

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження технологій управління проектом створення мобільного додатку
для навчання користувачів у форматі мікро-льорнінгу»

Студента 2-го курсу групи УПз-21

Дениса РУДЯКА
(Ім'я, ПРІЗВИЩЕ)

(підпис студента)

Науковий керівник:

к.е.н., доцент
(науковий ступінь, вчене звання)

Анна КОЛОМІЄЦЬ
(Ім'я, ПРІЗВИЩЕ)

(дата)

(підпис)

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач
кафедри технологій
управління

(підпис)

Віктор МОРОЗОВ
(Ім'я, ПРІЗВИЩЕ)

(дата)

Київ – 2025

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Освітній рівень Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Управління проектами

ЗАТВЕРДЖУЮ

Завідувач кафедри

професор Віктор МОРОЗОВ

«29» вересня 2025 року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Денис РУДЯК

Група: УПз-21

1. Тема кваліфікаційної роботи: «Дослідження технологій управління проектом створення мобільного додатку для навчання користувачів у форматі мікро-льорнінгу»

Затверджена Протоколом №15 від 16.06.2025 року.

2. Строк подання студентом готової роботи – 10.12.2025 р.

3. Цільова установка та вихідні дані до роботи: дослідженні сучасних методів та інструментів управління проектами у сфері цифрових освітніх технологій, аналізі існуючих мобільних платформ мікро-льорнінгу та концепцій створення інформаційних систем, а також у розробці науково обґрунтованої моделі реалізації мобільного додатку для персоналізованого навчання. До вихідних даних належить вивчення теоретичних основ мікро-льорнінгу, принципів побудови адаптивних навчальних систем, методологій управління програмними проектами та особливостей архітектурних підходів до мобільних застосунків.

У рамках роботи передбачається обґрунтування вибору методології **Agile (Scrum)** як найбільш придатної для ітеративної розробки освітнього цифрового продукту; формування структури команди та її ролей; визначення механізмів управління ризиками та ресурсами; проведення аналізу вимог і проектування архітектури системи; а також створення прототипу інтерфейсу мобільного додатку для демонстрації основних функціональних можливостей.

4. Зміст роботи: Зміст роботи охоплює теоретичні, аналітичні та практичні аспекти створення мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу.

5. Перелік графічного матеріалу: цілі та завдання проекту, дерево проблем, дерево цілей, бізнес-аналітичні моделі: SWOT, PEST. Основні ролі, команди та обов'язки. Концептуальна модель бази даних, фізична модель бази даних.

Визначення вимог до проєкту, WBS проєкту. Концептуальна та архітектурна модель мобільного додатку. Паспорт проєкту. Алгоритм системи прийняття рішень по плану навчання користувача. Компоненти продукту, дизайн-концепція мобільного додатку.

6. Календарний план виконання роботи

№ з/п	Назва частин роботи	Виконання роботи
1	Вивчення літературних джерел з предмету дослідження	02.10.25-11.10.25
2	Збір і вивчення матеріалів	12.10.25-19.10.25
3	Складання розгорнутого плану кваліфікаційної роботи	20.10.25-23.10.25
4	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	24.10.25-25.10.25
5	Підготовка розділу 1	26.10.25-02.11.25
6	Підготовка розділу 2	03.11.25-14.03.25
7	Підготовка розділу 3	15.11.25-23.11.25
8	Підготовка розділу 4	24.11.25-05.12.25
9	Оформлення кваліфікаційної роботи	25.11.25-02.12.25
10	Передача кваліфікаційної роботи науковому керівникові	02.12.25
11	Попередній захист кваліфікаційної роботи	10.12.25-14.12.25
12	Передача кваліфікаційної роботи рецензенту для рецензування	15.12.25

Дата видачі завдання «30» вересня 2025 р.

Керівник роботи доцент Анна КОЛОМІЄЦЬ

(посада, ім'я, прізвище)

(підпис)

Завдання прийняла до виконання студент групи УПз-21

Денис РУДЯК

(ім'я, прізвище)

ЗМІСТ

АНОТАЦІЯ	6
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
РОЗДІЛ 1. ТЕОРЕТИЧНЕ ПІДГРУНТЯ ТА ДОСЛІДЖЕННЯ ЩОДО ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ	12
1.1 Актуальність персоналізованого навчання та мікро-льорнінгу	12
1.2 Сучасні тренди та цифрові рішення у сфері мобільного навчання	14
1.3 Аналіз існуючих додатків	15
1.4. Постановка задачі у вигляді паспорту проєкту	18
1.5. Формування наукової новизни та іновайційності	20
1.6. SWOT-аналіз та PEST-аналіз проєкту	21
РОЗДІЛ 2. АНАЛІЗ, МОДЕЛЮВАННЯ ТА СТРУКТУРУВАННЯ ВИМОГ ДО СИСТЕМИ	26
2.1 Ідентифікація потреб та характеристик цільової аудиторії	26
2.2 Визначення функціональних та нефункціональних вимог до системи	28
2.3. Побудова концептуальної моделі інформаційної системи	33
2.4 Проєктування структури бази даних мобільного додатку	36
2.5 Формування користувацьких ролей та сценаріїв поведінки	41
Use Case 1. Реєстрація та автентифікація користувача	41
Use Case 2. Формування персоналізованого навчального маршруту	42
Use Case 3. Проходження модуля	43
Use Case 4. Оцінювання результатів та відстеження прогресу	43
Use Case 5. Управління навчальним контентом	43
Use Case 6. Адміністрування системи	43
2.6 Аналіз вимог до безпеки та захисту персональних даних	43
2.7 OBS проєкту	46
2.8 WBS проєкту	48
2.9 Розрахунок економічної ефективності проєкту	53
2.10 Модель адаптивного керування навчальним планом	57
РОЗДІЛ 3. ВИБІР ТЕХНОЛОГІЇ УПРАВЛІННЯ ТА ПРОЄКТУВАННЯ АРХІТЕКТУРИ	61
3.1 Вибір технології управління проєктом створення мобільного додатку з мікро-льорнінгу	61
3.2 Структурна організація програмного продукту	64
3.3 Моделювання поведінки системи на основі сценаріїв використання	65

3.4 Архітурне проектування програмного забезпечення	69
3.5 Вибір технологічного стеку та інструментів реалізації	72
3.6 Проектування АРІ та взаємодії між компонентами системи	74
3.7 Забезпечення масштабованості та підтримованості системи	77
РОЗДІЛ 4. МОДЕЛЮВАННЯ КОРИСТУВАЦЬКОЇ ВЗАЄМОДІЇ ТА ТЕХНОЛОГІЇ ТЕСТУВАННЯ СИСТЕМИ	80
4.1 Розробка концепції взаємодії користувача з інтерфейсом додатку	80
4.2 Сценарії використання	84
4.3 Планування тестування та критерії якості	87
4.3 Оцінка очікуваної ефективності впровадження системи	89
4.4 Юзабіліті-тестування та оцінка користувацького досвіду	92
ВИСНОВКИ	95
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	96
ДОДАТКИ	100

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

«Дослідження технологій управління та проєктування мобільного додатку для персоналізованого навчання у форматі мікро-льорнінгу»

Студент: Рудяк Д.Т.

Науковий керівник: Коломієць А.С. Рік захисту: 2025

Мета роботи – провести комплексне дослідження технологій управління проєктом та методів проєктування інформаційних систем для створення та впровадження мобільного додатку персоналізованого навчання у форматі мікро-льорнінгу. У роботі здійснено аналіз підходів до розроблення мобільних освітніх застосунків, оцінювання їхньої життєздатності, а також побудову архітектури та функціональної моделі системи.

Ціль проєкту – спроектувати мобільний додаток, що забезпечує персоналізовані освітні підходи, адаптивне подання навчального контенту та можливості інтеграції з аналітичними сервісами для відстеження прогресу користувачів. Система спрямована на підвищення якості навчання, збільшення залученості користувачів та створення гнучкого середовища для мікро-льорнінгу в освітніх і корпоративних практиках.

Практична цінність роботи полягає у створенні концепції мобільного додатку, який дозволяє автоматизувати процеси подачі навчального контенту, персоналізації навчальних маршрутів, аналізу результатів та підвищення ефективності користувацької взаємодії. Запропоноване рішення може бути адаптоване для різних сфер дистанційного навчання, корпоративного тренінгу, підвищення кваліфікації та розвитку навичок співробітників.

Наукова новизна роботи полягає в обґрунтуванні вибору гнучких технологій управління проєктами (Scrum, Kanban) для розроблення мобільного освітнього застосунку, застосуванні сучасних підходів до архітектурного проєктування та формалізації моделі персоналізованого мікро-льорнінгу. У роботі розроблено концептуальну та логічну моделі інформаційної системи, визначено механізми адаптивного навчання,

сформовано структуру бази даних та описано моделі взаємодії між компонентами системи.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі проведено аналіз предметної області, досліджено сучасні тенденції розробки мобільних додатків, оцінено економічну доцільність проєкту, виконано SWOT та PEST-аналіз, сформовано наукову новизну та інноваційність рішення.

У другому розділі визначено вимоги до системи, виконано моделювання користувацьких ролей і сценаріїв, побудовано концептуальну модель системи, структуру бази даних, а також сформовано OBS та WBS проєкту.

У третьому розділі здійснено вибір технології управління, спроектовано архітектуру програмного забезпечення, розроблено API, описано принципи масштабованості та підтримованості системи.

У четвертому розділі представлено концепцію UI/UX, сценарії взаємодії користувача, описано методи тестування, критерії оцінювання якості та юзабіліті-тестування.

За результатами роботи зроблено висновки щодо ефективності застосування гнучких методологій управління проєктами, обґрунтовано вибір архітектурних рішень, визначено стратегії подальшого вдосконалення системи та її масштабування. Результати роботи підтверджують перспективність розвитку мобільного навчання у форматі мікро-льорнінгу та демонструють можливість успішного впровадження розробленого рішення в освітній та корпоративний сектор.

Ключові слова: мікро-льорнінг, персоналізоване навчання, управління проєктами, мобільний додаток, інформаційна система, адаптивне навчання, UI/UX дизайн, архітектура ПЗ.

Робота містить 101 сторінку без додатків, 21 рисунок та 8 таблиць. Додатки складають 6 сторінок.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

UI (User Interface) – інтерфейс користувача; зовнішній вигляд додатку, який взаємодіє з користувачем.

UX (User Experience) – досвід користувача; загальне враження від використання додатку, його зручність, інтуїтивність.

API (Application Programming Interface) – інтерфейс прикладного програмування; набір інструментів і протоколів для взаємодії між компонентами програмного забезпечення.

DB (Database) – база даних; організоване сховище інформації, що використовується для збереження навчального контенту та даних користувачів.

JSON (JavaScript Object Notation) – формат обміну даними, який використовується для зберігання і передачі структурованої інформації.

MVP (Minimum Viable Product) – мінімально життєздатний продукт; базова версія продукту з мінімальним набором функцій.

REST – архітектурний стиль для створення вебсервісів; використовується для взаємодії з сервером через HTTP.

IDE (Integrated Development Environment) – інтегроване середовище розробки, що використовується для написання та тестування коду.

ПЗ (Програмне забезпечення) – сукупність програм для реалізації функціоналу додатку.

Сценарій використання (Use case) – опис взаємодії користувача з системою для досягнення певної цілі.

Фреймворк – програмна структура, що полегшує розробку застосунків

Мікро-льорнінг – підхід до навчання, що базується на поданні навчального контенту у вигляді невеликих, легко засвоюваних фрагментів.

Персоналізоване навчання – адаптація навчального процесу відповідно до індивідуальних потреб, рівня та темпу засвоєння користувача.

ВСТУП

У сучасному світі динамічний розвиток цифрових технологій спричиняє кардинальні зміни у сфері освіти. Традиційні форми навчання поступово втрачають свою ефективність на тлі зростаючого попиту на гнучкі, доступні й персоналізовані освітні рішення. Одним із перспективних підходів у цьому напрямі є мікро-льорнінг – метод навчання, що передбачає подання інформації у вигляді коротких, чітко структурованих навчальних модулів, адаптованих до потреб користувача. Особливої актуальності мікро-льорнінг набуває в умовах використання мобільних пристроїв, що забезпечують безперервність освітнього процесу незалежно від місця й часу.

Зі зростанням інтересу до персоналізованого навчання виникає необхідність у створенні інтелектуальних систем, здатних адаптувати навчальний контент відповідно до індивідуальних особливостей, рівня підготовки та освітніх цілей кожного користувача. Комбінація мікро-льорнінгу та персоналізації дає змогу підвищити ефективність засвоєння знань, мотивацію до навчання та залученість користувачів.

Метою роботи є розробка проекту мобільного додатку, що реалізує концепцію персоналізованого навчання у форматі мікро-льорнінгу. У межах роботи буде здійснено аналіз сучасних рішень у цій сфері, визначено технічні та функціональні вимоги до системи, спроектовано її інформаційну структуру, базові компоненти архітектури та логіку взаємодії користувача з системою.

Для досягнення цієї мети необхідно:

- Проаналізувати сучасний стан галузі мобільного навчання та тенденції її розвитку;
- Здійснити огляд існуючих програм-аналогів, що реалізують мікро-льорнінг або персоналізоване навчання;
- Провести порівняльний аналіз програм-аналогів з урахуванням потенційних переваг і недоліків проєктованого рішення;

- Сформулювати функціональні та нефункціональні вимоги до мобільного додатку на основі проведеного аналізу;
- Виконати структурне та архітектурне проектування інформаційної системи;
- Розробити базові інструкції щодо передбачуваної взаємодії користувача з системою.

Структурно робота складається з чотирьох розділів, кожен з яких послідовно розкриває ключові аспекти розробки проєкту мобільного додатку для персоналізованого навчання у форматі мікро-льорнінгу.

Об'єктом дослідження є процес розробки проєкту інформаційної системи для мобільного навчання з використанням принципів мікро-льорнінгу та персоналізації освітнього контенту.

Предметом дослідження є методи формування вимог, моделювання архітектури та проектування мобільного додатку, що забезпечує персоналізоване подання навчального матеріалу у вигляді мікро-уроків з урахуванням потреб користувача.

Методи дослідження:

- аналітичні методи (аналіз наукових джерел, огляд сучасних технологій мобільного навчання, аналіз програм-аналогів);
- методи системного аналізу (визначення вимог, класифікація функціональних компонентів, побудова моделі предметної області);
- методи структурного та об'єктно-орієнтованого моделювання (використання UML-діаграм, побудова архітектурних схем, створення conceptual та logical models);
- методи проєктного менеджменту (SWOT-аналіз, PEST-аналіз, формування OBS та WBS, оцінювання економічної ефективності);
- прототипування та UX-дослідження (побудова wireframes, сценаріїв користувача, юзабіліті-оцінювання).

Наукова новизна дослідження полягає у формалізації підходів до управління проєктом створення мобільного додатку для мікронавчання та у

розробленні адаптивної моделі планування освітніх сценаріїв, що забезпечує персоналізацію навчального процесу на основі аналізу поведінкових даних користувачів.

Практичне значення одержаних результатів. Результати роботи мають практичне значення для створення мобільного додатку, який забезпечить користувачам доступ до персоналізованого навчання у зручному, короткоформатному форматі. Проєкт додатку може бути використаний як основа для реалізації системи, здатної адаптувати навчальний контент відповідно до індивідуальних цілей, рівня знань та навчального темпу користувача. Запропоноване рішення сприятиме підвищенню ефективності засвоєння матеріалу, залученості до освітнього процесу та гнучкості навчання в умовах обмеженого часу.

РОЗДІЛ 1. ТЕОРЕТИЧНЕ ПІДГРУНТЯ ТА ДОСЛІДЖЕННЯ ЩОДО ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1 Актуальність персоналізованого навчання та мікро-льорнінгу

Освітні технології зазнали радикальних трансформацій упродовж останніх десятиліть, що зумовлено цифровізацією, розвитком мобільних пристроїв та зміною моделей споживання інформації. Однією з ключових тенденцій є перехід від масового одноманітного навчання до індивідуалізованих освітніх траєкторій. Персоналізоване навчання передбачає адаптацію змісту, темпу та форм подання матеріалу до індивідуальних особливостей користувача: рівня попередніх знань, стилю навчання, мотивації, доступного часу тощо. Це дозволяє підвищити залученість, зменшити когнітивне навантаження та зробити освітній процес більш ефективним і гнучким.

Мікронавчання є ще одним важливим трендом сучасної освіти. Воно ґрунтується на поданні навчального матеріалу у вигляді коротких, чітко сфокусованих інформаційних блоків, які можна опрацювати за кілька хвилин. Такий формат відповідає особливостям сприйняття інформації в умовах високого темпу життя та постійної фрагментації уваги. Короткі модулі полегшують повторення та закріплення знань, знижують когнітивне навантаження та дають користувачеві відчутний ефект «малих досягнень», що позитивно впливає на мотивацію.

Особливо актуальним є поєднання персоналізованого навчання та мікронавчання у мобільному середовищі. Мобільні додатки надають широкі можливості для адаптації контенту, відстеження поведінки користувачів, організації гнучких навчальних маршрутів, використання елементів гейміфікації та інтерактивних інтерфейсів. Завдяки цьому навчання стає доступним будь-де та будь-коли, а освітній процес можна організувати в режимі «коротких сесій», інтегрованих у повсякденну активність користувача. Для покоління, орієнтованого на швидкий результат,

автономність та комфорт, такий формат є максимально природним та привабливим.

Аналіз сучасних наукових публікацій і практичних рішень показує, що персоналізоване навчання та мікро-льорнінг активно досліджуються як в академічному середовищі, так і в індустрії. Значна кількість робіт присвячена окремим аспектам: побудові адаптивних навчальних систем, розробці рекомендаційних алгоритмів, впливу мікромодулів на ефективність запам'ятовування, застосуванню гейміфікації для підвищення мотивації, використанню мобільних платформ у навчанні дорослих тощо. Існують численні мобільні додатки для вивчення мов, підготовки до тестів, професійного розвитку, що використовують мікроформат та базові механізми персоналізації (наприклад, рівневий підбір завдань, адаптацію складності чи повторення на основі помилок).

Водночас проведений аналіз показує низку обмежень існуючих рішень. Багато з них зосереджені лише на одному аспекті – наприклад, на мовному навчанні, тренувальних тестах або повторенні термінів – без побудови цілісної моделі індивідуальної освітньої траєкторії. Персоналізація часто зводиться до простого урахування рівня складності чи частоти повторень, без комплексного аналізу цілей користувача, його навчальних преференцій, доступного часу, типових патернів поведінки та динаміки прогресу. Особливою проблемою є слабка інтеграція аналітики: не всі системи надають прозорі механізми моніторингу результатів і немає достатньо гнучких інструментів для автоматизованого формування навчальних маршрутів саме у форматі мікро-льорнінгу.

Ще одним недопрацюванням, виявленим у рамках аналізу, є недостатня увага до поєднання кількох важливих компонентів в одному рішенні: персоналізованих планів навчання, мікромодулів, розгалуженої аналітики, гейміфікованих механік та зручного мобільного інтерфейсу. Більшість наявних продуктів реалізують лише окремі елементи цієї концепції, що обмежує їхню гнучкість і здатність повністю адаптуватися до потреб різних

категорій користувачів. Відсутність універсальної, модульної та масштабованої системи персоналізованого мікронавчання для мобільного середовища створює помітну нішу для подальших досліджень і розробок.

Таким чином, поєднання персоналізованого навчання та мікро-льорнінгу в межах єдиного мобільного додатку, орієнтованого на адаптивне формування навчальних маршрутів, глибоку аналітику навчальної активності та інтеграцію елементів гейміфікації, є актуальним і науково, і практично. Саме відсутність комплексних рішень такого типу у наявних розробках та публікаціях визначає наукову новизну та практичну значущість даної роботи, а також обґрунтовує доцільність її виконання.

1.2 Сучасні тренди та цифрові рішення у сфері мобільного навчання

Цифровізація освіти зумовлює активне впровадження мобільного навчання (mobile learning) – формату, що передбачає використання мобільних пристроїв (смартфонів, планшетів) для доступу до навчального контенту будь-де і будь-коли [9, 10, 11, 12].

Цей підхід підтримує концепцію «навчання протягом усього життя» (lifelong learning) і робить освітній процес більш доступним та гнучким.

До ключових освітніх трендів, що визначають розвиток мобільного навчання, належать:

- Мікро-льорнінг – подача контенту невеликими логічно завершеними блоками (до 5–7 хвилин). Такий формат сприяє кращому засвоєнню матеріалу та знижує когнітивне навантаження.
- Адаптивне навчання – технології, які підлаштовують контент і завдання під рівень знань, інтереси та прогрес користувача.
- Гейміфікація – використання ігрових механік (бали, досягнення, рейтинги) для підвищення мотивації до навчання.
- Інтерактивність – включення відео, вбудованих запитань, симуляцій та мінітестів, що сприяють залученню користувача.

- Навчання з використанням штучного інтелекту (AI-assisted learning) – автоматизовані рекомендації, перевірка знань, чат-боти-консультанти.
- Lifelong learning – підхід до навчання впродовж усього життя, орієнтований на постійне самовдосконалення.
- LXP – Learning Experience Platforms – нове покоління систем управління навчанням, що орієнтуються не лише на контент, а й на досвід користувача. Вони забезпечують більш гнучке управління траєкторією навчання та поєднують різні формати: текст, відео, ігри, соціальні елементи.

Ці тенденції формують високі вимоги до мобільних освітніх рішень.

Зокрема, сучасний додаток має забезпечувати:

- персоналізовану траєкторію навчання;
- швидкий і легкий доступ до навчального контенту;
- можливість навчатися в «мікропаузах» (під час очікування, поїздок, відпочинку тощо);
- зручний і інтуїтивний інтерфейс.

Ключові переваги мобільного навчання:

- Гнучкість – навчання можна здійснювати у зручний для користувача час і темп.
- Персоналізація – можливість адаптації контенту до індивідуальних потреб.
- Мультимедійність – використання відео, аудіо, інтерактивної графіки для покращення засвоєння матеріалу.
- Оперативність – доступ до актуальної інформації в режимі реального часу.
- Мотивація – завдяки ігровим елементам, змагальним механізмам та візуалізації прогресу.

1.3 Аналіз існуючих додатків

Аналіз програмних аналогів є важливим етапом у проектуванні нового

продукту, оскільки дозволяє виявити успішні підходи, типові помилки конкурентів та унікальні функції, які можна або запозичити, або вдосконалити. Розглянемо найбільш релевантні мобільні додатки, що реалізують окремі компоненти концепції персоналізованого мікро-льорнінгу.

1) **Duolingo** – це мобільний додаток для вивчення іноземних мов, який поєднує мікро-уроки з елементами гейміфікації. Додаток подає матеріал у вигляді коротких вправ, що охоплюють лексику, граматику, читання та аудіювання. В основі – система рівнів, балів та візуального прогресу. Алгоритм адаптує рівень складності до успішності користувача. Користувачі отримують нагороди за щоденне навчання та мають змогу повторювати матеріал згідно з інтервальним методом закріплення знань.

2) **Quizlet** – це цифрова платформа для запам'ятовування термінів, понять і визначень, яка використовує метод флеш-карток. Користувачі можуть створювати власні набори або користуватись готовими. Додаток підтримує тести, ігри, інтерактивні вправи, а також функцію «розумного навчання», яка визначає слабкі місця й повторює складні елементи. Інтерфейс орієнтований на швидке, модульне вивчення, що зручно для мікро-льорнінг

3) **EdApp** – це мобільна платформа мікро-навчання, орієнтована переважно на корпоративне середовище. Додаток дозволяє проходити короткі, інтерактивні навчальні модулі, адаптовані під конкретні ролі чи задачі працівників. Підтримується гейміфікація, візуалізація прогресу, push-нагадування та вбудована аналітика для HR-відділів. Користувачі навчаються у власному темпі, а контент адаптується під пристрій і поведінкові звички

4) **Khan Academy** – це освітня платформа, яка надає доступ до безкоштовного контенту з різних предметів у форматі коротких відеоуроків, інтерактивних вправ та тестів. Додаток включає адаптивну систему навчання: завдання автоматично підлаштовуються до результатів користувача. Система прогресу та значків стимулює регулярне навчання, а можливість перегляду тем з мобільного пристрою підтримує концепцію навчання в зручний час.

5) **Brilliant** – це мобільний додаток для навчання математики, логіки, природничих наук та інформатики через вирішення інтерактивних задач. Замість пасивного споживання інформації користувач навчається, розв'язуючи поступово ускладнені проблеми. Система адаптується до прогресу користувача, пропонуючи контент, що відповідає рівню знань. Візуальний стиль додатку підтримує зосереджене, покрокове навчання з використанням логіки та індукції.

6) **Coursera (мобільна версія)** – це навчальна платформа, яка пропонує повноформатні онлайн-курси від університетів та компаній. Через мобільний додаток користувачі можуть переглядати відеолекції, проходити тести та завантажувати матеріали. Курси здебільшого не орієнтовані на мікро-льорнінг, однак дозволяють навчатися у власному темпі. Є система рекомендацій на основі інтересів і попередньої активності.

В табл. 1.1 наведено переваги та особливості конкурентів.

Таблиця 1.1

Переваги та особливості конкурентів

Характеристика	<i>Duolingo</i>	<i>Quizlet</i>	<i>EdApp</i>	<i>Khan Academy</i>	<i>Brilliant</i>	<i>Coursera</i>
Мікро-льорнінг	+	+	+	–	+	–
Персоналізація	+/-	–	+/-	+	+	+
Гейміфікація	+	+/-	+	–	–	–
Адаптивність контенту	+	+/-	–	+	+	+
Офлайн-доступ	+	+	+	+	+/-	+/-
Основна сфера	Іноземні мови	Запам'ятовування понять	Профільне навчання	Освітні предмети	STEM напрями	Вища освіта

Проведений аналіз програмних аналогів демонструє, що кожен із розглянутих мобільних додатків реалізує окремі компоненти концепції персоналізованого мікро-льорнінгу, однак жоден із них не поєднує всі ключові елементи в єдиній, цілісній системі. Існуючі рішення успішно використовують мікромодулі (Duolingo, EdApp, Quizlet), адаптивні алгоритми (Khan Academy, Brilliant), гейміфікацію (Duolingo, EdApp), або персоналізовані рекомендації (Coursera), проте кожен із продуктів

фокусується переважно на своїй вузькій сфері та обмеженій моделі взаємодії з користувачем. Бракує комплексних рішень, що забезпечували б одночасно глибоку персоналізацію, гнучке формування навчальних маршрутів, повноцінну адаптивність, інтерактивність, аналітику та зручний мобільний формат, орієнтований на короткі навчальні сесії. Це свідчить про наявність суттєвої прогалини на ринку та підтверджує доцільність розроблення нового мобільного застосунку, який інтегрує сильні сторони існуючих платформ та долає їхні обмеження [2, 3, 4, 5].

1.4. Постановка задачі у вигляді паспорту проєкту

У результаті проведеного теоретичного аналізу було встановлено, що сучасна система освіти перебуває в активному процесі трансформації, спричиненої цифровізацією, зростаючими вимогами до гнучкості навчання та потребою в індивідуальному підході до здобувачів знань. Серед ключових тенденцій, які формують нові вимоги до освітніх рішень, варто виокремити персоналізацію навчального процесу, мікро-льорнінг як ефективну форму подачі матеріалу, гейміфікацію, а також широке використання мобільних технологій.

Аналіз освітніх платформ і мобільних додатків, що реалізують концепції персоналізованого або мікроформатного навчання, показав, що хоча багато з них частково підтримують згадані підходи, вони не забезпечують цілісної інтеграції таких компонентів, як:

- подання матеріалу у вигляді логічно завершених, коротких блоків (мікро-уроків);
- адаптація контенту до рівня підготовки, інтересів і навчального стилю користувача;
- можливість гнучкого, асинхронного навчання через мобільний пристрій;
- простота у використанні й інтуїтивно зрозумілий інтерфейс, орієнтований на користувача.

Це свідчить про наявність вакантної ніші на ринку мобільних освітніх застосунків, що дає підстави для створення нового проєкту, який об'єднає зазначені функціональні переваги в єдиному рішенні. Метою такого продукту є підвищення ефективності засвоєння знань за рахунок індивідуалізації навчального процесу, що є особливо актуальним у контексті самостійного навчання, корпоративної освіти та неформального підвищення кваліфікації.

З огляду на викладене, доцільним є формулювання наступного комплексу завдань, що підлягають вирішенню в межах роботи:

1. Провести аналіз тенденцій розвитку персоналізованого та мобільного навчання у форматі мікро-льорнінгу.
2. Дослідити функціональні можливості існуючих програмних продуктів, що реалізують схожі підходи.
3. Сформувати перелік вимог до проєктованої інформаційної системи з урахуванням потреб цільової аудиторії та результатів аналізу аналогів.
4. Розробити концептуальну модель інформаційної системи, що реалізує персоналізоване навчання у форматі мікро-льорнінгу.
5. Спроектувати архітектуру програмного забезпечення, визначити компоненти, структуру бази даних та логіку взаємодії користувача з додатком.
6. Сформулювати основи сценаріїв використання мобільного додатку та підходи до оцінки його ефективності на етапі майбутнього впровадження.

В таблиці 1.2. зображений фрагмент паспорту проєкту, повна версія паспорту проєкту зображена в додатках (див. Додаток А, таб. А.1)

Таблиця 1.2

Фрагмент паспорту проєкту

Цілі проєкту	Створення мобільного додатку для персоналізованого навчання у форматі мікро-льорнінгу. Адаптивна подача навчального контенту відповідно до індивідуальних потреб користувача. Структурування контенту у вигляді коротких мікроурокув. Збереження та аналіз прогресу користувача. Рекомендації щодо подальшого навчання на основі результатів. Мотивація через гейміфікацію (бейджі, рівні, бали). Інтуїтивно зрозумілий та зручний мобільний інтерфейс.
Зацікавлені	Замовник: Інвестор

сторони проекту	Команда проекту: Project Manager. Системний аналітик. Програмісти (Backend, Mobile) UI/UX дизайнер. Інженер з якості (QA). Системний адміністратор Технічна підтримка. Контент-менеджер. Маркетолог. Бізнес-Аналітик. Користувачі: Люди, які прагнуть вивчати новий матеріал за допомогою персоналізованого плану у форматі коротких фактів та уроків в ігровому форматі
------------------------	---

Отже, узагальнення матеріалу, викладеного у даному розділі, дозволяє стверджувати, що на ринку освітніх мобільних додатків спостерігається дефіцит рішень, які б у комплексі задовольняли потреби користувачів у персоналізованому та мікро-форматному навчанні. Існуючі програмні продукти реалізують лише окремі функції, водночас не забезпечуючи належного рівня адаптивності, зручності та індивідуалізації навчального процесу.

Запропонований у межах даної роботи проект мобільного додатку має на меті поєднати ключові переваги сучасних цифрових освітніх рішень з новими підходами до подачі контенту, персоналізації та користувацького досвіду. Завдяки цим перевагам, потенційна розробка має усі передумови для конкурентоспроможності та популярності серед цільової аудиторії мобільних освітніх платформ.

1.5. Формування наукової новизни та іновайційності

Наукова новизна даного дипломного проекту полягає у системному дослідженні та методологічному обґрунтуванні застосування сучасних технологій управління проектами в процесі створення мобільного навчального додатку, що функціонує відповідно до концепції мікро-льорнінгу. У роботі розглянуто проект створення програмного продукту не лише як технічне завдання, а як складну соціотехнічну систему, у межах якої поєднуються процеси управління, розробки, педагогічного проектування та кінцевого використання.

Уперше для даного класу освітніх мобільних застосунків обґрунтовано доцільність адаптації гнучких підходів до управління проектами з урахуванням специфіки мікро-льорнінгу, який характеризується

фрагментованістю контенту, високими вимогами до інтерактивності та необхідністю постійного оперативного оновлення навчальних матеріалів. Запропонований підхід дозволяє забезпечити узгодженість між освітніми цілями, функціональними вимогами та обмеженнями проєкту за термінами, вартістю та ресурсами.

Наукова новизна також полягає у формуванні інтегрованої моделі управління проєктом створення мобільного додатку, яка поєднує класичні процеси управління проєктами з елементами педагогічного дизайну та аналізу користувачького досвіду. У межах даної моделі враховано особливості поведінки користувачів мобільних навчальних застосунків, зокрема короткотривалі сесії навчання, необхідність підтримки мотивації та індивідуальні траєкторії засвоєння знань.

Інноваційність проєкту полягає у розробці концепції мобільного навчального додатку, у якому навчальний контент поділено на логічно завершені мікромодулі, управління створенням та розвитком яких здійснюється за допомогою сучасних технологій управління проєктами. Такий підхід забезпечує можливість гнучкої модифікації функціональності додатку, масштабування навчального контенту та адаптації продукту до змінних вимог користувачів і зовнішнього середовища.

Окрім того, інноваційність проєкту визначається орієнтацією на персоналізоване навчання, що реалізується шляхом використання механізмів оцінювання прогресу користувачів та адаптації навчальних сценаріїв. Застосування технологій управління проєктом у такому контексті дозволяє забезпечити контроль якості освітнього продукту на всіх етапах його життєвого циклу – від ініціації та планування до впровадження й подальшої підтримки.

Таким чином, наукова новизна та інноваційність проєкту полягають у комплексному поєднанні методологій управління проєктами, цифрових освітніх технологій та концепції мікро-льорнінгу, що формує науково обґрунтовану основу для створення ефективного, адаптивного та

конкурентоспроможного мобільного навчального додатку.

1.6. SWOT-аналіз та PEST-аналіз проєкту

Для комплексної оцінки доцільності та життєздатності проєкту створення мобільного додатку для навчання користувачів у форматі мікро-льорнінгу доцільно застосувати методи стратегічного аналізу, зокрема SWOT-аналіз та PEST-аналіз. Зазначені інструменти дозволяють визначити внутрішні сильні та слабкі сторони проєкту, а також проаналізувати зовнішні фактори, що впливають на його реалізацію.

SWOT-аналіз передбачає дослідження чотирьох груп факторів: сильних сторін (Strengths), слабких сторін (Weaknesses), можливостей (Opportunities) та загроз (Threats).

Сильні сторони (Strengths):

- використання формату мікро-льорнінгу, що відповідає сучасним освітнім потребам користувачів;
- орієнтація на мобільну платформу, яка забезпечує доступність навчання у будь-який час;
- можливість персоналізації навчального контенту;
- застосування сучасних технологій управління проєктом, що сприяють підвищенню якості продукту;
- потенціал масштабування функціональності додатку.

Слабкі сторони (Weaknesses):

- залежність від стабільного доступу до інтернету та мобільних пристроїв;
- обмеження обсягу навчального матеріалу через формат мікро-льорнінгу;
- необхідність постійного оновлення та актуалізації контенту;
- потреба у залученні фахівців з педагогічного дизайну;
- початкові витрати на розробку та тестування програмного забезпечення.

Можливості (Opportunities):

- зростання попиту на дистанційне та мобільне навчання;
- інтеграція з існуючими освітніми платформами та сервісами;
- розширення цільової аудиторії за рахунок різних вікових та професійних груп;
- впровадження гейміфікації та адаптивного навчання;
- можливість комерціалізації продукту через підписки або корпоративні рішення.

Загрози (Threats):

- високий рівень конкуренції на ринку освітніх мобільних додатків;
- швидкі зміни технологій мобільної розробки;
- ризики інформаційної безпеки та захисту персональних даних;
- зниження мотивації користувачів до довготривалого використання додатку
- правові обмеження у сфері використання освітнього контенту.

Таким чином, SWOT-аналіз показує, що за умови ефективного управління ризиками та використання наявних можливостей проєкт має потенціал успішної реалізації.

PEST-аналіз використовується для дослідження макросередовища проєкту та охоплює політичні (Political), економічні (Economic), соціальні (Social) та технологічні (Technological) фактори.

Політичні фактори (Political):

- державна підтримка цифровізації освіти;
- регуляторні вимоги щодо захисту персональних даних користувачів;
- освітні стандарти та нормативно-правові акти у сфері електронного навчання;
- можливі зміни у законодавстві щодо ІТ-послуг.

Економічні фактори (Economic):

- рівень платоспроможності цільової аудиторії;

- витрати на розробку, впровадження та підтримку мобільного додатку;
- розвиток ринку цифрових освітніх послуг;
- можливість залучення інвестицій або грантового фінансування.

Соціальні фактори (Social):

- зростання потреби у безперервному навчанні;
- популяризація самоосвіти та онлайн-навчання;
- зміна стилю споживання інформації у бік коротких форматів;
- рівень цифрової грамотності населення.

Технологічні фактори (Technological):

- розвиток мобільних технологій та платформ;
- доступність інструментів кросплатформної розробки;
- впровадження штучного інтелекту для персоналізації навчання;
- підвищені вимоги до кібербезпеки програмних продуктів.

За результатами проведеного PEST-аналізу встановлено, що зовнішнє макросередовище загалом є сприятливим для реалізації проєкту створення додатку для навчання користувачів у форматі мікро-льорнінгу. Політичні та нормативно-правові чинники, зокрема курс на цифровізацію освіти та розвиток електронних освітніх сервісів, створюють позитивні передумови для впровадження інноваційних навчальних рішень. Економічні фактори свідчать про наявність ринкового попиту на цифрові освітні продукти, а також про потенційну можливість диверсифікації джерел фінансування проєкту.

Соціальні тенденції, пов'язані зі зростанням потреби у безперервному навчанні, мобільності та використанні коротких форматів подачі інформації, додатково підсилюють актуальність концепції мікро-льорнінгу, роблячи її однією з провідних моделей сучасної цифрової освіти. Зміни у поведінці користувачів, зростання попиту на персоналізовані освітні траєкторії та очікування швидкого доступу до знань у будь-який час і в будь-якому місці зумовлюють необхідність створення зручних, адаптивних та високоінтерактивних мобільних рішень.

Паралельно з цим динамічний розвиток інформаційних технологій, мобільних операційних систем, фреймворків для кросплатформеної розробки та інструментів аналітики відкриває широкі можливості для реалізації складних алгоритмів персоналізації, оптимізації навчальних процесів та забезпечення високого рівня користувацького досвіду. Сукупність цих соціальних і технологічних чинників створює сприятливе середовище для успішної розробки програмного продукту, здатного відповідати актуальним вимогам ринку освітніх послуг.

Узагальнюючи викладене, можна стверджувати, що дослідження соціально-технологічного контексту проекту дає змогу чітко визначити його цінність, потенціал та обмеження. Ці завдання створюють основу для подальших етапів роботи – аналізу вимог, розроблення архітектури, вибору технологічного стеку та практичної реалізації мобільного додатку. Їхнє чітке формулювання забезпечує структурованість дослідження, послідовність розробки та відповідність кінцевого продукту.

РОЗДІЛ 2. АНАЛІЗ, МОДЕЛЮВАННЯ ТА СТРУКТУРУВАННЯ ВИМОГ ДО СИСТЕМИ

2.1 Ідентифікація потреб та характеристик цільової аудиторії

На початковому етапі розробки будь-якого інформаційного продукту одним із найважливіших завдань є ідентифікація цільової аудиторії. Саме розуміння її особливостей, очікувань, освітніх потреб та поведінкових характеристик дозволяє сформулювати обґрунтовані вимоги до майбутньої системи, а також забезпечити її релевантність, зручність і ефективність використання.

Цільова аудиторія проєктованого мобільного додатку – це користувачі, які мають потребу у швидкому, адаптивному та персоналізованому навчанні в зручному форматі.

До цієї категорії належать переважно:

- студенти вищих навчальних закладів;
- учні старших класів закладів загальної середньої освіти;
- молоді спеціалісти, які потребують підвищення кваліфікації;
- працівники корпоративного сектору, зацікавлені у швидкому опануванні навичок;
- усі охочі до саморозвитку у вільний від основної діяльності час (Lifelong learners).

Соціально-демографічні характеристики цільової аудиторії:

- Вік: переважно 16–35 років;
- Рівень освіти: середня, середньо-спеціальна, вища або процес її здобуття;
- Цифрова грамотність: середній або високий рівень володіння мобільними пристроями, додатками;
- Часовий ресурс: обмежений, що зумовлює потребу в стислому, модульному навчанні.

Мотиваційні чинники:

- бажання опанувати нову навичку або предмет;
- потреба у підготовці до тестів, іспитів або співбесід;
- прагнення до самореалізації та саморозвитку;
- професійна необхідність у постійному оновленні знань (особливо в ІТ, фінансах, медицині).

Поведінкові патерни користувача:

- фрагментарне використання додатка протягом дня;
- очікування миттєвого результату після завершення модуля;
- бажання мати контроль над процесом навчання (вибір тем, темпу, тривалості);
- схильність порівнювати власні результати з іншими користувачами.

Прогнозовані бар'єри у використанні додатку:

- інформаційне перевантаження (у разі надмірно об'ємного контенту);
- надмірно складна або нецікава подача матеріалу;
- технічні труднощі (повільна робота, баги, незрозумілий інтерфейс);
- відсутність системної мотивації.

На рис. 2.1 зображено причини та наслідки в необхідності створення даного продуктового рішення

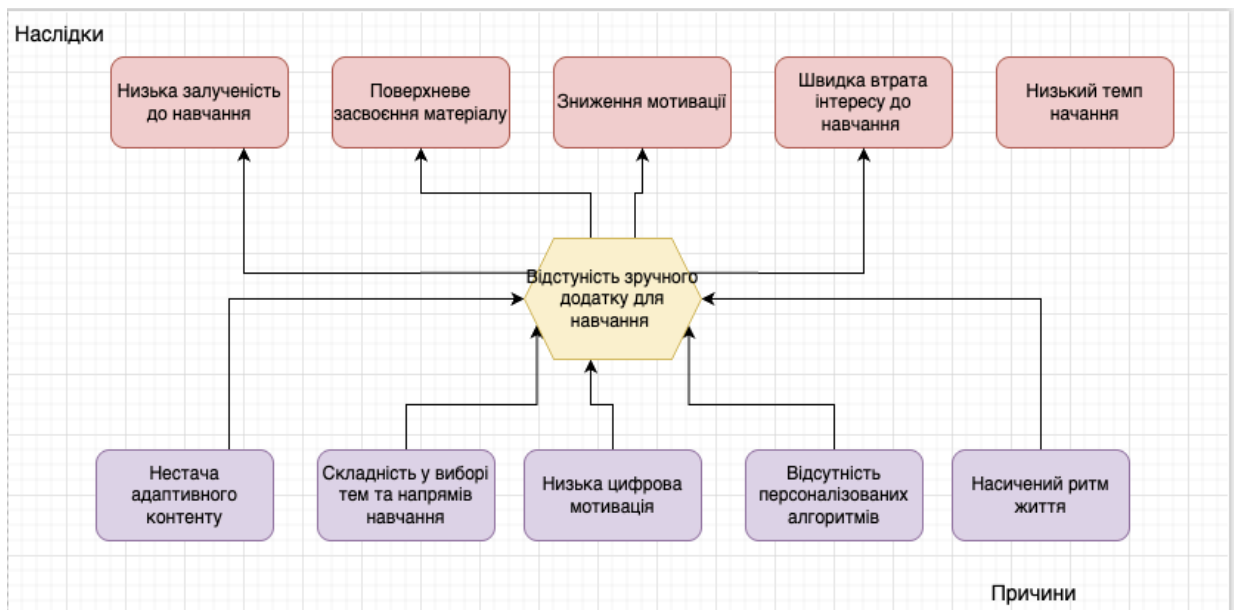


Рис. 2.1. Дерево причин та наслідків

2.2 Визначення функціональних та нефункціональних вимог до системи

Процес розробки будь-якого програмного продукту передбачає послідовне проходження кількох ключових етапів, одним із яких є формалізація вимог до інформаційної системи. Цей етап має принципове значення, оскільки саме від якості сформульованих вимог залежить успішність подальшого проектування, реалізації, впровадження та експлуатації програмного забезпечення. У випадку розробки мобільного додатку для персоналізованого навчання у форматі мікро-льорнінгу, визначення вимог дозволяє забезпечити повну відповідність продукту освітнім цілям, поведінковим особливостям користувачів та специфіці подачі навчального контенту [9-10].

Вимоги до системи прийнято класифікувати на дві великі категорії: функціональні (що описують, які дії виконує система з точки зору користувача) та нефункціональні (що стосуються якості реалізації цих функцій – зручності, надійності, продуктивності тощо). Такий поділ дозволяє комплексно охопити не лише поведінкову логіку системи, але й загальні очікування щодо її якості та ефективності використання.

У межах даного підрозділу буде здійснено систематизацію функціональних можливостей, необхідних для реалізації концепції персоналізованого мобільного навчання, а також окреслено нефункціональні вимоги, що забезпечують комфорт, продуктивність і безпечність роботи з додатком.

Функціональні вимоги

Функціональні вимоги визначають основну поведінку системи – тобто набір дій, які вона має виконувати у відповідь на взаємодію користувача або інші тригери. Для мобільного додатку персоналізованого навчання у форматі мікро-льорнінгу визначено такі ключові функції:

1. Реєстрація та автентифікація користувачів. Система має підтримувати створення облікового запису з мінімальним набором

персональних даних. Реалізація входу може відбуватись через email, номер телефону або інтеграцію з обліковими записами Google, Apple, Facebook тощо.

2. Формування профілю користувача. Користувач має можливість вказати свій рівень підготовки, цілі навчання, бажану тематику. Ці дані використовуються для персоналізації навчального шляху.

3. Доступ до структурованого навчального контенту. Контент має бути поділений на мікроуроки – логічно завершені модулі тривалістю 3–10 хвилин, що можуть включати відео, текст, ілюстрації та завдання.

4. Адаптація навчального процесу. Система повинна динамічно підлаштовувати рекомендований контент відповідно до прогресу користувача, його помилок та темпу виконання завдань.

5. Інтерактивність та самоперевірка. Кожен навчальний модуль має супроводжуватись тестовими завданнями або вправами для самоперевірки знань. Результати впливають на подальші рекомендації.

6. Відстеження прогресу навчання. Передбачено візуалізацію проходження курсів, кількості пройдених уроків, відсотка завершення теми, рівня досягнень тощо.

7. Система гейміфікації. Заохочення користувача до регулярного навчання шляхом додавання системи балів, бейджів, рівнів, таблиць лідерів тощо.

8. Push-сповіщення та нагадування. Автоматичні повідомлення щодо нових уроків, незавершених завдань, рекомендованих повторень або мотиваційних нагадувань.

9. Можливість офлайн-доступу (опційно). Користувач повинен мати можливість попередньо завантажити навчальні модулі для перегляду без доступу до Інтернету.

Нефункціональні вимоги

Нефункціональні вимоги визначають якісні характеристики системи, що безпосередньо впливають на зручність, ефективність та довіру користувачів

до продукту.

1. Юзабіліті (зручність користування). Інтерфейс має бути інтуїтивно зрозумілим, з мінімалістичним дизайном, орієнтованим на мобільні пристрої. Важливо забезпечити швидке освоєння додатку без необхідності проходження додаткових інструкцій.

2. Продуктивність. Всі дії користувача повинні виконуватись швидко, без помітних затримок. Завантаження навчального модуля не повинно перевищувати 2–3 секунди.

3. Надійність та стабільність роботи. Додаток має працювати без збоїв, втрати даних або зупинок. Передбачається обробка виняткових ситуацій і збереження проміжного прогресу.

4. Безпека та конфіденційність. Персональні дані користувача повинні зберігатися захищено. Передбачено використання HTTPS, шифрування, автентифікації та політик обмеження доступу до даних.

5. Кросплатформеність. Підтримка роботи на основних мобільних операційних системах (Android, iOS). Додаток має бути однаково доступний та повнофункціональний на різних пристроях.

6. Масштабованість. Система має бути здатною до обслуговування великої кількості користувачів без зниження продуктивності або стабільності.

7. Можливість локалізації. Підтримка кількох мов інтерфейсу для адаптації під різні ринки або регіони.

8. Інтеграція з іншими сервісами. Потенційна можливість підключення зовнішніх платформ (Google Classroom, LMS, календарі, сповіщення через месенджер).

Формалізація цілей і завдань інформаційної системи

Формалізація цілей і завдань інформаційної системи є ключовим етапом у процесі проектування програмного продукту, оскільки вона забезпечує концептуальне обґрунтування доцільності розробки та визначає вектор подальшої технічної реалізації. Чітке визначення цілей дозволяє уникнути розбіжностей між очікуваннями користувачів і функціональними

можливостями системи, а також слугує основою для оцінки ефективності майбутнього продукту [6].

У контексті даної роботи інформаційна система реалізується у вигляді мобільного додатку, функціональне призначення якого полягає в організації процесу персоналізованого навчання у форматі мікро-льорнінгу.

Головна ціль інформаційної системи: створити інтерактивний, адаптивний та доступний мобільний додаток, що дозволяє користувачам ефективно опановувати навчальний матеріал у форматі коротких мікроуроків, з урахуванням їхнього рівня знань, індивідуальних освітніх цілей і стилю навчання.

Підцілі інформаційної системи:

- Забезпечення персоналізації навчального процесу – адаптація контенту до потреб і особливостей кожного користувача на основі початкового тестування, прогресу та інтересів.
- Організація подачі навчального матеріалу у форматі мікро-льорнінгу – поділ освітнього контенту на короткі, логічно завершені модулі, які легко інтегруються в повсякденний ритм користувача.
- Мотивація користувача до регулярного навчання – впровадження гейміфікованих елементів: досягнення, бейджі, рейтинги, щоденні цілі тощо.
- Забезпечення зручності доступу до освітнього контенту через мобільний пристрій
- Швидкий доступ до уроків, оптимізація під різні розміри екранів, можливість перегляду офлайн (опційно).

Моніторинг та аналіз прогресу навчання – візуалізація результатів, відображення темпів опрацювання матеріалів, надання рекомендацій для повторення чи поглиблення тем.

Підтримка інтерактивної взаємодії з користувачем – надання зворотного зв'язку після тестування, повідомлень, підказок та корисної статистики.

Основні завдання інформаційної системи:

1. Розробити структуру даних, що дозволяє зберігати користувацький профіль, прогрес, результати тестування та історію навчання;
2. Забезпечити можливість керування навчальним контентом: додавання, оновлення, категоризація мікроуроків;
3. Реалізувати механізм реєстрації та автентифікації користувачів;
4. Розробити інтерфейс, який забезпечує інтуїтивно зрозумілу взаємодію з додатком;
5. Запровадити систему адаптивного навчання на основі відповідей, швидкості виконання завдань і самоперевірки;
6. Впровадити модуль push-повідомлень та гейміфікації;
7. Забезпечити захист персональних даних відповідно до вимог інформаційної безпеки;

Отже, узагальнюючи всі цілі функціонального навантаження додатку приведено дерево цілей та підцілей на рис. 2.2.



Рис. 2.2. Дерево цілей та підцілей

Очікуваний результат реалізації системи: у результаті розробки проекту буде створено функціональну модель мобільного додатку, що забезпечує персоналізоване навчання у зручному, мобільному форматі. Система буде здатна адаптуватися до навчального стилю користувача, формувати рекомендації, підтримувати зацікавлення та сприяти підвищенню ефективності засвоєння матеріал.

2.3. Побудова концептуальної моделі інформаційної системи

Концептуальна модель інформаційної системи є інструментом абстрактного опису структури, функцій і взаємозв'язків її основних елементів. Вона дозволяє уявити систему як організовану сукупність взаємопов'язаних підсистем і процесів, які разом забезпечують досягнення поставленої мети – у цьому випадку, реалізацію персоналізованого освітнього середовища через мобільний додаток.

Інформаційна система розглядається як складне функціональне утворення, компоненти якого взаємодіють між собою, створюючи цілісність, що перевищує суму її частин. Взаємодія елементів може бути як лінійною (послідовною), так і ієрархічною (наприклад, адміністративна структура управління навчальним процесом) або мережевою (взаємозв'язок між модулями контенту та користувачами).

Основна система:

Інформаційна система мобільного навчання у форматі мікро-льорнінгу

Надсистема:

- Система вищої освіти, освітні онлайн-платформи, цифрові сервіси для самонавчання.
- Освітні політики, стандарти цифрової педагогіки.
- Системи сертифікації (наприклад, Coursera, Udemy).

Підсистеми:

- Мобільний застосунок – основна платформа для навчання
- Модулі навчального контенту – освітні мікроуроки, тести, пояснення
- Аналітичний модуль – персоналізація на основі прогресу
- Функціональні модулі мобільного застосунку
- Апаратна частина та обладнання
- Бек-офіс та Технічна підтримка – забезпечення функціонування та реагування на запити.

На рис. 2.3 наведено концептуальну модель ІС.

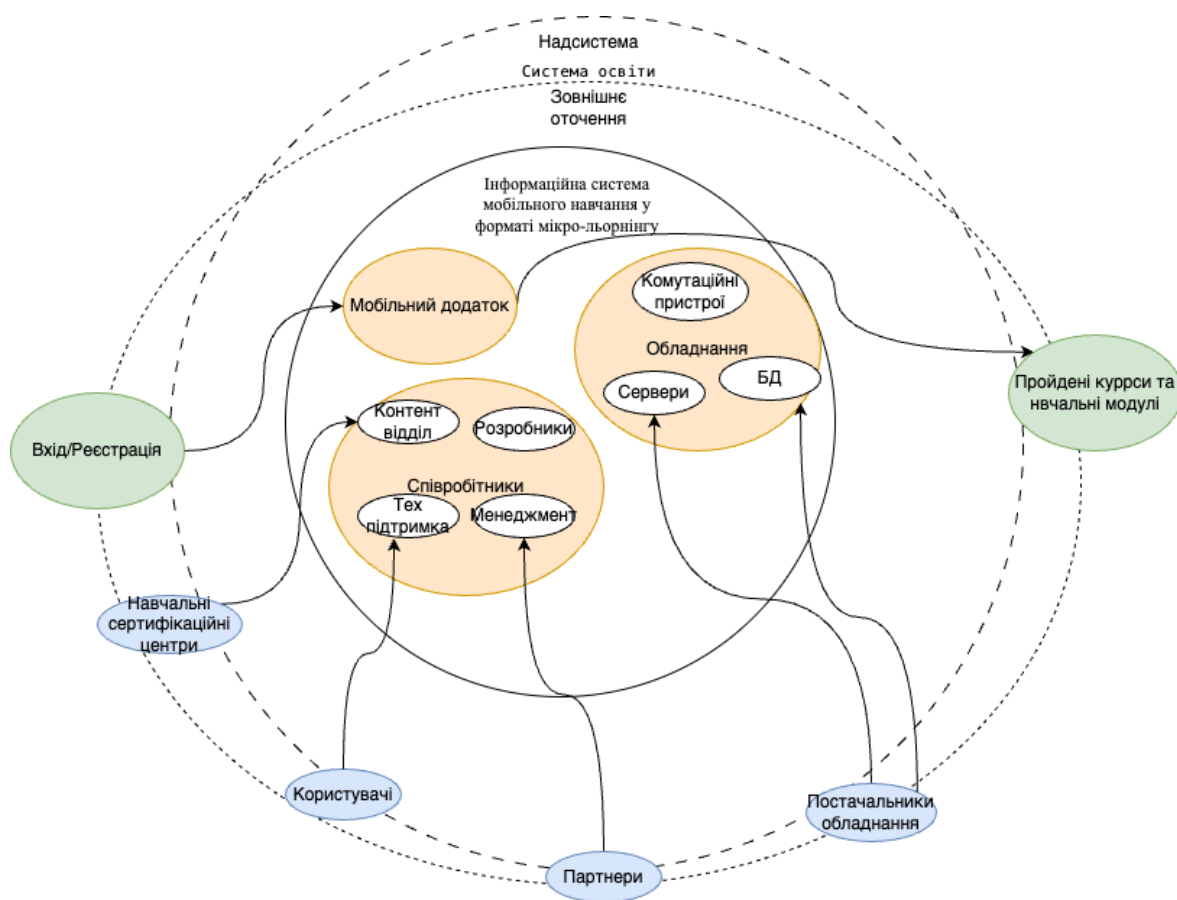


Рис. 2.3. Концептуальна модель ІС

У межах функціонування інформаційної системи персоналізованого мобільного навчання взаємодіють кілька ключових суб'єктів і зовнішніх факторів, кожен з яких відіграє специфічну роль у забезпеченні ефективності, стабільності та практичної цінності продукту.

Перш за все, користувачі – це основні споживачі системи. До цієї категорії належать учні, студенти, молоді спеціалісти та працівники, які прагнуть до самонавчання або підвищення кваліфікації. Вони взаємодіють із навчальним контентом, проходять мікроуроки та тести, отримують рекомендації, фіксують прогрес і надають зворотний зв'язок, що в подальшому сприяє вдосконаленню системи.

Освітні установи та викладачі можуть виступати як автори або куратори навчального контенту. Вони здатні адаптувати навчальні модулі до конкретних програм, аналізувати успішність студентів на основі статистики, а також інтегрувати мобільний додаток у традиційний освітній процес як додатковий або основний інструмент.

Розробники та команда технічної підтримки відіграють ключову роль у забезпеченні працездатності, оновлюваності та адаптивності системи. Вони відповідають за реалізацію функціоналу, тестування програмного коду, усунення помилок, оновлення безпеки та розширення можливостей відповідно до зворотного зв'язку користувачів.

У розгортанні продукту важливу роль відіграють маркетингові партнери, які забезпечують поширення мобільного додатку серед цільової аудиторії, формування попиту, брендингову підтримку та популяризацію серед нових користувачів.

Інвестори та бізнес-замовники забезпечують фінансову підтримку розробки, масштабування та впровадження системи. Вони також можуть формувати стратегічні вимоги до функціональності, визначати ключові показники ефективності (KPI) та пріоритети у розвитку продукту.

Нарешті, цифрові платформи (зокрема, Google Play та App Store) слугують технологічним середовищем для розміщення, розповсюдження та технічної модерації мобільного додатку. Вони накладають вимоги до безпеки, стабільності, швидкодії, UX/UI-рішень, а також контролюють відповідність додатку чинним стандартам публікації.

Таким чином, інформаційна система є відкритою багаторівневою структурою, у якій успішність функціонування залежить від скоординованої взаємодії усіх внутрішніх та зовнішніх суб'єктів.

Підпроцеси в межах системи:

1. Реєстрація / вхід до системи – Користувач створює профіль, вводить освітні цілі, проходить вступний тест (якщо передбачено)
2. Головна панель / інтерфейс навігації – Користувач обирає

тематичний модуль, отримує доступ до коротких блоків навчального матеріалу

3. Перегляд та проходження мікроуроків – Мікроуроки тривають 2–5 хвилин, містять текст, відео або інтерактивний елемент
4. Виконання тестових завдань – Після кожного модуля доступне коротке самотестування
5. Аналіз прогресу та надання рекомендацій – Візуалізація результатів, динаміка проходження, формування персоналізованих порад
6. Надання гейміфікованих винагород – Нарахування бейджів за активність, streak-серії, досягнення нових рівнів
7. Надсилання push-сповіщень – Нагадування про непереглянуті уроки, рекомендації щодо повторення пройденого матеріалу
8. Підтримка користувачів – У разі технічних труднощів або помилок користувач звертається до служби підтримки

Взаємодія системи з середовищем:

- Освітнє середовище впливає на зміст навчального матеріалу
- Ринок EdTech визначає очікування щодо функцій, доступності, адаптивності
- Користувач формує поведінкові патерни, на які система реагує адаптацією контенту
- Відгуки користувачів ззовні впливають на розширення функціоналу

Результат взаємодії з системою:

В результаті проходження повного циклу навчання користувач отримує:

- персоналізовану траєкторію навчання;
- знання, адаптовані до його рівня;
- візуалізацію прогресу та зворотний зв'язок;
- мотиваційні елементи (бейджі, нагадування, рекомендації);
- можливість навчатися в будь-якому місці у зручний час.

2.4 Проєктування структури бази даних мобільного додатку

Проєктування структури бази даних є одним з ключових етапів у

розробці будь-якої інформаційної системи, зокрема мобільного додатку для персоналізованого навчання. База даних (БД) слугує основою для зберігання, обробки та взаємодії з навчальним контентом, користувацькими даними, результатами тестування та індивідуальними налаштуваннями. Від її структури залежить ефективність функціонування системи, швидкість доступу до інформації, масштабованість та зручність підтримки продукту[33].

Основна мета проектування бази даних – створити логічну і фізичну модель, яка забезпечуватиме цілісність, цілеспрямованість і надійність управління освітніми даними.

Концептуальні основи моделювання

На концептуальному рівні база даних орієнтована на зберігання ключових об'єктів системи, таких як:

- Користувачі – особи, які проходять навчання;
- Навчальні модулі (уроки) – одиниці мікро-льорнінгу;
- Тести – інструменти самоперевірки;
- Прогрес навчання – хронологія і результати проходження матеріалу;
- Досягнення – гейміфіковані елементи мотивації;
- Налаштування – параметри персоналізації інтерфейсу та поведінки додатку.

Ці сутності мають чітко виражені зв'язки між собою, що дозволяє побудувати нормалізовану реляційну модель даних. На рис. 2.4. зображено концептуальну модель БД.

Логічна модель баз даних

Логічна модель бази даних – це формалізоване представлення структур, які будуть реалізовані у фізичній СУБД, але без прив'язки до конкретної реалізації чи синтаксису SQL. Вона описує сутності (таблиці), атрибути (поля) та взаємозв'язки між ними, необхідні для функціонування мобільного додатку у сфері персоналізованого навчання у форматі мікро-льорнінгу.

На цьому рівні проектування важливо уникнути надмірності (редундантності), забезпечити цілісність даних і підготувати модель до подальшої нормалізації. Первинні ключі (Primary Keys) гарантують унікальність кожного запису у таблиці та використовуються для ідентифікації об'єктів. Наприклад, `user_id` у таблиці `Users` є первинним ключем, що ідентифікує кожного користувача.

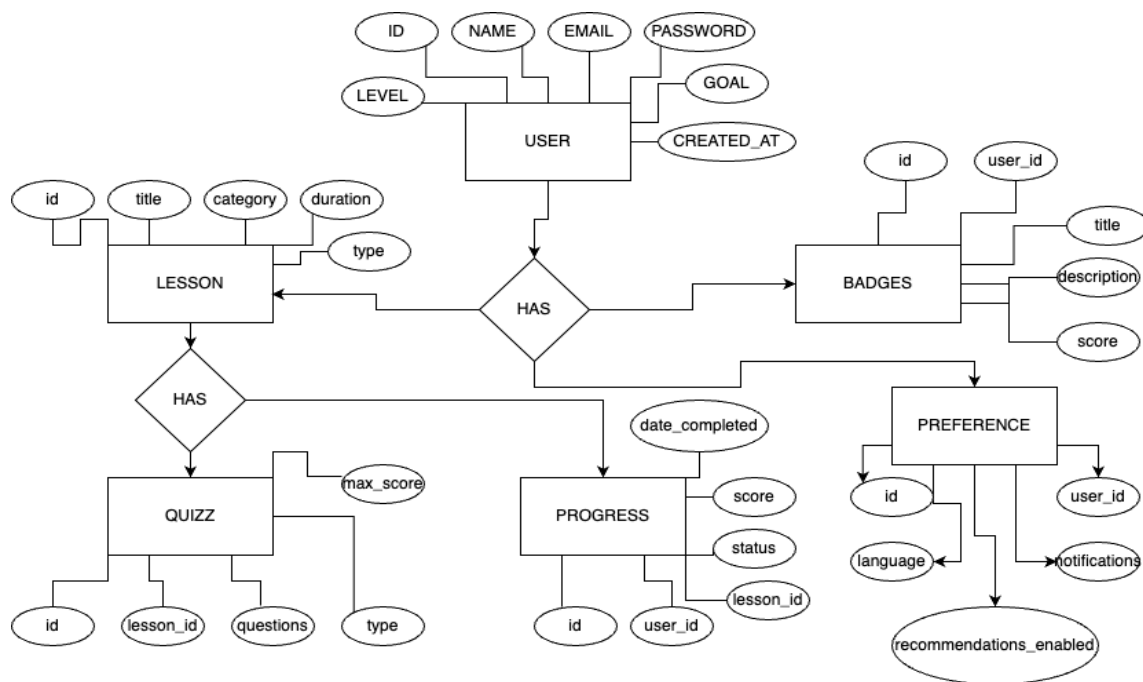


Рис. 2.4. Концептуальна модель бази даних

Зовнішні ключі (Foreign Keys) створюють зв'язки між таблицями, наприклад, зв'язок `lesson_id` у таблиці `Quizzes` із `lesson_id` у `Lessons`. Вони гарантують, що будь-який запис у дочірній таблиці відповідає наявному запису в батьківській, і тим самим забезпечують цілісність зв'язків у базі даних. Візуальне представлення логічної моделі БД зображено на рис. 2.5.

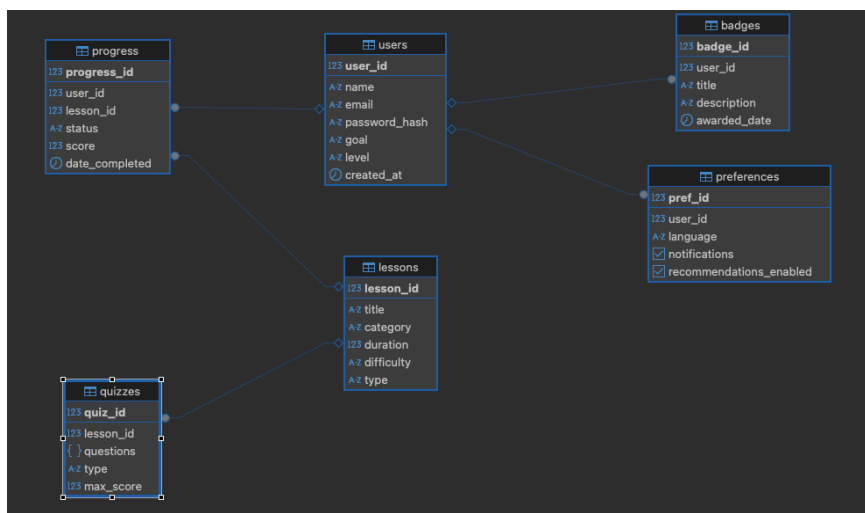


Рис. 2.5. Логічна модель бази даних

Фізична модель баз даних

Фізична модель бази даних є найнижчим рівнем абстракції в процесі проєктування інформаційної системи. Вона безпосередньо пов'язана з реалізацією логічної структури бази в обраній системі управління базами даних (СУБД) і враховує специфіку зберігання даних, типи даних, індексацію, обмеження цілісності, продуктивність, захист і масштабованість. Деталі реалізації представлені у фрагменті програмного коду (див. Додаток Б, рис. Б.1).

У фізичній моделі визначається синтаксис створення таблиць, встановлюються типи даних для кожного поля, реалізуються первинні та зовнішні ключі, а також правила поведінки при оновленні або видаленні пов'язаних записів.

1. Lessons:
 - a. **Primary key** – LessonId.
 - b. Містить інформацію про мікроуроки: назва, категорія, тривалість, складність, тип.
2. Quizzes:
 - a. **Primary key** – QuizId.
 - b. **Foreign key** – LessonId → Lessons(LessonId).
 - c. Містить питання та тип тестування, прив'язане до уроку.
3. Progress:
 - a. **Primary key** – ProgressId.
 - b. **Foreign key** – UserId → Users(UserId).

- c. **Foreign key** – LessonId → Lessons(LessonId).
- d. Фіксує результат проходження уроків (бал, статус, дата завершення).
- 4. Preferences:
 - a. **Primary key** – PrefId.
 - b. **Foreign key** – UserId → Users(UserId).
 - c. Персоналізовані налаштування користувача: мова, сповіщення, рекомендації.

Цей список логічно структурує зв'язки між основними компонентами інформаційної системи та чітко визначає, як зберігаються та пов'язуються між собою навчальні, поведінкові та персоналізовані дані користувача. На рис. 2.6 приведено вихідний файл коду із створенням таблиць БД.

```

● CREATE TABLE Users (
  user_id SERIAL PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100) UNIQUE,
  password_hash TEXT,
  goal TEXT,
  level VARCHAR(50),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

● CREATE TABLE Lessons (
  lesson_id SERIAL PRIMARY KEY,
  title VARCHAR(150),
  category VARCHAR(100),
  duration INT, -- у хвиликах
  difficulty VARCHAR(50),
  type VARCHAR(50)
);

● CREATE TABLE Quizzes (
  quiz_id SERIAL PRIMARY KEY,
  lesson_id INT REFERENCES Lessons(lesson_id),
  questions JSONB,
  type VARCHAR(50),
  max_score INT
);

● CREATE TABLE Progress (
  progress_id SERIAL PRIMARY KEY,
  user_id INT REFERENCES Users(user_id),
  lesson_id INT REFERENCES Lessons(lesson_id),
  status VARCHAR(50),
  score INT,
  date_completed DATE
);

● CREATE TABLE Badges (
  badge_id SERIAL PRIMARY KEY,
  user_id INT REFERENCES Users(user_id),
  title VARCHAR(100),
  description TEXT,
  awarded_date DATE
);

```

Рис. 2.6. Код зі створення таблиць БД

Таким чином, у межах цього розділу було проведено всебічний аналіз вимог до інформаційної системи мобільного навчання у форматі мікро-льорнінгу. Сформульовано передумови та мету її створення,

побудовано діаграму причинно-наслідкових зв'язків, що обґрунтовує актуальність і функціональну доцільність системи. Також розроблено концептуальну модель інформаційної системи, яка відображає її структурну організацію. Особливу увагу приділено проектуванню бази даних: було сформовано концептуальну, логічну та фізичну моделі, а також надано SQL-структури таблиць, необхідні для її реалізації у практичному середовищі.

2.5 Формування користувацьких ролей та сценаріїв поведінки

Ефективність функціонування мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу значною мірою залежить від коректного визначення ролей користувачів та моделювання їхньої взаємодії з інформаційною системою. Формування користувацьких ролей дозволяє структурувати функціональні можливості системи, забезпечити розмежування доступу до ресурсів та підвищити зручність користування додатком [34].

У межах досліджуваного проєкту передбачено виділення кількох основних користувацьких ролей, кожна з яких виконує визначені функції та взаємодіє з системою відповідно до своїх потреб та повноважень.

Кінцевий користувач (Learner) є основною роллю системи та безпосередньо споживачем навчального контенту. До основних функцій даної ролі належать реєстрація та автентифікація в системі, вибір навчальних тем, проходження мікронавчальних модулів, виконання інтерактивних завдань і тестів, а також перегляд результатів власного прогресу. Поведінка користувача характеризується короткими, але регулярними сесіями взаємодії з додатком, що відповідає концепції мікро-льорнінгу.

Контент-менеджер (Content Manager) відповідає за створення, редагування та актуалізацію навчального контенту. До його функцій входить формування мікромодулів, структурування навчальних матеріалів, додавання

тестових завдань та налаштування логіки подання інформації. Дана роль забезпечує відповідність контенту освітнім цілям та стандартам якості.

Адміністратор системи (Administrator) здійснює управління користувачами, налаштування параметрів системи та контроль коректної роботи додатку. Основними завданнями адміністратора є управління обліковими записами, контроль доступу, моніторинг активності користувачів та забезпечення інформаційної безпеки системи.

На основі визначених користувацьких ролей було змодельовано основні сценарії поведінки, які відображають типові послідовності дій користувачів у процесі роботи з мобільним додатком. Ключовими сценаріями є сценарій первинного входу користувача до системи, сценарій вибору навчального курсу, сценарій проходження мікронавчального модуля, сценарій оцінювання результатів навчання та сценарій адміністрування системи.

Кожен сценарій описує послідовність взаємодії між користувачем та системою, включаючи ініціацію дій, обробку запитів і відображення результатів. Моделювання сценаріїв поведінки забезпечує цілісне уявлення про функціонування системи та слугує основою для подальшого проектування архітектури програмного забезпечення та розробки інтерфейсу користувача.

Отже, формування користувацьких ролей та сценаріїв поведінки дозволяє підвищити рівень персоналізації навчального процесу, оптимізувати структуру функціональних можливостей додатку та забезпечити ефективну взаємодію користувачів з системою відповідно до концепції мікро-льорнінгу.

Для формалізації взаємодії користувачів із мобільним додатком персоналізованого навчання у форматі мікро-льорнінгу було розроблено набір основних сценаріїв використання (Use Cases). Use Case-аналіз дозволяє чітко визначити функціональні межі системи, сформулювати вимоги до її поведінки та забезпечити орієнтацію процесу проектування на потреби кінцевих користувачів.

Use Case 1. Реєстрація та автентифікація користувача

Мета: Отримання доступу до функціональних можливостей системи

Сценарій розпочинається з ініціації користувачем процесу реєстрації або входу до системи. Користувач вводить ідентифікаційні дані, після чого система здійснює перевірку коректності введеної інформації та автентифікує користувача. У разі успішної автентифікації відбувається перехід до головного інтерфейсу додатку. Альтернативні потоки включають обробку помилок введення та відновлення доступу до облікового запису.

Use Case 2. Формування персоналізованого навчального маршруту

Мета: Адаптація навчального контенту до потреб користувача

Після первинного входу користувач проходить коротке опитування або діагностичне тестування. На основі отриманих даних система формує індивідуальний навчальний маршрут, що включає набір мікромодулів, рекомендованих для вивчення. Сценарій передбачає динамічну зміну маршруту залежно від результатів навчання та активності користувача.

Use Case 3. Проходження модуля

Мета: Засвоєння навчального матеріалу

Користувач обирає навчальний мікромодуль та переглядає відповідний навчальний контент, який може містити текстову, графічну або мультимедійну інформацію. Після ознайомлення з матеріалом користувач виконує коротке інтерактивне завдання або тест. Система фіксує результат виконання та зберігає його у базі даних.

Use Case 4. Оцінювання результатів та відстеження прогресу

Мета: Контроль власних навчальних досягнень

Сценарій передбачає перегляд користувачем статистики проходження навчальних модулів, результатів тестування та загального прогресу. Система візуалізує дані у вигляді графіків або індикаторів та надає рекомендації щодо подальшого навчання.

Use Case 5. Управління навчальним контентом

Мета: Забезпечення актуальності навчального матеріалу

Контент-менеджер здійснює створення, редагування та видалення

навчальних мікромодулів. Сценарій включає перевірку коректності введених даних та налаштування параметрів доступу до контенту.

Use Case 6. Адміністрування системи

Мета: Забезпечення коректної роботи системи

Адміністратор здійснює управління користувачами, контроль доступу, аналіз системних журналів та налаштування параметрів безпеки. У разі виникнення помилок адміністратор має можливість виконувати коригувальні дії для відновлення стабільної роботи системи.

2.6 Аналіз вимог до безпеки та захисту персональних даних

У процесі розробки мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу однією з ключових умов його успішного функціонування є забезпечення належного рівня інформаційної безпеки та захисту персональних даних користувачів. Оскільки система передбачає обробку особистої інформації, результатів навчальної діяльності та даних про поведінку користувачів, питання безпеки набувають першочергового значення.

Аналіз вимог до безпеки системи розпочинається з визначення основних категорій даних, що зберігаються та обробляються в межах мобільного додатку. До них належать ідентифікаційні дані користувачів (облікові записи, контактна інформація), навчальні результати, історія взаємодії з контентом, а також технічні дані, що використовуються для забезпечення коректної роботи системи. Втрата або несанкціонований доступ до такої інформації може призвести до порушення конфіденційності та зниження довіри користувачів до додатку.

З метою мінімізації зазначених ризиків у системі повинні бути реалізовані механізми автентифікації та авторизації користувачів. Логіку автентифікації користувача подано у фрагменті програмного коду (див. Додаток Б, рис. Б.2). Автентифікація забезпечує перевірку особи користувача під час входу до системи, тоді як авторизація визначає рівень доступу до

функціональних можливостей відповідно до ролі користувача. Такий підхід дозволяє обмежити доступ до критичних ресурсів системи та забезпечити їх використання виключно уповноваженими особами.

Важливим аспектом захисту інформації є забезпечення цілісності та конфіденційності даних під час їх передачі та зберігання. Для цього необхідно застосовувати сучасні методи криптографічного захисту, що унеможливають перехоплення або модифікацію даних сторонніми особами. Крім того, система повинна передбачати засоби резервного копіювання та відновлення інформації у разі виникнення технічних збоїв або втрати даних.

Окрему увагу слід приділити захисту персональних даних відповідно до чинних нормативно-правових вимог. Система має забезпечувати обробку персональної інформації з урахуванням принципів законності, мінімізації обсягу даних та інформованої згоди користувачів. Реалізація відповідних механізмів дозволяє забезпечити відповідність програмного продукту вимогам законодавства у сфері захисту інформації та персональних даних.

Таким чином, аналіз вимог до безпеки та захисту персональних даних засвідчує необхідність комплексного підходу до проектування мобільного додатку. Загрози та механізми захисту персональної інформації зображено в таб. 2.1.

Таблиця 2.1

Вимоги та механізми захисту персональних даних користувачів

<i>Вимога до системи</i>	<i>Потенційна загроза</i>	<i>Механізм захисту</i>
Забезпечення конфіденційності персональних даних користувачів	Несанкціонований доступ до облікових записів	Автентифікація користувачів, надійні паролі
Розмежування доступу за ролями	Зловживання правами доступу	Авторизація на основі ролей користувачів
Захист даних під час	Перехоплення даних у	Використання

передачі	мережі	захищених протоколів передавання
Цілісність навчальних та облікових даних	Модифікація або підміна інформації	Контроль цілісності, серверна валідація даних
Захист бази даних	Викрадення або пошкодження даних	Обмеження доступу, резервне копіювання
Доступність сервісу	Відмова в обслуговуванні	Моніторинг системи, відновлення після збоїв
Захист персональних даних відповідно до законодавства	Порушення вимог щодо обробки даних	Мінімізація збору даних, інформована згода
Журналювання подій безпеки	Неможливість виявлення інцидентів	Логуювання дій користувачів

Впровадження ефективних механізмів безпеки сприяє підвищенню надійності системи, захисту інтересів користувачів та формуванню довіри до цифрового навчального середовища.

2.7 OBS проєкту

Організаційна структура проєкту (Organizational Breakdown Structure, OBS) створення мобільного додатку для персоналізованого навчання у форматі мікро-льорнінгу відображає функціональну модель управління та чіткий розподіл відповідальності між учасниками за рівнями компетенцій проєктної команди. Запропонована OBS забезпечує ефективну взаємодію між управлінськими, аналітичними, технічними та підтримуючими ролями в межах своїх обов'язків та вертикалей

На верхньому рівні організаційної структури розташований **власник продукту**, який визначає стратегічні цілі проєкту, формує бачення кінцевого програмного продукту та встановлює пріоритети розвитку функціональних можливостей мобільного додатку. Власник продукту відповідає за відповідність проєкту бізнес-цілям і освітнім потребам користувачів.

Безпосереднє керівництво виконанням проєкту здійснює **проєкт-менеджер**, який підпорядковується власнику продукту. Проєкт-менеджер відповідає за планування робіт, організацію взаємодії між учасниками команди, контроль термінів виконання завдань, управління ризиками та ресурсами, а також за комунікацію між зацікавленими сторонами. Проєкт-менеджеру підпорядковуються дві основні функціональні групи, що формують організаційну структуру проєкту – **Tech Team** та **Business Delivery Team**.

Tech Team відповідає за технічну реалізацію мобільного додатку.

До її складу входять:

- **Backend Developer** – реалізує серверну логіку системи, забезпечує обробку даних, створення API, взаємодію з базами даних та впровадження механізмів безпеки.
- **Frontend/Mobile Developer** – відповідає за розробку клієнтської частини додатку, побудову інтерфейсу, інтеграцію з бекендом та адаптацію UI до мобільних платформ.
- **DevOps Engineer** – забезпечує автоматизацію процесів розгортання, безперервну інтеграцію та доставку (CI/CD), налаштування хмарної інфраструктури, моніторинг і стабільність роботи системи.

Business Delivery Team виконує завдання, пов'язані з аналітикою, проектуванням користувацького досвіду та створенням навчального контенту.

У її склад входять:

- **Business Analyst** — здійснює аналіз потреб користувачів, збір та формалізацію вимог, координує взаємодію між власником продукту, технічною командою та контентним напрямом.
- **UI/UX Designer** — відповідає за дизайн інтерфейсу, розробку макетів, прототипування, юзабіліті-дослідження та забезпечення відповідності інтерфейсу принципам користувацько-орієнтованого дизайну.
- **Content-Manager** – створює, структурує та оновлює навчальний

контент, адаптує матеріали до концепції мікро-льорнінгу та забезпечує їх відповідність навчальним цілям.

У межах проєкту окремо буде функціонувати **Support Manager**, який буде виокремлений як окрема бізнес вертикаль, коли проєкт буде доведений до релізу. Він буде відповідати за технічну й інформаційну підтримку кінцевих користувачів, обробку звернень, вирішення типових проблем і передавання зворотного зв'язку команді розробки для подальшого вдосконалення системи.

Запропонована OBS забезпечує структуровану взаємодію між технічними та бізнес-напрямами, сприяє ефективному розподілу ролей, оптимізації комунікацій та підвищенню керованості проєкту. Загальна схема OBS представлена на рисунку 2.7.

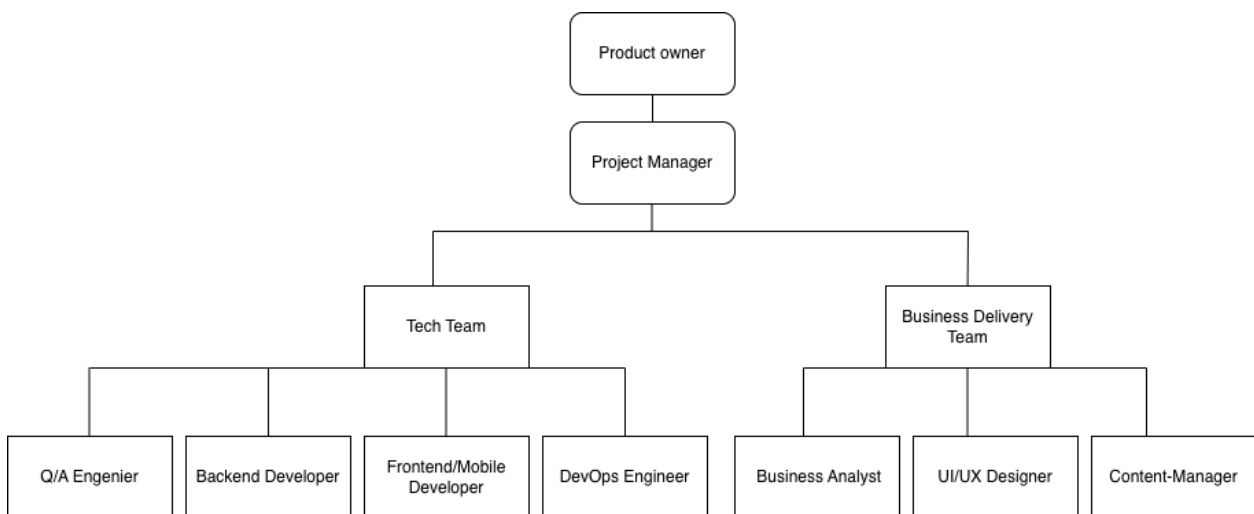


Рис.2.7 OBS-проєкту

Таким чином, представлена OBS проєкту забезпечує ефективний розподіл ролей, чітку ієрархію підпорядкування та створює організаційні умови для успішної реалізації мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу

2.8 WBS проєкту

Для забезпечення ефективного планування та управління процесом створення мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу в межах даної роботи було побудовано структуру декомпозиції робіт проєкту (Work Breakdown Structure, WBS). WBS дозволяє ієрархічно представити повний обсяг робіт проєкту, розподілити їх на логічно пов'язані етапи та забезпечити прозорість виконання проєктних задач протягом усього життєвого циклу програмного продукту.

На верхньому рівні WBS розглядається проєкт розробки мобільного навчального додатку як цілісний об'єкт управління. Наступний рівень структури охоплює основні етапи реалізації проєкту, кожен з яких включає комплекс взаємопов'язаних робіт, спрямованих на досягнення конкретних проміжних результатів.

Першим етапом у структурі WBS є планування та аналіз проєкту. На цьому етапі визначаються стратегічні цілі проєкту, аналізуються потреби та очікування цільової аудиторії, а також оцінюються потенційні ризики та наявні ресурси. Результатом даного етапу є сформована концепція проєкту, часові рамки виконання робіт і базові управлінські рішення.

Другий етап WBS пов'язаний із концептуальним та логічним проєктуванням системи. У межах цього етапу здійснюється формування функціональних і нефункціональних вимог до мобільного додатку, розробка концептуальної моделі інформаційної системи, проєктування структури бази даних та вибір технологій і інструментів розробки. Даний етап створює теоретичну та технічну основу для подальшої реалізації програмного продукту.

Наступним етапом є реалізація програмного забезпечення, що включає безпосередню розробку серверної та клієнтської частин мобільного додатку, реалізацію бізнес-логіки системи та механізмів персоналізації навчального контенту. Також у цей етап охоплює тестування та контроль якості програмного продукту. У межах цього етапу проводиться модульне та інтеграційне

тестування, перевіряється відповідність реалізованого функціоналу визначеним вимогам, а також здійснюється виявлення й усунення дефектів. Це забезпечує стабільність, надійність та готовність системи до впровадження.

Подальшим етапом є впровадження системи, яке включає підготовку додатку до експлуатації, проведення пілотного запуску, навчання користувачів та аналіз результатів тестового використання. На основі отриманого зворотного зв'язку вносяться необхідні коригування перед повномасштабним розгортанням системи.

Завершальним етапом WBS є завершення проєкту та супровід, що передбачає підготовку фінальної документації, передачу програмного продукту в експлуатацію, організацію технічної підтримки та офіційну здачу проєкту замовнику. Даний етап забезпечує формальне завершення проєкту та створює умови для подальшого функціонування системи.

Таким чином, побудована WBS проєкту забезпечує структурований підхід до організації робіт, дозволяє ефективно планувати та контролювати виконання завдань і слугує основою для подальшого управління строками, ресурсами та відповідальністю в процесі створення мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу. WBS зображено на рис. 2.8

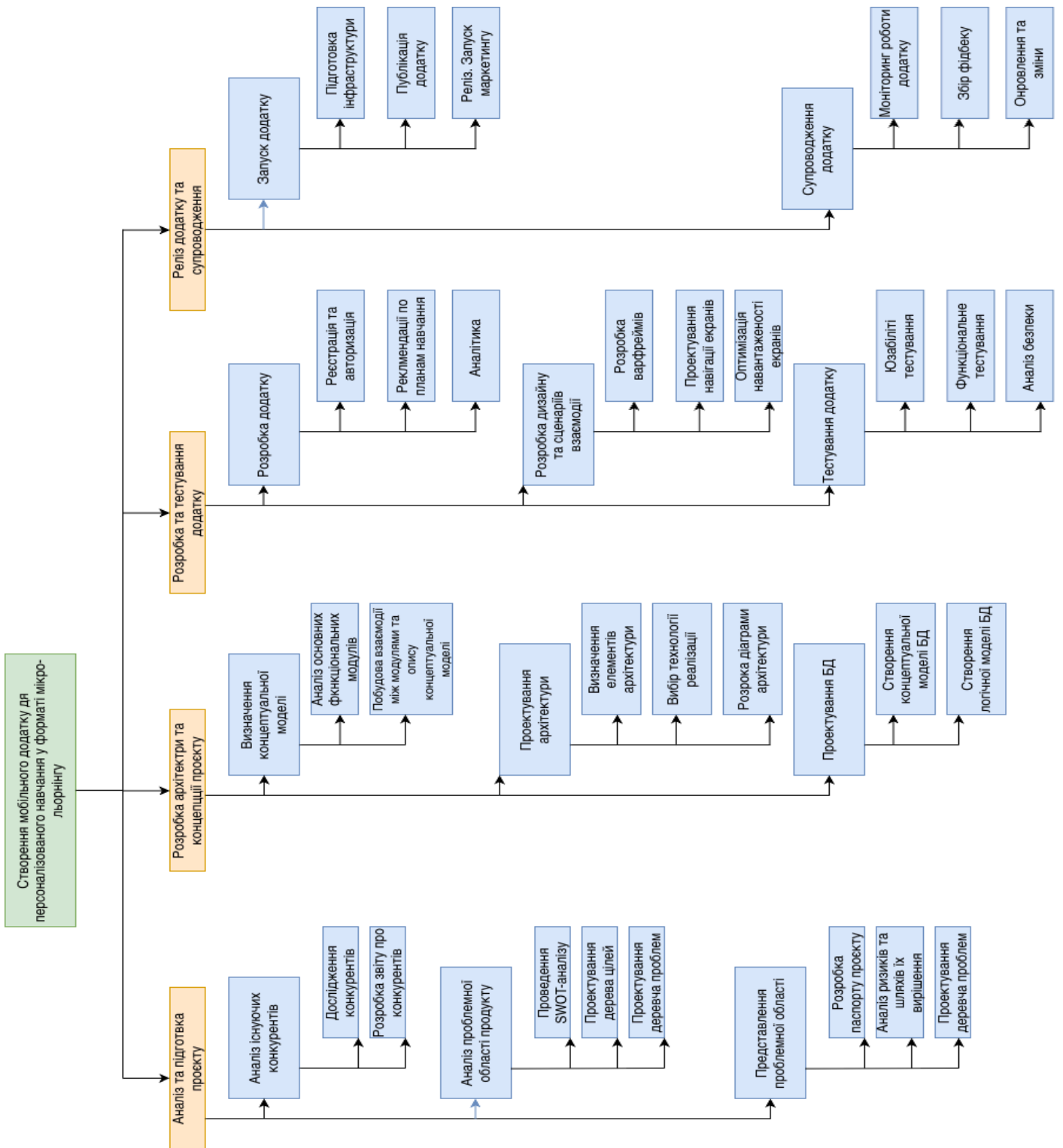


Рис.2.8 WBS-проекту

Додатково слід відзначити, що побудована структура декомпозиції робіт проєкту забезпечує можливість деталізації кожного етапу до рівня окремих керованих задач, що можуть бути оцінені за тривалістю, трудомісткістю та необхідними ресурсами. Така деталізація створює передумови для формування календарного плану проєкту, побудови мережевого графіка та визначення критичного шляху реалізації проєкту. Окрім цього, WBS проєкту є основою для розподілу відповідальності між учасниками команди відповідно до організаційної структури, що дає змогу забезпечити узгодженість між обсягом робіт і виконавцями. Поєднання WBS та OBS дозволяє сформувати прозору систему управління проєктом, у якій кожен елемент робіт пов'язаний із очікуваним результатом. В таблиці 2.2. зображено віхи проєкту

Таблиця 2.2

Представлення віх проєкту

Віха проєкту	Дата завершення	Опис	Очікуваний результат
Віха 1: Завершення аналітичного етапу та паспорту проєкту	05.09.2025	Проведення аналізу ринку EdTech, вивчення конкурентів, виконання SWOT- та PEST-аналізу; побудова дерева проблем та дерева цілей; визначення вимог та формування паспорта проєкту; опис цільової аудиторії та ризиків.	проблема та цілі проєкту; сформований документ вимог (FR/NFR); паспорт проєкту; аналітичний звіт про ринок та конкурентів.
Віха 2: Затвердженн я концептуаль ної моделі та архітектури	26.09.2025	Розробка концептуальної моделі додатку, визначення модулів додатку, проєктування архітектури, створення концептуальної та логічної моделі бази даних..	затверджена концептуальна модель сервісу; логічна структура БД та зв'язки; архітектурна діаграма

Віха проєкту	Дата завершення	Опис	Очікуваний результат
Віха 3: Завершення UX/UI дизайну та сценаріїв взаємодії	24.10.2025	Створення користувацьких сценаріїв (Use Cases), розробка UX-флоу, навігаційних карт, інтерфейсних макетів та інтерактивних прототипів додатку. Визначення вимог до доступності та мобільної адаптивності.	UX-сценарії та навігаційні карти; дизайн-макети усіх основних екранів; інтерактивний прототип додатку; затверджена система дизайну (UI Kit).
Віха 4: Реалізація MVP додатку (клієнтська та серверна частини)	24.12.2025	Розробка базових модулів: реєстрація, авторизація, персоналізований навчальний маршрут, проходження мікромодулів, система прогресу, базова аналітика. Впровадження API, підключення БД, реалізація механізму персоналізації контенту.	обочий прототип (MVP) мобільного додатку; реалізовані ключові функції: навчальні модулі, маршрут, прогрес; базова аналітика та рекомендації; інтегрований клієнт і сервер.
Віха 5: Тестування, інтеграція та оптимізація додатку	16.01.2026	Повне функціональне, інтеграційне, нефункціональне та юзабіліті тестування. Аналіз продуктивності, перевірка безпеки, оптимізація швидкодії та зручності. Усунення дефектів та UX-помилки.	фіналізована версія додатку без критичних помилок; протокол тестування та звіт про виявлені/усунені помилки; підтверджена відповідність вимогам; готовність продукту до релізу.
Віха 6: Розгортання, публікація та супровід додатку	30.01.2026	Підготовка серверної інфраструктури, налаштування DevOps-процесів, деплой бекенду, публікація мобільного додатку в маркетплейсах (за потреби), запуск маркетингової кампанії. Налагодження системи моніторингу та збору зворотного зв'язку.	публікований та доступний мобільний додаток; стабільна робота після деплою; отримання перших користувацьких відгуків; підготовлена дорожня карта для наступних релізів.

2.9 Розрахунок економічної ефективності проєкту

Економічна ефективність є одним із ключових критеріїв оцінювання доцільності розробки та впровадження мобільного додатку. Проведення економічного аналізу дозволяє визначити, наскільки інвестиції у створення такого продукту є обґрунтованими, а також оцінити доцільність, вигоду в процесі його експлуатації. У межах оцінювання враховуються як витрати на реалізацію проєкту, так і очікувані фінансові надходження. Нематеріальні переваги, що виникають при впровадженні системи в освітній або корпоративний сектор.

Витрати на розробку додатку включають одноразові інвестиції, пов'язані з проведенням аналітичних робіт, моделюванням бізнес-вимог, проєктуванням архітектури, створенням клієнтської та серверної частин системи, реалізацією механізмів персоналізації, інтеграцією з хмарними сервісами, формуванням навчального контенту та виконанням усіх етапів тестування. До експлуатаційних витрат належать оплата хмарної інфраструктури, технічна підтримка, оновлення контенту, моніторинг продуктивності, маркетингові активності та супровід користувачів. Сукупність зазначених витрат формує початкову інвестицію та регулярні операційні витрати протягом життєвого циклу додатку.

Економічні вигоди проєкту можуть формуватися за рахунок різних моделей монетизації, зокрема преміум-підписки, платних мікрокурсів, доступу до персоналізованих навчальних маршрутів, інтеграції з корпоративними освітніми платформами або розміщення рекламного контенту. Додатково важливими є непрямі вигоди – підвищення ефективності навчання, збільшення залученості користувачів, покращення якості освітніх процесів та можливість масштабування продукту на інші сегменти ринку.

Для оцінювання доцільності інвестування доцільно застосувати стандартний аналіз фінансових показників. Одним із базових критеріїв є чистий приведений дохід (**NPV**), який визначається як різниця між сумою

дисконтованих грошових надходжень та початковими витратами:

$$NPV = \sum_{t=1}^n \frac{CF_t}{(1+r)^t} - I, \quad (3.1)$$

де CF_t – грошовий потік у періоді t ,

r – дисконтна ставка,

I – початкові інвестиції.

Позитивне значення NPV свідчить про економічну доцільність проєкту. Для загальної оцінки ефективності потрібно застосувати загальний показник рентабельності інвестицій (ROI):

$$ROI = \frac{\Pi_{\text{чистий}}}{I} \times 100\%, \quad (3.2)$$

де $\Pi_{\text{чистий}}$ – є сумарним прибутком за період аналізу та розрахунків

Одним із заключних параметрів аналізу економічної доцільності є період окупності (PP), який показує час, який необхідно для повернення вкладених коштів:

$$PP = \frac{I}{CF_{\text{річний}}}, \quad (3.3)$$

де $CF_{\text{річний}}$ – середній грошовий потік за рік.

Отже, беручи до уваги стратегію економічної оцінки можна визначити, що для виконання розробки MVP мобільного додатку впродовж 4 місяців залучається команда:

- Project Manager – 55 000 грн/місяць
- Backend Developer – 65 000 грн/місяць
- Backend Developer – 65 000 грн/місяць
- Frontend Developer – 60 000 грн/місяць
- DevOps Engineer – 70 000 грн/місяць

- Business Analyst – 55 000 грн/місяць
- UI/UX Designer – 50 000 грн/місяць
- Content-Manager – 35 000 грн/місяць
- Support Manager – 30 000 грн/місяць

Загальні витрати на команду будуть складати сума всіх заробітніх плат помножена на тремін розробки MVP, тобто

$$V_{\text{команди}} = 485,000 \times 4 = 1,940,000 \text{ грн}$$

(3.4)

Додаткові витрати:

- хостинг та інфраструктура – 35 000 грн
- маркетинг початкового етапу – 50 000 грн
- OpenAI API, Firebase, аналітика – 27 000 грн
- непередбачувані витрати (10%): – 194 000 грн

Загальну суму додаткових витрат буде складати:

$$V_{\text{додаткові}} = 35,000 + 50,000 + 27,000 + 194,000 = 306\,000 \text{ грн}$$

(3.5)

Виходячи із вхідних даних загальна вартість стартових інвестицій в проєкт буде складати 2 246 000 грн. За стратегією юніт-економіки додатку, вона буде формуватись на основі щомісячної підписки з вартістю 120 грн/міс, що складатиме 100 грн чистого прибутку після комісій AppStore. Очікуваний приріст активних користувачів додатку буде складати:

- у 1-й рік – 1 000 платних користувачів
- у 2-й рік – 2 000 користувачів
- у 3-й рік – 3 500 користувачів

То при розрахунку загальних річних потоків виходить, що за 1-й рік додаток буде мати дохід 1 200 000 грн. За 2-й рік 2 400 000 грн та за 3-й рік 4 200 000 грн. Відповідно загальний NPV при дисконтній ставці 10% буде складати 3 986 696 грн. Чистий прибуток за 3 роки буде складати 5 554 000

грн, виходячи з всіх розрахунків ROI складатиме 247%, що свідчить про досить високу економічну вигоду і швидку окупність стартових інвестицій, які будуть складати 10 місяців і 2 тижні. Графік окупності зображений на рис. 2.9

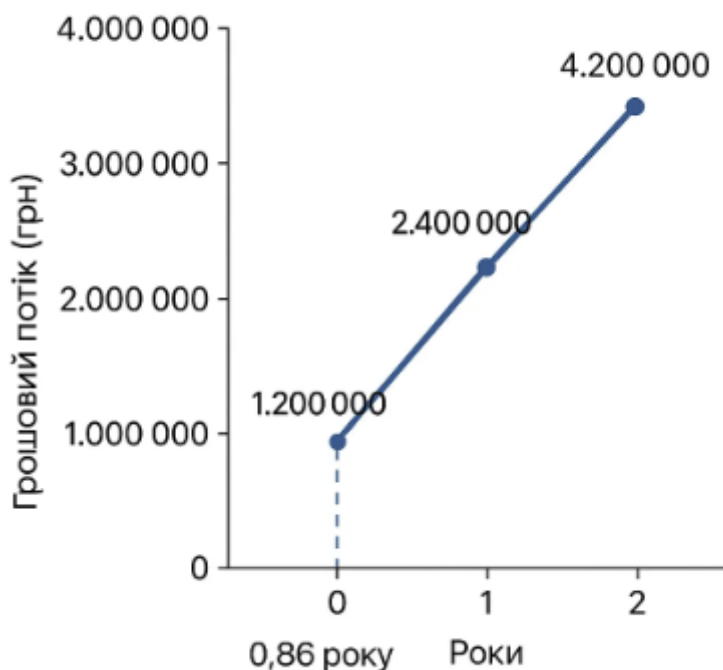


Рис. 2.9 Графік окупності додатку

Отримані результати свідчать про високу економічну привабливість проєкту. Значення $NPV > 0$, значення ROI, який перевищує 200 % і період окупності, який складає менше 1 року. Це підтверджує, що мобільний додаток має реальний потенціал комерційного успіху, здатний приносити стабільний прибуток та ефективно масштабуватися на ринку освітніх технологій.

При аналізі економічної ефективності важливо враховувати не лише оптимістичний сценарій розвитку продукту, але й можливі ризики, що можуть вплинути на темпи зростання користувачької бази, рівень витрат або стабільність грошових потоків.

До ключових ризиків проєкту належать:

- **Ризики ринку** — повільніший, ніж прогнозовано, темп

залучення платних користувачів, зміна поведінки споживачів, конкуренція з великими платформами (Coursera, Duolingo, EdApp).

- **Технологічні ризики** — можливі затримки в розробці, збільшення витрат на інфраструктуру, потреба в оптимізації алгоритмів персоналізації.
- **Операційні ризики** — зростання витрат на підтримку, необхідність збільшення команди, збільшення вартості API (OpenAI, Firebase).
- **Правові та регуляторні ризики** — посилення вимог до обробки даних користувачів, потреба сертифікації.

Для врахування зазначених ризиків доцільно застосувати коефіцієнт коригування грошових потоків, що зменшує очікуваний дохід у разі їхнього впливу. Зазвичай для проєктів EdTech використовується коригувальний фактор 0,85–0,9, що відповідає зниженню прогнозованих доходів на 10–15 %. З урахуванням цих умов потрібно провести вторний розрахунок NPV, що скадатиме 3 511 492 грн, що на ~12 % нижче від початкової оцінки, однак усе ще має дуже позитивні показники. Відповідно чистий прибуток після коригувань по дисконтній ставці буде складати 4 618 000 грн. Виходячи з цього ми можемо провести розрахунок відкорегованого ROI.

$$ROI = \frac{4\,618\,000}{2\,246\,000} \times 100\% \approx 205\% \quad (3.6)$$

Період окупності з урахуванням знижених грошових потоків збільшується, але лише незначною мірою, період окупності після корегування складатиме 1 рік та 1 місяць .

2.10 Модель адаптивного керування навчальним планом

У системах персоналізованого навчання важливо забезпечити адаптивність навчального плану, тобто здатність автоматично визначати

оптимальний наступний навчальний модуль для користувача. Запропонована модель прийняття рішень ґрунтується на аналізі трьох основних параметрів: успішності користувача, часових характеристик навчання та його індивідуального профілю. Модель не є математично складною, але описує чіткий формальний механізм, на основі якого система визначає наступний крок у навчальному процесі користувача. В рамках поточного проєкту було сформоване чітке математичне представлення функціоналу персоналізації та адптивності начального контенту, який можна описати формулою:

$$NextModule = \Omega(S, T, P), \quad (3.7.)$$

де

Ω – правила прийняття рішень побудовані на основі внутрішніх алгоритмів системи.

S – рівень успішності користувача у спів-відношенні до внутрішніх балів додатку.

T – часові характеристики користувача, час за який був пройдений модуль чи тест.

P – індивідуальний профіль. Налаштування користувача, його інтереси та вибір поточного курсу дисципліни.

В поточній реалізації система має досить гнучку механіку підбору рекомендацій та навчального контенту. Наявність поточних 4 змінних у математичному представленні дозволяє мати велику варіативність сценаріїв, що спряє високому інтересу користувача та його мотивації для продовження навчання. Відповідно лінійне представлення загальної логіки персоналізації дозволить нам в майбутньому масштабувати та видозмінювати алгоритми, що буде безперечно перевагою в гнучкості та стабільності роботи всієї системи. Складність алгоритму підбору рекомендацій рівнопропорційно буде залежити від кількості змінних, які будуть застосовуватись для аналізу та агрегації персоналізованого плану навчання.

В таблиці 2.3 представлені сценарії поведінки системи на основі вихідних результатів користувача, де детально описано сценарії поведінки посиляючись на вхідні значення змінних.

Таблиця 2.3

Логіка адаптивного вибору наступного навчального кроку

Умови (параметри S, T, P)	Інтерпретація стану користувача	Рішення системи (наступний крок)
$S \geq 80\%$	Висока успішність, матеріал засвоєно добре	Перехід до модуля вищої складності
$S \geq 80\%$ і T швидкий	Висока успішність та швидке виконання завдань	Підвищення складності на два рівні, пришвидшення темпу
$50\% \leq S < 80\%$	Середній рівень успішності, розуміння матеріалу	Видача модуля аналогічної тематики, структури
$50\% \leq S < 80\%$ і T повільний	Середня успішність і повільне виконання	Зменшення навантаження: коротші або спрощені мікромодулі
$S < 50\%$	Низька успішність, матеріал не засвоєно	Повернення до базових пояснень, базових тренувань
$S < 50\%$ і T нестабільний	Слабкі результати та нерівномірний темп навчання	Рекомендація мікропрактики, адаптивних підказок та нагадувань
P: ціль = швидке освоєння	Користувач хоче пройти курс швидше	Система збільшує темп, більше інтенсивних модулів
P: ціль = повторення	Користувач прагне закріпити знання	Пріоритет на ревізії, повторних модулях, тестах

Р: низький стартовий рівень	Початківець, слабка база знань	Зниження стартової складності. Початкові та базові модулі
------------------------------------	--------------------------------	---

Основною перевагою даного алгоритму є чітка лінійна та прогнозована модель поведінки, вона сприяє майбутньому масштабуванню та розвитку. Схематичне зображення алгоритму представлено на рис. 2.10., деталі реалізації програмного коду наведено в додатку Б (див. Додаток Б, рис. Б.3).

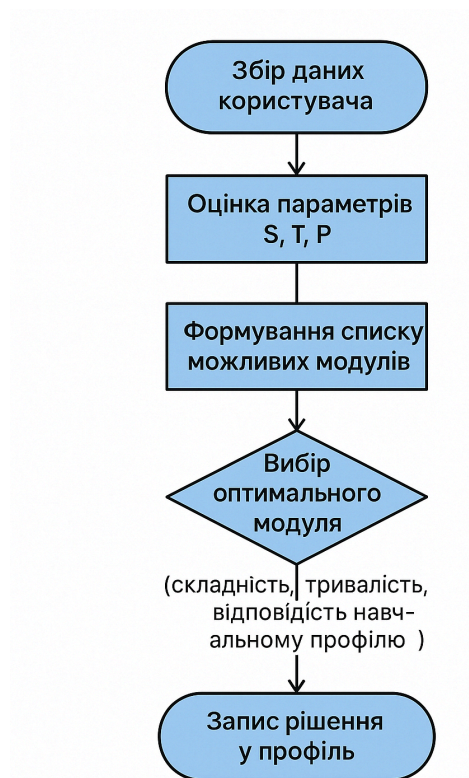


Рис. 2.10. Алгоритм підбору навчальних модулів для користувача

Запропонована модель забезпечує персоналізацію навчання без використання складних ML-алгоритмів, оскільки базується на простих правилах, що враховують успішність, темп та індивідуальні характеристики користувача. Вона є легкою у реалізації та масштабуванні, дозволяє гнучко оновлювати сценарії навчання та розширювати функціональність без змін архітектури. Модель також створює основу для подальшого впровадження AI-рішень і підвищує ефективність мікро-льорнінгу завдяки адаптації темпу

й складності контенту до потреб кожного користувача [35].

РОЗДІЛ 3. ВИБІР ТЕХНОЛОГІЇ УПРАВЛІННЯ ТА ПРОЄКТУВАННЯ АРХІТЕКТУРИ

3.1 Вибір технології управління проєктом створення мобільного додатку з мікро-льорнінгу

Вибір технології управління проєктом є одним із ключових рішень, що визначає ефективність процесу розроблення програмного продукту, рівень керованості робіт та досягнення поставлених цілей у визначені терміни. Проєкт створення мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу має низку специфічних характеристик, які безпосередньо впливають на вибір відповідної технології управління.

До таких характеристик належать динамічність та варіативність вимог до функціональності системи, необхідність постійного вдосконалення навчального контенту, орієнтація на користувацький досвід, а також залежність якості продукту від зворотного зв'язку кінцевих користувачів. Формат мікро-льорнінгу передбачає створення коротких навчальних модулів, які можуть змінюватися або доповнюватися відповідно до результатів аналізу ефективності навчання, що ускладнює фіксацію повного переліку вимог на початковому етапі проєкту.

Аналіз традиційних каскадних технологій управління проєктами свідчить, що вони базуються на послідовному виконанні етапів із мінімальною можливістю внесення змін після завершення попереднього етапу. Такий підхід є доцільним для проєктів зі стабільними та чітко визначеними вимогами, проте в умовах розроблення мобільного освітнього додатку може призвести до зниження гнучкості, ускладнення впровадження змін та затримок у реалізації функціональних покращень.

З урахуванням зазначених особливостей, для управління проєктом створення мобільного додатку з мікро-льорнінгу обрано гнучку технологію

управління, що ґрунтується на ітеративному та інкрементальному підходах. Дана технологія передбачає поділ процесу розробки на короткі ітерації, у межах яких здійснюється планування, реалізація визначеного набору функцій, тестування та оцінка результатів. Такий підхід дозволяє поетапно формувати функціонально завершений продукт та адаптувати його до поточних потреб користувачів.

Використання гнучкої технології управління є особливо доцільним для реалізації мікро-льорнінгу, оскільки вона забезпечує можливість швидкого коригування структури навчальних модулів, складності завдань та алгоритмів персоналізації. Кожна ітерація дає змогу оцінити ефективність реалізованих елементів з точки зору зручності використання і навчальної результативності, що є критичним для освітніх цифрових продуктів.

Крім того, обрана технологія управління сприяє підвищенню прозорості виконання робіт та ефективній взаємодії між учасниками проєкту. Поєднання гнучкого підходу з використанням інструментів структуризації, таких як WBS та OBS, дозволяє поєднати ітеративну розробку з чітким розподілом відповідальності, плануванням ресурсів і контролем термінів виконання завдань. Це особливо важливо для проєктів, у яких одночасно виконуються роботи з проєктування архітектури, розробки програмного коду та створення навчального контенту.

Застосування гнучкої технології управління також знижує рівень проєктних ризиків, оскільки потенційні помилки або невідповідності вимогам виявляються на ранніх етапах розробки. Регулярні ітерації та перевірка результатів дозволяють своєчасно коригувати напрям розвитку проєкту та мінімізувати витрати на усунення критичних недоліків.

Таким чином, вибір гнучкої технології управління проєктом для створення мобільного додатку є науково обґрунтованим і доцільним. Обраний підхід забезпечує адаптивність процесу розробки, орієнтацію на потреби користувачів, ефективне управління змінами та створює передумови для формування якісного й конкурентоспроможного цифрового освітнього

продукту.

Використання гнучкої технології управління забезпечує адаптацію функціональності додатку до особливостей мікро-льорнінгу, зокрема тривалості навчальних сесій, структури навчальних мікромодулів та індивідуальних траєкторій навчання користувачів. Регулярне отримання зворотного зв'язку дозволяє коригувати навчальний контент, рівень складності матеріалів та інтерфейсні рішення на ранніх етапах розробки.

Крім того, обрана технологія управління проектом добре узгоджується з використанням інструментів структуризації робіт і ресурсів, таких як WBS та OBS, що забезпечує прозорість планування, чіткий розподіл відповідальності та контроль виконання завдань. Такий підхід підвищує керованість проекту, зменшує ризики та створює умови для послідовного розвитку додатку.

У межах гнучких технологій управління проектами найбільш поширеними методами є Scrum та Kanban, які широко застосовуються під час розроблення програмного забезпечення. Обидва підходи спрямовані на підвищення адаптивності процесу розробки, прозорості виконання робіт і якості кінцевого результату, однак мають різні акценти у плануванні та організації діяльності команди.

Методологія Scrum ґрунтується на ітеративній моделі розробки, де робота поділяється на фіксовані часові ітерації – спринти. Для кожного спринту формується перелік завдань, виконання яких завершується створенням інкременту програмного продукту з чітко визначеною функціональністю. Scrum передбачає чітко визначені ролі, регулярні зустрічі та формалізовані події, що забезпечує контроль термінів, прогнозованість результатів і системний розвиток продукту [13].

Застосування Scrum є доцільним у проектах, де можливо спланувати обсяг робіт на короткий проміжок часу та отримувати стабільні інкременти функціональності. Для освітнього мобільного додатку це дозволяє послідовно впроваджувати нові навчальні модулі, механізми персоналізації, аналітики та елементи інтерфейсу, зберігаючи контроль за якістю реалізації.

Метод Kanban, на відміну від Scrum, не передбачає фіксованих ітерацій і базується на безперервному потоці виконання завдань. Основна увага в Kanban приділяється візуалізації процесу роботи, обмеженню кількості паралельних задач та оптимізації часу їх виконання. Такий підхід є ефективним для процесів підтримки, вдосконалення існуючих функцій та оперативного реагування на зміни. Kanban доцільно застосовувати у випадках, коли пріоритети часто змінюються, а завдання мають нерівномірний характер. Для освітніх цифрових систем цей підхід добре підходить на етапах супроводу продукту, адаптації навчального контенту та впровадження дрібних функціональних покращень на основі зворотного зв'язку користувачів. З урахуванням специфіки проекту створення мобільного додатку з мікро-льорнінгу, який включає етапи аналізу, проектування, розробки та тестування нового функціоналу, більш доцільним є використання Scrum як основної технології управління проектом. Scrum забезпечує структурованість процесу розробки, можливість планування і контролю реалізації функціональних інкрементів та чітке узгодження результатів кожного етапу. Водночас використання окремих елементів Kanban може бути доцільним на етапах підтримки та доопрацювання системи після впровадження, зокрема для управління запитами користувачів і дрібними змінами навчального контенту. Таким чином, у межах даного проекту основним підходом управління обрано Scrum із можливим поєднанням елементів Kanban, що забезпечує баланс між структурованістю та гнучкістю процесу розробки.

3.2 Структурна організація програмного продукту

У цьому підрозділі детально розглядається логічна структура мобільного додатку, що виступає основною складовою системи персоналізованого навчання у форматі мікро-льорнінгу. Програмний продукт розробляється як багаторівнева, багатокomпонентна система, що забезпечує гнучкість, масштабованість і підтримку високого рівня користувацького

досвіду. Архітектурна модель ґрунтується на принципах модульності, інкапсуляції функціональності та реюзабельності (повторного використання) коду, що є базовими положеннями сучасного програмного проектування.

Функціонально система поділяється на низку автономних, але взаємопов'язаних модулів, кожен із яких відповідає за реалізацію певної логіки додатку. Зокрема, передбачено:

1. **Модуль автентифікації та реєстрації**, який забезпечує створення користувацьких профілів, збереження персональних даних та захист доступу через механізми хешування паролів;

2. **Модуль навчального контенту**, що відповідає за динамічне завантаження, структурування та відображення мікроуроків;

3. **Модуль тестування**, призначений для створення й проходження коротких тестів після кожного навчального блоку, з автоматичним підрахунком балів;

4. **Модуль прогресу**, який здійснює моніторинг навчальних досягнень користувача, накопичення статистики й відображення аналітики у зручному вигляді (графіки, таблиці, діаграми);

5. **Модуль нагород і бейджів**, що реалізує елементи гейміфікації, мотивуючи користувачів до проходження курсів;

6. **Модуль персональних налаштувань**, який дозволяє змінювати мову інтерфейсу, активувати або вимикати push-сповіщення, а також керувати алгоритмами.

3.3 Моделювання поведінки системи на основі сценаріїв використання

Процес моделювання поведінки системи є критичним етапом у проектуванні будь-якого програмного продукту, особливо коли йдеться про користувацько орієнтовані інформаційні системи. У контексті мобільного додатку для персоналізованого мікро-льорнінгу такий підхід дозволяє описати логіку взаємодії між користувачем і системою через конкретні

сценарії використання.

Сценарій використання (use case) – це формалізований опис типових дій, які виконує користувач для досягнення певної мети в системі. Ці сценарії відображають функціональні вимоги до програмного забезпечення з позиції кінцевого користувача. Для їх моделювання застосовується нотація UML (Unified Modeling Language), зокрема діаграми варіантів використання та діаграми активностей.

Основні сценарії використання системи:

1. Реєстрація та авторизація користувача. Користувач створює обліковий запис, вводячи персональні дані, або входить у вже створений акаунт за допомогою електронної пошти чи соціальних мереж.

2. Перегляд навчального контенту. Після входу користувач має змогу обрати навчальний модуль відповідно до своєї мети, рівня знань або рекомендованого алгоритмом тракту. Контент подається у вигляді мікроуроків.

3. Проходження тестування. Після завершення навчального блоку користувач проходить короткий тест на закріплення знань. Результати фіксуються у профілі.

4. Моніторинг прогресу. Користувач має змогу переглядати загальний прогрес, отримані результати тестів, рівень активності та досягнуті бейджі.

5. Отримання рекомендацій. Система надає персоналізовані рекомендації щодо наступних уроків або зміни навчального курсу залежно від попередніх досягнень.

6. Налаштування персонального профілю. Користувач може змінити мову, дозволити чи заборонити повідомлення, активувати адаптивний режим навчання.

На рис 3.2. приведено схему основних функціональних Use Cases.

Таким чином, моделювання поведінки системи через сценарії використання дозволяє не лише конкретизувати функціональну модель додатку, а й забезпечити ефективне технічне завдання для подальшої

реалізації архітектури, інтерфейсів та бізнес-логіки системи. Це підвищує відповідність розробленого програмного продукту очікуванням користувачів та забезпечує більшу гнучкість у процесі його масштабування.



Рис. 3.2. Схема основних Use-Cases

Перелік альтернативних *use-cases* в рамках роботи системи:

1. Реєстрація / Вхід

Основний потік: Користувач вводить email і пароль, проходить перевірку, входить у систему.

Альтернативні потоки:

- 1А. Невірні дані: Система повідомляє про помилку автентифікації, пропонує відновлення пароля.
- 1В. Email вже зареєстровано: При повторній спробі реєстрації система блокує дію, пропонує авторизацію.
- 1С. Тимчасова недоступність сервера: Виводиться повідомлення з проханням повторити пізніше.

2. Перегляд навчального контенту

Основний потік: Користувач обирає курс, мікроуроки завантажуються

Альтернативні потоки:

- 2А. Курс недоступний: Виводиться повідомлення «Курс тимчасово недоступний», пропонується альтернатива.
- 2В. Контент не підходить за рівнем: Система автоматично коригує курс згідно з налаштуваннями персоналізації.
- 2С. Відсутнє інтернет-з'єднання: Система надає офлайн-уроки, якщо були попередньо збережені.

3. Проходження тесту

Основний потік: Користувач відповідає на запитання, система підраховує бал.

Альтернативні потоки:

- 3А. Пропуск запитання: Дозволяється перейти до наступного, система знижує загальний бал.
- 3В. Час на тест завершився: Відбувається автоматичне завершення тесту з оцінкою доступних відповідей.
- 3С. Нестабільне з'єднання: Система зберігає введені відповіді локально та надсилає їх пізніше.

4. Моніторинг прогресу

Основний потік: Користувач переглядає успішність, діаграми, бейджі.

Альтернативні потоки:

- 4А. Дані не оновлено: Система виводить останні збережені локально результати та повідомляє про проблему з оновленням.
- 4В. Зміна рівня: Якщо рівень користувача змінився (наприклад, після кількох невдач), система адаптує навчальний маршрут.

5. Отримання рекомендацій

Основний потік: Система аналізує прогрес, надає поради.

Альтернативні потоки:

- 5А. Низький бал кілька разів поспіль: Пропонується пройти короткий вступний модуль повторно.

- 5В. Користувач проігнорував рекомендації: Система відстежує й адаптує рекомендації на майбутнє.

6. Налаштування профілю

Основний потік: Користувач змінює мову, push-сповіщення тощо.

Альтернативні потоки:

- 6А. Неправильний формат введених даних: Наприклад, email без @ -

виводиться повідомлення про помилку.

- 6В. Зміна мови недоступна: Система повідомляє про технічну неможливість зміни мови наразі. Ці альтернативні потоки підвищують реалістичність моделювання, слугують основою для технічної специфікації та тест-кейсів, і є необхідними для забезпечення стійкості та зручності користування системою в реальних умовах.

3.4 Архітектурне проектування програмного забезпечення

Архітектурне проектування програмного забезпечення є ключовим етапом розробки, що забезпечує структурну цілісність системи та визначає принципи її масштабованості, гнучкості, підтримованості та безпеки.

Архітектура програмного продукту виступає абстрактною моделлю високого рівня, яка описує компоненти системи, їхню ієрархію, взаємозв'язки та способи взаємодії.

Зважаючи на специфіку мобільного додатку для персоналізованого мікро-льорнінгу, проєкт обрано у вигляді багаторівневої клієнт-серверної архітектури, яка умовно поділяється на такі основні рівні:

1. Рівень представлення (Presentation Layer)

Цей рівень відповідає за інтерфейс користувача – взаємодію з навчальним контентом, перегляд тестів, результатів, рекомендацій тощо. Він реалізується у вигляді мобільного застосунку, розробленого для iOS/Android з використанням кросплатформеного фреймворку (наприклад, Flutter або React Native).

Особливості:

- адаптивний дизайн;
- інтерактивні елементи (графіки, бейджі, повідомлення);
- підтримка багатомовності.

2. Логічний рівень (Application / Logic Layer)

Цей рівень містить бізнес-логіку додатку – алгоритми персоналізації, перевірки тестів, нарахування балів, аналіз прогресу.

Ключові задачі:

- маршрутизація запитів;
- авторизація та автентифікація;
- адаптивне формування рекомендацій;
- обробка введених даних;
- логування дій користувача.

У цьому шарі також реалізується взаємодія із зовнішніми API – наприклад, аналітичними платформами або системами push-сповіщень.

3. Рівень доступу до даних (Data Access Layer)

Забезпечує доступ до бази даних, реалізує CRUD-операції (створення, читання, оновлення, видалення), обробляє запити до сховища даних.

Цей рівень відповідає за:

- зберігання профілів користувачів, уроків, результатів тестів;
- забезпечення цілісності зв'язків (через зовнішні ключі);
- шифрування чутливих даних (наприклад, паролів).

4. Рівень зберігання даних (Database Layer)

На цьому рівні функціонує СКБД (наприклад, PostgreSQL), що зберігає структуровані дані відповідно до логічної та фізичної моделі БД.

Передбачено:

- нормалізовану структуру таблиць;
- підтримку транзакцій;
- резервне копіювання;
- індексацію основних атрибутів для підвищення швидкодії.

5. Сервіси та інтеграції (External Services Layer)

Оскільки сучасні додатки взаємодіють з іншими платформами, передбачено підтримку таких інтеграцій:

- push-сповіщення (Firebase Cloud Messaging);
- аналітика користувацької поведінки (Google Analytics, Amplitude);
- сервіси електронної пошти;
- можливе підключення до LMS-систем (Learning Management System).

Архітектурний стиль. Застосовується модульно-орієнтована архітектура (modular architecture), де кожен функціональний блок реалізується у вигляді незалежного модуля, що спрощує тестування, оновлення та масштабування [36].

Безпека. Система передбачає:

- шифрування паролів користувачів (hash-функції);
- захист API через токенизацію (JWT);
- обмеження доступу за ролями (RBAC – Role-Based Access Control).

Таким чином, архітектура додатку відповідає сучасним вимогам до масштабованих, безпечних та зручних систем, що забезпечують персоналізований підхід до навчання. Такий підхід дозволяє адаптувати систему до зростаючої кількості користувачів і функцій без втрати продуктивності або надійності. Візуальна архітектурна схема зображена на рис. 3.3.

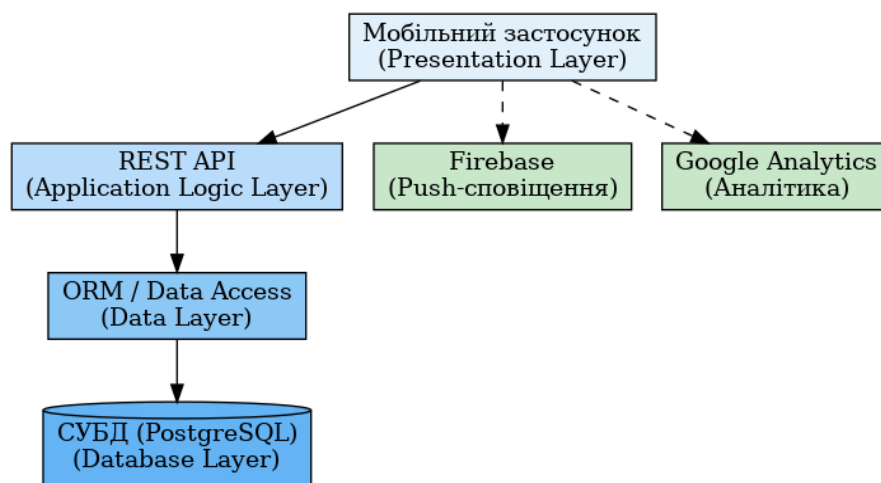


Рис. 3.3. Високорівнева архітектурна схема додатку

3.5 Вибір технологічного стеку та інструментів реалізації

Процес розробки сучасного мобільного додатку вимагає ретельного вибору технологічного стеку, який забезпечить ефективність, масштабованість, безпеку та зручність у подальшій підтримці. Враховуючи функціональні вимоги до системи персоналізованого мікро-льорнінгу, цільову аудиторію та обсяг планованої функціональності, було обрано такі інструменти та платформи:

1. Фронтенд (Flutter / Dart). Обрано як основний інструмент розробки мобільного інтерфейсу завдяки:

- підтримці кросплатформеності (iOS та Android з єдиною базою коду);
- високій продуктивності та нативному рендерингу;
- активній спільноті та великій кількості готових віджетів;
- зручному управлінню станом через BLoC або Provider.

Альтернативою може бути React Native, однак Flutter надає кращий контроль над UI та вищу швидкодію в мобільних додатках з високим навантаженням.

2. Бекенд. Серверна частина реалізується на основі Node.js з використанням фреймворку Nest.js, що забезпечує:

- високу швидкодію обробки запитів;
- можливість легкої реалізації REST API;
- зручну інтеграцію з базою даних та зовнішніми сервісами;
- підтримку масштабованої асинхронної логіки.

3. База даних (PostgreSQL). Вибір обумовлений необхідністю роботи зі структурованими даними, складними зв'язками між таблицями та підтримкою транзакцій.

Переваги:

- підтримка реляційної моделі;
- високий рівень безпеки та стабільності;
- наявність інструментів резервного копіювання;
- підтримка JSONB для напівструктурованих даних.

4. Сервіси персоналізації та аналітики:

- Firebase Cloud Messaging (FCM) – для реалізації push-сповіщень;
- Firebase Authentication – спрощена авторизація через email, Google або інші провайдери;
- Google Analytics for Firebase – відстеження взаємодії користувача з додатком, аналіз ефективності навчальних модулів;
- OpenAI / ML-інструменти (опційно) – для побудови алгоритмів персоналізації в подальшому.

5. Інструменти розробки та контролю версій:

- Visual Studio Code (легке та швидке IDE з підтримкою Flutter та Node.js);
- Git + GitHub / GitLab (для керування версіями коду, спільної розробки, CI/CD);
- Postman (тестування REST API та перевірка взаємодії між фронтендом і бекендом).

6. Дизайн та прототипування:

Figma – інструмент для створення інтерактивних прототипів, макетів екранів та UI-дизайну з можливістю командної роботи та адаптації під різні розміри екранів.

У цьому розділі було детально розглянуто ключові аспекти, пов'язані з проєктуванням архітектури мобільного додатку для персоналізованого навчання у форматі мікро-льорнінгу. Насамперед була сформована структурна організація програмного продукту, яка передбачає модульну побудову системи з виділенням окремих функціональних підсистем – автентифікації, роботи з контентом, тестування, аналітики та персоналізації. Такий підхід дозволяє забезпечити масштабованість і легку підтримку ПЗ.

Особливу увагу було приділено моделюванню поведінки системи на основі сценаріїв використання (use cases). Розроблені типові сценарії охоплюють ключові етапи взаємодії користувача з додатком – від реєстрації до отримання рекомендацій, а також включають альтернативні потоки, що підвищують гнучкість та реалістичність системної поведінки. Використання діаграм варіантів використання дозволяє наочно представити взаємозв'язки між акторами та функціональними можливостями системи.

У межах архітектурного проектування було обґрунтовано вибір багаторівневої архітектури, яка включає рівень представлення, логіки, доступу до даних і бази даних. Така архітектура є сучасним стандартом для мобільних застосунків і забезпечує розподіл відповідальностей між компонентами системи, що значно полегшує її розробку, тестування та подальший розвиток.

Завершальним етапом стало обґрунтування вибору технологічного стеку. Для фронтенду обрано кросплатформену технологію Flutter, яка дозволяє ефективно реалізувати єдиний інтерфейс для iOS та Android. Для серверної частини рекомендовано використовувати Node.js з Nest, а для управління даними – СКБД PostgreSQL. Таким чином, результати, отримані в цьому розділі, формують повноцінну технічну основу для подальшої реалізації проєкту, забезпечуючи узгодженість між логікою, архітектурою та вибраними інструментами розробки.

3.6 Проєктування API та взаємодії між компонентами системи

Проєктування прикладного програмного інтерфейсу (API) та механізмів взаємодії між компонентами системи є ключовим етапом у створенні мобільного додатку. Коректно спроектоване API забезпечує узгоджену взаємодію клієнтської та серверної частин системи, масштабованість програмного продукту, а також спрощує подальший супровід і розвиток функціональностіх.

У межах даного проєкту архітектура системи передбачає поділ на

клієнтську частину (мобільний додаток), серверну частину (бекенд) та рівень зберігання даних. Взаємодія між цими складовими здійснюється через програмний інтерфейс, який реалізує доступ до основних функціональних можливостей системи, зокрема управління користувачами, навчальним контентом, результатами навчання та персоналізованими рекомендаціями.

API проекту спроектовано за принципом чіткої ізоляції відповідальностей, що дозволяє розмежувати логіку обробки запитів, бізнес-логіку та роботу з базою даних. Клієнтська частина повідомляє сервер про дії користувача шляхом виклику відповідних API-методів, а сервер, у свою чергу, виконує обробку запиту, звернення до бази даних і повертає структуровану відповідь у встановленому форматі.

Важливим аспектом проектування API є стандартизація форматів обміну даними. У проєкті використовується єдиний структурований формат передавання інформації, що забезпечує однозначну інтерпретацію даних на стороні клієнта та мінімізує ризик помилок під час інтеграції компонентів системи. Кожен API-метод має чітко визначене призначення, параметри запиту та структуру відповіді [21].

Для забезпечення безпеки взаємодії між компонентами системи у процесі проектування API передбачено механізми автентифікації та авторизації користувачів. Доступ до ресурсів системи визначається роллю користувача та його правами, що дозволяє обмежити несанкціонований доступ до конфіденційних даних і функціональних можливостей додатку.

Окрему увагу приділено проектуванню API для підтримки персоналізованого навчання. Серверна частина системи обробляє дані про активність користувачів, результати проходження мікронавчальних модулів і на їх основі формує рекомендації щодо подальшого навчального маршруту. Така логіка реалізується на рівні серверних компонентів, що дозволяє централізовано управляти алгоритмами персоналізації.

Таким чином, спроектований API виступає ключовим елементом зв'язку між компонентами системи та забезпечує логічну цілісність, безпеку і

масштабованість мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу. Реалізація чітко визначених інтерфейсів взаємодії створює основу для подальшого розвитку системи та інтеграції з зовнішніми сервісами.

API системи поділяється на тематичні групи відповідно до основних функціональних підсистем мобільного додатку:

- API управління користувачами
- API навчального контенту
- API навчального процесу
- API персоналізації та рекомендацій
- API результатів навчання та досягнень

Такий поділ дозволяє забезпечити модульність, зручність масштабування та спростити підтримку системи. В таблиці 3.1 та в таблиці 3.2 представлений детальний опис структури та поведінкових особливостей API управління користувача та API навчального контнету. Також додатково можна ознайомитись зі структурою API навчального процесу, API персоналізації та рекомендацій, API результатів навчання та досягнень в додатку Б (див. Додаток Б, таб. А.2, А.3, А.4).

Таблиця 3.1

Характеристика API управління користувача

Параметр	Опис
<i>Призначення API</i>	Забезпечення автентифікації, авторизації та управління профілем користувача
<i>Типові операції</i>	Реєстрація нового користувача; вхід до системи; отримання та оновлення даних профілю; вихід із системи
<i>Структура запиту</i>	Ідентифікаційні дані користувача;

	облікові дані (логін, пароль); метадані сеансу роботи
<i>Структура відповіді</i>	Статус виконання операції; унікальний ідентифікатор користувача; дані профілю; маркер доступу (token)

Таблиця 3.2

Характеристика API навчального контенту

Параметр	Опис
<i>Призначення API</i>	Забезпечення автентифікації, авторизації та управління профілем користувача
<i>Типові операції</i>	Реєстрація нового користувача; вхід до системи; отримання та оновлення даних профілю; вихід із системи
<i>Структура відповіді</i>	Статус виконання операції; унікальний ідентифікатор користувача; дані профілю; маркер доступу (token)

3.7 Забезпечення масштабованості та підтримуваності системи

Однією з ключових вимог до сучасних програмних продуктів, зокрема мобільних додатків, є здатність системи ефективно масштабуватися та забезпечувати довгострокову підтримуваність. Масштабованість дозволяє системі адаптуватися до зростання кількості користувачів, обсягу навчального контенту та інтенсивності обробки даних, тоді як підтримуваність забезпечує можливість оперативного внесення змін, виправлення помилок і розвитку функціональних можливостей без суттєвих витрат ресурсів [21-22].

У межах даного проєкту забезпечення масштабованості системи

досягається завдяки використанню модульної архітектури, де клієнтська частина, серверні компоненти та база даних є логічно розділеними та взаємодіють між собою через чітко визначені програмні інтерфейси. Такий підхід дає змогу незалежно розвивати окремі компоненти системи, зменшуючи ризик виникнення помилок при модифікації програмного коду.

Збільшення навантаження на систему, пов'язане з ростом кількості користувачів або частоти звернень до навчального контенту, враховується шляхом проєктування серверної частини з можливістю горизонтального масштабування. Це дозволяє розподіляти обробку запитів між кількома екземплярами серверних компонентів та забезпечувати стабільну роботу додатку в умовах пікових навантажень. Окрему увагу приділено оптимізації роботи з базою даних та ефективному управлінню збереженням і обробкою інформації.

Підтримуваність системи забезпечується шляхом застосування принципів зрозумілої структури програмного коду, розмежування функціональних обов'язків компонентів та використання стандартизованих підходів до проєктування програмного забезпечення. Документування архітектурних рішень і програмних інтерфейсів сприяє швидкому ознайомленню нових розробників із проєктом та зменшує час, необхідний для впровадження змін.

Важливу роль у забезпеченні підтримуваності відіграє можливість розширення функціональності системи без порушення її наявної логіки роботи. Запровадження нових навчальних модулів, алгоритмів персоналізації або аналітичних інструментів може здійснюватися шляхом додавання окремих компонентів або сервісів, що мінімізує вплив змін на вже реалізовані функції. Таким чином, реалізовані архітектурні рішення спрямовані на забезпечення масштабованості та підтримуваності мобільного додатку. Це створює передумови для стабільної експлуатації системи, її подальшого розвитку та адаптації до змінних потреб користувачів і вимог освітнього середовища.

Додатково, з метою забезпечення масштабованості та підтримуваності системи, у процесі проектування та реалізації програмного забезпечення застосовуються практичні принципи сучасної програмної інженерії. Зокрема, використовується принцип модульності, який передбачає побудову системи у вигляді логічно незалежних компонентів із чітко визначеними інтерфейсами взаємодії. Такий підхід дозволяє розширювати або модифікувати окремі модулі без впливу на роботу всієї системи. У розробленні програмного забезпечення також враховуються принципи **SOLID**, що сприяють підвищенню якості коду та зменшенню складності його супроводу. Принцип єдиної відповідальності полягає в тому, що кожен програмний компонент відповідає за чітко визначену функцію, що полегшує тестування та налагодження. Принцип відкритості/закритості забезпечує можливість розширення функціональності системи без необхідності внесення змін у вже реалізований код .

Принцип підстановки гарантує коректну взаємозамінність компонентів системи, що підвищує її гнучкість. Принцип розділення інтерфейсів дозволяє уникнути надмірних залежностей між компонентами шляхом надання доступу лише до необхідних функцій. Принцип інверсії залежностей сприяє зменшенню зв'язності між модулями та підвищує стабільність архітектури системи. Застосування наведених принципів у поєднанні з продуманою архітектурою дозволяє забезпечити довготривалу підтримуваність програмного продукту, зменшити витрати на його подальший розвиток та створити технічні передумови для масштабування мобільного додатку.

РОЗДІЛ 4. МОДЕЛЮВАННЯ КОРИСТУВАЦЬКОЇ ВЗАЄМОДІЇ ТА ТЕХНОЛОГІЇ ТЕСТУВАННЯ СИСТЕМИ

4.1 Розробка концепції взаємодії користувача з інтерфейсом додатку

Розробка концепції взаємодії користувача (UX – User Experience) з інформаційною системою є одним з ключових факторів, що визначає прийнятність і успішність цифрового продукту на ринку. У сучасних мобільних додатках користувачі очікують інтуїтивної, швидкої та естетично привабливої взаємодії, що мінімізує когнітивне навантаження та скорочує шлях до досягнення цілей [15, 16, 17].

У межах даного проєкту, концепція взаємодії базується на таких принципах:

- Контекстність: система адаптує інтерфейс і контент відповідно до цілей, рівня підготовки та звичок користувача.
- Консистентність: усі екрани додатку мають уніфікований стиль, логіку розташування елементів, шрифти, кольорову палітру, що формує стабільне враження від взаємодії.
- Мінімізація кліків: для виконання будь-якої базової дії користувачу не повинно знадобитись більше 3 переходів.
- Пріоритет мобільності: уся навігація оптимізована під одноручне користування смартфоном. Очікувана структура інтерфейсу:
- Домашній екран з інформацією про активні курси, поточний прогрес, нові рекомендації.
- Екран навчального модуля, що дозволяє послідовно переглядати мікроуроки.
- Тестовий модуль, вбудований після кожного блоку контенту для оперативної перевірки знань.
- Панель профілю, що включає розділ з персональними налаштуваннями, досягненнями, мовними та функціональними опціями.
- Сторінка прогресу, де візуалізовано динаміку навчання за допомогою

графіків, бейджів, балів.

Інтерфейс також має передбачати адаптацію для осіб з особливими освітніми потребами, зокрема підтримку масштабування, зміну контрастності, альтернативний текст до зображень тощо [14]. На рис.4.1. представлено порядок взаємодії користувача з екранами мобільного додатку. На рис. 4.2. представлена повна карта екранів додатку.

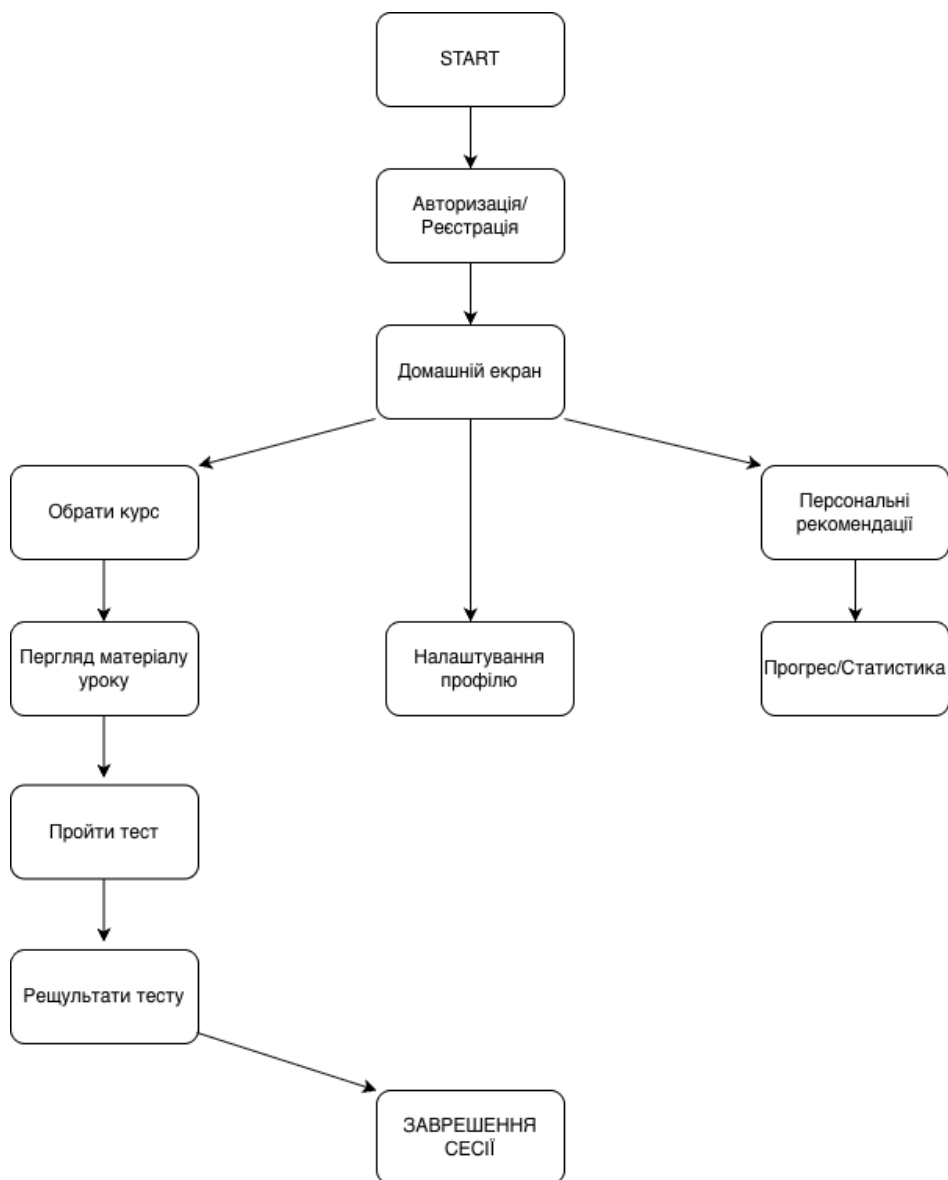


Рис. 4.1. Схематична карта взаємодії користувача з екранами додатку

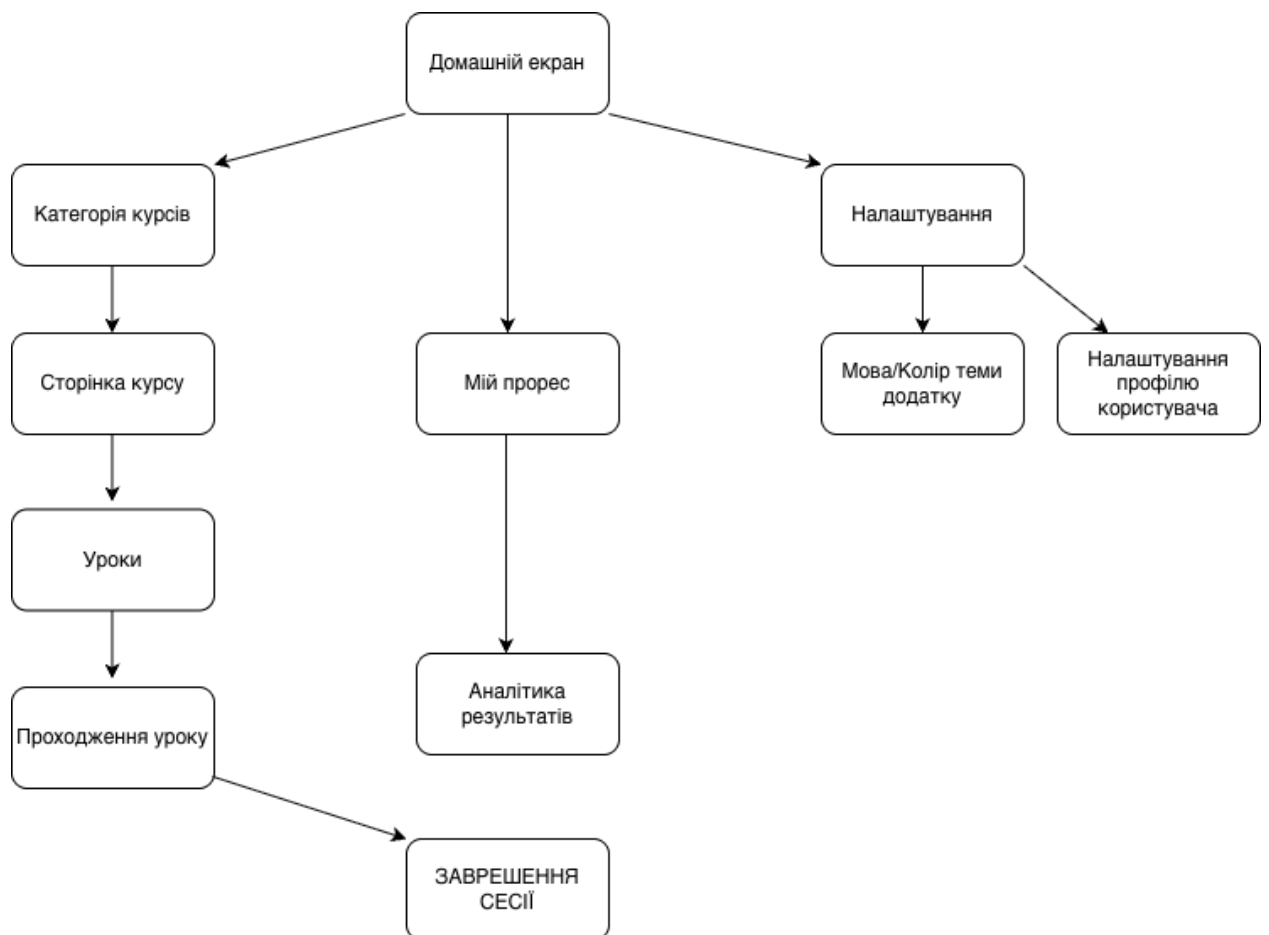


Рис. 4.2. Карта Екранів мобільного додатку

У процесі планування взаємодії користувача з інтерфейсом важливим кроком є створення базових прототипів майбутніх екранів – так званих вайрфреймів (wireframes). Це низькорівнева візуалізація структури сторінки, яка відображає логіку розміщення основних елементів інтерфейсу без акценту на стилізацію чи графічне оформлення.

Метою проєктування вайрфреймів у межах цього проєкту є візуальне змодельовання ключових сценаріїв взаємодії користувача з мобільним додатком мікро-льорнінгу. Це дає змогу:

- оцінити ергономіку інтерфейсу на ранніх етапах;
- уникнути помилок проєктування перед переходом до UI-дизайну;
- забезпечити злагоджену роботу між розробниками, дизайнерами та аналітиками.

У межах роботи було створено базові вайрфрейми таких екранів (рис.4.3, рис. 4.4):

- Екран авторизації – з формою для входу, кнопкою реєстрації та альтернативними способами входу;
- Головний екран – з рекомендаціями курсів, прогресом та доступом до навігації;
- Екран уроку – з навчальним контентом, таймером, елементами взаємодії (вправа, відео, аудіо);
- Екран проходження тесту – із запитаннями, варіантами відповідей, індикатором прогресу;
- Екран статистики – з графіками успішності, отриманими бейджами, прогресом по темах;
- Екран налаштувань профілю – для зміни мови, теми, сповіщень, цілей навчання.

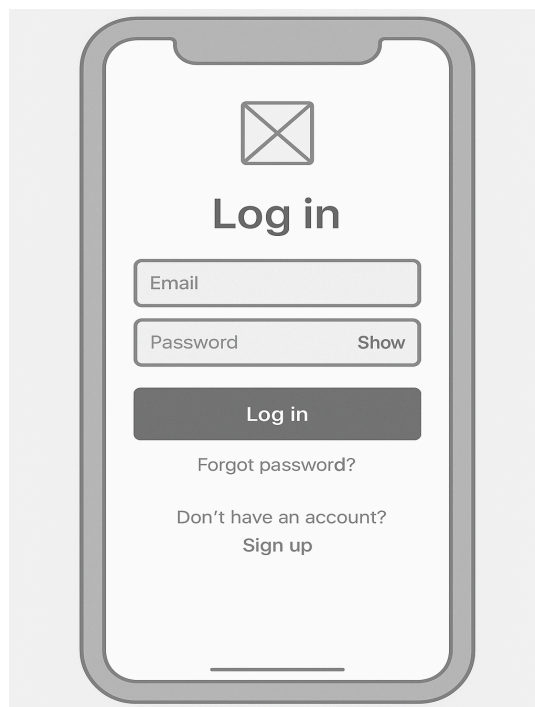


Рис. 4.3. Вайрфрейм екрану авторизації користувача

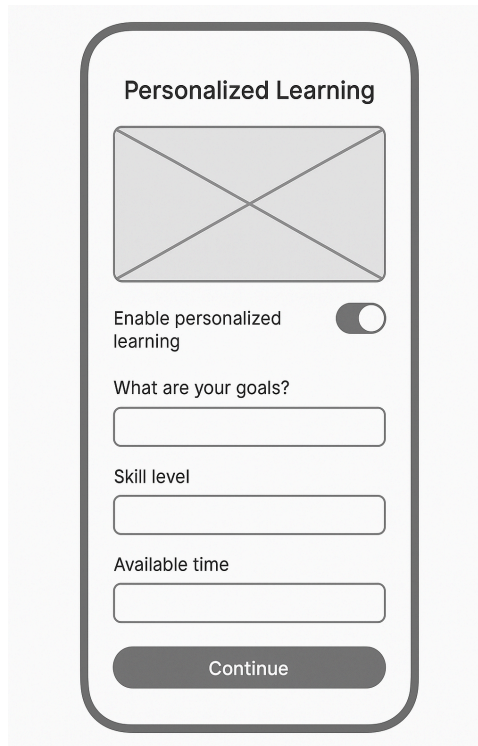


Рис. 4.4. Вайрфрейм екрану налаштувань персоналізованого навчання
Кожен вайрфрейм був розроблений у відповідності до принципів

UX-дизайну:

- інтуїтивність (мінімальна кількість кроків до ключових дій),
- доступність (контрастність, розмір шрифту),
- послідовність (однакова логіка у всіх модулях),
- адаптивність (універсальність для смартфонів різних розмірів).

4.2 Сценарії використання

Сценарії використання (use cases) відіграють ключову роль у формалізації логіки взаємодії користувача з інформаційною системою. Вони відображають типові дії, які виконує користувач для досягнення певної цілі в рамках системи, а також визначають функціональні вимоги до майбутнього мобільного додатку. У даному проєкті сценарії виступають інструментом концептуального моделювання, оскільки практична реалізація прототипу не передбачена.

На основі цільової аудиторії (учні, студенти, фахівці, що самостійно навчаються), було визначено шість базових сценаріїв використання, що

описують основну поведінку користувача в межах інформаційної системи.

Сценарій 1: Реєстрація / Вхід у систему

Перший контакт користувача з додатком починається з етапу ідентифікації. Користувач має можливість зареєструватися, створивши обліковий запис за допомогою електронної пошти або сторонніх сервісів (наприклад, Google). Після підтвердження реєстрації виконується вхід до системи.

Основні кроки:

- відкриття екрану входу;
- введення персональних даних (email, пароль);
- підтвердження реєстрації або авторизація;
- перехід до головної сторінки додатку.

Сценарій 2: Вибір навчального курсу

Після авторизації користувач потрапляє на домашній екран, де має змогу переглядати доступні навчальні курси. Вибір курсу може здійснюватись вручну або автоматично на основі рекомендаційної системи.

Основні кроки:

- ознайомлення з рекомендованими або популярними курсами;
- перегляд опису, тематики, рівня складності;
- натискання кнопки «Почати курс».

Цей сценарій реалізує персоналізацію освітнього шляху, дозволяючи адаптувати навчання до індивідуальних інтересів.

Сценарій 3: Перегляд мікроуроків

Навчальний контент подається у вигляді послідовних коротких модулів (мікроуроків). Користувач переглядає відео, читає текстовий матеріал або виконує інтерактивні завдання.

Основні кроки:

- відкриття мікроуроку;
- перегляд інформаційного матеріалу (відео, текст, слайди);
- виконання коротких дій (наприклад, підтвердження розуміння,

збереження нотаток).

Такий формат відповідає сучасним трендам у сфері мобільного навчання та підтримує принцип «навчання за потребою».

Сценарій 4: Проходження тестування

Після завершення певного блоку контенту користувач має можливість пройти тестування для закріплення знань. Це може бути тест із закритими або відкритими питаннями, з фіксацією часу виконання та автоматичним підрахунком балів.

Основні кроки:

- запуск тесту;
- послідовне проходження запитань;
- надсилання результатів;
- отримання зворотного зв'язку (оцінка, пояснення відповідей, рекомендації).

Сценарій 5: Перегляд результатів і статистики

Після тестування користувач може переглядати загальний прогрес у курсі, статистику правильних відповідей, отримані нагороди (бейджі), а також загальну динаміку навчання.

Основні кроки:

- відкриття вкладки «Прогрес» або «Аналітика»;
- перегляд графіків, списку завершених модулів, рейтингів;
- аналіз своїх досягнень і слабких місць;
- отримання системних рекомендацій.

Сценарій 6: Управління налаштуваннями профілю

Користувач має змогу змінювати параметри свого акаунта, у тому числі мову інтерфейсу, сповіщення, навчальні цілі, рівень складності матеріалів тощо.

Основні кроки:

- відкриття екрану налаштувань;
- вибір потрібної опції (мова, тема, режим навчання);

- збереження змін.

4.3 Планування тестування та критерії якості

У контексті розробки інформаційної системи будь-якого рівня складності важливу роль відіграє планування процедур тестування та визначення критеріїв оцінки функціональності. Хоча в межах цього проєкту не передбачено створення функціонального прототипу, проєктна документація повинна містити чітке уявлення про те, яким чином буде здійснюватися контроль якості програмного продукту на подальших етапах реалізації.

Тестування – це структурований процес перевірки відповідності програмного забезпечення технічним вимогам і очікуванням кінцевих користувачів. Метою є виявлення можливих дефектів, забезпечення коректності логіки роботи системи, а також перевірка її стабільності, зручності використання та безпеки.

Типи тестування, що заплановані до проведення після реалізації системи:

1. Функціональне тестування. Перевірка відповідності фактичної поведінки додатку заявленим функціональним вимогам. Охоплює сценарії реєстрації, навігації, взаємодії з мікроуроками, тестування знань, обробки аналітики тощо.

2. Тестування юзабіліті (зручності використання). Аналіз інтерфейсу з точки зору інтуїтивності, зрозумілості та простоти взаємодії. Зокрема, враховується кількість кроків до цілі, читабельність тексту, логіка переходів між екранами.

3. Тестування сумісності. Перевірка стабільної роботи додатку на різних операційних системах (Android, iOS), версіях пристроїв, типах екранів (смартфони, планшети).

4. Безпекове тестування. Особливо актуальне у випадку збереження персональних даних користувача. Перевірка захищеності від несанкціонованого доступу, правильність реалізації хешування паролів,

безпека API-запитів.

5. Навантажувальне тестування (performance testing). Оцінювання здатності додатку обробляти одночасну взаємодію великої кількості користувачів без зниження продуктивності.

Ключові критерії якості системи

У процесі планування тестування важливо враховувати основні атрибути якості, які будуть використовуватись як метрики успішності функціональної реалізації системи:

- Коректність – усі функції працюють відповідно до технічного завдання.
- Стабільність – відсутність збоїв і помилок у межах стандартного сценарію використання.
- Юзабіліті – інтерфейс легко засвоюється та не викликає труднощів у нових користувачів.
- Масштабованість – архітектура дозволяє розширення функціональності без повного перероблення.
- Безпека – дані користувача захищені згідно з галузевими стандартами.
- Адаптивність – коректна робота інтерфейсу на різних пристроях.
- Швидкодія – мінімальний час відгуку системи при завантаженні курсів, тестів, аналітики.

Методи та інструменти для тестування (рекомендовано для реалізації у майбутньому):

- Unit Testing – для перевірки окремих функціональних модулів (напр., обчислення балів у тесті).
- Manual Testing (ручне тестування) – для тестування сценаріїв, пов'язаних із взаємодією користувача.
- Automated Testing – за допомогою фреймворків (наприклад, Jest, Selenium) для перевірки стабільності після оновлень.
- A/B тестування – у разі впровадження альтернативних інтерфейсних рішень.

На рис. 4.5 зображено інтерації всіх процесів в рамках тестування

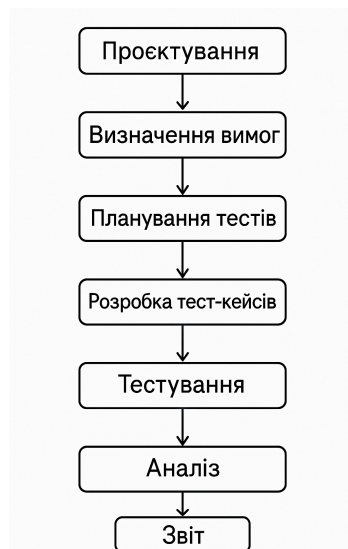


Рис. 4.5. Схема процесу тестування

4.4 Оцінка очікуваної ефективності впровадження системи

Незважаючи на відсутність повноцінної реалізації інтерфейсу, результати концептуального проектування мобільного додатку для персоналізованого мікро-льорнінгу дозволяють зробити попередні висновки щодо його потенційної ефективності в умовах реального освітнього середовища.

Проектна модель системи, що охоплює архітектурну структуру, модулі функціональності, базу сценаріїв використання, логіку взаємодії користувача, а також детальну специфікацію бази даних, створює підґрунтя для практичного застосування системи в освітньому процесі. Це дає змогу сформулювати оцінку очікуваного впливу на ключові показники якості навчання.

Очікувані ефекти від впровадження додатку:

1. Підвищення рівня персоналізації навчання. Завдяки можливості адаптації контенту під потреби конкретного користувача, система дозволяє враховувати темп засвоєння, інтереси та попередні результати. Це сприяє зростанню ефективності засвоєння матеріалу.

2. Зростання внутрішньої мотивації до самонавчання. Елементи гейміфікації (бейджі, прогрес-бари, рейтинги), а також миттєвий зворотний зв'язок створюють стимул до регулярного використання додатку та досягнення нових цілей.

3. Зменшення когнітивного навантаження за рахунок мікро-формату. Поділ навчального матеріалу на короткі модулі (мікроуроки) дозволяє користувачу краще зосередитися, уникати втоми та поступово формувати довготривалі знання.

4. Посилення аналітичної підтримки прийняття рішень. Інформаційна система, завдяки модулю прогресу та звітності, надає користувачу інструменти саморефлексії: графіки динаміки, детальні звіти з тестів, аналіз слабких зон. Це дозволяє оптимізувати власну траєкторію навчання.

5. Гнучкість і доступність. Реалізація додатку у форматі мобільного рішення забезпечує доступ до навчального контенту в будь-який час і в будь-якому місці, що підвищує зручність використання та охоплення аудиторії.

На рис. 4.6. представлено графік очікуваного впливу додатку на цільову аудиторію.



Рис. 4.6. Графік очікуваного впливу додатку на освітній процес користувача

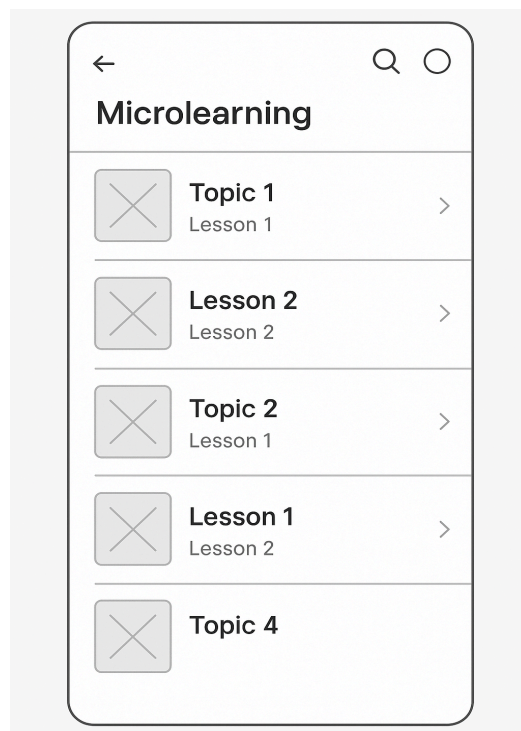


Рис. 4.7. Варфрейм з теми та доступними уроками для користувача

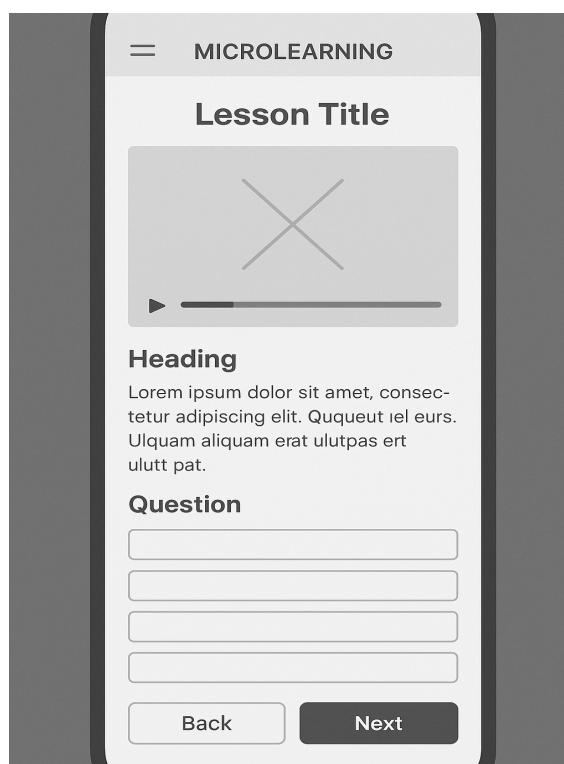


Рис. 4.8. Варфрейм з екраном проходження уроку



Рис. 4.9. Приклад стартової інструкції для користувача

4.5 Юзабіліті-тестування та оцінка користувацького досвіду

Юзабіліті-тестування є одним із ключових етапів оцінки якості мобільного додатку з персоналізованого навчання у форматі мікро-льорнінгу, оскільки ефективність навчального процесу значною мірою залежить від зручності, інтуїтивності та доступності користувацького інтерфейсу. Для освітніх мобільних застосунків особливо важливо забезпечити мінімальний когнітивний бар'єр, швидку навігацію та комфортну взаємодію користувача з навчальним контентом у межах короткотривалих сесій.

Метою юзабіліті-тестування є виявлення проблем інтерфейсу, оцінка зручності виконання основних сценаріїв використання та визначення відповідності додатку очікуванням і потребам цільової аудиторії. У контексті мікро-льорнінгу тестування спрямоване на перевірку того, наскільки легко

користувач може розпочати навчальну сесію, перейти між мікромодулями, отримати зворотний зв'язок та оцінити власний прогрес.

У процесі юзабіліті-тестування були визначені ключові сценарії взаємодії користувачів із системою, зокрема реєстрація та вхід, вибір навчального модуля, проходження мікронавчального контенту, виконання тестових завдань та перегляд статистики результатів. Для кожного сценарію оцінювалися такі показники, як кількість дій для досягнення цілі, зрозумілість навігації, час виконання завдання та наявність помилкових дій.

Оцінка користувацького досвіду здійснювалася на основі якісних та кількісних критеріїв. До кількісних показників належать час взаємодії з інтерфейсом, кількість помилок та рівень завершення сценаріїв. Якісна оцінка базується на суб'єктивному сприйнятті користувачами зручності використання додатку, зрозумілості подачі інформації та загального рівня задоволеності роботою з системою.

Особливу увагу приділено відповідності інтерфейсу принципам мобільного UX-дизайну, таким як мінімалізм, читабельність елементів, адаптація до різних розмірів екранів та забезпечення швидкого доступу до основних функцій. Такий підхід дозволяє зменшити навантаження на користувача та підвищити ефективність освоєння навчального матеріалу у форматі мікро-льорнінгу. Результати юзабіліті-тестування стали основою для внесення коригувань у структуру інтерфейсу, оптимізації навігаційних елементів та вдосконалення сценаріїв взаємодії. Отриманий зворотний зв'язок дозволив підвищити зручність використання мобільного додатку та покращити загальний користувацький досвід, що є важливим чинником успішного впровадження цифрової освітньої системи. Таким чином, проведення юзабіліті-тестування та комплексна оцінка користувацького досвіду забезпечили відповідність мобільного додатку з мікро-льорнінгу вимогам зручності, доступності та ефективності навчального процесу. У табл. 4.1. представлені основні критерії оцінки якості інтерфейсу в рамках тестування продукту.

Критерії оцінки якості інтерфейсу

<i>Критерій UX</i>	<i>Опис критерію</i>	<i>Показники оцінювання</i>
Зрозумілість інтерфейсу	Наскільки інтуїтивно користувач розуміє елементи керування та їх призначення	Кількість помилкових дій, потреба в підказках
Простота навігації	Легкість переходу між екранами та навчальними модулями	Кількість кроків для досягнення цілі
Швидкість виконання завдань	Час, необхідний для виконання типових сценаріїв використання	Середній час виконання сценарію
Зручність читання контенту	Комфорт сприйняття тексту та навчальних матеріалів	Читабельність, розмір шрифту, контрастність
Відповідність мікро-льорнінгу	Прийнятність інтерфейсу до коротких навчальних сесій	Час проходження мікромодуля
Доступність функцій	Швидкий доступ до основних функцій додатку	Кількість дій до запуску навчання

Результати юзабіліті-тестування також підтверджують важливість ітеративного підходу до розробки, зокрема використання методології Scrum. Завдяки включенню UX-оцінювання на кожному етапі життєвого циклу продукту стає можливим системно вдосконалювати інтерфейс, оперативно реагувати на зауваження користувачів та адаптувати функціональні рішення відповідно до реальних сценаріїв використання. Такий підхід забезпечує безперервне підвищення якості продукту, стабільність його роботи та відповідність ключовим принципам користувацько-орієнтованого дизайну, що є критичним чинником для успішного впровадження та масштабування мобільного додатку. Застосування ітеративних циклів дозволяє формувати гнучкий механізм перевірки гіпотез, оптимізувати навантаження на користувача та покращувати структуру навчального контенту на основі фактичних даних взаємодії [30, 31, 32].

ВИСНОВКИ

У результаті виконання даної кваліфікаційної роботи було досягнуто поставленої мети — розроблено проєкт мобільного додатку, що реалізує концепцію персоналізованого навчання у форматі мікро-льорнінгу та відповідає сучасним вимогам до мобільних освітніх інформаційних систем.

У межах дослідження виконано всі завдання. Зокрема:

- Проаналізовано сучасний стан галузі мобільного навчання та тенденції її розвитку, визначено ключові напрями розвитку мікро-льорнінгу та персоналізованих освітніх технологій.
- Здійснено огляд і порівняльний аналіз програм-аналогів, виявлено їхні переваги й недоліки, що дозволило визначити конкурентні характеристики для майбутнього додатку.
- Сформульовано функціональні та нефункціональні вимоги до системи, включно з вимогами до безпеки, масштабованості, продуктивності та UX.
- Розроблено структурне й архітектурне проєктування інформаційної системи, створено концептуальну та логічну моделі бази даних, спроєктовано API та взаємодію між компонентами системи.
- Створено сценарії взаємодії користувача із системою, визначено основні Use Case-и, ролі та логіку поведінкових маршрутів.
- Сформовано OBS та WBS, що забезпечують організаційну та структурну основу управління проєктом.
- Проведено оцінювання економічної ефективності, що підтвердило доцільність і фінансову перспективність реалізації продукту.

Одним із ключових результатів є розроблення адаптивної моделі прийняття рішень, яка забезпечує персоналізацію навчального процесу на основі показників успішності, темпу навчання та індивідуальних характеристик користувача. Вона формує рекомендації щодо вибору наступного мікромодуля та створює основу для подальшого впровадження штучного інтелекту в освітню систему.

У роботі також систематизовано підхід до управління проектом, застосовано сучасні гнучкі методології, зокрема Scrum (з частковими елементами Kanban), що підвищують ефективність планування та контроль за виконанням робіт.

Практичне значення результатів полягає в можливості використання розробленого проекту як готової основи для створення повноцінного мобільного додатку, орієнтованого на персоналізоване короткоформатне навчання. Пропоновані архітектурні рішення, модель адаптивного навчання та UX-підходи можуть застосовуватися в освітніх платформах різного призначення — від самостійного навчання до корпоративних систем підвищення кваліфікації.

Отже, всі цілі кваліфікаційної роботи досягнуті. Розроблений проект мобільного додатку підтверджує можливість створення сучасної, адаптивної, зручної та ефективної системи мікро-льорнінгу. Його впровадження дозволить користувачам отримувати якісний персоналізований контент, підвищувати власну навчальну результативність та ефективно використовувати короткі проміжки часу для розвитку.

Таким чином, виконана робота створює міцне підґрунтя для подальшої розробки, розширення функціоналу, інтеграції інтелектуальних технологій та масштабування системи у межах освітніх та корпоративних сфер.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Coursera [Електронний ресурс] – Режим доступу до ресурсу: <https://www.coursera.org/> (дата звернення: 15.05.2025)
2. Duolingo [Електронний ресурс] – Режим доступу до ресурсу: <https://www.duolingo.com/> (дата звернення: 16.05.2025)
3. Khan Academy [Електронний ресурс] – Режим доступу до ресурсу: <https://www.khanacademy.org/> (дата звернення: 16.05.2025)
4. EdApp by SafetyCulture [Електронний ресурс] – Режим доступу до ресурсу: <https://www.edapp.com/> (дата звернення: 16.05.2025)
5. Firebase Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://firebase.google.com/docs> (дата звернення: 17.05.2025)
6. NestJS Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.nestjs.com/> (дата звернення: 17.05.2025)
7. PostgreSQL Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/> (дата звернення: 18.05.2025)
8. OpenAI GPT API Docs [Електронний ресурс] – Режим доступу до ресурсу: <https://platform.openai.com/docs> (дата звернення: 18.05.2025)
9. The Microlearning Breakthrough: Creating Training That Learners Will Use [Електронний ресурс] – Режим доступу до ресурсу: <https://trainingindustry.com/articles/content-development/microlearning-an-advantageous-and-effective-training-method/> (дата звернення: 19.05.2025)
10. Mobile Learning: Transforming the Delivery of Education and Training [Електронний ресурс] – Режим доступу до ресурсу: <https://unesdoc.unesco.org/ark:/48223/pf0000219641> (дата звернення: 19.05.2025)
11. Principles of Instructional Design in Mobile Environments [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/50224316_Universal_Instructional_Design_Principles_for_Mobile_Learning (дата звернення: 20.05.2025)

12. User Experience in Mobile Learning Applications: A Systematic Review [Электронный ресурс] – Режим доступа до ресурсу: <https://dl.acm.org/doi/10.1145/3313831.3376418> (дата звернення: 20.05.2025)
13. Agile Project Management for Software Development [Электронный ресурс] – Режим доступа до ресурсу: <https://www.scrum.org/resources/what-is-scrum> (дата звернення: 20.05.2025)
14. Kanban Guide for Agile Teams [Электронный ресурс] – Режим доступа до ресурсу: <https://kanban.university/kanban-guide/> (дата звернення: 21.05.2025)
15. Mobile App Design Best Practices [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.android.com/design> (дата звернення: 21.05.2025)
16. Human–Computer Interaction: UX Methods and Tools [Электронный ресурс] – Режим доступа до ресурсу: <https://www.interaction-design.org/literature/topics/human-computer-interaction> (дата звернення: 21.05.2025)
17. Machine Learning for Adaptive Educational Systems [Электронный ресурс] – Режим доступа до ресурсу: <https://www.sciencedirect.com/science/article/pii/S074756322100240X> (дата звернення: 21.05.2025)
18. Evaluation of Mobile Learning Applications Using UX Metrics [Электронный ресурс] – Режим доступа до ресурсу: <https://www.tandfonline.com/doi/abs/10.1080/10494820.2020.1821860> (дата звернення: 22.05.2025)
19. Learning Experience Design Foundations [Электронный ресурс] – Режим доступа до ресурсу: <https://www.nngroup.com/articles/learning-experience-design/> (дата звернення: 22.05.2025)
20. RESTful API Design Principles [Электронный ресурс] – Режим доступа до ресурсу: <https://restfulapi.net/> (дата звернення: 23.05.2025)
21. Cloud Architecture for Scalable Mobile Apps [Электронный ресурс] – Режим

- доступу до ресурсу: <https://cloud.google.com/architecture> (дата звернення: 23.05.2025)
22. Best Practices for Secure Mobile App Development [Електронний ресурс] – Режим доступу до ресурсу: <https://owasp.org/www-project-mobile-security/> (дата звернення: 23.05.2025)
23. Buchem, I., & Hamelmann, H. (2010). Microlearning: A strategy for ongoing professional development. *eLearning Papers*, 21, 1–15.
24. Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3), e1355. <https://doi.org/10.1002/widm.1355>.
25. Panigrahi, R., Srivastava, P. R., & Sharma, D. (2024). *E-learning personalization: Trends, challenges and opportunities. Education and Information Technologies*, 29, 4263–4287.
26. Kaur, G., Singh, D., & Patel, J. R. (2022). Usability and user experience in mobile learning: A systematic review. *Education and Information Technologies*, 27, 6011–6034.
27. Sharples, M., Pea, R., McAndrew, P., et al. (2021). *Innovating pedagogy 2021*. The Open University.
28. OECD. (2024). *Education at a glance 2024: OECD indicators*. OECD Publishing.
29. IEEE. (2021). *IEEE Std 7010 — Recommended practice for assessing the impact of autonomous and intelligent systems on human well-being*. IEEE.
30. Schwaber K., Sutherland J. *The Scrum Guide : The Definitive Guide to Scrum : The Rules of the Game* [Електронний ресурс]. – 2020. – Режим доступу: <https://scrumguides.org/scrum-guide.html> (дата звернення: 20.05.2025).
31. Nielsen J., Budiu R. *Mobile usability*. – Berkeley : New Riders, 2013. – 216 p.
32. Shneiderman B., Plaisant C., Cohen M. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. – Pearson, 2017. – 672 p.
33. Pressman R., Maxim B. *Software Engineering: A Practitioner's Approach*. – McGraw-Hill, 2019. – 928 p.
34. Larman C. *Applying UML and Patterns*. – Prentice Hall, 2005. – 736 p
35. Biggs J., Tang C. *Teaching for Quality Learning at University*. – McGraw-Hill Education, 2011. – 418 p.

36. Bass L., Clements P., Kazman R. *Software Architecture in Practice*. – Addison-Wesley, 2021. – 624 p.

ДОДАТКИ

Додаток А

Таблиця А.1

Паспорту проєкту

Цілі проєкту	<ol style="list-style-type: none">1. Створення мобільного додатку для персоналізованого навчання у форматі мікро-льорнінгу.2. Адаптивна подача навчального контенту відповідно до індивідуальних потреб користувача.3. Структурування контенту у вигляді коротких мікроуроків.4. Збереження та аналіз прогресу користувача.5. Рекомендації щодо подальшого навчання на основі результатів.6. Мотивація через гейміфікацію (бейджі, рівні, бали).7. Інтуїтивно зрозумілий та зручний мобільний інтерфейс.
Зацікавлені сторони проєкту	<p>Замовник: Інвестор</p> <p>Команда проєкту:</p> <ol style="list-style-type: none">1. Project Manager2. Системний аналітик3. Програмісти (Backend, Frontend)4. UI/UX дизайнер5. Інженер з якості (QA)6. Системний адміністратор7. Технічна підтримка8. Контент-менеджер9. Бізнес-Аналітик <p>Користувачі: Люди, які прагнуть вивчати новий матеріал за допомогою персоналізованого плану у форматі коротких фактів та уроків в ігровому форматі</p>

Результат проекту	<ol style="list-style-type: none"> 1. Проаналізовано галузь мобільного навчання з акцентом на персоналізацію освітнього процесу та мікро-льорнінг як сучасний формат подачі знань; 2. Виконано аналіз програм-аналогів, виявлено їхні обмеження та окреслено переваги запропонованої концепції; 3. Сформульовано функціональні й нефункціональні вимоги, враховуючи потреби кінцевих користувачів, гнучкість системи та зручність взаємодії; 4. Спроектовано архітектуру системи з описом ключових модулів, що охоплюють автентифікацію, навчальний контент, тестування, аналітику прогресу, персоналізацію; 5. Розроблено концептуальну, логічну та фізичну моделі бази даних, які забезпечують структуроване зберігання та обробку навчальної інформації; 6. Сформовано діаграми та візуальні моделі, зокрема: діаграму варіантів використання, карту екранів, user flow, а також макети (wireframes) основних інтерфейсів додатку; 7. Сформульовано план тестування та критерії оцінки якості, які можуть бути використані на етапі реалізації та впровадження; 8. Оцінено очікувану ефективність додатку, що полягає в підвищенні мотивації до навчання, зменшенні когнітивного навантаження та підтримці індивідуального освітнього треку користувача.
Життєвий цикл ПЗ	<p>Проект передбачає реалізацію протягом 6-8 місяців.</p> <p>Етапи проекту:</p> <ol style="list-style-type: none"> 1.1.Планування та аналіз проекту <ol style="list-style-type: none"> 1.1.1. Визначення обсягу та цілей проекту 1.1.2. Визначення вимог та очікувань користувачів 1.1.3. Аналіз потенційних ризиків та ресурсів 1.1.4. Розробка графіка виконання проекту 1.1.5. Встановлення бюджету та ресурсів 1.2.Концептуальне проектування <ol style="list-style-type: none"> 1.2.1. Розробка концепції інформаційної системи 1.2.2. Визначення функціональних на нефункціональних вимог до системи 1.2.3. Проведення необхідних досліджень 1.2.4. Побудова всіх необхідних діаграм 1.2.5. Створення структурної схеми БД 1.2.6. Вибір технологій та інструментів розробки 1.2.7. Розробка документації проекту

Життєвий цикл ПЗ	<p>1.3. Реалізація</p> <p>1.3.1. Розробка програмного забезпечення</p> <p>1.3.2. Розробка інтерфейсу користувача</p> <p>1.3.3. Розробка плану тестування</p> <p>1.3.4. Оптимізація та вдосконалення ІС</p> <p>1.3.5. Розробка інструкцій для ІС</p> <p>1.3.5.1. Інструкція користувача</p> <p>1.3.5.2. Інструкція адміністратора</p> <p>1.4. Тестування</p> <p>1.4.1. Проведення модульних тестів ІС</p> <p>1.4.2. Виконання інтеграційних тестів</p> <p>1.4.3. Перевірка на відповідність функціональним та технічним вимогам</p> <p>1.4.4. Виконання тестів на навантаження</p> <p>1.4.5. виправлення помилок та дефектів</p> <p>1.5. Впровадження</p> <p>1.5.1. Підготовка документації для впровадження</p> <p>1.5.2. Навчання персоналу щодо використання ІС</p> <p>1.5.3. Проведення пілотного запуску системи</p> <p>1.5.4. Вирішення проблем, виявлених під час пілотного запуску</p> <p>1.5.5. Повне впровадження системи на всіх робочих місцях</p> <p>1.6. Завершення та підтримка</p> <p>1.6.1. Формування завершальної документації проєкту</p> <p>1.6.2. Передача системи у підтримку та експлуатацію</p> <p>1.6.3. Забезпечення технічної підтримки користувачам</p> <p>1.6.4. Фінальна презентація проєкту замовнику</p> <p>9. Здача проєкту замовнику</p>
------------------	--

Таблиця А.2

Характеристика API взаємодії з навчальним матеріалом

Параметр	Опис
<i>Призначення API</i>	Фіксація дій користувача під час проходження навчального матеріалу
<i>Типові операції</i>	Початок проходження модуля; збереження результатів виконання завдань; завершення модуля
<i>Структура запиту</i>	Ідентифікатор користувача; ідентифікатор мікромодуля; результати завдань; час виконання; статус проходження

Таблиця А.3

Характеристика API модуля персоналізації навчання

Параметр	Опис
<i>Призначення API</i>	Формування та адаптація індивідуального навчального плану користувача
<i>Типові операції</i>	Отримання персоналізованих рекомендацій; оновлення навчального плану; адаптація рівня складності контенту
<i>Структура відповіді</i>	Перелік рекомендованих мікромодулів; рівень пріоритетності; обґрунтування рекомендацій; очікуваний результат навчання

Таблиця А.4

Характеристика API модуля персоналізації навчання

Параметр	Опис
<i>Призначення API</i>	Відображення та аналіз результатів навчальної діяльності користувача
<i>Типові операції</i>	Кількість пройдених модулів; рівень успішності; динаміка навчального прогресу; рекомендації щодо подальших дій
<i>Структура відповіді</i>	Агреговані числові показники; часові інтервали; індикатори прогресу

Програмний код, таблиці та інструкції

```
CREATE TABLE Progress (  
  progress_id SERIAL PRIMARY KEY,  
  user_id INT REFERENCES Users(user_id),  
  lesson_id INT REFERENCES Lessons(lesson_id),  
  status VARCHAR(50),  
  score INT,  
  date_completed DATE  
);  
  
CREATE TABLE Badges (  
  badge_id SERIAL PRIMARY KEY,  
  user_id INT REFERENCES Users(user_id),  
  title VARCHAR(100),  
  description TEXT,  
  awarded_date DATE  
);  
  
CREATE TABLE Preferences (  
  pref_id SERIAL PRIMARY KEY,  
  user_id INT REFERENCES Users(user_id),  
  language VARCHAR(50),  
  notifications BOOLEAN,  
  recommendations_enabled BOOLEAN  
);
```

Рис. Б.1 Код із створення таблиць БД

```
@Injectable()  
export class AuthService {  
  constructor(  
    private userService: UsersService,  
    private jwtService: JwtService,  
  ) {}  
  
  async validateUser(email: string, password: string): Promise<any> {  
    const user = await this.userService.findByEmail(email)  
    if (user && await bcrypt.compare(password, user.password)) {  
      // eslint-disable-next-line @typescript-eslint/no-unused-vars  
      const [password, ...result] = user  
      return result  
    }  
    return null  
  }  
  
  async login(user: any) {  
    const payload = { username: user.email, sub: user.id }  
    return {  
      access_token: this.jwtService.sign(payload),  
    }  
  }  
  
  async register(userDto: any) {  
    const hashedPassword = await bcrypt.hash(userDto.password, 10)  
    return this.userService.create({  
      ...userDto,  
      password: hashedPassword,  
    })  
  }  
}
```

Рис. Б.2 Код для модуля авторизації

```

function getNextLearningStep(S, T, P = {}) {
  /** Базова рекомендація за замовчуванням */
  let recommendation = {
    action: 'відати модуль аналогічної складності',
    difficultyChange: 'same',
    tempo: 'normal',
    tags: [],
    explanation: 'Базова рекомендація за замовчуванням.';
  };

  // 1. Логіка за успішністю S та часом T (згідно Таблиці 2.2)

  if (S ≥ 80 && T ≡ 'fast') {
    // S ≥ 80% і T швидкий
    recommendation = {
      action: 'перейти до модуля значно вищої складності',
      difficultyChange: 'up2',
      tempo: 'faster',
      tags: ['high_score', 'fast_time'],
      explanation:
        'Висока успішність і швидке виконання – підвищення складності на два рівні, пришвидш
    };
  } else if (S ≥ 80) {
    // S ≥ 80%
    recommendation = {
      action: 'перейти до модуля вищої складності',
      difficultyChange: 'up1',
      tempo: 'normal',
      tags: ['high_score'],
      explanation:
        'Висока успішність – матеріал засвоєно добре, рекомендується підвищення складності.';
    };
  }
}

```

Рис. Б.3 Код для модуля рекомендацій по навчальному плану користувача