеження стану системи, аналіз логів, сповіщення про збої.

Функції:

- Збір логів з усіх мікросервісів (ELK stack, Loki)
- Побудова метрик і графіків (Prometheus + Grafana)
- Alerting про помилки або зниження продуктивності

Базуючись на обраній архітектурі розпишемо основні компоненти системи у табл. 2.1:

Таблиця 2.1

Компоненти системи

Компонент	Опис
User Interface	Веб-інтерфейс користувача, що взаємодіє з API Gateway
Admin Interface	Панель адміністратора для керування тарифами, заявками, перегляду аналітики
Authentication Component	Реалізація авторизації, токенів, сесій
Subscription Component	Вся логіка роботи з тарифами, замовленнями та підписками
Billing Component	Формування рахунків, інтеграція з платіжними сервісами

Analytics Component	Збір, обробка та вивід звітів
Support Component	Робота зі зверненнями клієнтів
Notification Component	Генерація та відправка повідомлень через SMS/Email
API Gateway	Проксі-сервер для маршрутизації запитів
Service Discovery	Виявлення доступних сервісів, балансування
Logging & Monitoring	Збір логів, моніторинг працездатності системи

Основні сценарії взаємодії:

Сценарій 1. Реєстрація нового користувача

Даний сценарій описує процес створення нового облікового запису користувача у системі. Реєстрація є першою точкою входу в систему та базовим компонентом користувацької взаємодії.

Компоненти, що беруть участь:

- User Interface
- API Gateway
- Authentication Component
- User Service
- Notification Component

Хід взаємодії:

1. Користувач заповнює форму реєстрації на клієнтському інтерфейсі (User Interface), вводячи email, пароль та інші необхідні дані.
2. Дані надсилаються до API Gateway, який аутентифікує запит і перенаправляє його до Authentication Component.
3. Authentication Component здійснює валідацію введених даних (перевірка унікальності email, міцності пароля).

4. Після перевірок, створюється запис користувача через User Service, де генерується унікальний ID та зберігаються основні атрибути (ім'я, email, хеш пароля).
5. Notification Component надсилає лист підтвердження реєстрації на вказаний email користувача.
6. Користувач отримує повідомлення про успішну реєстрацію й автоматично входить до системи через сесію.

Сценарій 2. Авторизація та доступ до особистого кабінету

Процес авторизації забезпечує контроль доступу до внутрішніх функцій системи, зокрема персонального кабінету користувача.

Компоненти, що беруть участь:

- User Interface
- API Gateway
- Authentication Component
- User Service

Хід взаємодії:

1. Користувач надає свої облікові дані (email + пароль) у формі авторизації на User Interface.
2. Запит надсилається через API Gateway до Authentication Component, де проводиться перевірка логіну й пароля.
3. У разі успіху формується JWT-токен, що містить ID користувача та строк дії сесії.
4. Усі подальші запити до системи з боку клієнта супроводжуються цим токеном для авторизації.
5. Інформація про користувача (ім'я, підписка, статус акаунту) довантажується через User Service та відображається в інтерфейсі.

Сценарій 3. Оформлення підписки

Даний сценарій є ключовим для бізнес-функціоналу, адже формує дохід компанії.

Компоненти, що беруть участь:

- User Interface
- API Gateway
- Subscription Component
- Billing Component
- Notification Component

Хід взаємодії:

1. Користувач у своєму кабінеті на User Interface обирає бажаний тарифний план і ініціює підписку.
2. Запит надсилається до Subscription Component, який перевіряє можливість підписки (наприклад, чи вже активна інша підписка).
3. Якщо підписка дозволена, запит перенаправляється до Billing Component, який формує рахунок або запит до зовнішнього платіжного шлюзу.
4. Після успішної оплати Subscription Component змінює статус підписки у своїй базі, а Notification Component надсилає електронне підтвердження користувачу.
5. У випадку автоматичного продовження підписки, ці дії можуть виконуватись у фоновому режимі за заданим графіком.

Сценарій 4. Надсилання звернення до служби підтримки

Забезпечення зворотного зв'язку є критичним для якості обслуговування клієнтів.

Компоненти, що беруть участь:

- User Interface
- API Gateway
- Support Component
- Notification Component
- Admin Panel

Хід взаємодії:

1. Користувач у розділі «Допомога» заповнює форму звернення із зазначенням теми та описом проблеми.
2. Запит надсилається через API Gateway до Support Component, де створюється нова заявка та присвоюється статус «нове».
3. Notification Component надсилає листа-підтвердження користувачу, що звернення зареєстроване.
4. Через Admin Panel оператор підтримки бачить список заявок, відкриває нові звернення, змінює статуси та відповідає користувачеві.
5. Усі відповіді та зміни статусів відображаються в інтерфейсі користувача в режимі реального часу.

Сценарій 5. Перегляд аналітики в адмін-панелі

Цей сценарій описує адміністративний доступ до агрегованих даних про стан системи та поведінку користувачів.

Компоненти, що беруть участь:

- Admin Panel
- API Gateway
- Analytics Component
- Subscription Component
- Billing Component

Хід взаємодії:

1. Адміністратор відкриває Admin Panel та переходить у розділ «Аналітика».
2. Інтерфейс ініціює запит через API Gateway до Analytics Component.
3. Analytics Component запитує агреговані дані з Subscription Component (кількість підписок), Billing Component (дохід за період), і власного сховища (історичні графіки).
4. Зібрана інформація обробляється, кешується та повертається у вигляді діаграм і таблиць для візуалізації адміністратору.

Зобразимо діаграми діяльності для основних користувацьких флоу: авторизації та замовлення сервісу. На рис. 2.7 показана блок-схема діяльності авторизації

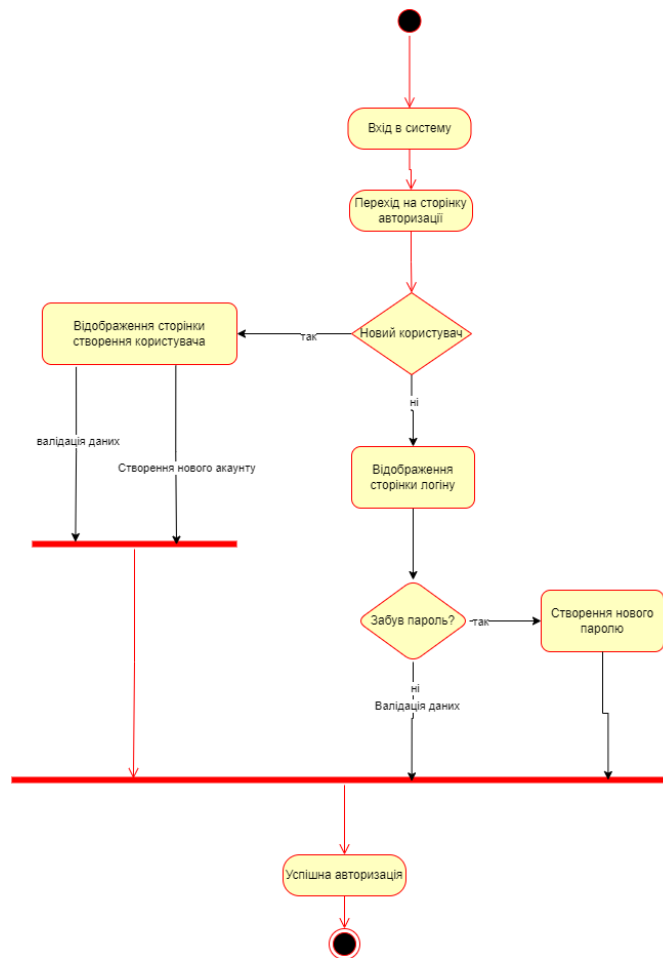


Рис. 2.7. – Діаграма діяльності авторизації

Процес починається зі входу у систему, далі показуємо форму авторизації. В залежності від вибору користувача показуємо форму реєстрації чи форму логіну.

На рис. 2.8 зображено схему діяльності для замовлення сервісу. Після успішної авторизації, користувач обирає сервіс із заданого переліку, переходить на форму оплати, сплачує за сервіс. У разі успішної оплати, адміністратор сервісу створює запит на підключення. Технічна команда обробляє цей запит та проводить необхідні дії. Далі у разі успішного підключення технічна команда

змінює статус запиту на “Виконано”. У разі якщо оплата не пройшла, замовлення відміняється.

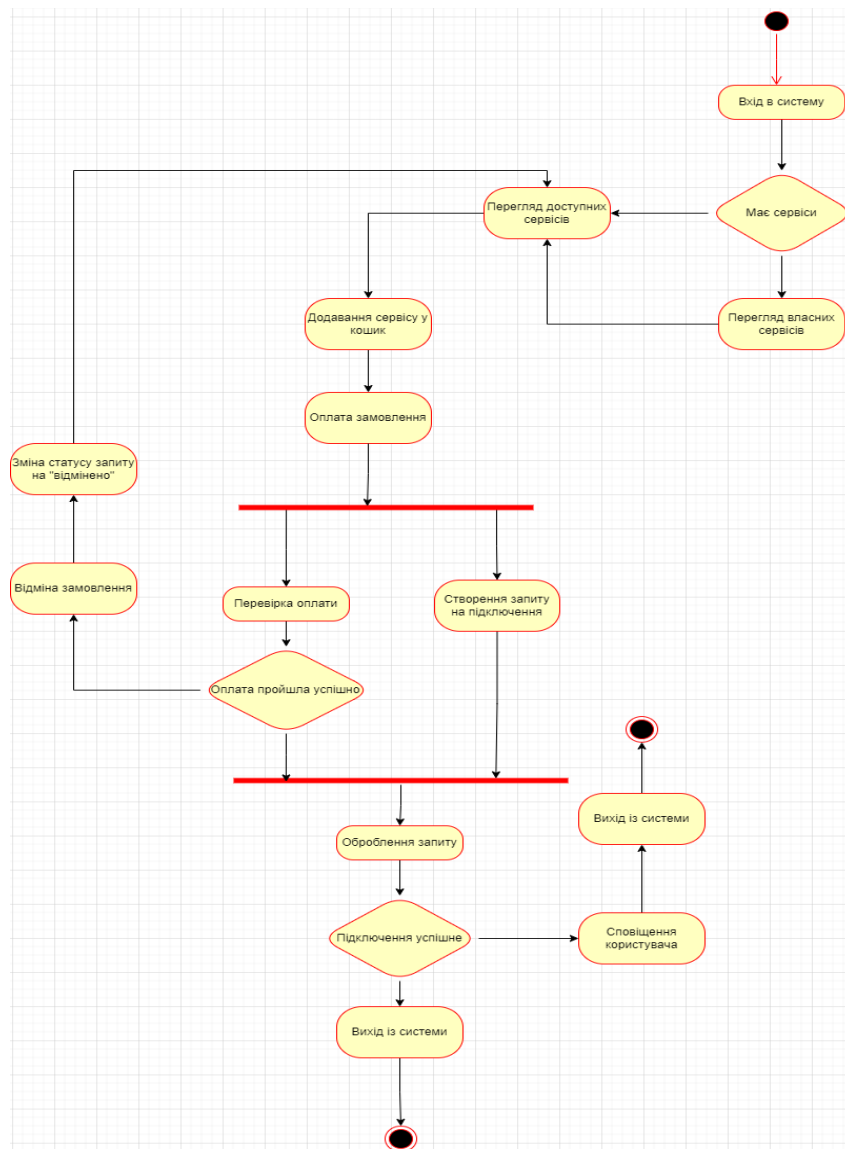


Рис. 2.8. – Діаграма діяльності замовлення сервісу

Оскільки у нашій системі приймають участь декілька активних виконавців, то зобразимо діаграму прецедентів (рис. 2.9.) для кращого розуміння взаємодії користувачів із системою. На діаграмі представимо такі ролі як користувач, адміністратор та технічна команда. Основні дії включають: перегляд доступних сервісів, замовлення сервісу, створення заявки на підключення, оброблення заявки, відміна замовлення, моніторинг статусу сервісів, перегляд покриття, підтвердження замовлення.

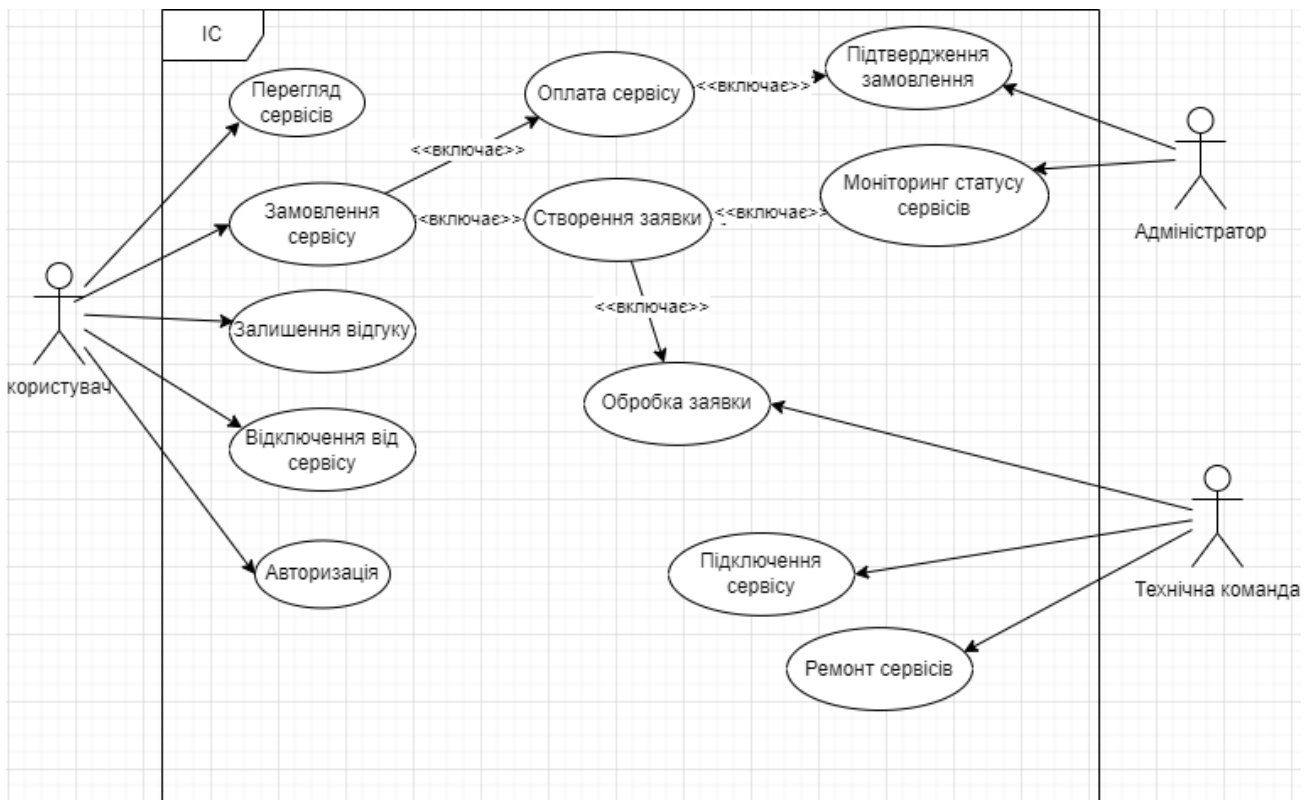


Рис. 2.9. – Діаграма прецедентів

Далі для нашої системи зобразимо діаграму класів, вона є важливим інструментом в об'єктно-орієнтованому моделюванні, що використовується для візуалізації структури системи. Вона показує класи, їх атрибути, методи та взаємозв'язки між класами. Діаграма класів дозволяє розробникам зрозуміти логічну структуру системи, сприяє визначенню функціональних можливостей кожного класу і допомагає виявити можливі залежності та взаємодії між різними компонентами системи.

В нашій діаграмі класів (рис. 2.10.) буде відображено основні сутності системи інтернет-сервісу для телекомунікаційної компанії. Основними класами будуть "Користувач", "Адміністратор", "Технічна команда", "Сервіс", "Замовлення" та "Запит на підключення". Клас "Користувач" буде містити інформацію про користувачів системи, їхні атрибути та методи. Клас "Адміністратор" представлятиме адміністраторів, які мають змогу створювати запити на підключення сервісів. Клас "Технічна команда" включатиме

інформацію про команду, що обробляє запити. Класи "Сервіс" і "Замовлення" відобразатимуть сервіси, які пропонуються користувачам, та замовлення цих сервісів відповідно. Клас "Запит на підключення" буде представляти запити, які створюються адміністраторами та обробляються технічною командою.

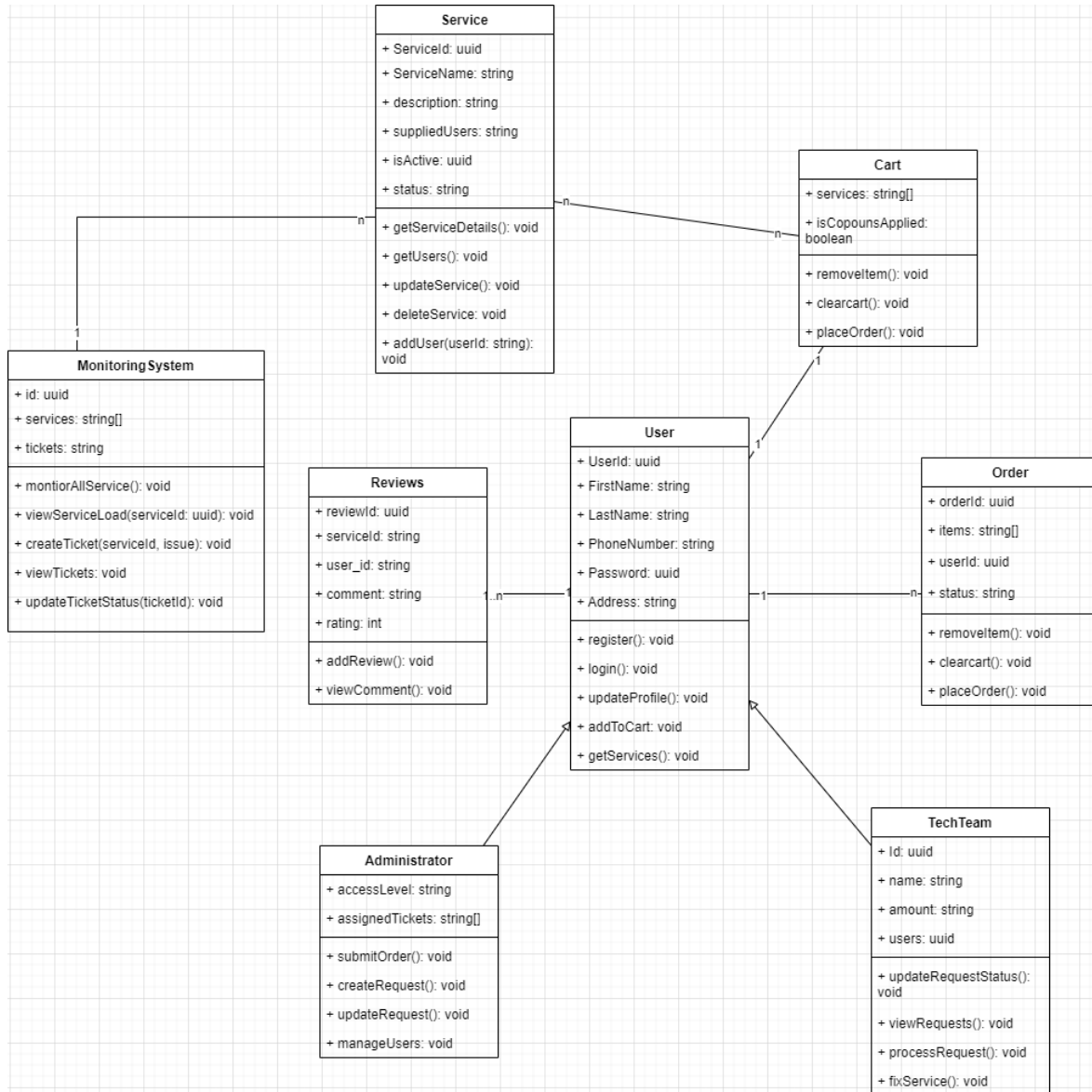


Рис. 2.10. – Діаграма класів

Архітектура системи програми забезпечує комплексний підхід до управління товарами, користувачами та замовленнями. Використання діаграми прецедентів та блок-схеми діяльності дозволяє чітко визначити ролі користувачів і їхні взаємодії з системою. Діаграма бази даних відображає логічну структуру

даних та зв'язки між таблицями, що гарантує цілісність та ефективне управління інформацією. Такий підхід дозволяє створити надійну та функціональну систему для електронної комерції.

2.4 Проектування бази даних

Розробка концептуальної моделі бази даних є важливим етапом у створенні інтернет-сервісу. Модель визначає основні сутності та їх взаємозв'язки, що дозволяє забезпечити ефективне зберігання та обробку даних. Основні сутності бази даних включають в себе наступне;

Користувачі (Users)

- **UserID**: Унікальний ідентифікатор користувача.
- **FirstName**: Ім'я користувача.
- **LastName**: Прізвище користувача.
- **Email**: Електронна пошта користувача.
- **Password**: Пароль користувача.
- **PhoneNumber**: Номер телефону користувача.
- **Address**: Адреса користувача.
- **Role**: Роль користувача (напр. клієнт, адміністратор).
- **DateRegistered**: Дата реєстрації користувача.

Послуги (Services)

- **ServiceID**: Унікальний ідентифікатор послуги.
- **ServiceName**: Назва послуги.
- **Description**: Опис послуги.
- **Price**: Вартість послуги.
- **ServiceType**: Тип послуги (напр. мобільний зв'язок, інтернет, телебачення).

Заявка (Ticket)

- **TicketID**: Унікальний ідентифікатор заявки в підтримку.
- **UserID**: Ідентифікатор користувача, який подав заявку (посилання на сутність Користувачі).
- **ServiceId**: Сервіс
- **Description**: Опис проблеми.
- **Status**: Статус заявки (напр. відкрита, у процесі, закрита).

- **CreatedDate:** Дата створення заявки.
- **ResolvedDate:** Дата вирішення заявки (якщо вирішено).

Замовлення (Orders)

- **OrderID:** Унікальний ідентифікатор замовлення.
- **UserID:** Ідентифікатор користувача, який зробив замовлення (посилання на сутність Користувачі).
- **ServiceID:** Ідентифікатор замовленої послуги (посилання на сутність Послуги).
- **OrderDate:** Дата замовлення.
- **Status:** Статус замовлення (напр. нове, обробляється, завершене).

Огляди (Reviews)

- **ReviewID:** Унікальний ідентифікатор огляду.
- **UserID:** Ідентифікатор користувача, який залишив огляд (посилання на сутність Користувачі).
- **ServiceID:** Ідентифікатор послуги, яку оглядають (посилання на сутність Послуги).
- **Rating:** Оцінка (звичайно від 1 до 5).
- **Comment:** Коментар користувача.
- **ReviewDate:** Дата огляду.

Спроекуємо діаграму Чена для зображення концептуальної моделі баз даних.

Діаграма Чена зображена у Додатку Е.

Мета інфологічного (концептуального) моделювання – забезпечення найбільш природних для людини способів збору та представлення тієї інформації, яку передбачається зберігати в базі даних. Ця модель підтримує функціональність додатку, ефективно організовуючи дані.

Далі розробимо логічну модель бази даних (рис. 2.12.). На етапі побудови логічної моделі було додано додаткові сутності для подолання несумісностей для логічної моделі з реляційною: було видалено двосторонні зв'язки багато до багатьох, також було розкладено складні поля в окремі сутності.

Було додано додаткові сутності для зміни відношень “багато-до-багатьох”.

- TECH_USER

Атрибути: TeamId - ідентифікатор команди, UserId - ідентифікатор користувача, JoinedDate - дату приєднання до команди.

- USER_ORDER

Атрибути: OrderId - ідентифікатор замовлення, UserId - ідентифікатор користувача

Також дані є нормалізованими: один з важливих кроків у розробці бази даних – це нормалізація даних для забезпечення оптимальної структури щоб зменшити дублювання інформації та досягти максимальної ефективності. У цій роботі нормалізація до 3НФ.

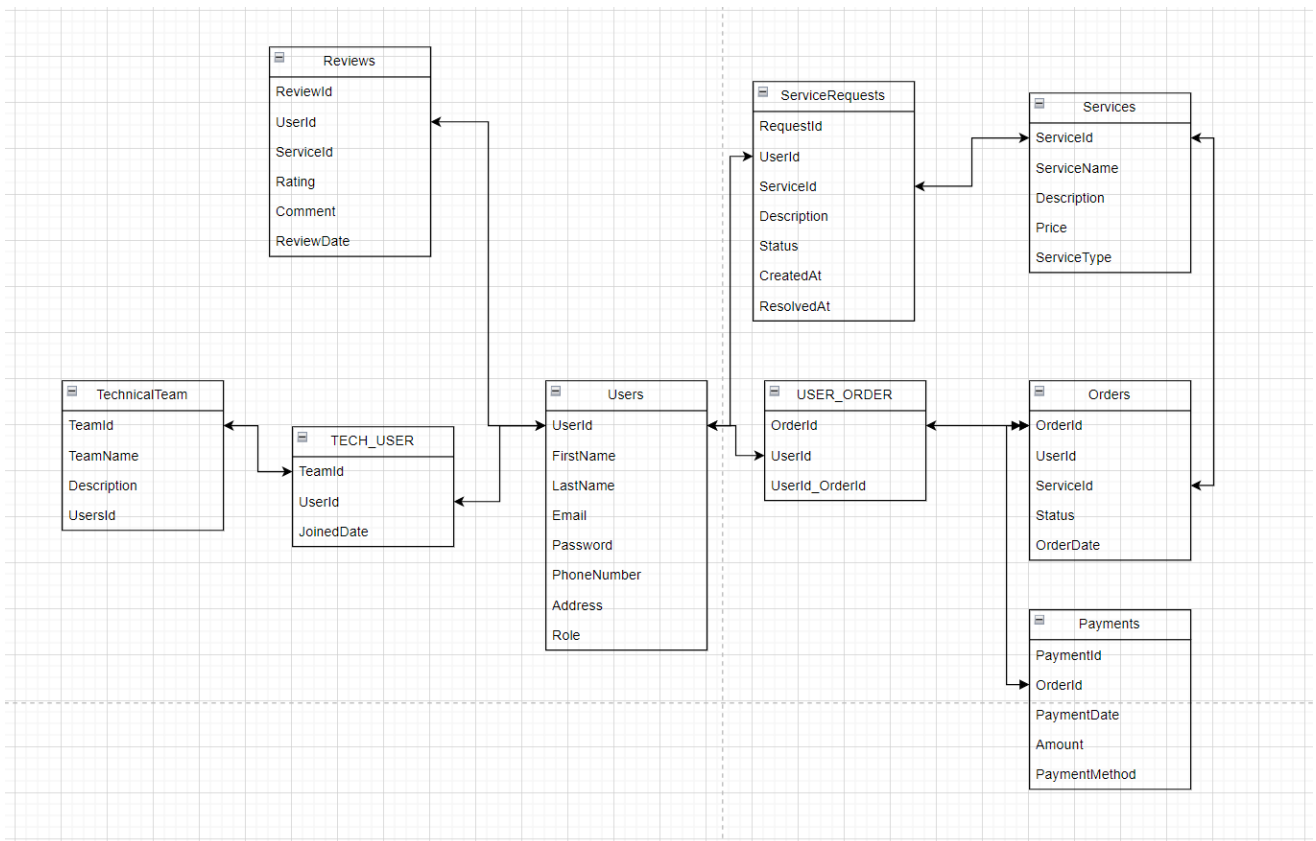


Рис. 2.12. – Логічна модель бази даних

На основі представленої моделі бази даних можна зробити висновок, що структура бази даних проекту забезпечує повну та ефективну обробку інформації, необхідної для функціонування інтернет-сервісу телекомунікаційної компанії.

РОЗДІЛ 3. ПЛАНУВАННЯ ТА УПРАВЛІННЯ ЕЛЕМЕНТАМИ ПРОЄКТУ

Ефективне управління IT-проектом передбачає не лише технічну реалізацію продукту, але й комплексне планування усіх ключових складових життєвого циклу — від ініціації до завершення. У цьому розділі детально розглянуто процес планування проекту створення інтернет-системи для телекомунікаційної компанії, що охоплює структуру робіт, командну організацію, часові межі, бюджет та розподіл відповідальності. Усе планування реалізовано за допомогою Microsoft Project, що забезпечує інтегроване управління строками, ресурсами та контрольними точками.

3.1 Життєвий цикл проекту та визначення фаз

Життєвий цикл проекту (Project Life Cycle) — це сукупність логічно взаємопов'язаних фаз, які визначають послідовність робіт з моменту ініціації проекту до його завершення. Він є основною структурною основою, яка визначає процеси, завдання, контрольні точки та очікувані результати на кожному етапі. Згідно зі стандартами PMBOK® (Project Management Body of Knowledge), життєвий цикл проекту поділяється на п'ять основних процесних груп:

1. ініціація (Initiating),
2. планування (Planning),
3. виконання (Executing),
4. моніторинг і контроль (Monitoring & Controlling),
5. завершення (Closing).

Для проекту розробки інтернет-сервісу в межах телеком-компанії застосовано адаптований життєвий цикл (Табл 3.1), який дозволяє одночасно зберігати структурність, керованість і гнучкість реалізації функціоналу.

Життєвий цикл проєкту

Фаза життєвого циклу	Цілі та задачі	Результати
1	2	3
Ініціація	<ul style="list-style-type: none"> • Офіційно визначити початок проєкту; • Оцінити доцільність реалізації ідеї; • Визначити основних стейкхолдерів та ключові очікування; • Отримати схвалення на подальшу розробку. 	<ul style="list-style-type: none"> • Створення бізнес-кейсу проєкту; • Опис проблемної області та цілей проєкту; • Ідентифікація зацікавлених сторін (stakeholders); • Формування паспорт проєкту / Project Charter; • Призначення керівника проєкту (Project Manager); • Прийняття рішення про старт (Go/No-Go decision).
Планування	<ul style="list-style-type: none"> • Створити комплексний план реалізації проєкту; • Визначити обсяг, бюджет, терміни, ресурси, ризики; • Узгодити очікування замовника із командою. 	<ul style="list-style-type: none"> • Декларація обсягу проєкту (Scope Statement); • WBS (Work Breakdown Structure) — ієрархічна структура робіт; • Календарний план (Gantt-діаграма, milestones); • Оцінка бюджету та ресурсів; • План управління ризиками; • План комунікацій; • Матриця відповідальності (RACI); • План управління якістю.

1	2	3
Виконання	<ul style="list-style-type: none"> • Реалізувати план проєкту згідно з визначеним обсягом та строками; <p>Забезпечити продуктивну роботу команди та комунікацію зі стейкхолдерами.</p>	<ul style="list-style-type: none"> • Реалізація запланованих робіт; • Ведення щоденних стендапів / Scrum-митингів (у разі Agile); • Випуск інкрементів продукту, прототипів, релізів; • Ведення журналу ризиків та проблем; • Збір зворотного зв'язку від стейкхолдерів; <p>Оновлення документації, за потреби.</p>
Моніторинг і контроль	<ul style="list-style-type: none"> • Відстежувати прогрес проєкту та відхилення від плану; • Забезпечити дотримання бюджету, графіку, якості; <p>Приймати рішення щодо змін і коригувань.</p>	<ul style="list-style-type: none"> • Регулярні звіти про стан проєкту (status reports); • Метрика виконання: Earned Value, SPI, CPI тощо; • Контроль обсягу, часу, вартості, ризиків; • Затвердження змін у проєкті (change requests); • Проведення внутрішніх перевірок якості; <p>Виявлення і усунення відхилень.</p>
Завершення	<ul style="list-style-type: none"> • Формалізувати завершення проєкту; • Підвести підсумки, задокументувати результати; <p>Передати кінцевий продукт замовнику.</p>	<ul style="list-style-type: none"> • Акт прийому-передачі проєкту; • Фінальний звіт проєкту (Final Report); • Проведення ретроспективи або «уроку, отриманих у проєкті» (lessons learned); • Закриття контрактів та фінансових зобов'язань; • Аргументація досягнення бізнес-цілей або пояснення невідповідностей; • Архівація документації.

3.2 Побудова ієрархічної структури робіт (WBS)

Ієрархічна структура робіт (WBS – Work Breakdown Structure) — це декомпозиція всіх робіт проєкту на ієрархічну структуру, що відображає повний обсяг робіт, необхідний для досягнення цілей проєкту. Вона дозволяє визначити, оцінити, спланувати та розподілити бюджет на всі необхідні роботи в межах проєкту, а також визначити необхідні ресурси — відповідно до обсягу, графіка та вартості. Повні або часткові провали проєктів часто мають спільну причину — погано розроблену або взагалі відсутню структуру WBS. [30] Структур робіт представлено на Рисунку 3.1.

WBS є базовим інструментом проєктного менеджменту, рекомендованим стандартом PMBOK і використовується для:

- визначення повного обсягу проєкту;
- формування підґрунтя для оцінки вартості, часу та ресурсів;
- побудови календарного плану;
- розробки RACI-матриці та призначення відповідальностей

Переваги WBS структури включає в себе:

1. WBS допомагає планувальникам проєкту визначити обсяг роботи та знайти найкращий спосіб її деталізації.
2. WBS надає загальний огляд проєкту, який зацікавлені сторони можуть переглядати на будь-якому рівні деталізації.
3. Заява про обсяг охоплює лише загальні межі проєкту. WBS дозволяє побачити деталізовані завдання, що формують повний обсяг проєкту.
4. Члени команди цінують чіткі інструкції щодо того, що вони мають доставити. Робочий пакет чітко визначає обсяг завдання, а його зв'язок з іншими елементами WBS підвищує залученість, демонструючи, як їхня робота сприяє успіху проєкту.

5. Якщо для якогось результату немає відповідного підсумкового завдання або робочого пакету у WBS — це сигнал, що ви ще не ідентифікували всю необхідну роботу в проєкті.
6. WBS допомагає точніше оцінити строки реалізації та вартість проєкту.
7. Під час виконання проєкту робочі пакети та підсумкові завдання можуть бути ще не розпочатими, у процесі або завершеними. Розбиття роботи на менші компоненти дозволяє точніше відстежувати прогрес на кожному етапі. [30]

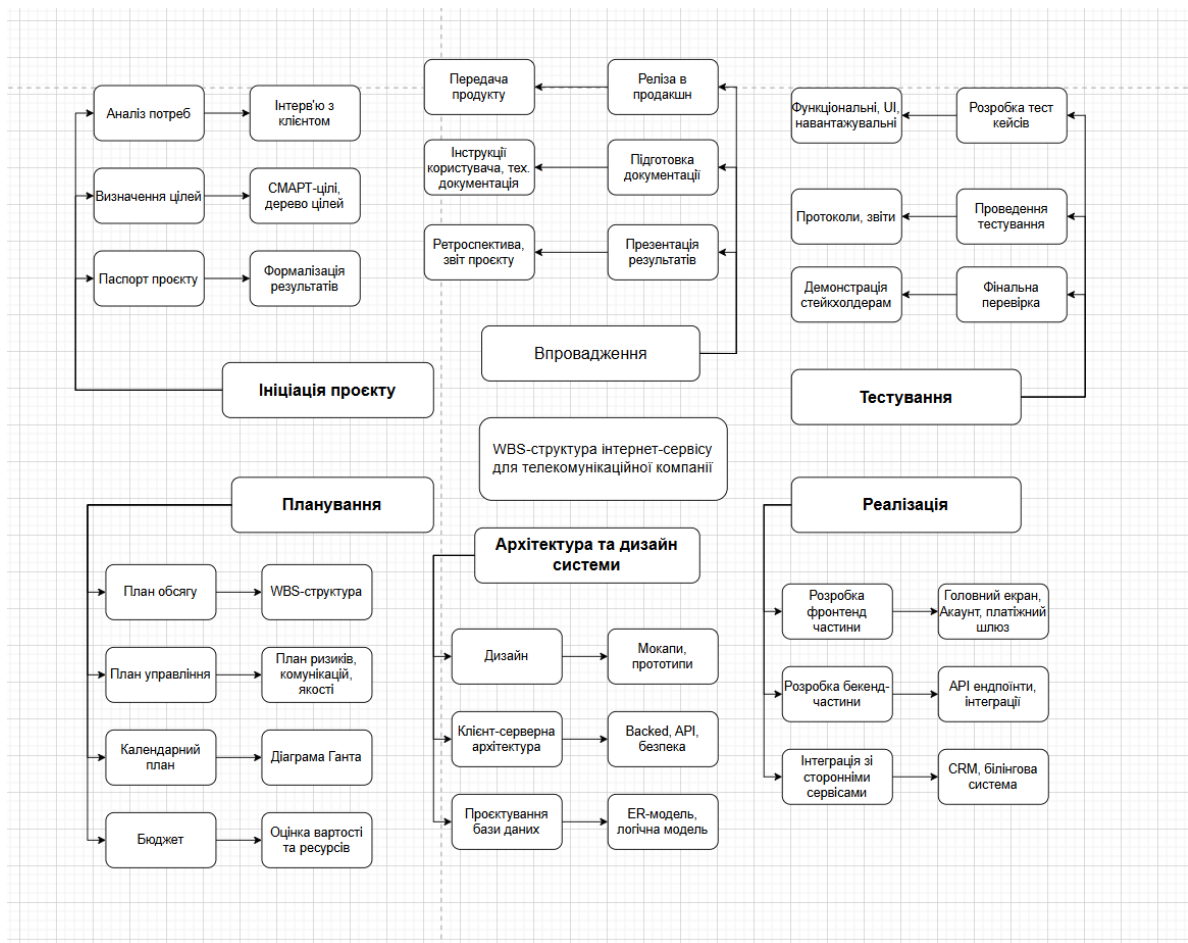


Рис. 3.1. – WBS структура проєкту

3.3 Організаційна структура управління проєктом

Ефективне управління ІТ-проєктами передбачає використання відповідної організаційної структури, яка визначає ієрархію, ролі та зони відповідальності

членів команди. В управлінні проєктами виділяють такі основні типи організаційних структур:

1. Функціональна структура – базується на спеціалізації та чіткому поділі за функціями (наприклад, розробка, тестування, управління). Головною перевагою є висока спеціалізація працівників, але така структура може призводити до недостатньої координації між відділами.



Рис. 3.2. – Функціональна структура управління проєктом

2. Проєктна структура – орієнтована на конкретні проєкти. Команда формується на період виконання конкретного проєкту і підпорядковується безпосередньо керівнику проєкту. Це дозволяє досягати високої гнучкості та чіткості відповідальності, однак супроводжується складністю в управлінні ресурсами після завершення проєкту.



Рис. 3.3. – Проектна структура управління проектом

3. Матрична структура – поєднує переваги функціональної та проектної структур. Працівники підпорядковуються одночасно функціональним керівникам та керівникам проектів, що дозволяє ефективно управляти ресурсами та забезпечує високий рівень координації. Проте матрична структура може створювати подвійне підпорядкування і потенційні конфлікти.

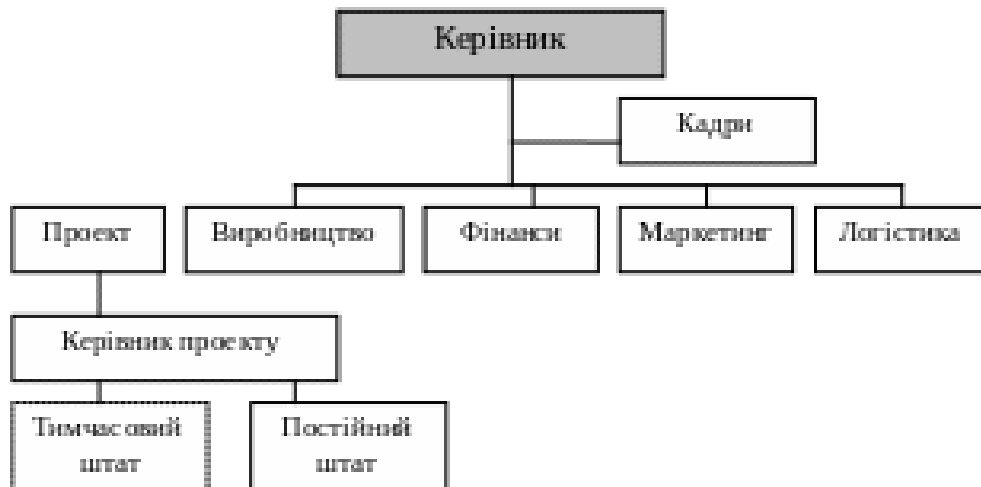


Рис. 3.4. – матрична структура управління проектом

Для нашого проєкту, що реалізується за методологією Water-scrum-fall, найбільш ефективною буде саме матрична організаційна структура, адже вона найкраще поєднує переваги гнучкості Scrum та структурованості Waterfall.

Візуалізація ієрархії структури команди зображено на рисунку 3.5.

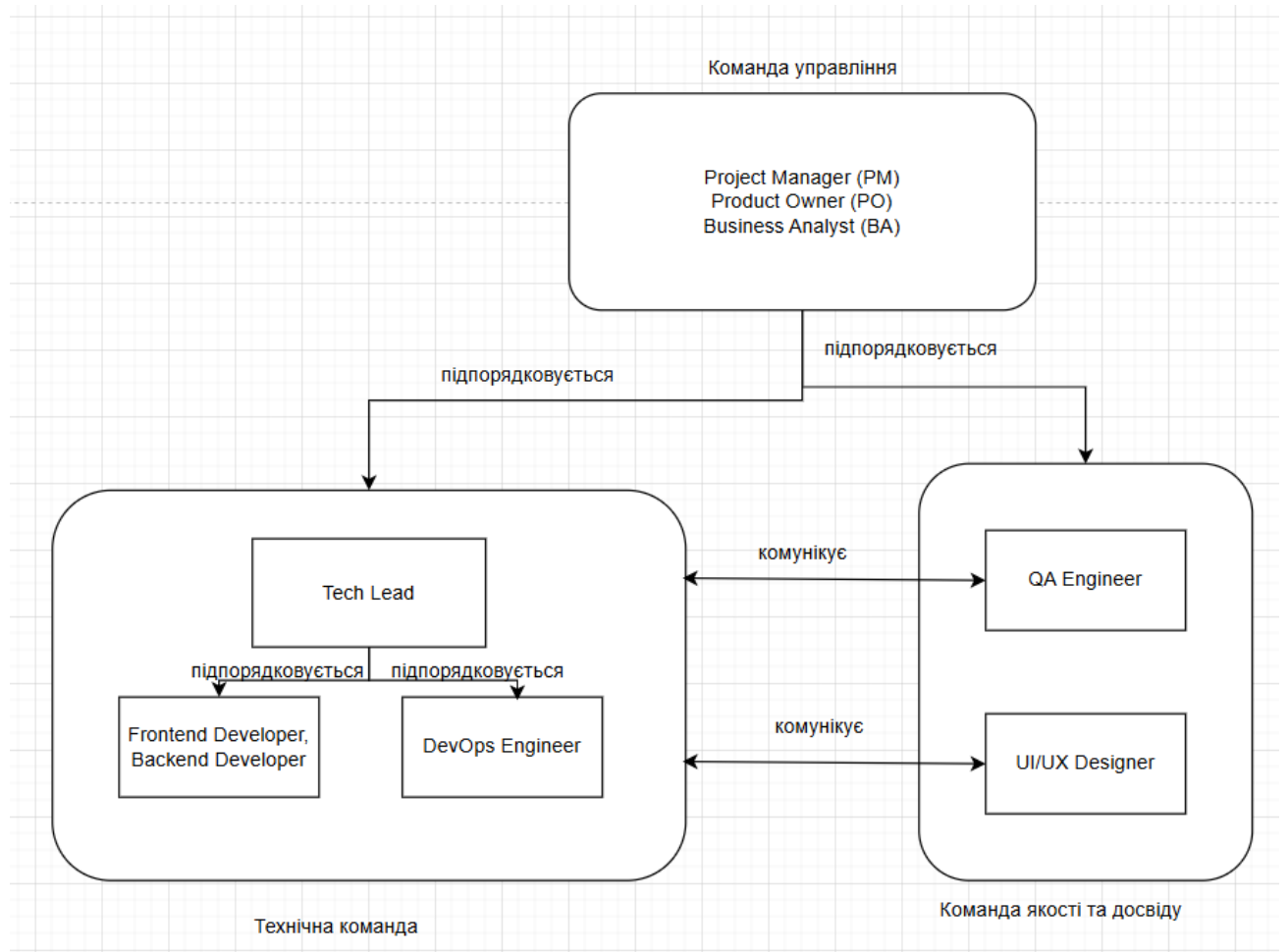


Рис. 3.5. – Ієрархія структури команди

Опис ролей:

Project Manager (PM): Ключова роль, яка відповідає за управління всіма аспектами проєкту, координацію роботи команди та досягнення стратегічних цілей проєкту. Забезпечує контроль за реалізацією проєкту відповідно до плану, строків та бюджету.

Обов'язки:

1. Розробка плану проєкту, календарного графіку та контрольних віх.

2. Координація та контроль роботи учасників команди.
3. Управління ризиками, ведення журналу ризиків, розробка стратегій їх мінімізації.
4. Підготовка регулярних звітів для замовників і стейкхолдерів.
5. Вирішення конфліктів та оптимізація робочих процесів.

Product Owner (PO): Відповідає за бачення продукту з точки зору бізнесу та кінцевих користувачів. Формує вимоги, визначає пріоритетність завдань, забезпечує відповідність розробки стратегії та цілям компанії. Завдяки цій ролі проєкт відповідає реальним потребам користувачів та генерує бізнес-цінність.

Обов'язки:

1. Формування та уточнення продуктових вимог.
2. Пріоритезація задач (Product Backlog) для досягнення максимального бізнес-ефекту.
3. Проведення демо-зустрічей, збір зворотного зв'язку від користувачів.
4. Комунікація з командою розробки для передачі бачення продукту.
5. Забезпечення відповідності продукту бізнес-цілям проєкту.

Business Analyst (BA): Аналітична роль, яка забезпечує глибоке розуміння ринку, конкурентів, цільової аудиторії. Допомагає трансформувати бізнес-потреби в зрозумілі та чітко задокументовані вимоги до функціоналу, що зменшує ризик непорозумінь та забезпечує якісну реалізацію продукту.

Обов'язки:

1. Аналіз ринку, конкурентів та потреб кінцевих користувачів.
2. Формування документації (специфікацій) щодо вимог до системи.
3. Створення user-stories та сценаріїв використання (use-cases).
4. Підготовка аналітичних звітів та рекомендацій щодо вдосконалення системи.
5. Комунікація з усіма зацікавленими сторонами для уточнення деталей реалізації.

Tech Lead (TL): Технічний лідер команди, відповідає за архітектурні рішення, вибір технологій, а також технічну якість розробки. Забезпечує стабільність, масштабованість і продуктивність кінцевого рішення, допомагає зменшити технічні ризики проєкту.

Обов'язки:

1. Вибір та узгодження архітектури системи.
2. Прийняття ключових технічних рішень, аналіз можливих ризиків.
3. Керівництво командою розробників (frontend, backend, DevOps).
4. Контроль дотримання стандартів якості коду, проведення рев'ю коду.
5. Комунікація з РМ та РО щодо технічних аспектів реалізації.

Frontend Developer: Відповідає за створення та підтримку зручного, ефективного та естетичного інтерфейсу користувача. Це підвищує користувацьку задоволеність, зручність використання та залученість користувачів до роботи з системою.

Обов'язки:

1. Розробка інтерфейсу веб-додатку згідно з дизайн-макетами.
2. Адаптація інтерфейсу під різні типи пристроїв та браузерів.
3. Реалізація інтерактивних елементів та динамічних компонентів.
4. Взаємодія з backend для отримання та передачі даних (REST API).
5. Оптимізація продуктивності інтерфейсу та часу завантаження сторінок.

Backend Developer: Забезпечує надійну реалізацію бізнес-логіки, інтеграцію із зовнішніми та внутрішніми системами, що дозволяє сервісу працювати швидко, стабільно та ефективно.

Обов'язки:

1. Реалізація бізнес-логіки додатку та алгоритмів.
2. Розробка та документування API (REST/GraphQL) для взаємодії з фронтендом та інтеграціями.
3. Інтеграція з внутрішніми системами компанії (білінг, CRM тощо).

4. Забезпечення безпеки додатку, захисту даних та відповідності GDPR.
5. Оптимізація продуктивності сервера та бази даних.

UI/UX Designer: Відповідає за розробку прототипів інтерфейсу та користувацького досвіду. Це забезпечує інтуїтивність, зручність та задоволеність користувачів при роботі з продуктом.

Обов'язки:

1. Створення мокапів та прототипів користувацького інтерфейсу.
2. Дизайн інтерфейсів відповідно до сучасних стандартів UI/UX.
3. Проведення юзабіліті-тестувань, збір та аналіз зворотного зв'язку.
4. Підготовка дизайн-системи (гайдлайнів, бібліотек компонентів).
5. Комунікація з командою розробників для реалізації дизайну.

QA Engineer: Забезпечує високу якість реалізації продукту, виявляючи помилки та невідповідності на ранніх стадіях розробки, що мінімізує ризики, пов'язані із запуском недопрацьованого продукту.

Обов'язки:

1. Розробка та документування тест-кейсів і тест-планів.
2. Проведення ручного та автоматизованого тестування продукту.
3. Виявлення, реєстрація та контроль виправлення дефектів.
4. Забезпечення відповідності продукту функціональним і нефункціональним вимогам.
5. Взаємодія з командою розробників щодо покращення якості продукту.

DevOps Engineer: Забезпечує безперервність процесу розробки, інтеграції та доставки продукту. Це дозволяє мінімізувати ризики при розгортанні та забезпечує стабільність роботи системи у продакшн.

Обов'язки:

1. Налаштування та підтримка середовищ розробки, тестування і розгортання.
2. Реалізація процесів Continuous Integration та Continuous Delivery (CI/CD).

3. Моніторинг продуктивності та стабільності роботи додатку.
4. Управління інфраструктурою (Docker, Kubernetes, AWS).
5. Забезпечення автоматизації та оптимізації процесів розгортання продукту.

Підструктури команди та їх цілі:

Управлінська підструктура. Мета: ефективне планування, стратегічне управління та координація виконання проєкту для забезпечення досягнення бізнес-цілей і термінів реалізації.

Технічна підструктура. Мета: забезпечення розробки технічно якісного, стабільного і продуктивного програмного продукту, здатного до подальшого масштабування та підтримки.

Дизайнерсько-тестувальна підструктура. Мета: створення максимально зручного, інтуїтивно зрозумілого інтерфейсу і забезпечення високого рівня якості кінцевого продукту через ретельне тестування і контроль.

Для візуалізації розподілу відповідальностей використовується матриця відповідальності RACI — один із найпоширеніших інструментів у проєктному менеджменті, рекомендований стандартом PMBOK. Вона дозволяє встановити, хто:

- R (Responsible) — виконує роботу;
- A (Accountable) — несе відповідальність за результат;
- C (Consulted) — консультує або надає експертизу;
- I (Informed) — інформується про результати або прогрес.

Матриця відповідальності зображено на рисунку 3.2.

Таблиця 3.2.

Таблиця відповідальності (RACI)

Етапи / Завдання проекту	PM	PO	BA	TL	FE	BE	UX	QA	DevOps
1. Формування цілей проекту	A	R	C	I	I	I	I	I	I
2. Збір та аналіз вимог	C	A	R	I	I	I	I	I	I
3. Побудова WBS, планування	R	C	C	C	I	I	I	I	I
4. Визначення архітектури	I	I	C	A/R	I	R	I	I	C
5. Розробка мокапів та UI	I	C	C	C	I	I	A/R	C	I
6. Розробка фронтенду	I	I	I	C	A/R	I	C	I	I
7. Розробка бекенду	I	I	I	C	I	A/R	I	I	C
8. Інтеграція з білінговою системою	I	I	C	R	I	A/R	I	I	C
9. Написання тест-кейсів	I	I	I	I	I	I	I	A/R	I
10. Проведення тестування	I	I	I	C	I	I	I	A/R	I

11. CI/CD та деплой сервісу	I	I	I	C	I	I	I	C	A/R
12. Презентація продукту стейкхолдерам	R	A	C	I	I	I	I	I	I
13. Завершення проекту, ретроспектива	A	R	C	C	I	I	I	C	C

3.4 Планування часу та ресурсів

Планування часу та ресурсів — один із найважливіших процесів управління проектом, що безпосередньо впливає на його успішність. Згідно зі стандартом PMBOK® Guide, процеси управління часом охоплюють: визначення активностей, встановлення їх послідовності, оцінку тривалості, створення розкладу та контроль виконання. Планування ресурсів, у свою чергу, передбачає визначення необхідних людських, матеріальних та технічних ресурсів для виконання кожного завдання та їх раціональний розподіл у часі.

Недостатньо чітке планування може призвести до затримок, перевищення бюджету, перевантаження команди та ризику зриву проекту. Визначимо часові віхи проекту (Табл. 3.3), що необхідно для встановлення контрольних точок у межах життєвого циклу, за якими можна оцінювати прогрес виконання робіт.

Часові віхи проєкту

№	Назва віхи	Опис	Орієнтовна дата
1	Формалізація ідеї проєкту	Первинне обґрунтування потреби, визначення бізнес-проблеми	22.01.2024
2	Затвердження ініціації проєкту	Презентація ідеї, опис проблеми, призначення Project Manager, погодження з керівництвом	01.02.2024
3	Завершення аналізу потреб та визначення цілей	SMART-цілі, дерево цілей, початкові очікування від функціоналу	05.02.2024
4	Розробка та затвердження паспорта проєкту	Формальний документ з описом цілей, обсягів, стейкхолдерів, обмежень	07.02.2024
5	Побудова WBS	Ієрархічна структура робіт з розбивкою на підсистеми та робочі пакети	08.02.2024
6	SWOT-аналіз, аналіз ризиків	Оцінка ризиків проєкту, визначення загроз, формування початкових стратегій їх мінімізації	09.02.2024
7	Завершення планування проєкту	Календарний план, план комунікацій, бюджет, ресурси, діаграма Ганта	12.02.2024
8	Формування та призначення проєктної команди	Назначення ролей: UI/UX, BE, FE, QA, DevOps, BA, PO	13.02.2024
9	Створення макетів інтерфейсу (UI/UX)	Прототипування користувацьких сценаріїв, навігаційної структури, затвердження мокапів	16.02.2024
10	Затвердження архітектури системи	Вибір технологій, створення діаграми архітектури, API-структури, безпекових вимог	20.02.2024
11	Розробка схеми бази даних	Концептуальна та логічна моделі, Проектування таблиць, зв'язків, індексів	21.02.2024

Продовження табл. 3.4.

12	Початок спринтів: розробка MVP	Визначено backlog, перший інкремент: авторизація, дашборд, профіль користувача	22.02.2024
13	Завершення 1-го спринту	Демонстрація MVP 1, отримання зворотного зв'язку, тестування ядра системи	07.03.2024
14	Завершення 2-го спринту	Додано: управління тарифами, історія оплат, звернення до підтримки	21.03.2024
15	Завершення 3-го спринту	Додано: інтеграція з CRM, e-mail, смс шлюзами	04.04.2024
16	Завершення 4-го спринту	Адмін-панель, аналітичні панелі, логування подій	18.04.2024
17	Завершення інтеграції з зовнішніми сервісами	Інтеграція з білінговою системою, обробка API-запитів	20.04.2024
18	Початок комплексного тестування	Проведення функціонального, навантажувального та регресійного тестування	22.04.2024
19	Успішне завершення тестування	Усі критичні баги усунено, продукт готовий до впровадження	25.04.2024
20	Підготовка технічної документації	Інструкції користувача, документація API, схеми БД, керівництво з безпеки	26.04.2024
21	Реліз системи у продакшн	Завершальне впровадження, підключення до основних доменів та середовищ	29.04.2024
22	Презентація результатів	Демонстрація функціоналу замовнику, збір фінального фідбеку	30.04.2024
23	Ретроспектива та оцінка ефективності	Аналіз досягнутих цілей, часових/бюджетних відхилень, формування звіту Lessons Learned	02.05.2024
24	Закриття проєкту	Офіційне завершення, архівація даних, вивід команди з проєкту	03.05.2024

Внесемо визначені часові віхи у програмне забезпечення Microsoft Project для подальшого планування та візуалізації календарного графіка реалізації проєкту (рис. 3.6). Це дозволить автоматизувати побудову діаграми Ганта, врахувати залежності між задачами та контролювати дотримання термінів виконання.

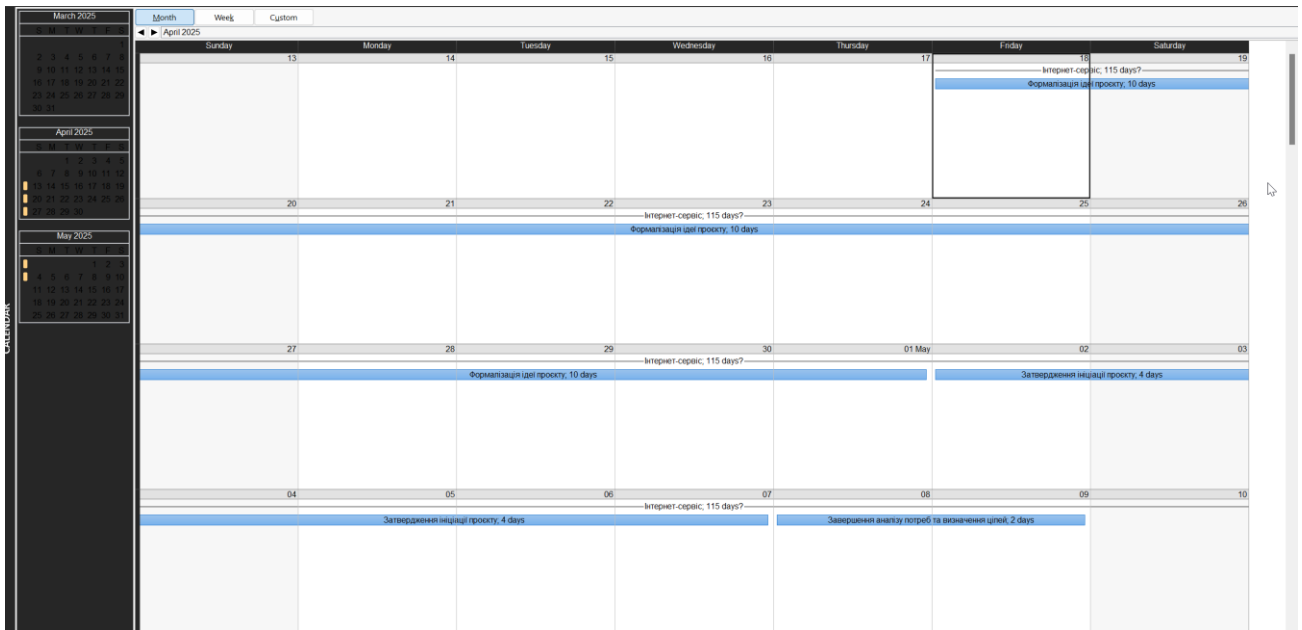


Рис. 3.6. – календар проєкту у програмі MS Project

На основі попередньо сформованої ієрархічної структури робіт (WBS) та визначених часових віх, кожному завданню було призначено тривалість, ресурси та залежності, що забезпечить можливість точного розрахунку термінів завершення проєкту та ефективного контролю за його реалізацією (Рис. 3.7).

Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
	Інтернет-сервіс	115 days?	Fri 18.04.25	Thu 25.09.25			
	Планування	22 days?	Fri 18.04.25	Mon 19.05.25		Project Manager;P	
	Формалізація ідеї	10 days	Fri 18.04.25	Thu 01.05.25			
	Затвердження ініціа	4 days	Fri 02.05.25	Wed 07.05.25	2		
	Завершення аналізу	2 days	Thu 08.05.25	Fri 09.05.25	3		
	розробка та затверд	1 day?	Mon 12.05.25	Mon 12.05.25	4		
	Побудова WBS	1 day?	Tue 13.05.25	Tue 13.05.25	5		
	SWOT-аналіз, аналіз	3 days	Wed 14.05.25	Fri 16.05.25	6		
	Завершення планув:	1 day?	Mon 19.05.25	Mon 19.05.25	7		
	Створення архітектур	8 days?	Tue 20.05.25	Thu 29.05.25	1	Developer;Technic	
	Формування та приз	3 days	Tue 20.05.25	Thu 22.05.25			
	Створення макетів і	4 days	Fri 23.05.25	Wed 28.05.25	10		
	Затвердження архіт:	1 day?	Thu 29.05.25	Thu 29.05.25	11		
	Впровадження	82 days?	Fri 30.05.25	Mon 22.09.25	1;9	Developer	
	Розробка схеми баз	1 day?	Fri 30.05.25	Fri 30.05.25			
	Спринт	80 days	Tue 03.06.25	Mon 22.09.25			
	Тестування	26 days	Tue 12.08.25	Tue 16.09.25	20	Testers	
	Завершення проєкту	7 days?	Wed 17.09.25	Thu 25.09.25	24	Project Manager	
	Підготовка технічно	3 days	Wed 17.09.25	Fri 19.09.25			
	Реліз системи у про	1 day?	Mon 22.09.25	Mon 22.09.25	47		
	Презентація результ	1 day?	Tue 23.09.25	Tue 23.09.25	48		
	Ретроспектива та о	1 day?	Wed 24.09.25	Wed 24.09.25	49		
	цінка ефективності						
	Закриття проєкту	1 day?	Thu 25.09.25	Thu 25.09.25	50		

Рис. 3.7. – часові віхи у програмі MS Project

У рамках планування було побудовано діаграму Ганта (рис. 3.8) допомогою програмного забезпечення MS Project, що дозволяє відобразити часові межі завдань, послідовність їх виконання, залежності між роботами, а також ключові контрольні точки. Побудова діаграми дала змогу чітко структурувати етапи реалізації проєкту, визначити критичний шлях і виявити потенційні часові конфлікти.

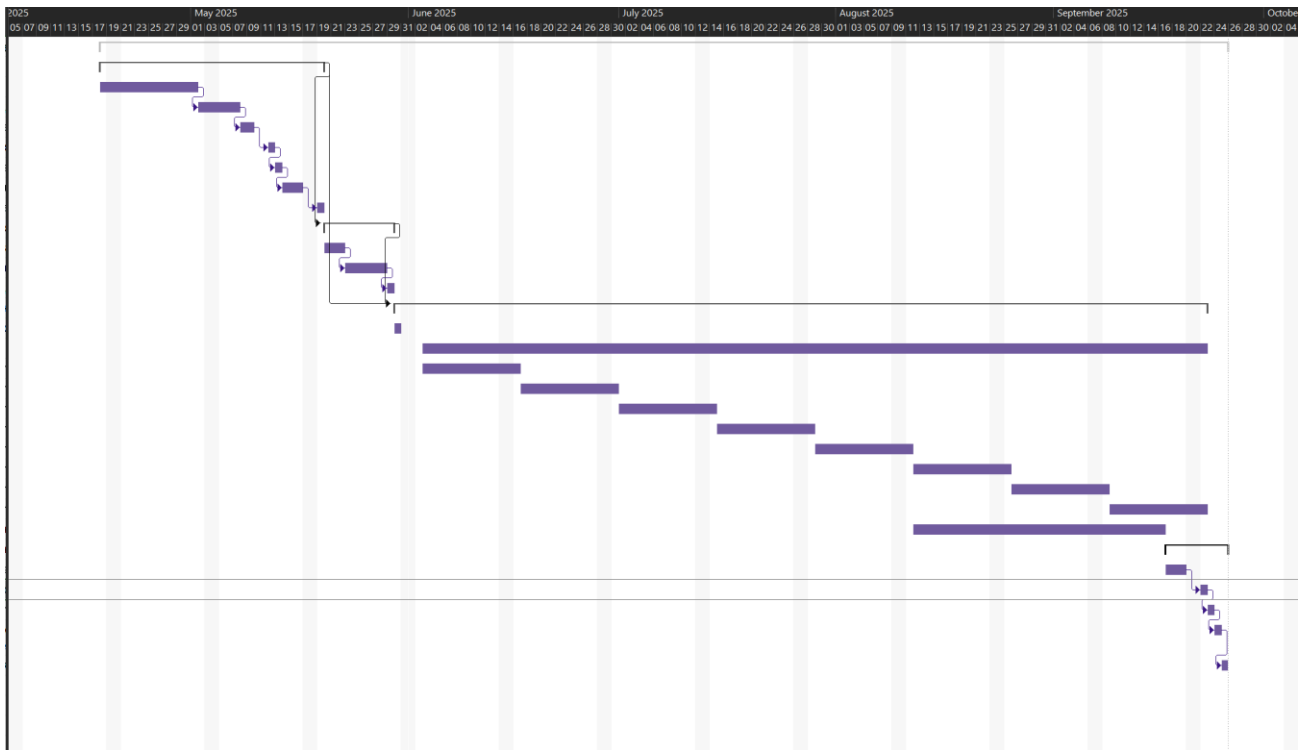


Рис. 3.8. – діаграма Ганта

У процесі планування ресурсів в управлінні проектами важливо оцінити вартість людських ресурсів. Це дозволяє оцінити бюджет проекту. Нижче подано розрахунок витрат на персонал на основі матриці відповідальності (RACI) (Табл .3.4).

Таблиця 3.4

Оцінка вартості людських ресурсів

Роль	К-сть місяців участі	Місячна ставка, грн	Вартість, грн
Project Manager	4	90 000	360 000
Product Owner	4	80 000	320 000
Business Analyst	3	65 000	195 000
Tech Lead	4	100 000	400 000
Frontend Developer	4	70 000	280 000

Продовження табл. 3.4.

Backend Developer	4	80 000	320 000
QA Engineer	3	60 000	180 000
UI/UX Designer	2	60 000	120 000
DevOps Engineer	2	75 000	150 000
Разом	—	—	2 325 000 грн

В даному випадку у нас стався конфлікт ресурсів бюджету. Фактична потреба у зарплатному фонді перевищує загальні початкові інвестиції на 825 000 грн. Вирішемо конфлікт ресурсів (Табл. 3.5).

Таблиця 3.5

Вирішення конфлікту людських ресурсів

Захід оптимізації	Економія	Пояснення
1. Зменшення тривалості участі UI/UX Designer	– 60 000	Залучити лише на етап прототипування
2. Поєднання ролей BA + PO	– 300 000	Один фахівець виконує дві функції
3. Заміна Tech Lead на Senior Dev	– 100 000	Структурування через досвідченого розробника
4. Залучення QA на фіксовану часткову ставку	– 70 000	Зменшено з 3 до 2 місяців
5. Використання open-source CI/CD рішень	– 90 000	Зниження потреби в DevOps

6.Зменшення часу розробки оптимізації процесів шляхом	- 200 000	Оптимізація процесів зменшує час розробки та дозволяє залучати менше розробників
Разом зекономлено	– 1 200 000 грн	
Оновлений бюджет на персонал	1 400 000 грн	Дорівнює загальним початковим інвестиціям

Візуалізуємо бюджет проекту за допомогою інструменту MS Project (рис. 3.9).

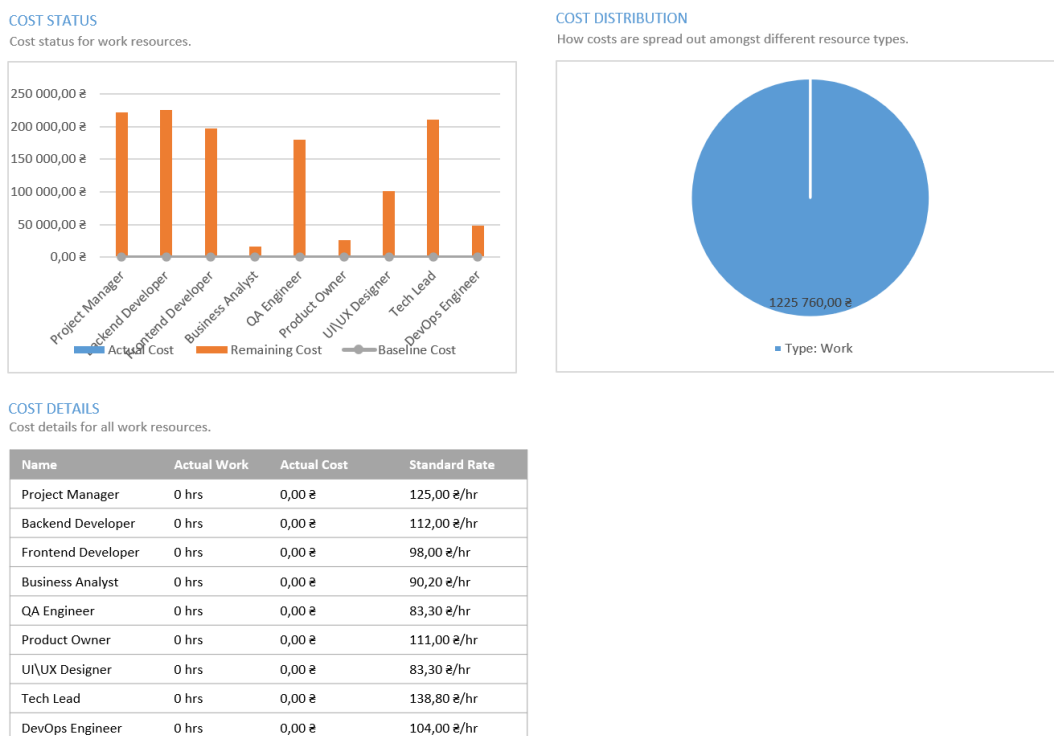


Рис. 3.9. – бюджет проекту за MS Project

3.5 Контроль та моніторинг виконання проєкту

Контроль та моніторинг виконання є критичною фазою управління проєктом, що забезпечує досягнення запланованих цілей у межах бюджету, строків і заданої якості. Згідно з PMBOK [4] Guide, ефективне відстеження прогресу потребує систематичного збору, аналізу та оцінки фактичних результатів щодо плану, а також прийняття своєчасних коригуючих дій. У цьому пункті було описано інструменти моніторингу, які використовуються в проєкті, а також механізми контролю виконання завдань, витрат і ризиків.

Контроль і моніторинг проєкту забезпечують безперервне спостереження за виконанням ключових завдань, своєчасне виявлення відхилень та прийняття управлінських рішень щодо їх усунення. Для цього використовуються сучасні цифрові інструменти управління, що дозволяють централізовано відстежувати статус усіх компонентів проєкту. Проєкт розробки інтернет-сервісу для телекомунікаційної компанії реалізується в системі управління проєктами **Jira** (рис. 3.10.), що забезпечує:

- зручну структурування завдань (Epic → Story → Task),
- контроль за спринтами та відслідковування швидкості команди (Velocity),
- можливість вести журнал ризиків та змін.

Додатково для візуалізації прогресу використовується діаграма Ганта у Microsoft Project, а також Google Sheets для фінансового моніторингу витрат і виконання бюджету.

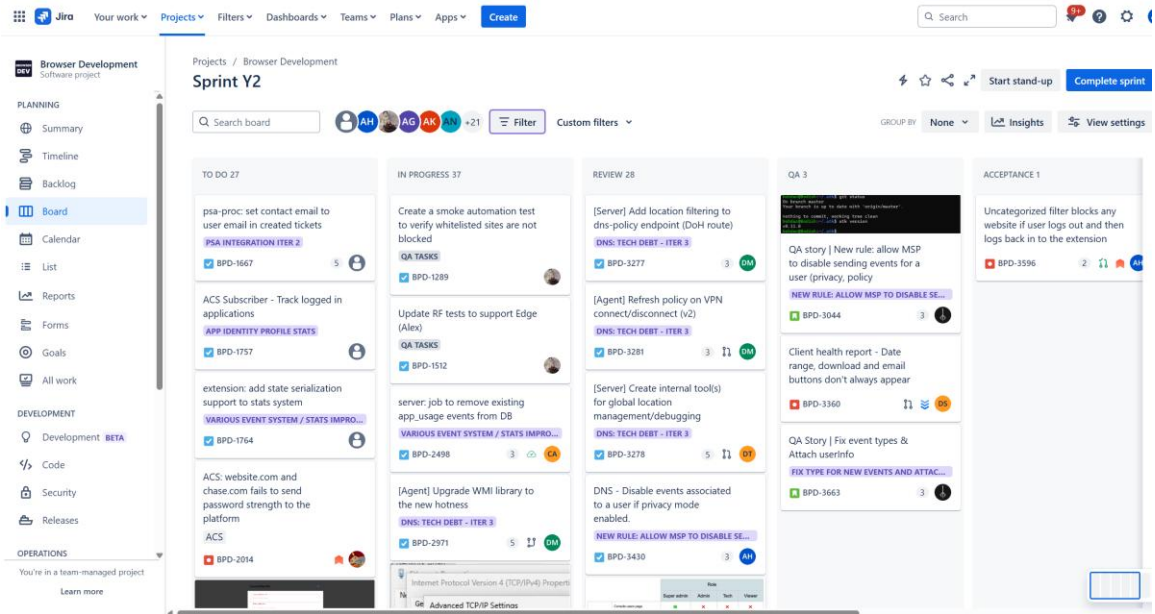


Рис. 3.10. – Управління Проектом в JIRA

Для забезпечення ефективного моніторингу проекту було підібрано набір інструментів, що відповідають ключовим компонентам контролю. Застосування цих засобів дозволяє оперативно відстежувати виконання завдань, строки, бюджет, ризики та забезпечує прозору комунікацію із зацікавленими сторонами (табл. 3.6).

Таблиця 3.6

Інструменти контролю проекту

Компонент контролю	Інструмент	Функція
Контроль завдань	Jira Software	Ведення backlog, трекінг задач, статуси (To Do → In Progress → Done)
Контроль строків	Microsoft Project, Gantt Chart	Візуалізація етапів, контроль за строками, аналіз критичного шляху
Контроль якості	TestRail, Google Sheets	Збір результатів тестування, перевірка відповідності вимогам

Контроль бюджету	Google Sheets / Excel	Звітність витрат за категоріями, контроль перевищення бюджету
Контроль ризиків	Jira (risk log), Excel	Ведення журналу ризиків, моніторинг запобіжних дій
Зміни та рішення	Confluence / Google Docs	Документація змін, обґрунтування рішень, затвердження
Звітність для замовника	PowerPoint, PDF	Регулярні звіти про статус (Status Report) зі статусом задач та проблем

Ключові підходи до моніторингу

1. Щоденні стендапи — короткі наради команди для обговорення прогресу та проблем.
2. Щотижневі статус-мітинги з керівником проєкту та стейкхолдерами.
3. Оцінка освоєного обсягу (Earned Value Analysis):
 - SPI (Schedule Performance Index)
 - CPI (Cost Performance Index)
4. Моніторинг ризиків через індикатори впливу/ймовірності (Risk Heatmap).
5. Sprint Review + Retrospective — оцінка виконаного функціоналу, зворотний зв'язок.
6. Ведення журналу змін (Change Log) — для фіксації змін обсягу, строків чи бюджету.

РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

У рамках даного розділу було здійснено реалізацію ключових складових програмного продукту, що включають програмну логіку, користувацький інтерфейс та бізнес-функціонал ІС. На початковому етапі було описано структуру програмного забезпечення інтернет сервісу, далі було побудовано мокапи інтерфейсу, які дозволили сформувавши загальну концепцію навігації, структуру екранів та сценарії взаємодії користувача з системою. Реалізовані компоненти інтегруються у єдину інтернет-систему, що відповідає вимогам проєкту та забезпечує користувацьку й технічну ефективність.

4.1 Опис структури програмного забезпечення

Для кожного з сервісів оберемо відповідну мову програмування, фреймворк та інструменти з урахуванням функціональних вимог, швидкодії, можливості масштабування та підтримки у спільноті розробників. Розглянемо програмну реалізацію кожного мікросервісу системи, а також загальні принципи їхньої взаємодії (рис. 4.1).

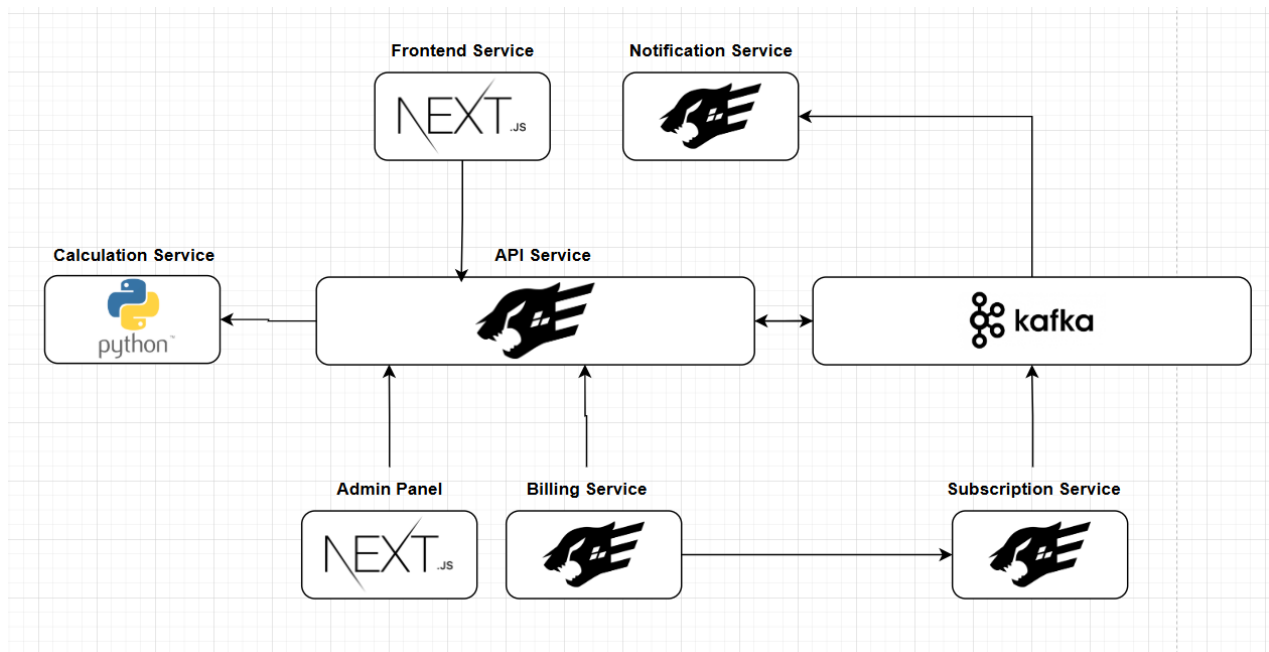


Рис. 4.1. – Опис архітектури інтернет сервісу

1. Auth Service

- Мова програмування: Node.js (TypeScript)
- Фреймворк: Express.js + JWT
- Причина вибору: Node.js дозволяє обробляти велику кількість одночасних запитів, що важливо для авторизаційного сервісу. TypeScript забезпечує типобезпеку, що знижує ризики помилок.
- Інтеграція: Генерує токени доступу, які використовуються іншими сервісами через заголовки авторизації (Bearer).

2. User Service

- Мова: Python
- Фреймворк: FastAPI
- Причина вибору: FastAPI забезпечує високу продуктивність та просту декларативну роботу з API, має сучасну систему валідації та автоматичну генерацію Swagger документації.
- Інтеграція: Обслуговує запити через API Gateway; передає дані до Analytics Service.

3. Subscription Service

- Мова: Java
- Фреймворк: Spring Boot
- Причина вибору: Spring Boot є стабільною платформою для реалізації складної бізнес-логіки, надає зручні механізми інтеграції з базами даних та мікросервісами (через Spring Cloud).
- Інтеграція: Отримує дані з Billing Service та передає оновлення до Notification Service.

4. Billing Service

- Мова: Javascript
- Бібліотека: Nestjs

- Причина вибору: Nestjs забезпечує надзвичайну продуктивність, низьке споживання пам'яті та швидку обробку обчислень, що є критичним для сервісу тарифікації.
- Інтеграція: Працює із Subscription Service, передає звіти до Analytics Service.

5. Support Service

- Мова: Python
- Фреймворк: Django REST Framework
- Причина вибору: Django забезпечує повнофункціональну backend-систему з ORM, автентифікацією та панеллю адміністратора, що пришвидшує розробку.
- Інтеграція: Взаємодіє із Notification Service та Admin Panel через API.

6. Notification Service

- Мова: Node.js
- Фреймворк: NestJS
- Причина вибору: NestJS дозволяє реалізовувати event-driven логіку та працювати з чергами повідомлень. Має вбудовану підтримку RabbitMQ.
- Інтеграція: Отримує повідомлення з інших сервісів через брокер подій (наприклад, RabbitMQ) та надсилає електронні листи або SMS.

8. API Gateway

- Мова: JavaScript (Node.js)
- Фреймворк: Nestjs + middleware (Rate Limiter, CORS, Auth)
- Причина вибору: Nestjs дозволяє швидко реалізувати проксуючий сервер з кастомною маршрутизацією.
- Інтеграція: Усі зовнішні запити проходять через API Gateway. Реалізовано централізовану автентифікацію, логування та обмеження швидкості.

9. Admin Panel

- Фронтенд: Nestjs + Material Design

- Причина вибору: Nestjs забезпечує динамічний UI, Redux — зручне управління станом, а Material Design дозволяє швидко створювати інтерфейси адміністративного рівня.
- Інтеграція: Використовує API від Support, Analytics, Billing, Subscription Service.

10. User Web Interface

- Фронтенд: Nextjs
- Причина вибору: Nextjs легкий у використанні, швидкий у рендерінгу та ідеально підходить для створення SPA, що взаємодіє з API.
- Інтеграція: Взаємодіє з Auth, Subscription, Notification та Support через API Gateway.

Взаємодія між сервісами

Взаємодія мікросервісів реалізується за допомогою двох моделей:

- Синхронна — HTTP/REST через API Gateway (наприклад, Auth → User, Subscription → Billing)
- Асинхронна — через брокер подій (RabbitMQ або Kafka), наприклад для Notification Service та подій Billing → Analytics.

Дані зберігаються у відповідних локальних базах даних кожного сервісу. Для взаємного доступу реалізовано обмежені контракти (DTO) для API. Це дозволяє знизити зв'язаність компонентів і спростити оновлення.

Загальна інфраструктура сервісів оркеструється через Docker і Kubernetes, що забезпечує масштабованість, ізоляцію середовищ і контроль за розгортанням.

Хостинг та інфраструктура

Для хостингу серверів та забезпечення надійної інфраструктури буде використовуватись AWS (Amazon Web Services). Основними причинами вибору AWS є:

- Масштабованість: AWS забезпечує легке масштабування ресурсів відповідно до потреб сервісу, що дозволяє ефективно управляти навантаженням та забезпечувати безперебійну роботу системи.
- Надійність: AWS має високу надійність завдяки своїм розподіленим дата-центрам, що дозволяє мінімізувати ризик простоїв та втрати даних.
- Безпека: AWS пропонує широкий спектр інструментів та сервісів для забезпечення безпеки даних та управління доступом, що є критично важливим для телекомунікаційних компаній.
- Гнучкість: AWS підтримує широкий спектр технологій та фреймворків, що дозволяє легко інтегрувати різні компоненти системи та забезпечувати їх ефективну роботу.

При виборі технологій враховувались такі чинники, як продуктивність, підтримка асинхронної обробки, масштабованість, доступність розробників та активність спільноти. Мікросервіси функціонують незалежно один від одного, взаємодіючи через REST API або брокери повідомлень, що дозволяє досягнути високої гнучкості, ізольованості та надійності в роботі системи. Для покращення продуктивності використовується Redis для кешування даних, що зменшує навантаження на базу даних і прискорює доступ до часто використовуваних даних. Контейнеризація з Docker і оркестрація з Kubernetes забезпечують легке розгортання та масштабування системи. Використання GitLab CI/CD автоматизує процеси інтеграції та розгортання, що дозволяє безперервно впроваджувати зміни. Хмарні провайдери (AWS, Google Cloud, Azure) забезпечують масштабованість і надійність інфраструктури, дозволяючи швидко реагувати на зміни у навантаженні.

4.2 Реалізація інтерфейсу IT-сервісу.

Побудуємо мокапи інтерфейсу, що дозволить сформуванати загальну концепцію навігації, структуру екранів та сценарії взаємодії користувача з системою. Мокапи інтерфейсу зображені в додатках В-Д. На основі даних прототипів було розроблено фінальний дизайн інтерфейсу. На рис. 4.2 зображено лендінг сервісу, що включає в себе кнопку особистого кабінету, модуль «Передзвоніть мені», картки сервісів та панель навігації.



Рис. 4.2. – Головна сторінка сервісу

Інтерфейс персонального акаунту (рис. 4.3.) розроблено з урахуванням принципів зручності (usability), логічної навігації та швидкого доступу до ключових функцій.

Основні елементи інтерфейсу:

Панель навігації (ліва частина екрану):

Забезпечує швидкий доступ до основних розділів, таких як:

- «Мій профіль»
- «Підписки»
- «Оплати»

- «Підтримка»
- «Налаштування»

Центральна панель (основна інформація):

Виводиться актуальний баланс користувача, що відображає залишок коштів або заборгованість. Поруч із балансом відображається вибраний метод оплати (наприклад, банківська карта або платіжна система).

Секція активних сервісів (нижня частина центральної панелі):

Перелічуються всі активні підписки, їхній поточний статус, а також сумарна місячна вартість обраних послуг. Користувач може змінювати тариф, призупинити або скасувати підписку.

Звіт пропускної здатності:

У вигляді графіку або таблиці відображається поточне навантаження на сервіси (наприклад, середній рівень трафіку, використання обсягу даних), що дозволяє користувачеві оцінювати ефективність послуги.

Дизайн передбачає мінімалістичний стиль, акцент на важливу інформацію, інтерактивні підказки та адаптацію під мобільні пристрої.

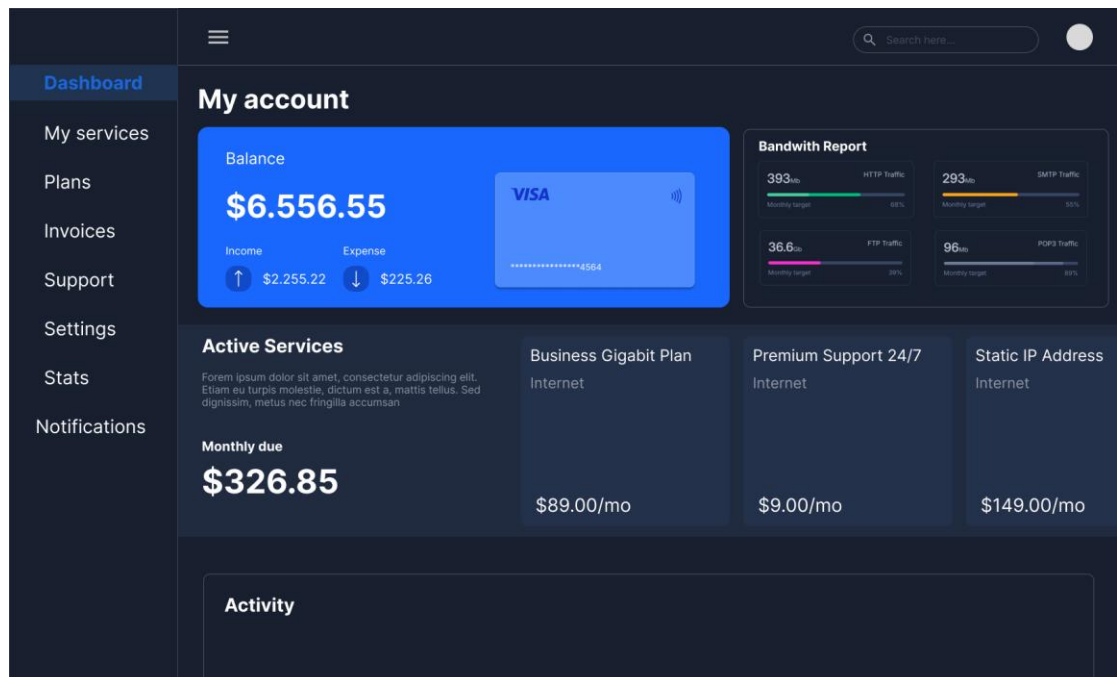


Рис. 4.3. – Прототип персонального кабінету користувача

Інтерфейс адміністративної панелі (рис. 4.4.) розроблений для забезпечення зручного доступу до аналітичних показників, управління підписками, користувачами та сервісами. Панель орієнтована на менеджерів і операторів компанії.

Основні елементи адмін-панелі:

Графік продажів (основний верхній блок):

Відображає загальну динаміку підписок та доходів за обраний період (день, тиждень, місяць). Реалізовано у вигляді інтерактивної лінійної або стовпчастої діаграми.

Інформаційні панелі (dashboard-картки):

- Продажі за регіонами — карта або список, розподіл за географією.
- Кількість зареєстрованих користувачів — показник активного зростання бази.
- Загальний дохід — сума надходжень за поточний місяць.

Панелі розміщені горизонтально після основного графіка та оновлюються у реальному часі.

Таблиця сервісів (нижня частина):

У табличному вигляді наведено перелік всіх тарифних планів, їх статус (активний/деактивований), ціни, кількість користувачів, які підключили кожен сервіс. Доступна функція фільтрації, редагування й деактивації тарифу.

Фільтри й пошук:

Забезпечено зручний інтерфейс для швидкого пошуку користувача, послуги або періоду аналітики.

Адмін-панель також містить журнал активності, механізм управління ролями користувачів, та можливість експорту звітів у форматі PDF/CSV.

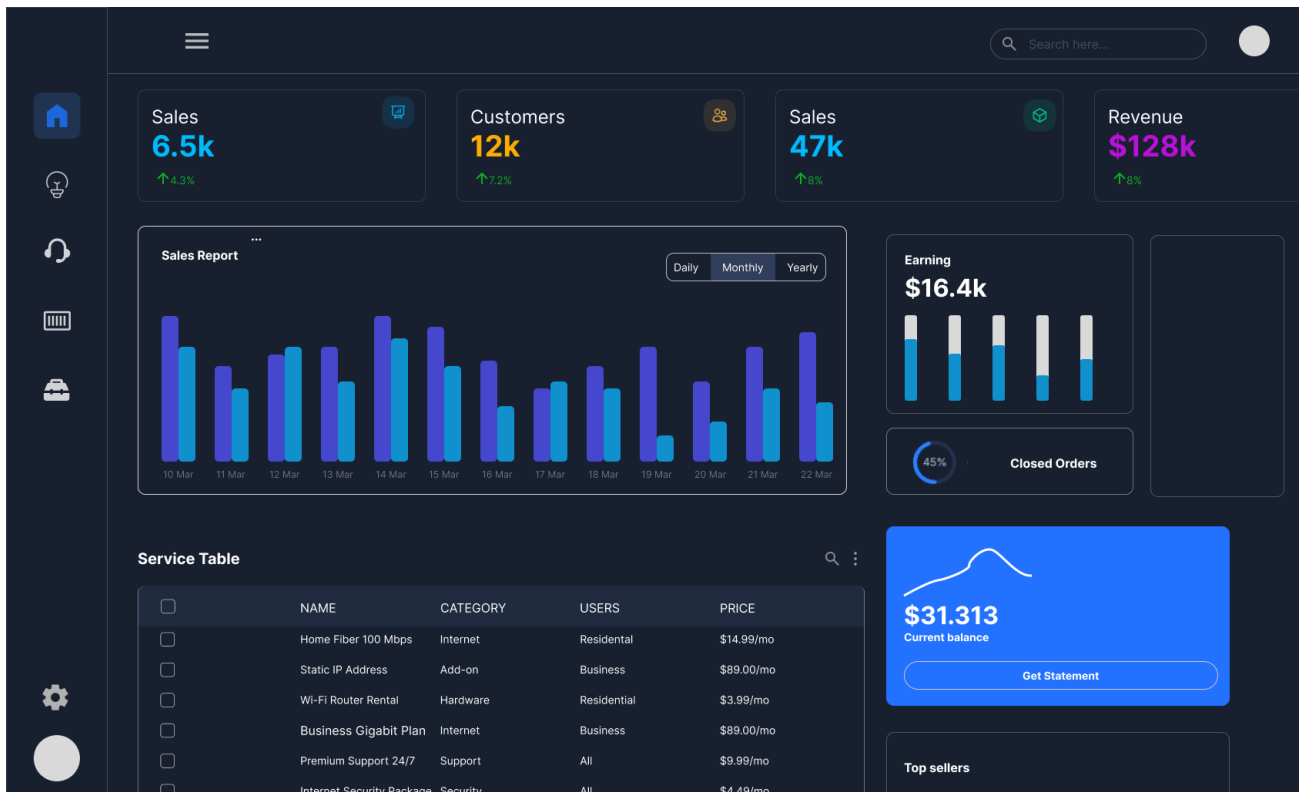


Рис. 4.4. – прототип адмін панелі

4.3 Тестування програмного забезпечення

Для перевірки функціональності, стабільності та надійності реалізованих модулів системи було проведено функціональне тестування з використанням **тест-кейсів**, що охоплюють ключові користувацькі сценарії. Кожен тест-кейс описує вхідні умови, очікувану поведінку системи, а також результат, який дозволяє оцінити відповідність фактичної реалізації заданим вимогам.

Тестування проводилося вручну на прототипі веб-інтерфейсу. Було обрано 5 ключових функціональних кейсів, які покривають основні процеси: реєстрація, авторизація, розрахунок послуг, перевірка покриття, надсилання звернення. Для кожного з них створено таблицю з формалізованим описом дій.

Тест-кейс 1: Реєстрація нового користувача.

Мета: перевірити створення облікового запису з коректним збереженням у базі даних.

Тест Кейс 1

Поле	Значення
Вхідні дані	Ім'я: testuser; Email: test@example.com; Пароль: Test1234
Дії	<ol style="list-style-type: none"> 1. Відкрити сторінку реєстрації 2. Ввести дані 3. Натиснути «Зареєструватися»
Вихідні дані	Новий запис у базі: { "email": "test@example.com", "role": "user" }
Очікуваний результат	Відображення повідомлення про успішну реєстрацію; редирект на сторінку входу

Тест-кейс 2: Авторизація користувача

Мета: перевірити вхід користувача з правильними обліковими даними та генерацію токена.

Тест Кейс 2

Поле	Значення
Вхідні дані	Email: test@example.com; Пароль: Test1234
Дії	<ol style="list-style-type: none"> 1. Відкрити форму входу 2. Ввести логін і пароль 3. Натиснути «Увійти»
Вихідні дані	JWT-токен, userID, redirect на /dashboard
Очікуваний результат	Успішний вхід у систему, доступ до особистого кабінету

Тест-кейс 3: Розрахунок вартості підключення

Мета: перевірити правильність виконання розрахунку вартості підключення на основі введених параметрів.

Тест кейс 3

Поле	Значення
Вхідні дані	Адреса: вул. Київська, 15 Поверх: 3 Тип послуги: «Домашній інтернет»
Дії	1. Відкрити сторінку розрахунку 2. Заповнити дані 3. Натиснути «Розрахувати»
Вихідні дані	Вартість: 549 грн
Очікуваний результат	Відображення правильної ціни з урахуванням вхідних параметрів

Тест-кейс 4: Перевірка покриття

Мета: перевірити коректне відображення карти покриття та визначення рівня доступності послуги.

Тест кейс 4

Поле	Значення
Вхідні дані	Район: «Солом'янський», будинок №12
Дії	1. Перейти на сторінку «Карта покриття» 2. Обрати адресу
Вихідні дані	Статус: «Покриття доступне» Коеф. покриття: 85%
Очікуваний результат	Відображення точної інформації на карті з актуальними даними

Тест-кейс 5: Надсилання звернення у підтримку

Мета: перевірити функціональність відправлення звернення з форми підтримки.

Тест кейс 5

Поле	Значення
Вхідні дані	Ім'я: Іван; Email: ivan@example.com ; Тема: «Зник інтернет»
Дії	1. Відкрити сторінку підтримки2. Заповнити форму3. Натиснути «Надіслати»
Вихідні дані	Повідомлення: «Ваше звернення надіслано. ID: #3421»
Очікуваний результат	Дані успішно збережено у базі; користувач бачить підтвердження

Результати ручного функціонального тестування підтвердили працездатність основних модулів системи. Усі тест-кейси виконано успішно: введені дані були оброблені правильно, відповіді системи відповідали очікуваним результатам.

Отримані результати свідчать про відповідність реалізованого програмного забезпечення початковим функціональним вимогам.

Надалі можливе розширення тестування через впровадження автоматизованих сценаріїв із використанням бібліотек Cypress, Jest або Postman Collection для API.

ВИСНОВКИ

Магістерська робота присвячена дослідженню процесів управління проектом розробки інтернет-сервісу для телекомунікаційної компанії. Актуальність теми обґрунтована стрімким розвитком цифрових технологій у телеком-секторі та необхідністю впровадження сучасних ІТ-рішень для покращення взаємодії з клієнтами, автоматизації бізнес-процесів і забезпечення гнучкості в умовах високої конкуренції.

У процесі реалізації поставленої мети було вирішено всі поставлені завдання:

- Проведено всебічний аналіз предметної області, ринку телекомунікаційних послуг та конкурентного середовища.
- Сформовано технічне завдання та паспорт проекту, в якому визначено мету, функціональні та нефункціональні вимоги, зацікавлені сторони та очікувані результати реалізації.
- Виконано SWOT-аналіз, що дозволив систематизувати внутрішні сильні та слабкі сторони, а також зовнішні чинники, які можуть вплинути на успіх проекту.
- Обґрунтовано вибір гібридної методології управління (Water-Scrum-Fall), що поєднує структурованість каскадного підходу з гнучкістю Agile, та описано життєвий цикл розробки відповідно до стандартів PMBOK.
- Побудовано організаційну структуру управління проектом та визначено ролі учасників команди. За допомогою матриці RACI чітко розподілено відповідальність між учасниками проекту.
- Розроблено ієрархічну структуру робіт (WBS), календарний план, систему часових віх та діаграму Ганта, що забезпечили можливість ефективного планування і контролю ходу реалізації.

- Здійснено оцінку бюджету проєкту, визначено критичні статті витрат, проведено розрахунок показників економічної ефективності (NPV, IRR, ROI, період окупності), що підтвердили фінансову доцільність проєкту.
- Побудовано модель ризиків проєкту, проведено кількісний та якісний аналіз з визначенням стратегії реагування на кожен ідентифікований ризик.
- Розроблено архітектуру інтернет-сервісу на основі мікросервісного підходу. Кожен компонент системи спроектовано з урахуванням масштабованості, ізоляції та ефективності. Для кожного мікросервісу підібрано оптимальні технології реалізації.
- Реалізовано інтерфейс користувача та адміністратора, враховуючи принципи UX/UI та адаптивного дизайну. Запропоновано чітку навігацію, інформативні дашборди та зручні механізми керування сервісами.
- Проведено функціональне тестування програмного забезпечення з використанням тест-кейсів, які охоплюють основні бізнес-процеси. Всі ключові функції системи пройшли успішну перевірку.

Розроблені рішення є прикладом ефективного поєднання класичних та сучасних підходів до управління ІТ-проєктами. Отримані результати можуть бути використані як основа для впровадження інтернет-сервісів у телекомунікаційних компаніях, а також адаптовані до інших проєктів цифрової трансформації. Матеріали роботи мають як теоретичне, так і практичне значення та можуть бути застосовані у професійній діяльності, навчальних курсах і подальших дослідженнях у галузі проєктного менеджменту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 *ISO - International Organization for Standardization.*
URL: <https://www.iso.org/obp/ui/#iso:std:iso:tr:21506:ed-1:v1:en>
- 2 Institute P. M. PMBOK Guide: The Project Management Body of Knowledge. Booksmith Publishing LLC, 2021.
- 3 Scrum Guide | Scrum Guides. *Home / Scrum Guides.*
URL: <https://scrumguides.org/scrum-guide.html> (date of access: 14.04.2025).
- 4 Telecom project management: Lessons learned and best practices: A review from Africa to the USA / Chinedu Alex Ezeigweneme та ін. *World Journal of Advanced Research and Reviews.* 2023. Т. 20, № 3. С. 1713–1730.
URL: <https://doi.org/10.30574/wjarr.2023.20.3.2707> (дата звернення: 15.04.2025).
- 5 Ericsson Mobility Report. URL: <https://www.ericsson.com/en/reports-and-papers/mobility-report> (дата звернення: 15.04.2025).
- 6 Fransman, M. (2001). Evolution of the telecommunications industry into the internet age. *Communications and Strategies*, 43, 57-113. Доступ: http://www.idate.org/fic/revue_telech/459/C&S43_FRANSMAN.pdf
- 7 Balashova E. Agile project management in telecommunications industry. *Revista Espacios.* 2017. С. 13–14.
- 8 Копчак, Ю., Лобунець, Т., & Луковський, Р. (2024). SWOT-АНАЛІЗ ЯК ВАЖЛИВИЙ ІНСТРУМЕНТ У РОЗРОБЦІ СТРАТЕГІЇ БІЗНЕСУ. *Економіка та суспільство*, (61). <https://doi.org/10.32782/2524-0072/2024-61-146>
- 9 Macleod, Les. (2013). Making SMART goals smarter. *Physician executive.* 38. 68-70, 72.
- 10 Opportunities for educational innovations in authentic project-based learning: understanding instructor perceived challenges to design for adoption / D. G. Rees Lewis та ін. *Educational Technology Research and Development.* 2019. Т. 67, №

4. С. 953–982. URL: <https://doi.org/10.1007/s11423-019-09673-4> (дата звернення: 16.04.2025).
- 11 Brandl F. J., Kagerer M., Reinhart G. A Hybrid Innovation Management Framework for Manufacturing – Enablers for more Agility in Plants. *Procedia CIRP*. 2018. Т. 72. С. 1154–1159.
URL: <https://doi.org/10.1016/j.procir.2018.04.022> (дата звернення: 17.04.2025).
- 12 D. West, “Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today: for Application Development & Delivery Professionals,” Forrester Research, vol. 26, pp. 1–17, 2011.
- 13 Durbin M., Niederman F. Bringing templates to life: overcoming obstacles to the organizational implementation of Agile methods. *International Journal of Information Systems and Project Management*. 2021. Т. 9, № 3. С. 1–18.
URL: <https://doi.org/10.12821/ijispm090301> (дата звернення: 17.04.2025).
- 14 Boehm B., Turner R. Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Software*. 2005. Т. 22, № 5. С. 30–39. URL: <https://doi.org/10.1109/ms.2005.129> (дата звернення: 17.04.2025).
- 15 Cooper R. G. Idea-to-Launch Gating Systems: Better, Faster, and More Agile. *Research-Technology Management*. 2017. Т. 60, № 1. С. 48–52.
URL: <https://doi.org/10.1080/08956308.2017.1255057> (дата звернення: 17.04.2025).
- 16 Cooper R. G., Sommer A. F. Agile–Stage-Gate for Manufacturers. *Research-Technology Management*. 2018. Т. 61, № 2. С. 17–26.
URL: <https://doi.org/10.1080/08956308.2018.1421380> (дата звернення: 17.04.2025).
- 17 AGILE PROJECT MANAGEMENT IN GOVERNMENTAL ORGANIZATIONS – METHODOLOGICAL ISSUES. *vol 6, issue 16*. 2020.

- T. 6, № 16. С. 262–275. URL: <https://doi.org/10.18769/ijasos.616037> (дата звернення: 17.04.2025).
- 18 R. Hassani, Y. El Bouzekri El Idrissi, and A. Abouabdellah, “Digital Project Management in the Era of Digital Transformation: Hybrid method,” Proceedings of the 2018 International Conference on Software Engineering and Information Management, pp. 98–103, 2018, doi: 10.1145/3178461.3178472
 - 19 E. C. Conforto and D. C. Amaral, “Agile project management and stage-gate model - A hybrid framework for technology-based companies,” Journal of Engineering and Technology Management, Т. 40, №. 1, С. 1–14, 2016, doi: 10.1016/j.jengtecman.2016.02.003. (дата звернення: 17.04.2025).
 - 20 M. R. Farokhad, J. R. Otegi-Olaso, L. S. Pinilla, N. T. Gandarias, and L. N. L. de Lacalle, “Assessing the Success of R&D Projects and Innovation Projects through Project Management Life Cycle,” Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, С. 1104–1110, 2019, doi: 10.1109/IDAACS.2019.8924298. (дата звернення: 17.04.2025).
 - 21 Z. T. Kosztyán, R. Jakab, G. Novák, and C. Hegedűs, “Survive IT! Survival analysis of IT project planning approaches,” Operations Research Perspectives, Т. 7, С. 1–17, 2020, doi: 10.1016/j.orp.2020.100170. (дата звернення: 17.04.2025).
 - 22 F. P. Zasa, A. Patrucco, and E. Pellizzoni, “Managing the Hybrid Organization: How Can Agile and Traditional Project Management Coexist?,” Research-Technology Management, Т. 64, №. 1, С. 54–63, 2021, doi: 10.1080/08956308.2021.1843331. (дата звернення: 17.04.2025).
 - 23 F. Copola Azenha, D. Aparecida Reis, and A. Leme Fleury, “The Role and Characteristics of Hybrid Approaches to Project Management in the Development of Technology-Based Products and Services,” Project Management Journal, Т. 52,

- №. 1, С. 90–110, 2021, doi: 10.1177/8756972820956884. (дата звернення: 17.04.2025).
- 24 P. R. Alves, A. Tereso, and G. Fernandes, “Project Management System Implementation in SMEs: A Case Study,” Proceedings of 33rd International Business Information Management Association Conference IBIMA 2019, С. 8322–8332, 2019.
- 25 K. Edwards, R. G. Cooper, T. Vedsmand, and G. Nardelli, “Evaluating the Agile-Stage-Gate Hybrid Model: Experiences From Three SME Manufacturing Firms,” International Journal of Innovation and Technology Management, Т. 16, №. 8, С. 229–263, 2020, doi: 10.1142/S0219877019500482. (дата звернення: 17.04.2025).
- 26 V. Yadav, M. Adya, D. Nath, and V. Sridhar, “Investigating an ‘Agile-Rigid’ Approach in Globally Distributed Requirements Analysis,” PACIS 2007 Proceedings. 12. Pacific Asia Conference on Information Systems, С. 150–165, 2007
- 27 T. Žužek, J. Kušar, L. Rihar, and T. Berlec, “Agile-Concurrent hybrid: A framework for concurrent product development using Scrum,” Concurrent Engineering, Т. 28, №. 4, С. 255–264, 2020, doi: 10.1177/1063293X20958541 (дата звернення: 17.04.2025).
- 28 A. Ziółkowski and T. Deręowski, “Hybrid Approach in Project Management – Mixing Capability Maturity Model Integration with Agile Practices,” Social Sciences, Т. 85, №. 3, С. 64–71, 2014, doi: 10.5755/j01.ss.85.3.8416 (дата звернення: 17.04.2025).
- 29 A. Gemino, B. Horner Reich, and P. M. Serrador, “Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice?,” Project Management Journal, Т. 52, №. 2, С. 161–175, 2021, doi: 10.1177/8756972820973082.
- 30 Utama W. A., Latief Y. Analysis cost of safety on project irrigation/channel base on WBS (Work Breakdown Structure). *IOP Conference Series: Materials Science*

and Engineering. 2021. T. 1098, № 2. C. 022027.

URL: <https://doi.org/10.1088/1757-899x/1098/2/022027> (дата звернення: 18.04.2025).

- 31 Stare A., Agile Project Management in Product Development projects. *Science Direct*. 2014, C.295-304.
- 32 The Guardian. Telecommunications industry.(2020) URL: <https://www.theguardian.com/business/telecoms>
- 33 M. Špundak, “Mixed Agile/Traditional Project Management Methodology – Reality or Illusion?,” *Procedia Social and Behavioral Sciences 27th IPMA World Congress*, T. 119, C. 939–948, 2014, doi: 10.1016/j.sbspro.2014.03.105.
- 34 G. Fernandes, S. Moreira, M. Araújo, E. B. Pinto, and R. J. Machado, “Project Management Practices for Collaborative University-Industry R&D: A Hybrid Approach,” *Procedia Computer Science International Conference on Enterprise Information Systems*, T. 138, C. 805–814, 2018, doi: 10.1016/j.procs.2018.10.105.
- 35 V. Vinekar, C. W. Slinkman, and S. Nerur, “Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View,” *Information Systems Management*, T. 23, №. 3, C. 31–42, 2006, doi: 10.1201/1078.10580530/46108.23.3.20060601/93705.4.
- 36 F. J. Brandl, M. Kagerer, and G. Reinhart, “A Hybrid Innovation Management Framework for Manufacturing – Enablers for more Agility in Plants,” *Procedia CIRP 51st CIRP Conference on Manufacturing Systems*, T. 72, C. 1154–1159, 2018, doi: 10.1016/j.procir.2018.04.022.
- 37 D. West, “Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today: for Application Development & Delivery Professionals,” *Forrester Research*, T. 26, C. 1–17, 2011.
- 38 M. Durbin and F. Niederman, “Bringing templates to life: overcoming obstacles to the organizational implementation of Agile methods,” *International Journal of*

- Information Systems and Project Management, T. 9, №. 3, C. 1–18, 2021, doi: 10.12821/ijispm090301.
- 39 B. Boehm and R. Turner, “Management Challenges to Implementing Agile Processes in Traditional Development Organizations,” *IEEE Software*, T. 22, №. 5, C. 30–39, 2005, doi: 10.1109/MS.2005.129.
- 40 R. G. Cooper, “Idea-to-Launch Gating Systems: Better, Faster, and More Agile,” *Research-Technology Management*, T. 60, №. 1, C. 48–52, 2017, doi: 10.1080/08956308.2017.1255057.
- 41 R. G. Cooper and A. F. Sommer, “Agile–Stage-Gate for Manufacturers,” *Research-Technology Management*, T. 61, №. 2, C. 17–26, 2018, doi: 10.1080/08956308.2018.1421380.
- 42 M. Bogdanova, E. Parashkevova, and M. Stoyanova, “Agile Project Management in Governmental Organizations - Methodological Issues,” *IJASOS- International E-Journal of Advances in Social Sciences*, T. 6, №. 16, C. 262–275, 2020, doi: 10.18769/ijasos.616037.
- 43 R. Hassani, Y. El Bouzekri El Idrissi, and A. Abouabdellah, “Digital Project Management in the Era of Digital Transformation: Hybrid method,” *Proceedings of the 2018 International Conference on Software Engineering and Information Management*, C. 98–103, 2018, doi: 10.1145/3178461.3178472.
- 44 E. C. Conforto and D. C. Amaral, “Agile project management and stage-gate model - A hybrid framework for technology-based companies,” *Journal of Engineering and Technology Management*, T. 40, №. 1, C. 1–14, 2016, doi: 10.1016/j.jengtecman.2016.02.003.
- 45 Z. T. Kosztyán, R. Jakab, G. Novák, and C. Hegedűs, “Survive IT! Survival analysis of IT project planning approaches,” *Operations Research Perspectives*, T. 7, C. 1–17, 2020, doi: 10.1016/j.orp.2020.100170.

ДОДАТКИ

Додаток А.

Лістинг алгоритму оптимального прокладання кабелю

```
import numpy as np

# Генеруємо дані для трьох районів з різним рівнем покриття
# Район 1: найбільше підключених будинків
# Район 2: середня кількість підключених будинків
# Район 3: найменше підключених будинків

# Параметри
alpha = 0.1 # коефіцієнт масштабування
radius = 500 # радіус аналізу в метрах

# Генерація тестових даних
np.random.seed(42)

def generate_area_data(total_houses, connected_houses, max_distance):
    distances = np.random.uniform(0, max_distance, total_houses)
    connectivity = np.zeros(total_houses)
    connected_indices = np.random.choice(total_houses, connected_houses,
replace=False)
    connectivity[connected_indices] = 1
    return distances, connectivity

# Район 1
total_houses_1 = 100
connected_houses_1 = 80
max_distance_1 = 50 # метри
distances_1, connectivity_1 = generate_area_data(total_houses_1,
connected_houses_1, max_distance_1)

# Район 2
total_houses_2 = 100
connected_houses_2 = 50
max_distance_2 = 100 # метри
distances_2, connectivity_2 = generate_area_data(total_houses_2,
connected_houses_2, max_distance_2)

# Район 3
```

```
total_houses_3 = 100
connected_houses_3 = 20
max_distance_3 = 150 # метри
distances_3, connectivity_3 = generate_area_data(total_houses_3,
connected_houses_3, max_distance_3)

# Функція для обчислення коефіцієнта корисності
def utility_coefficient(distances, connectivity, alpha):
    weights = np.exp(alpha * distances)
    utility = np.sum(weights * connectivity) / np.sum(weights)
    return utility

# Розрахунок коефіцієнтів корисності для трьох районів
utility_1 = utility_coefficient(distances_1, connectivity_1, alpha)
utility_2 = utility_coefficient(distances_2, connectivity_2, alpha)
utility_3 = utility_coefficient(distances_3, connectivity_3, alpha)

utility_1, utility_2, utility_3
```

```
import heapq

def heuristic(a, b):
    return abs(b[0] - a[0]) + abs(b[1] - a[1])

def a_star(graph, start, goal):
    frontier = []
    heapq.heappush(frontier, (0, start))
    came_from = {}
    cost_so_far = {}
    came_from[start] = None
    cost_so_far[start] = 0

    while frontier:
        current = heapq.heappop(frontier)[1]

        if current == goal:
            break

        for next in graph.neighbors(current):
            new_cost = cost_so_far[current] + graph.cost(current, next)
            if next not in cost_so_far or new_cost < cost_so_far[next]:
                cost_so_far[next] = new_cost
                priority = new_cost + heuristic(goal, next)
                heapq.heappush(frontier, (priority, next))
                came_from[next] = current

    return came_from, cost_so_far

# Usage:
# graph = Graph() # Define your graph here
# start, goal = (1, 1), (5, 5)
# came_from, cost_so_far = a_star(graph, start, goal)
```

Додаток В.

Лістинг класів інтернет-сервісу

```
class User:
    def __init__(self, name, email):
        self.name = name
        self.email = email

    def get_details(self):
        return f"Name: {self.name}, Email: {self.email}"

class Cart:
    def __init__(self):
        self.items = []

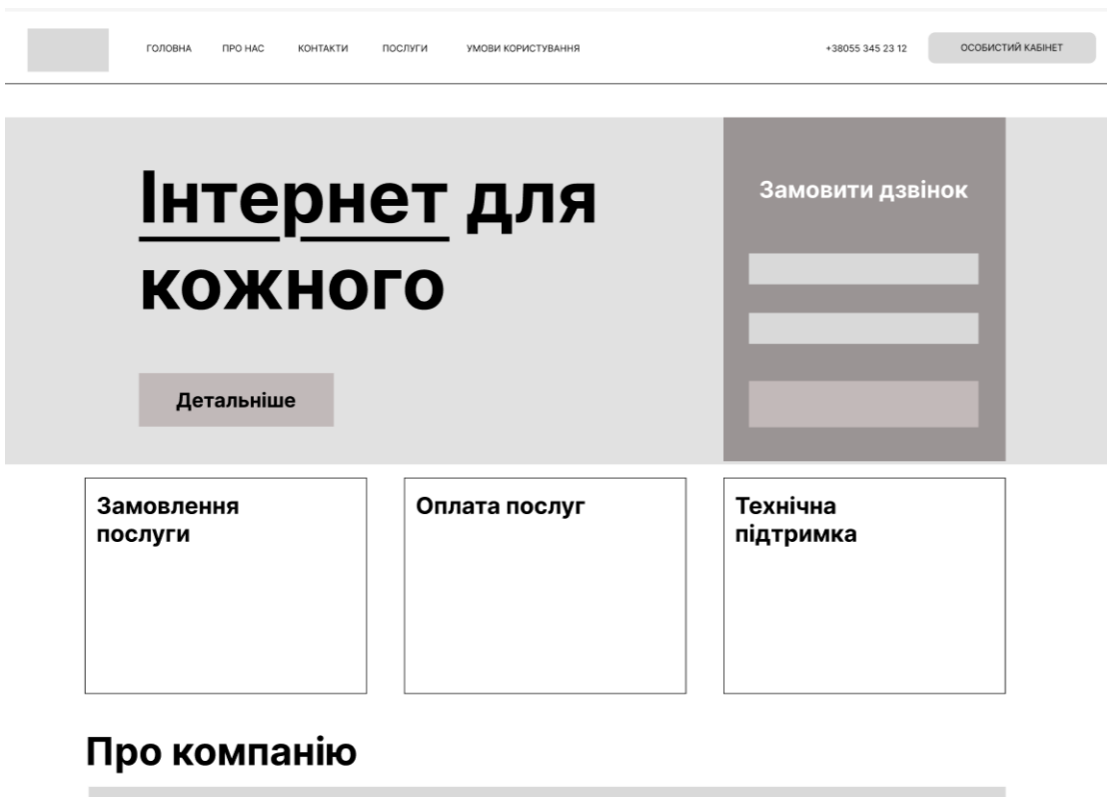
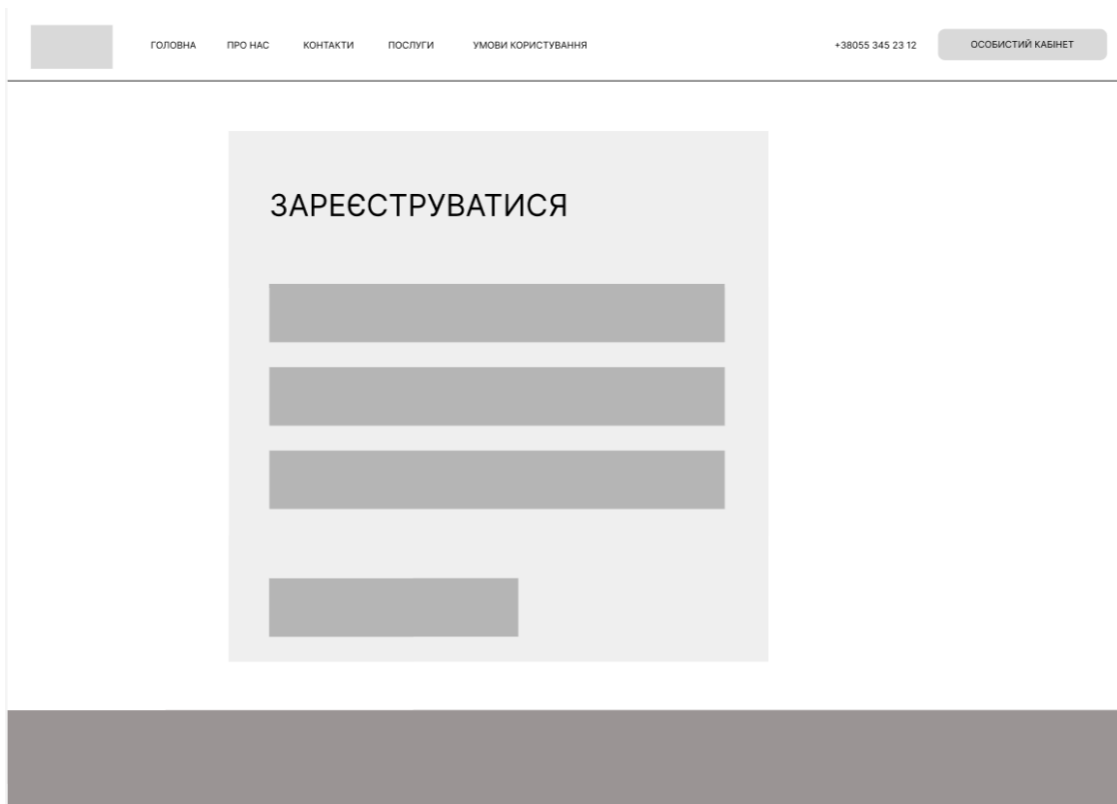
    def add_item(self, item):
        self.items.append(item)

    def remove_item(self, item):
        if item in self.items:
            self.items.remove(item)

    def get_items(self):
        return self.items

class Service:
    def __init__(self, name, price):
        self.name = name
        self.price = price

    def get_details(self):
        return f"Service: {self.name}, Price: {self.price}"
```



Додаток Е. Діаграма Чена

