

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп’ютерні науки
Освітньо-професійна програма «Управління проєктами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження процесів управління проєктом створення мобільного додатку для користування мережею коворкінгів «Coworkly»

Студентки 2-го курсу групи УПз-21

Науковий керівник:

Ганни ДЕМЧУК

(ім'я, прізвище)

к.т.н., доцент

(науковий ступінь, вчене звання)

Вадим ЗЮЗЮН

(ім'я, прізвище)

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: “До захисту в Екзаменаційній комісії”)

Завідувач кафедри
технологій
управління

Віктор МОРОЗОВ

(підпис)

(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 проєктами

ЗАТВЕРДЖУЮ

Завідувач кафедри

професор Віктор МОРОЗОВ

«29» вересня 2025 року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студентка: Ганна ДЕМЧУК

Група: УПз-21

1. Тема кваліфікаційної роботи: «Дослідження процесів управління проєктом створення мобільного додатку для користування мережею коворкінгів «Coworkly»

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: дослідження методів та інструментів управління проєктами, аналіз існуючих платформ і концепцій створення інформаційних систем; обґрунтування вибору методології Agile(Scrum); формування структури команди, управління ризиками та ресурсами, розробка прототипу платформи.

4. Зміст роботи: Обґрунтування доцільності та життєздатності проєкту. Визначення зацікавлених сторін, аналіз досліджень та публікацій. Проведення SWOT-аналізу, постановка задачі та розробка паспорту проєкту. Розробка інформаційного забезпечення проєкту, визначення функціональних та нефункціональних вимог, формування Use Case елементів. Аналіз методологій управління проєктами, вибір Agile, імплементація фреймворку Scrum.

Планування беклогу та спринтів. Ідентифікація та кількісний аналіз ризиків, розробка плану реагування. Розробка управління якістю проєкту. Опис структури програмного забезпечення, розробка концептуальної та логічної моделей бази даних, створення інтерфейсів додатку, формування пропозицій для успішного функціонування продукту.

5. Перелік графічного матеріалу: дерево цілей, дерево причин та наслідків, концептуальна модель IC, OBS проєкту, діаграма варіантів використання, діаграма Ганта, WBS проєкту, модель архітектури, концептуальна модель бази даних, логічна модель бази даних, інтерфейси ПЗ.

6. Календарний план виконання роботи

№ з/п	Назва частин роботи	Виконання роботи
1	Аналіз літературних джерел з предмету дослідження	01.10.25-10.10.25
2	Збір матеріалів для обґрунтування проблеми та доцільності дослідження	13.10.25-17.10.25
3	Складання плану кваліфікаційної роботи	20.10.25-22.10.25
4	Узгодження плану роботи з науковим керівником.	23.10.25-24.10.25
5	Підготовка розділу 1	27.10.25-31.10.25
6	Підготовка розділу 2	03.11.25-07.11.25
7	Підготовка розділу 3	10.11.25-14.11.25
8	Підготовка розділу 4	17.11.25-21.11.25
9	Погодження та внесення змін, рекомендацій від наукового керівника.	24.11.25-26.11.25
10	Оформлення кваліфікаційної роботи	27.11.25-28.11.25
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 Вадим ЗЮЗЮН

(посада, ім'я, прізвище)

_____ (підпис)

Завдання прийняла до виконання студентка групи УПз-21

Ганна ДЕМЧУК

(ім'я, прізвище)

_____ (підпис)

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження процесів управління проектом створення мобільного додатку для користування мережею коворкінгів «Coworkly»

Студентки: Демчук Ганни Максимівни

Науковий керівник: Зюзюн Вадим Ігорович

Рік захисту – 2025

Мета кваліфікаційної роботи полягає у дослідженні та реалізації процесів управління створенням продукту мобільного крос-платформенного додатку «Coworkly», спрямованого на подолання децентралізації ринку коворкінгів.

Ціль проекту – створення мобільного кросс-платформенного додатку, який забезпечить централізований пошук коворкінгів, функціонал гнучкого та миттєвого бронювання та інтеграцію AI-рекомендацій, згідно з визначеним бюджетом та вимогами якості.

Об'єктом дослідження є процеси розробки і впровадження мобільного додатку, спрямованого на централізацію послуг в сфері коворкінгів.

Предметом дослідження є методи та інструменти управління проектами, які забезпечують організацію, моніторинг і реалізацію складного функціоналу у мобільній розробці.

Практична цінність роботи полягає у створенні проекту мобільного додатку, готового до передачі команді розробників для етапу реалізації MVP. Продукт має високе функціональне значення, забезпечуючи централізацію ринку коворкінгів. Завдяки реалізації швидкого бронювання, актуального відображення завантаженості в реальному часі та AI-персоналізації, він усуває необхідність ручного опрацювання запитів та підвищує комфорт для користувачів, роблячи процес пошуку та оренди робочого місця зручним і швидким. Розроблена модель управління, що поєднує гнучкість Scrum і фінансову передбачуваність, може бути адаптована для управління іншими IT-

стартапами, а сформовані результати слугують практичним прикладом застосування інструментів ІТ-проектного менеджменту.

Кваліфікаційна робота містить анотацію, вступ, основну частину, яка включає чотири розділи, висновки, перелік використаних інформаційних джерел та додатки.

Перший розділ містить аналіз та огляд проблеми, останніх досліджень та публікацій в сфері коворкінгів, визначення зацікавлених сторін. В даному розділі проведено SWOT – аналіз, поставлено задачу та розроблено паспорт проекту.

Другий розділ охоплює розробку інформаційного забезпечення продукту, що в свою чергу містить розробку дерева причин та наслідків, концептуальної моделі системи, визначення функціональних та нефункціональних вимог проекту, формування Use Case елементів та створення діаграми варіантів використання.

Третій розділ містить проведений аналіз методологій управління проектами та присвячений методологій Agile та фреймворку Scrum. В даному розділі створено і визначено беклог продукту, розписано спринти, проведено кількісний аналіз ризиків, розроблено план реагування, бюджет проекту та описано план управління якістю.

Четвертий розділ складається з розробки додатку для користування, опис структури та архітектури програмного забезпечення, створення концептуальної та логічної моделі для бази даних. Також в цьому розділі наведено прототипи інтерфейсів для додатку, та сформовано перелік пропозицій для успішного використання продукту.

Робота завершується висновками щодо результатів дослідження та рекомендаціями для подальшого розвитку платформи.

Кваліфікаційна робота складається з 93 сторінок основного тексту, містить 21 рисунок, 10 таблиць та 4 сторінки додатків.

Ключові слова: управління проектами, коворкінги, мобільний додаток, Agile, гнучкі технології, Scrum.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ.....	11
1.1 Аналіз проблеми відсутності якісних інформаційних інструментів для планування та проведення коворкінгів.....	11
1.2 Аналіз останніх досліджень та публікацій.....	13
1.3 Визначення зацікавлених сторін проєкту.....	17
1.4 SWOT-аналіз проєкту.....	20
1.5 Постановка задачі дослідження, формулювання технічного завдання на розробку у вигляді паспорту проєкту.....	24
РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ	28
2.1 Побудова дерева причин та наслідків.....	28
2.2 Розробка концептуальної моделі інформаційної системи.....	28
2.3 Визначення функціональних та нефункціональних вимог до продукту ІТ проєкту.....	33
2.4 Формування Use Case елементів до функціональних вимог та побудова Use Case Diagram.....	35
РОЗДІЛ 3. ТЕХНОЛОГІЇ УПРАВЛІННЯ ПРОЄКТОМ ТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ КОРИСТУВАННЯ МЕРЕЖЕЮ КОВОРКІНГІВ «COWORKLY».....	37
3.1 Аналіз методологій управління проєктами.....	37
3.2 Імплементация фреймворку Scrum. Визначення User Story та формування беклогу продукту проєкту.....	40
3.3 Планування спринтів та розробка Діаграми Ганта.....	46
3.4 Управління ризиками проєкту.....	54
3.4.1 Вибір методології та структури управління.....	54
3.4.2 Ідентифікація та якісна оцінка ризиків.....	55
3.4.3 Кількісний аналіз та ранжування ризиків.....	56
3.4.4 Розробка протиризованих заходів (План реагування).....	59

3.5 Розробка бюджету проєкту	62
3.6 Розробка плану управління якістю в проєкті	63
РОЗДІЛ 4. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ КОРИСТУВАННЯ МЕРЕЖЕЮ КОВОРКІНГІВ «COWORKLY»	71
4.1 Опис структури програмного забезпечення та його архітектури	71
4.2 Розробка концептуальної та логічної моделей бази даних проєкту.....	74
4.3 Розробка інтерфейсів програмного забезпечення.....	77
4.4 Формування переліку пропозицій щодо успішного функціонування сервісу продукту	85
ВИСНОВКИ	87
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	89
ДОДАТКИ.....	94

ВСТУП

Сьогоднішній ринок праці динамічно трансформується під впливом технологічних інновацій та глобальних соціальних змін, зокрема, переходу до гібридного та віддаленого форматів роботи. Цей зсув вимагає нових, ефективних рішень для організації робочого простору, де коворкінги перетворюються з тимчасового тренду на невід'ємну частину сучасної бізнес-інфраструктури. В Україні, незважаючи на складні економічні та безпекові виклики, ринок коворкінгів демонструє потенціал до зростання, слугуючи центрами стабільності та відновлення для бізнесу та ІТ-спільноти. У цьому контексті виникає гостра потреба в інструментах, які б ефективно агрегували та управляли цим розподіленим ресурсом.

Актуальність кваліфікаційної роботи магістра обумовлена потребою трансформації робочого середовища, тобто перехід значної частини професій до гнучкого режиму праці, що підвищує попит на гнучкі робочі місця, що стимулює розвиток агрегаторів послуг коворкінгів; технологічна складність продукту, яка передбачає інтеграцію передових технологій, таких як обробка даних у реальному часі, для відображення завантаженості, та використання елементів штучного інтелекту для надання персоналізованих рекомендацій користувачам. Це створює високий ступінь ризику, що вимагає застосування структурованого та адаптивного підходу до управління проектом; потреба в гібридній моделі управління для забезпечення фінансової передбачуваності (яку вимагає стартап-інвестування) та одночасно гнучкості розробки. Відповідно виникла потреба у розробці мобільного додатку на кросплатформенній технології, який виступатиме єдиним централізованим хабом для користувачів коворкінгів. Додаток надасть користувачам можливість у реальному часі знаходити, порівнювати та бронювати робочі місця у партнерській мережі коворкінгів. Ключовою перевагою є використання модуля штучного інтелекту для персоналізації, який підвищує швидкість і точність вибору робочого місця, адаптуючи пропозиції під індивідуальні потреби користувача та його історію бронювань.

Метою роботи є дослідження та реалізація процесів управління створенням продукту мобільного крос-платформенного додатку «Coworkly», спрямованого на подолання децентралізації ринку коворкінгів. Досягнення мети забезпечується шляхом інтеграції передових технологій розробки програмного забезпечення та AI-персоналізації з адаптивною методологією Scrum, а також застосуванням стандартизованих підходів до управління якістю та мінімізації проєктних ризиків.

Для досягнення мети потрібно виконати наступні завдання:

1. Проаналізувати стан ринку коворкінгів, провести SWOT-аналіз та сформувану бізнес-модель проєкту, обґрунтовуючи його доцільність та життєздатність.

2. Розробити концептуальну модель, архітектуру системи та сформувану деталізований перелік функціональних і нефункціональних вимог до продукту.

3. Обґрунтувати вибір методології управління проєктом та розробити його детальний план реалізації.

4. Ідентифікувати, класифікувати та розробити стратегії реагування на ключові ризики проєкту, включаючи зовнішні та технічні ризики.

5. Обґрунтувати програмне забезпечення додатку і його архітектуру. Описати інфраструктуру ПЗ. Створити структури управління та сформувану перелік пропозицій щодо подальшого успішного функціонування, масштабування та диверсифікації монетизації сервісу.

Об'єктом дослідження є процеси розробки і впровадження мобільного додатку, спрямованого на централізацію послуг в сфері коворкінгів.

Предметом дослідження є методи та інструменти управління проєктами, які забезпечують організацію, моніторинг і реалізацію складного функціоналу у мобільній розробці.

Методи дослідження:

1. Аналіз останніх досліджень та публікацій: вивчення ринку, конкурентного середовища та SWOT аналіз.

2. Кейс-аналіз: вибір найбільш оптимальних технологій та інструментів розробки на основі успішного досвіду інших проєктів.

3. Фреймворк Scrum: застосовано для структурування обсягу робіт та планування ітераційної розробки.

4. Кількісний та якісний аналіз ризиків: оцінка ймовірності та впливу ризиків на фінансову та часову складові проєкту.

Практична цінність полягає у створенні проєкту мобільного додатку, готового до передачі команді розробників для етапу реалізації MVP. Продукт має високе функціональне значення, забезпечуючи централізацію ринку коворкінгів. Завдяки реалізації швидкого бронювання, актуального відображення завантаженості в реальному часі та AI-персоналізації, він усуває необхідність ручного опрацювання запитів та підвищує комфорт для користувачів, роблячи процес пошуку та оренди робочого місця зручним і швидким. Розроблена модель управління, що поєднує гнучкість Scrum і фінансову передбачуваність, може бути адаптована для управління іншими IT-стартапами, а сформовані результати слугують практичним прикладом застосування інструментів IT-проєктного менеджменту.

Апробація результатів:

1. Ziuziun V., Demchuk G. Informational architecture and UI/UX design of a mobile application for coworking spaces, 2025, 102-110. Section: Information Technology, Cyber Security and Computer Engineering. <https://doi.org/10.70286/eoss-24.11.2025.002>

РОЗДІЛ 1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1 Аналіз проблеми відсутності якісних інформаційних інструментів для планування та проведення коворкінгів

У сучасному цифровому середовищі, де гнучкі формати праці, зокрема коворкінги, набувають дедалі більшої популярності, виникає гостра потреба в якісних технологічних рішеннях. Зростання попиту на гібридні та дистанційні моделі роботи, особливо серед фрілансерів, стартапів, IT-спеціалістів та малого бізнесу, висвітлює низку проблем, пов'язаних із відсутністю ефективних інформаційних інструментів для організації робочого процесу в коворкінгах.

Основна проблемна область полягає в неефективній організації доступу до послуг коворкінгів. Вирішення таких завдань, як бронювання робочих місць, перегляд завантаженості в реальному часі, оплата послуг та вибір найближчої зручної локації, часто здійснюється вручну або через застарілі, неінтегровані веб-платформи, що є вкрай незручним для сучасного, вимогливого користувача.

Ключові проблеми, що свідчать про відсутність якісних інформаційних інструментів, можна сформулювати наступним чином:

1. Відсутність централізованої платформи: користувачі змушені витратити значний час на пошук інформації на десятках окремих сайтів, кожен з яких має власний інтерфейс та логіку бронювання. Це не лише ускладнює процес порівняння умов та вартості, але й створює ризик подвійних бронювань та призводить до втрати потенційних клієнтів для самих коворкінгів через їхню низьку видимість в інформаційному просторі.

2. Недостатня інформованість: існує гостра проблема браку актуальної та деталізованої інформації. Користувачам потрібні не лише дані про наявність вільних місць, а й специфічні деталі: чи є вільні місця у «тихій зоні», чи доступний конференц-зал з проєктором на конкретну годину, чи працює

кав'ярня. Відсутність таких даних у режимі реального часу робить планування робочого дня непередбачуваним та неефективним.

3. Складність у порівнянні: через різноманітність представлення інформації та відсутність єдиних стандартів, користувачам важко об'єктивно порівнювати вартість та умови. Один коворкінг може пропонувати погодинну оплату, інший – денний абонемент, а третій – пакет послуг. Без єдиної системи фільтрації та порівняння, вибір оптимального варіанту перетворюється на складний аналітичний процес, який вимагає ручного зведення даних.

4. Відсутність персоналізованого підходу: існуючі системи здебільшого функціонують як прості дошки оголошень і рідко пропонують персоналізовані рекомендації. Вони не враховують індивідуальні потреби користувача, як-от переваги щодо умов праці (наприклад, наявність додаткового монітора), типовий графік роботи чи бюджет. Це призводить до того, що користувач щоразу змушений проходити весь шлях пошуку «з нуля», замість того, щоб отримувати проактивні, релевантні пропозиції.

5. Низький рівень інтеграції: спостерігається погана інтеграція з екосистемою сучасних цифрових сервісів. Процес оплати часто вимагає ручного введення даних банківської картки, замість використання зручних методів в один клік (Apple Pay, Google Pay). Крім того, відсутність інтеграції з навігаційними системами змушує користувача вручну копіювати адресу і шукати маршрут в іншому додатку, що розриває користувацький досвід та створює додаткові незручності.

Таким чином, відсутність єдиного, зручного та інтелектуального інструменту для взаємодії з мережею коворкінгів створює значні бар'єри як для клієнтів, так і для власників просторів, що підтверджує високу актуальність та практичну доцільність розробки спеціалізованого мобільного додатку, здатного вирішити перелічені проблеми.

1.2 Аналіз останніх досліджень та публікацій

Після проведеного аналізу з попиту коворкінгів в Європі, інформація свідчить про зростання обсягу таких робочих просторів. Обсяг європейського ринку коворкінгів у 2025 році оцінюється в 7,23 млрд доларів США, а до 2030 року досягне 11,84 млрд доларів США, що свідчить про середньорічне зростання на рівні 10,37% протягом прогнозованого періоду майбутніх п'яти років [1]. Також під час проведення аналізу були виявлені тенденції ринку коворкінгів у Європі (рис. 1.1).

По-перше, важливим каталізатором розвитку ринку стали глобальні зміни у підходах до організації праці, спричинені пандемією COVID-19. Масовий перехід на віддалену роботу виявив як її переваги, так і недоліки, зокрема соціальну ізоляцію та зниження продуктивності в домашніх умовах. Як наслідок, компанії по всьому світу почали активно впроваджувати гібридні моделі, де коворкінги стали «третім місцем» між домом та центральним офісом. Згідно зі звітом JLL за 2024 рік, понад 50% великих корпорацій планують збільшити використання гнучких офісних просторів як частини своєї стратегії утримання талантів та оптимізації витрат на нерухомість.

По-друге, сучасні дослідження фіксують значну трансформацію цільової аудиторії коворкінгів. Якщо раніше основними клієнтами були фрілансери та стартапи, то сьогодні все більшу частку займають корпоративні клієнти. Такі компанії, як Microsoft, Amazon та Meta, використовують мережі коворкінгів для розміщення своїх розподілених команд, реалізуючи так звану «hub-and-spoke» [2] модель (центральний хаб та мережа сателітних офісів). Це дозволяє компаніям бути ближчими до талантів, скорочувати час на дорогу для співробітників та гнучко масштабувати кількість робочих місць.

По-третє, почали з'являтися перші спроби централізації у вигляді онлайн-платформ-агрегаторів (наприклад, Coworker.com, LiquidSpace). Їх існування підкреслює попит на централізовані рішення, але водночас вказує на незаповнену нішу. Більшість таких платформ мають суттєві обмеження: вони рідко пропонують синхронізацію з календарями коворкінгів у реальному часі,

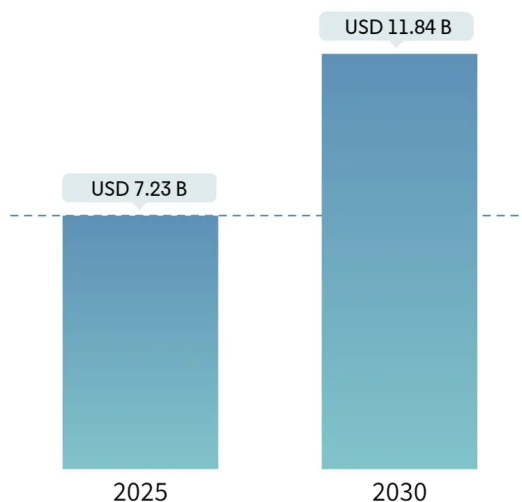
не мають інтегрованих інструментів для миттєвого бронювання та оплати і майже не використовують інтелектуальні алгоритми для персоналізації, що підтверджує необхідність створення більш технологічно просунутих систем.

Окремо варто відзначити унікальну роль, яку коворкінги відіграють в українському суспільстві, особливо після повномасштабного вторгнення. В умовах постійних відключень електроенергії та нестабільного інтернет-зв'язку, коворкінги перетворилися на своєрідні «пункти незламності» для бізнесу. Вони стали критично важливою інфраструктурою, що дозволяє тисячам спеціалістів та цілим компаніям продовжувати роботу, підтримуючи економіку країни. Цей досвід довів, що коворкінг – це не лише про гнучкість та економію, а й про стійкість та забезпечення безперервності бізнес-процесів у кризових умовах. Нижче представлено графік з проаналізованого джерела на рис.1.1.

Europe Coworking Spaces Market

Market Size in USD Billion

CAGR 10.37%



Source : Mordor Intelligence



Рис. 1.1. Графік зростання ринку коворкінгів в Європі з 2025 до 2030 року [1]

Для глибшого розуміння потреб користувачів та еволюції сервісів бронювання було проаналізовано історію створення та розвитку платформи «Croissant» [3]. Цей кейс наочно демонструє, як реальні проблеми користувачів трансформуються в успішний цифровий продукт.

Компанія Croissant, яка спеціалізується на наданні доступу до гнучких робочих просторів, була заснована у Нью-Йорку в 2015 році групою з чотирьох розробників. Початково засновники працювали над зовсім іншим стартап-проектом під назвою «Perqy». З метою економії коштів на оренду офісу, команда працювала у різноманітних кав'ярнях Манхеттена. Проте дуже швидко вони зіткнулися з типовими проблемами віддаленої роботи в непристосованих умовах:

- Складність пошуку вільних місць, особливо для командної роботи.
- Нестабільне Wi-Fi з'єднання та високий рівень шуму, що заважав концентрації.
- Соціальний тиск та необхідність постійно купувати каву або їжу (синдром «купівлі ще одного круасана»), щоб виправдати своє тривале перебування за столиком.

Саме ці виклики стали каталізатором зміни напрямку діяльності. Під час участі в хакатоні команда розробила першу ітерацію додатку Croissant. Початкова концепція полягала в наданні можливості погодинного бронювання столиків саме у кав'ярнях, що згодом трансформувалося у повноцінний агрегатор коворкінг-просторів. З моменту заснування компанія значно розширила свою мережу та географію присутності. На сьогодні Croissant надає доступ до робочих просторів у понад 70 містах світу, включаючи такі великі хаби, як Лондон, Берлін, Бостон та Сан-Франциско.

Цільовою аудиторією сервісу є переважно фрілансери, віддалені працівники та індивідуальні підприємці, які потребують професійного, тихого простору для роботи без необхідності укладання довгострокових договорів оренди [4].

Важливим аспектом розвитку компанії є її інноваційна складова. Croissant прагне надавати більше, ніж просто робочий стіл, плануючи використовувати аналітичні дані для надання рекомендацій щодо оптимізації творчості та продуктивності користувачів. Основна модель сервісу є яскравим прикладом «економіки на вимогу» [5], задовольняючи потребу у гнучких умовах праці на ринку, що стрімко трансформується. Croissant реалізує цю модель шляхом надання доступу до робочих просторів у реальному часі, що дозволяє монетизувати тимчасово вільні офісні площі, перетворюючи їх на ліквідний ресурс, доступний через мобільний додаток. Нижче наведено приклади підписок та інтерфейсів розглянутого додатку на рис. 1.2-1.3.

Аналіз кейсу Croissant підтверджує, що проблема пошуку якісного робочого місця є універсальною та актуальною. Проте, проєкт «Coworkly» пропонує додатковий крок в еволюції подібних сервісів, враховуючи сучасні технологічні:

1. AI-персоналізації: На відміну від ранніх версій аналогів, де користувач самотійно перебирає варіанти, «Coworkly» пропонує інтеграцію штучного інтелекту для проактивних рекомендацій, економлячи час клієнта.

2. Фокус на професійних просторах: Враховуючи досвід Croissant щодо шуму в кав'ярнях, «Coworkly» фокусується виключно на професійних коворкінгах та хабах, гарантуючи відповідність нефункціональним вимогам (NFR) щодо стабільного інтернету та наявності генераторів.

3. Гібридна модель бронювання: «Coworkly» поєднує можливість як підписки (для постійних користувачів), так і разового миттєвого бронювання (pay-as-you-go)[6], що є більш гнучким підходом для українських реалій. Модель "pay-as-you-go"[6] дозволяє користувачам Coworkly, особливо в умовах нестабільності, не прив'язуватися до місячних абонементів, а сплачувати лише за фактично відпрацьовані години в конкретній локації, що забезпечує максимальну фінансову ефективність для клієнта.

Crystal clear pricing

I'm based in

EXPLORER	CREATOR	LUMINARY
€ 29 /mo	€ 79 /mo	€ 149 /mo
10 hours per month Bring a guest for €5/hr Roll over unused hours NEW	40 hours per month 5 free guest hours Roll over unused hours NEW	120 hours per month 20 free guest hours Roll over unused hours NEW
Get Started	Get Started	Get Started

Interested in Croissant for your team? [Explore our team plans!](#)

Рис. 1.2. Ціни за підписки в додатку Croissant в Німеччині [3]

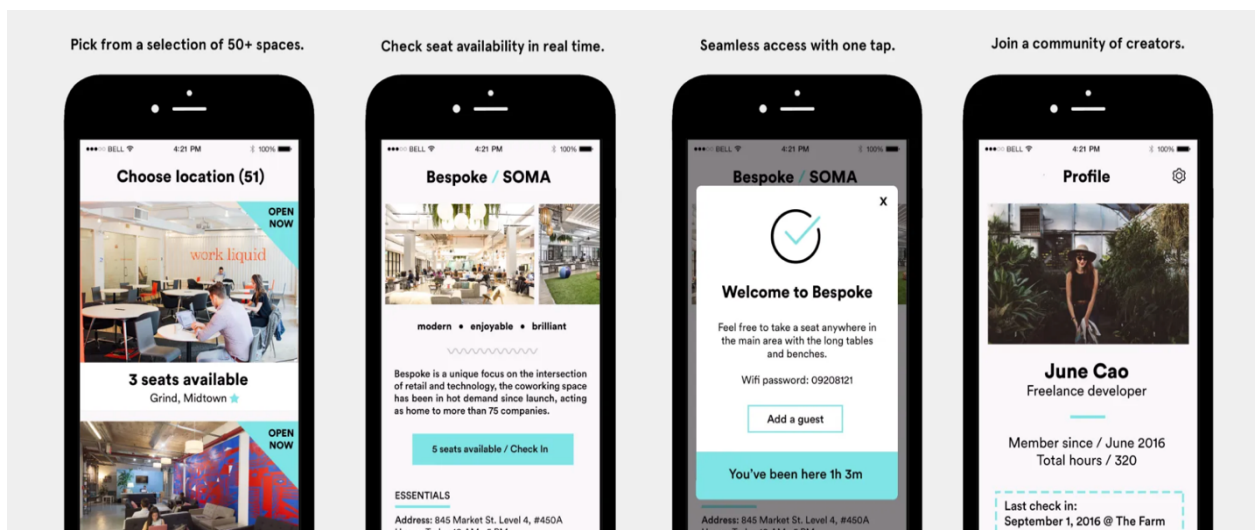


Рис. 1.3 Інтерфейс додатку Croissant [3]

1.3 Визначення зацікавлених сторін проєкту

Успішна реалізація будь-якого проєкту неможлива без чіткого визначення та аналізу зацікавлених сторін (стейкхолдерів [7]) – осіб, груп або організацій, які можуть впливати на проєкт або на яких проєкт може вплинути. Для проєкту розробки мобільного додатку «Coworkly» було ідентифіковано наступні ключові групи зацікавлених сторін.

1. Кінцеві користувачі: це основна група, для якої створюється продукт. До неї належать не лише фрілансери, стартапери та малі команди, але й ширші сегменти, такі як «цифрові кочівники», корпоративні співробітники на гібридному графіку та студенти.

- Інтереси та очікування: головний інтерес цієї групи полягає в отриманні зручного, швидкого та надійного інструменту для пошуку та бронювання робочих місць. Вони очікують інтуїтивно зрозумілий інтерфейс, прозору систему ціноутворення, актуальну інформацію про завантаженість, легкий процес оплати (включно з інтеграцією Apple Pay/Google Pay [8]) та корисну систему фільтрів (за локацією, ціною, зручностями). Персоналізовані рекомендації та чесні відгуки інших користувачів є для них додатковою цінністю.
- Рівень впливу – високий: задоволеність кінцевих користувачів та їхня готовність використовувати додаток безпосередньо визначають успіх проєкту на ринку. Їхній зворотний зв'язок є критично важливим для вдосконалення UX/UI дизайну та пріоритезації функціоналу.

2. Власники та адміністратори коворкінгів: ця група є ключовим бізнес-партнером, що надає послуги, які агрегує додаток.

- Інтереси та очікування: їх основна мета – максимізація завантаженості своїх локацій та залучення нового потоку клієнтів. Вони зацікавлені у зменшенні адміністративного навантаження на свій персонал (ручне опрацювання бронювань, прийом оплат). Від додатку вони очікують наявності зручної та функціональної адміністративної панелі, через яку можна легко керувати своїми просторами, оновлювати інформацію про доступність, ціни, фотографії та отримувати детальну аналітику щодо бронювань і доходів.
- Рівень впливу – високий: без їхньої участі та готовності інтегруватися в платформу, додаток не матиме цінності. Їхні вимоги до функціоналу адміністративної частини системи є пріоритетними.

3. Корпоративні клієнти: це нова, але стрімко зростаюча група стейкхолдерів [7], представлена HR-відділами та керівниками компаній, які впроваджують гнучкі та гібридні моделі роботи.

- Інтереси та очікування: вони прагнуть надати своїм співробітникам доступ до якісних робочих місць поза центральним офісом, оптимізуючи при цьому витрати. Їх цікавить можливість централізованого управління доступом для команди, встановлення лімітів на витрати та отримання єдиного рахунку за послуги для всієї компанії.
- Рівень впливу – середній на початковому етапі, але з високим потенціалом. Задоволення потреб цієї групи може стати ключовою конкурентною перевагою та забезпечити стабільний потік доходів у майбутньому.

4. Інвестори: особи або фонди, що надають фінансові ресурси для розробки, запуску та маркетингу проєкту.

- Інтереси та очікування: головний інтерес – отримання прибутку на інвестований капітал (ROI) [9]. Вони очікують на чіткий та життєздатний бізнес-план, прозору модель монетизації (наприклад, комісія з бронювань), докази ринкового попиту та потенціал для масштабування бізнесу. Інвестори вимагають регулярної звітності та досягнення ключових показників ефективності (KPI) [10].
- Рівень впливу – високий: вони впливають на стратегічні рішення, бюджет, терміни реалізації та загальний вектор розвитку проєкту.

5. Команда розробки: група, що безпосередньо реалізує проєкт: технічні спеціалісти (програмісти), UX/UI-дизайнери, аналітики, проєктний менеджер.

- Інтереси та очікування: їхня мета – створити якісний та технічно досконалий продукт. Вони зацікавлені у чітко сформульованих вимогах, реалістичних термінах виконання завдань та доступі до необхідних ресурсів (технологій, інструментів).

- Рівень впливу – високий: від їхньої кваліфікації, досвіду та злагодженої роботи залежить технічна реалізація, стабільність та якість кінцевого продукту.

1.4 SWOT-аналіз проєкту

Для комплексної оцінки стратегічних перспектив проєкту «Coworkly» та виявлення ключових факторів, що можуть вплинути на його успіх, було проведено SWOT-аналіз. Цей метод передбачає дослідження внутрішніх сильних та слабких сторін проєкту, а також зовнішніх можливостей та загроз [11].

Проведення даного аналізу є невід'ємним етапом стратегічного планування, оскільки цей метод дозволяє провести комплексну діагностику проєкту. Його корисність полягає у здатності систематизувати як внутрішні, так і зовнішні фактори, що впливають на життєздатність ініціативи.

З одного боку, аналіз внутрішнього середовища дозволяє об'єктивно оцінити конкурентні переваги проєкту, такі як інноваційна УТП (Унікальна Торгова Пропозиція [12]), та водночас виявити внутрішні бар'єри та зони ризику. З іншого боку, аналіз зовнішнього середовища надає системне бачення ринкових умов. Він дозволяє ідентифікувати сприятливі фактори, наприклад, зростання попиту на ринку, а також потенційні загрози, зокрема високий рівень конкуренції.

Таким чином, SWOT-аналіз слугує фундаментальною основою для подальшої розробки стратегії проєкту. Він надає повне бачення, що є критично необхідним для прийняття обґрунтованих управлінських рішень: як найефективніше використати сильні сторони для реалізації ринкових можливостей та які компенсаторні заходи необхідно розробити для мінімізації впливу загроз і нівелювання слабких сторін.

Нижче представлено таблицю з описом і деталями проведеного аналізу.

«SWOT – аналіз»

№	Strengths	Weaknesses	Opportunities	Threats
1	2	3	4	5
1	<i>Інноваційна УТП:</i> Інтеграція ІІІ для персоналізованих рекомендацій, що виділяє проект серед статичних каталогів.	<i>Виклик старту:</i> Проект є двосторонньою платформою, успіх якої залежить від одночасного залучення двох різних груп: кінцевих користувачів (попит) та коворкінг-партнерів (пропозиція).	<i>Стрімке зростання ринку:</i> Глобальний перехід на гібридні моделі роботи створює стабільно високий попит на гнучкі робочі простори.	<i>Висока конкуренція:</i> Можливий вихід на ринок великих міжнародних гравців з власними додатками та бюджетами.
2	<i>Вирішення проблеми ринку:</i> Централізована платформа для пошуку, бронювання та оплати, що вирішує проблему фрагментованості.	<i>Висока залежність від партнерів:</i> Якість сервісу критично залежить від готовності коворкінгів до співпраці та своєчасного оновлення ними інформації про доступність.	<i>Потенціал для розширення:</i> Можливість додавання суміжних сервісів (бронювання залів, кейтеринг, нетворкінг) для збільшення середнього чеку.	<i>Низькі бар'єри для входу:</i> Технологічна відносна простота створення базового агрегатора може призвести до появи "копій" проекту.
3	<i>Сучасний технологічний стек:</i> Використання Google Firebase забезпечує гнучкість, масштабованість та надійність при мінімізації витрат на інфраструктуру.	<i>Відсутність впізнаваності бренду:</i> Як новий гравець, «Coworkly» потребуватиме значних інвестицій у маркетинг для формування довіри та залучення перших клієнтів.	<i>Аналітика даних:</i> Можливість збирати та аналізувати унікальні дані про ринкові тренди, попит на робочі місця, ціноутворення та поведінку користувачів.	<i>Залежність від сторонніх API:</i> Зміна політики ціноутворення або умов використання OpenAI чи платіжних систем може вплинути на операційні витрати.

1	2	3	4	5
4	<i>В2В-орієнтація:</i> Архітектура задалегідь враховує потреби корпоративних клієнтів, відкриваючи стабільний канал монетизації.	<i>Відсутність контролю:</i> Оскільки «Coworkly» є інформаційним посередником, проєкт не має прямого контролю над якістю послуг, що надаються партнерами-коворкінгами (стабільність Інтернету, чистота, рівень сервісу).	<i>Можливості стратегічних партнерств:</i> Потенціал для інтеграції з іншими В2В-сервісами, наприклад бізнес-інкубаторами або освітніми проєктами для посилення ціннісної пропозиції.	<i>Ризики безпеки даних:</i> Платформа, що обробляє персональні та платіжні дані, є ціллю для кібератак, що несе репутаційні та фінансові ризики.

Для перетворення результатів SWOT-аналізу на конкретні стратегічні дії було розроблено розширений аналіз взаємовпливу, який наведено нижче (табл. 1.2) [13]. Цей інструмент дозволяє сформулювати стратегії чотирьох типів:

- SO (Maxi-Maxi): використання сильних сторін для реалізації можливостей.
- ST (Maxi-Mini): використання сильних сторін для нейтралізації загроз.
- WO (Mini-Maxi): використання можливостей для подолання слабких сторін.
- WT (Mini-Mini): мінімізація слабких сторін для уникнення загроз.

Таблиця 1.2

SWOT – аналіз (деталізація)

№	Код елементів SWOT	Характеристика розвитку взаємодії
1	2	3
1	S1 + O3	Використання інтегрованих AI-алгоритмів не лише для рекомендацій, а й для глибокого аналізу поведінкових даних, що дозволить пропонувати партнерам-коворкінгам унікальні аналітичні звіти про попит, монетизуючи дані.

1	2	3
2	S3 + O1	Масштабованість архітектури на базі Google Firebase дозволяє швидко адаптувати систему під стрімке зростання трафіку без необхідності дорогої перебудови інфраструктури на етапі масштабування.
3	S4 + O4	Архітектура, готова до B2B, дозволяє сформувати ексклюзивні партнерські пропозиції для великих IT-компаній та бізнес-інкубаторів, інтегруючи Coworkly в їхні корпоративні соцпакети.
4	W1 + O4	Подолання проблеми залучення перших користувачів шляхом стратегічного альянсу з однією з великих мереж коворкінгів, яка стане «якірним» партнером і приведе першу базу користувачів.
5	S1 + T2	Захист від появи технологічних «клонів» шляхом поглиблення AI-функціоналу, який складно скопіювати швидко, на відміну від базового функціоналу бронювання.
6	W2 + T1	Впровадження системи рейтингів та договорів для партнерів, щоб мінімізувати ризики низької якості сервісу, які можуть призвести до відтоку клієнтів до конкурентів.
7	S3 + T3	Гнучкість Firebase Cloud Functions дозволяє швидко переключитися на альтернативні AI-моделі або платіжні шлюзи у випадку різкого підвищення цін або зміни політик сторонніх сервісів.

Проведений аналіз взаємовпливу продемонстрував, що більшість ідентифікованих загроз та слабких сторін можуть бути ефективно нівельовані через реалізацію наявних можливостей. Загалом, аналіз підтверджує значний потенціал запропонованої інформаційної системи. Ключові сильні сторони, такі як AI-персоналізація забезпечують проєкту конкурентну перевагу. Водночас для мінімізації ризиків, пов'язаних із залежністю від партнерів та сторонніх API, проєкту необхідно впровадити стратегію диверсифікації технічних провайдерів та розробити жорсткі стандарти якості для коворкінгів-партнерів.

1.5 Постановка задачі дослідження, формулювання технічного завдання на розробку у вигляді паспорту проєкту

Паспорт проєкту – це стисла, структурована характеристика майбутнього проєкту, що дозволяє чітко побачити його ключові параметри: для кого і навіщо він створюється, які ресурси потрібні, хто бере участь, якими будуть основні етапи реалізації та кінцевий результат. Він допомагає швидко оцінити доцільність і життєздатність ініціативи ще до початку розробки.

Ознаки проєкту:

1. Мета: створення зручного мобільного додатку, який дозволить користувачам бронювати робочі місця у мережі коворкінгів «Coworkly», переглядати доступність, отримувати рекомендації на основі ШІ та оплачувати послуги онлайн.

2. Термін реалізації: 6 місяців.

3. Унікальний результат: повнофункціональний мобільний застосунок з елементами штучного інтелекту, орієнтований на український ринок.

4. Фінансування: власні інвестиції стартапу, з можливістю залучення венчурного капіталу або грантових коштів.

Цілі проєкту:

1. Створити інтуїтивно зрозумілий додаток, що спрощує процес пошуку та бронювання робочих місць у коворкінгах.

2. Інтегрувати елементи ШІ для персоналізації досвіду користувача.

3. Забезпечити централізоване управління мережею коворкінгів із боку адміністратора.

Зацікавлені сторони:

1. Кінцеві користувачі – фрілансери, стартапери, малі команди.

2. Власники коворкінгів – прагнуть збільшити завантаженість своїх локацій.

3. Інвестори – зацікавлені в розвитку цифрових сервісів.

4. Команда розробки – технічні спеціалісти, UX/UI-дизайнери, аналітики.

Результат проєкту:

Повністю готовий до використання мобільний додаток для Android та iOS з базовим набором функцій:

- авторизація;
- пошук локацій;
- бронювання;
- оплата;
- персоналізовані рекомендації;
- аналітика для власників коворкінгів.

Перелік задач:

1. Аналіз ринку та потреб користувачів.
2. Створення технічного завдання.
3. Проєктування UX/UI.
4. Розробка архітектури додатку.
5. Створення серверної частини.
6. Розробка мобільного додатку.
7. Інтеграція ШІ-модулів.
8. Тестування на різних пристроях.
9. Публікація в App Store та Google Play.
10. Збір зворотного зв'язку та підтримка.

Життєвий цикл проєкту.

1. *Ініціалізація.* На початковому етапі формується основна ідея мобільного додатку, визначаються цілі, завдання, очікувані результати та ключові зацікавлені сторони. Проводиться аналіз ринку: вивчаються потреби потенційних користувачів, конкуренти, тенденції у сфері коворкінг-сервісів і мобільних рішень. Також окреслюються загальні технічні можливості, в тому числі використання штучного інтелекту (ШІ) як елементу конкурентної переваги. На цьому етапі важливо обґрунтувати доцільність проєкту та отримати згоду на подальше планування.

2. *Планування.* Цей етап передбачає створення детального плану реалізації проекту. Розробляється технічне завдання (ТЗ), визначаються вимоги до функціоналу, інтерфейсу, інтеграції ШІ та зовнішніх сервісів. Складається графік робіт, формуються основні віхи (milestones), бюджет і ресурси. Визначаються ролі в команді (розробники, дизайнери, аналітики, ML-інженери). Також оцінюються можливі ризики – технічні, фінансові, часові – і готуються шляхи їх мінімізації.

3. *Реалізація.* В цей етап виконується безпосередня розробка продукту. Створюються макети інтерфейсу, реалізується серверна частина, бази даних, а також мобільний додаток. Інтегруються основні функції – реєстрація, авторизація, пошук та фільтрація локацій, бронювання, оплата. Водночас розробляються та інтегруються модулі штучного інтелекту: рекомендаційна система, аналітика завантаженості коворкінгів, прогнозування попиту. Кожен компонент поетапно перевіряється командою розробки.

4. *Тестування та контроль.* Цей етап включає комплексне тестування додатку: функціональне, юзабіліті, безпекове та навантажувальне. Особлива увага приділяється тестуванню ШІ-модулів – перевіряється точність рекомендацій, швидкість обробки запитів, коректність аналітики. Проводиться внутрішнє тестування та зовнішнє (бета) тестування з участю реальних користувачів. Виявлені помилки фіксуються та усуваються. Паралельно здійснюється контроль відповідності розробки затвердженим планам, бюджету та термінам.

5. *Впровадження.* Після успішного тестування продукт розміщується в App Store та Google Play. Налаштовується система збору аналітики, запуску оновлень і підтримки користувачів. Проводиться рекламна кампанія з метою залучення перших клієнтів. Забезпечується технічна підтримка, адаптація користувачів через інструкції та відповіді на запитання (FAQ). Власники коворкінгів підключають свої локації через адміністративну панель. Збирається зворотний зв'язок для оперативного вдосконалення функціоналу.

6. Завершення проєкту. Фінальний етап передбачає підбиття підсумків: проводиться оцінка досягнення цілей проєкту, аналіз виконання бюджету, часу, якості. Готується фінансово-технічна звітність. Команда узагальнює отриманий досвід і формує рекомендації для масштабування проєкту або розробки нових функцій. У разі успішного запуску – планується підтримка, оновлення, масштабування на інші регіони або країни.

Надалі (рис. 1.4.) буде представлено візуальну композицію цілей проєкту – дерево цілей. Це ієрархічна структура, яка містить орієнтири (результати), яких ми плануємо досягти під час реалізації проєкту або по його завершенню.

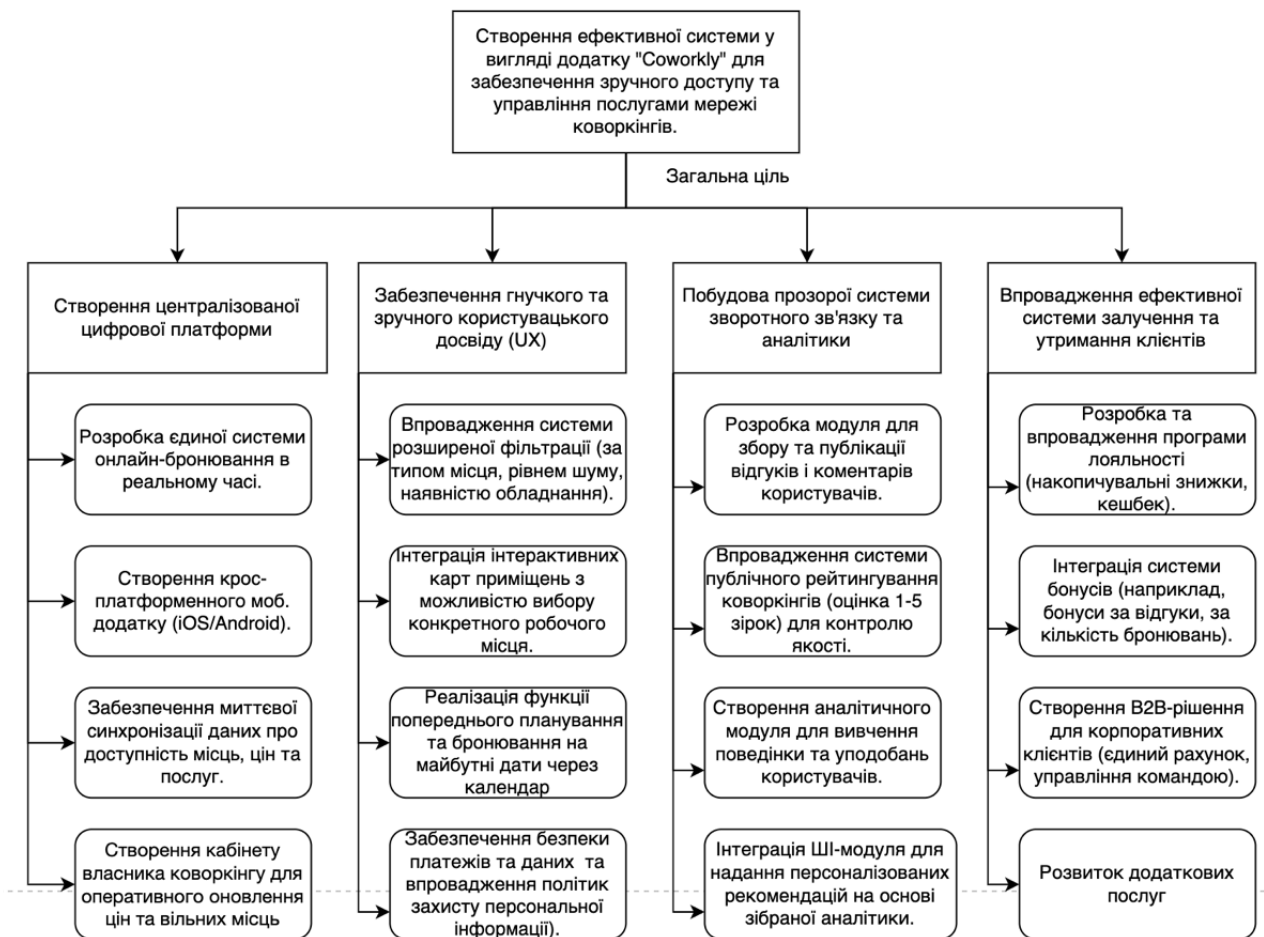


Рис. 1.4. Дерево цілей для проєкту «Coworkly»

Всі цілі, які будуть прописані в дереві цілей, повинні відповідати проблемам, що визначені в дереві проблем та дозволяти вирішувати ті проблеми, які планується вирішувати за рахунок даного проєкту [14].

РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

2.1 Побудова дерева причин та наслідків

Дерево причин та наслідків (рис. 2.1 наведено нижче) допомагає візуально проаналізувати кореневу проблему проєкту, зрозуміти її витoki та потенційні наслідки [14]. Воно дозволяє краще структурувати інформацію, виявити зв'язки між причинами та результатами, а також сформулювати ефективні шляхи вирішення. Такий підхід є корисним для ухвалення обґрунтованих управлінських рішень і планування подальших дій.

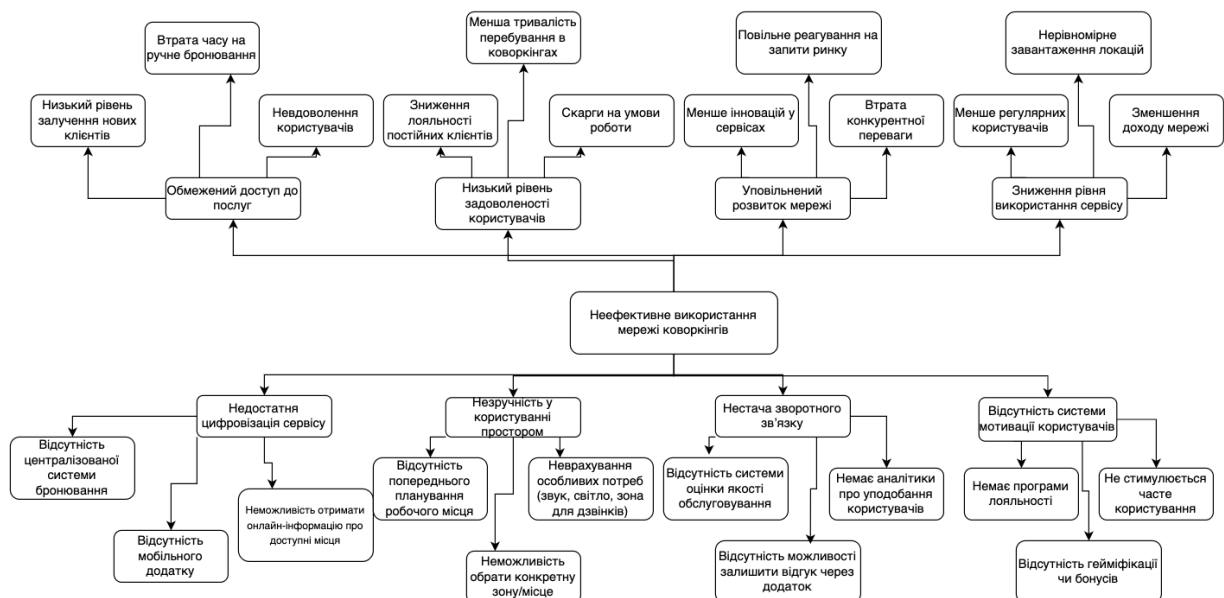


Рис. 2.1. Дерево причин та наслідків

2.2 Розробка концептуальної моделі інформаційної системи

Система: Мобільний додаток «Coworkly» для бронювання робочих місць у коворкінгах, з можливістю обрати та зберегти улюблене місце та отримати персоналізовані поради або нагадування за допомогою штучного інтелекту. Концептуальна система наведена на рис. 2.2.

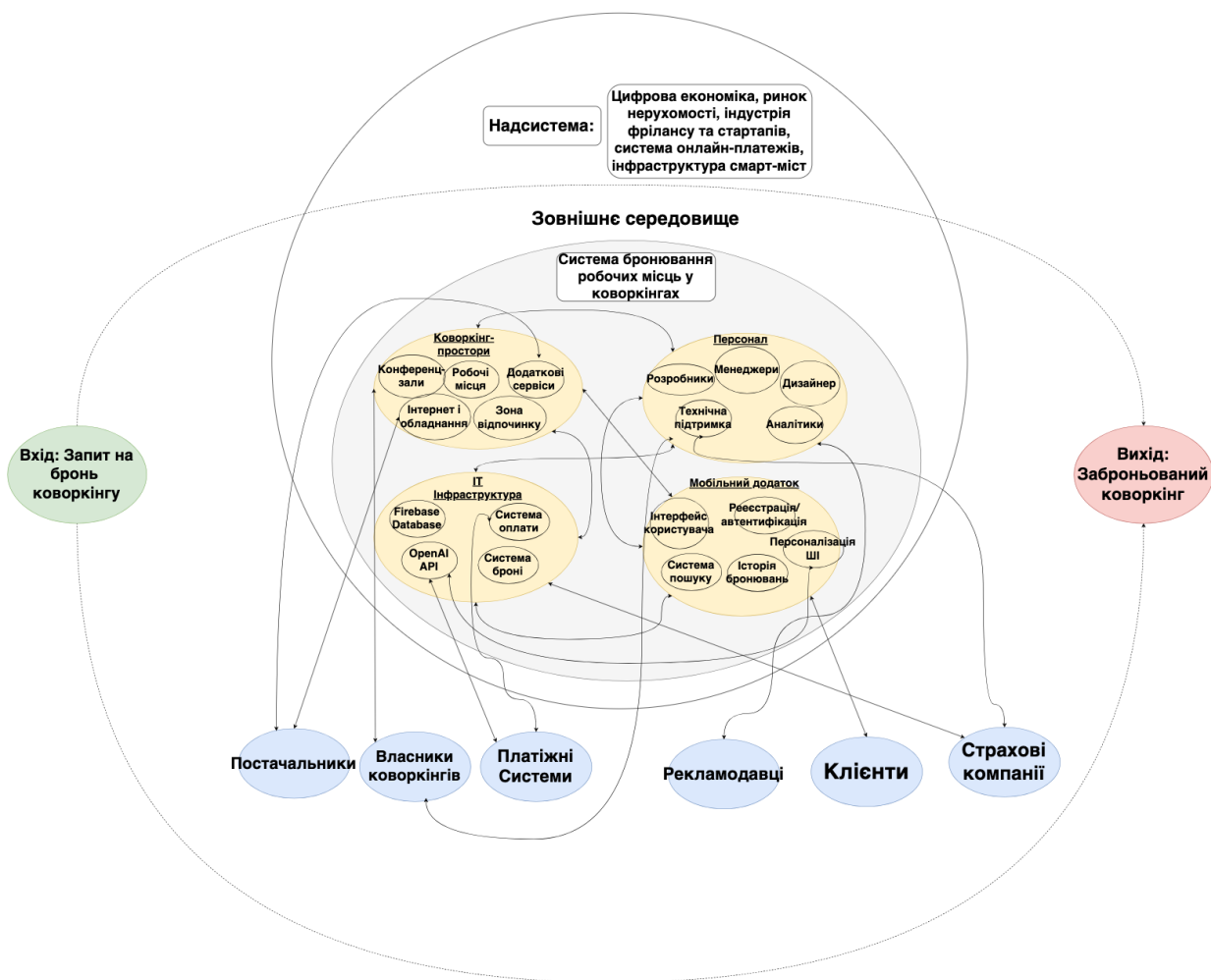


Рис. 2.2. Концептуальна модель інформаційної системи

Аналіз системи з фізичної точки зору:

Надсистема: цифрова економіка, ринок нерухомості, індустрія фрилансу та стартапів, система онлайн-платежів, інфраструктура смарт-міст.

Підсистеми:

1. Коворкінг-простори.
2. Мобільний додаток.
3. IT-інфраструктура (Firebase, OpenAI API [15]).
4. Персонал.

Зовнішні зв'язки системи з надсистемою:

1. Власники коворкінгів: розміщують інформацію, керують бронюванням.

2. Постачальники послуг постачають обладнання і додаткові сервіси в коворкінги.

3. Платіжні системи: обробляють онлайн-оплати (Google Pay, Apple Pay, банківські картки).

4. Страхові компанії: страхують інфраструктуру.

5. Клієнти: бронюють коворкінги, користуються послугами.

6. Рекламодавці: Розміщують рекламу в соціальних мережах і на банерах.

Перелік елементів для кожної підсистеми:

A) Коворкінги: 1. Робочі місця. 2. Конференц-зали. 3. Зона відпочинку. 4. Інтернет, обладнання. 5. Додаткові сервіси (кава, друк, шафки).

B) Мобільний додаток: 1. Інтерфейс користувача. 2. Реєстрація / автентифікація. 3. Система пошуку. 4. Історія бронювання. 5. Персоналізація ШІ.

C) IT-інфраструктура: 1. Firebase Database. 2. Система оплати. 3. Система броні. 4. OpenAI API.

D) Персонал: 1. Аналітики. 2. Технічна підтримка. 3. Менеджери. 4. Розробники. 5. Дизайнер.

E) Підпроцеси як елементи системи: 1. Реєстрація нового користувача. 2. Авторизація через email або соцмережі. 3. Пошук коворкінгів за параметрами. 4. Перегляд доступності місць. 5. Бронювання із зазначенням дати й часу. 6. Оплата послуг онлайн. 7. Отримання рекомендацій від ШІ. 8. Перегляд завантаженості. 9. Залишення оцінки та коментаря.

Надсистема для процесу і зовнішні зв'язки: результатом використання є комфортне та ефективно бронювання робочих місць для продуктивної праці. Зворотний зв'язок формується через систему оцінок, коментарів, історію бронювань, використання рекомендацій ШІ. Система має забезпечити інтуїтивний інтерфейс, точну інформацію про доступність, швидку оплату, персоналізовані пропозиції, збереження налаштувань користувача.

Концептуальна модель у цьому проєкті виконує одразу кілька важливих функцій. По-перше, вона дозволяє структурувати інформацію, виявити всі

необхідні елементи системи та визначити їх атрибути. По-друге, вона полегшує комунікацію між учасниками розробки: розробниками, дизайнерами, керівниками проєкту, бізнес-аналітиками тощо. Візуальна форма представлення моделі сприяє швидкому розумінню та ухваленню архітектурних рішень. По-третє, концептуальна модель є основою для подальшої побудови логічної моделі бази даних. Всі визначені сутності та зв'язки згодом трансформуються у таблиці, поля, ключі та обмеження.

Крім цього, концептуальна модель є гнучким інструментом для подальшого масштабування системи. Якщо в майбутньому з'явиться потреба додати нові функції, наприклад, підтримку інтеграції з календарями користувачів, модуль статистики або систему рекомендацій на базі штучного інтелекту, це можна буде зробити з мінімальними ризиками для вже наявної архітектури. Таким чином, концептуальна модель не лише забезпечує цілісне бачення інформаційної системи, а й створює міцне підґрунтя для її стійкої реалізації, розвитку та адаптації під потреби ринку. У випадку мобільного додатку «Coworkly», така модель стала ключовим кроком до забезпечення логічності, зручності використання та технічної узгодженості всіх функціональних компонентів.

Було також створено організаційну структуру проєкту (OBS), яка відображає членів команди, і їх зв'язок один з одним. Нижче можна побачити OBS для даного проєкту [16] (рис. 2.3).

Ця організаційна структура є типовою ієрархічною моделлю для розробки програмного забезпечення, яка забезпечує чітку вертикаль управління та горизонтальну співпрацю. На вершині цієї піраміди стоять спонсори проєкту та Інвестори, які є стратегічним органом, що визначає цілі та надає необхідні ресурси. Їхні вказівки транслуються проєктному менеджеру, який виступає центральним координатором, відповідальним за дотримання термінів, бюджету та забезпечення загального успіху проєкту.

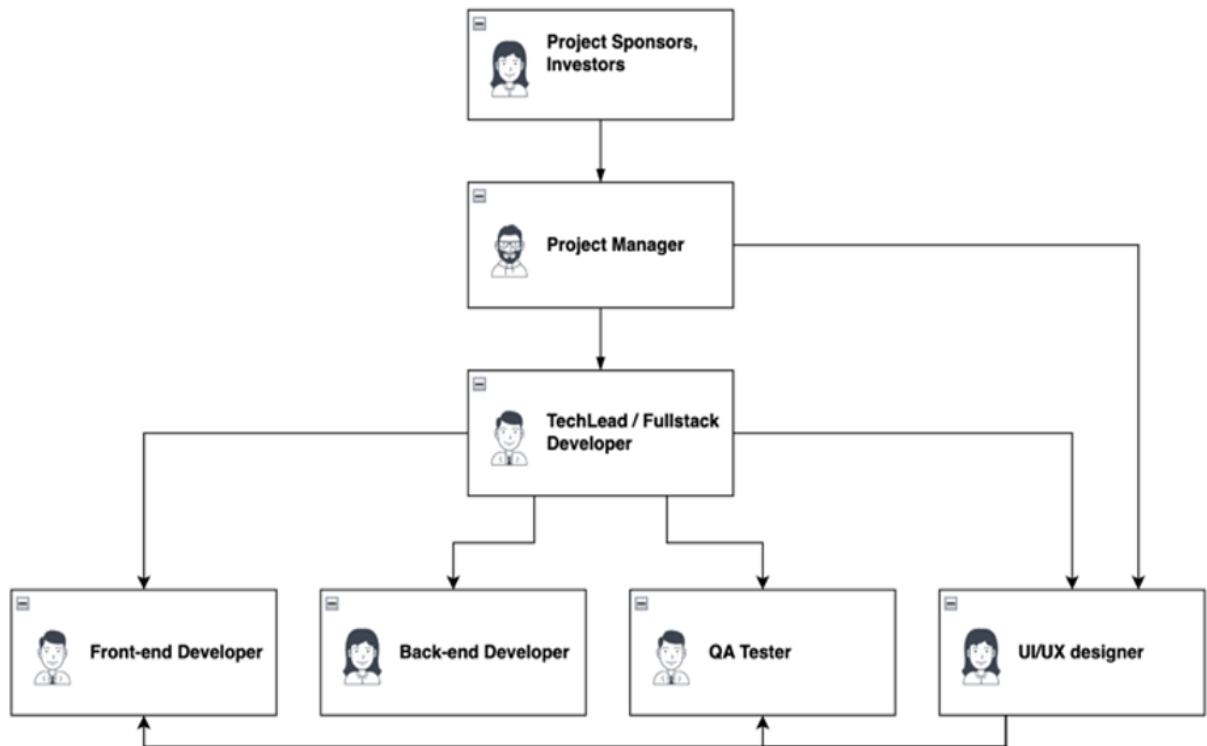


Рис. 2.3. OBS проекту «Coworkly»

У технічному аспекті, проєктному менеджеру підпорядковується технічний лідер / fullstack розробник, ключова особа, що несе відповідальність за архітектурну цілісність, технологічний стек та щоденне керівництво командою розробки. Під технічним лідерством працює основний виконавчий підрозділ, що включає спеціалістів за функціональним напрямом: front-end developer, який реалізує клієнтську частину та користувацький інтерфейс, та back-end developer, відповідальний за серверну логіку, базу даних та інтеграції.

Паралельно, критично важливу роль у забезпеченні якості виконує QA tester (тестувальник), який безпосередньо працює з розробниками, перевіряючи коректність та надійність створеного функціоналу. Окремо варто відзначити роль UI/UX designer'a, чиє підпорядкування проєктному менеджеру гарантує, що кінцевий продукт відповідатиме бізнес-цілям, тоді як його тісна співпраця з front-end розробником є запорукою точного втілення дизайну в код. Таким чином, ця модель забезпечує як ефективне управління (вертикаль спонсор-менеджер), так і якісне виконання (горизонталь розробка → тестування → дизайн).

2.3 Визначення функціональних та нефункціональних вимог до продукту ІТ проєкту

Функціональні вимоги:

1. *FR001*. Вибір мови інтерфейсу. Платформа повинна надавати можливість користувачу обрати бажану мову інтерфейсу (українську, англійську) під час реєстрації, а також змінювати її в будь-який момент через налаштування профілю.

2. *FR002*. Авторизація та реєстрація користувача. Користувач повинен мати змогу зареєструватись через email або соціальні мережі, а також проходити безпечну авторизацію (у т.ч. двофакторну).

3. *FR003*. Пошук доступних коворкінгів. Користувач повинен мати змогу здійснювати пошук коворкінгів за містом, районом, ціною, кількістю вільних місць, додатковими послугами.

4. *FR004*. Бронювання робочого місця. Користувач повинен мати змогу переглянути доступність конкретного робочого місця на календарі, вибрати дату й час, і здійснити бронювання.

5. *FR005*. Оплата послуг онлайн. Система повинна дозволяти оплату бронювання через банківські картки, Apple Pay/Google Pay та інші популярні платіжні сервіси.

6. *FR006*. Інтеграція з ШІ-модулем рекомендацій. Система повинна надавати персоналізовані рекомендації на основі історії бронювань, уподобань, часу відвідування тощо.

7. *FR007*. Перегляд завантаженості коворкінгів. Користувач має змогу бачити рівень зайнятості у реальному часі для кожного коворкінгу (наприклад, через інтерактивний графік або статус-індикатори).

8. *FR008*. Управління профілем користувача. Можливість редагувати особисту інформацію, переглядати історію бронювань, керувати способами оплати, налаштуваннями безпеки.

9. *FR009*. Оцінка та коментування коворкінгів. Користувачі можуть залишати оцінки та коментарі, що зберігаються у базі даних та можуть бути оброблені мовними моделями для аналізу.

Нефункціональні вимоги:

1. *NFR001*. Продуктивність. Система повинна обробляти запити користувачів із затримкою не більше 2 секунд при піковому навантаженні.

2. *NFR002*. Масштабованість. Архітектура системи повинна підтримувати горизонтальне масштабування для обслуговування великої кількості одночасних користувачів.

3. *NFR003*. Безпека. Додаток повинен відповідати стандартам безпеки OWASP Mobile Top 10: шифрування з'єднання, захист особистих даних, автентифікація тощо.

4. *NFR004*. Кросплатформеність. Продукт повинен працювати на Android та iOS з однаковим функціоналом та адаптивним інтерфейсом.

5. *NFR005*. Доступність. Інтерфейс має бути доступним для людей з обмеженими можливостями (контрастність, підтримка озвучування, масштабування тексту).

6. *NFR006*. Надійність. Система повинна забезпечувати безперебійну роботу з доступністю не менше 99,9% на місяць.

7. *NFR007*. Підтримка багатомовності та мультивалютності. Додаток повинен підтримувати українську та англійську мови інтерфейсу, а також забезпечувати відображення цін у кількох валютах (гривня, долар США, євро) з можливістю вибору користувачем під час використання додатку.

Наданий перелік вимог є ґрунтовною основою для розробки ІТ-продукту проєкту, чітко розмежовуючи його функціональні та нефункціональні аспекти. Функціональні вимоги зосереджені на безпосередній взаємодії користувача з платформою, забезпечуючи повноцінний цикл послуг: від вибору мови та безпечної авторизації до детального пошуку, миттєвого бронювання та безпечної онлайн-оплати послуг. Особливу увагу приділено модернізації, інтегруючи ШІ-модуль для персоналізованих рекомендацій, а також надаючи

користувачам можливість переглядати завантаженість коворкінгів у реальному часі та залишати оцінки. Водночас, Нефункціональні вимоги гарантують високу якість, надійність та стійкість системи, встановлюючи жорсткі стандарти продуктивності (обробка запитів до 2 секунд), безпеки (відповідність OWASP Mobile Top 10) та доступності (працездатність 99,9% та кросплатформеність на Android/iOS). Цей комплексний підхід свідчить про глибоке розуміння того, що успішний продукт має не лише виконувати свої основні функції, але й бути масштабованим, надійним і зручним для всіх категорій користувачів.

2.4 Формування Use Case елементів до функціональних вимог та побудова Use Case Diagram

Формування Use Case елементів є важливим етапом деталізації функціональних вимог до програмного забезпечення. Завдяки цьому забезпечується чітке розуміння взаємодії користувачів із системою, що дозволяє ефективно моделювати поведінку системи в різних сценаріях. Use Case описують типові дії користувачів (акторів), що призводять до досягнення певної цілі за допомогою системи. На основі сформованих сценаріїв будується діаграма варіантів використання (Use Case Diagram), яка візуально демонструє взаємозв'язки між акторами та функціональністю системи. Такий підхід сприяє виявленню можливих недоліків у вимогах, покращує комунікацію між замовниками та розробниками, а також полегшує подальше проектування системи. У цьому підрозділі здійснюється формалізація функціональних вимог у вигляді Use Case елементів та побудова відповідної Use Case діаграми (рис. 2.4).

Use Case Diagram (діаграма варіантів використання) показана далі, як користувачі взаємодіють із системою через її основні функції. У випадку проєкту Coworkly, діаграма демонструє, які дії може виконувати користувач, зокрема: реєстрація, пошук коворкінгів, бронювання місць, оплата, перегляд завантаженості, взаємодія з персоналізованими рекомендаціями тощо.

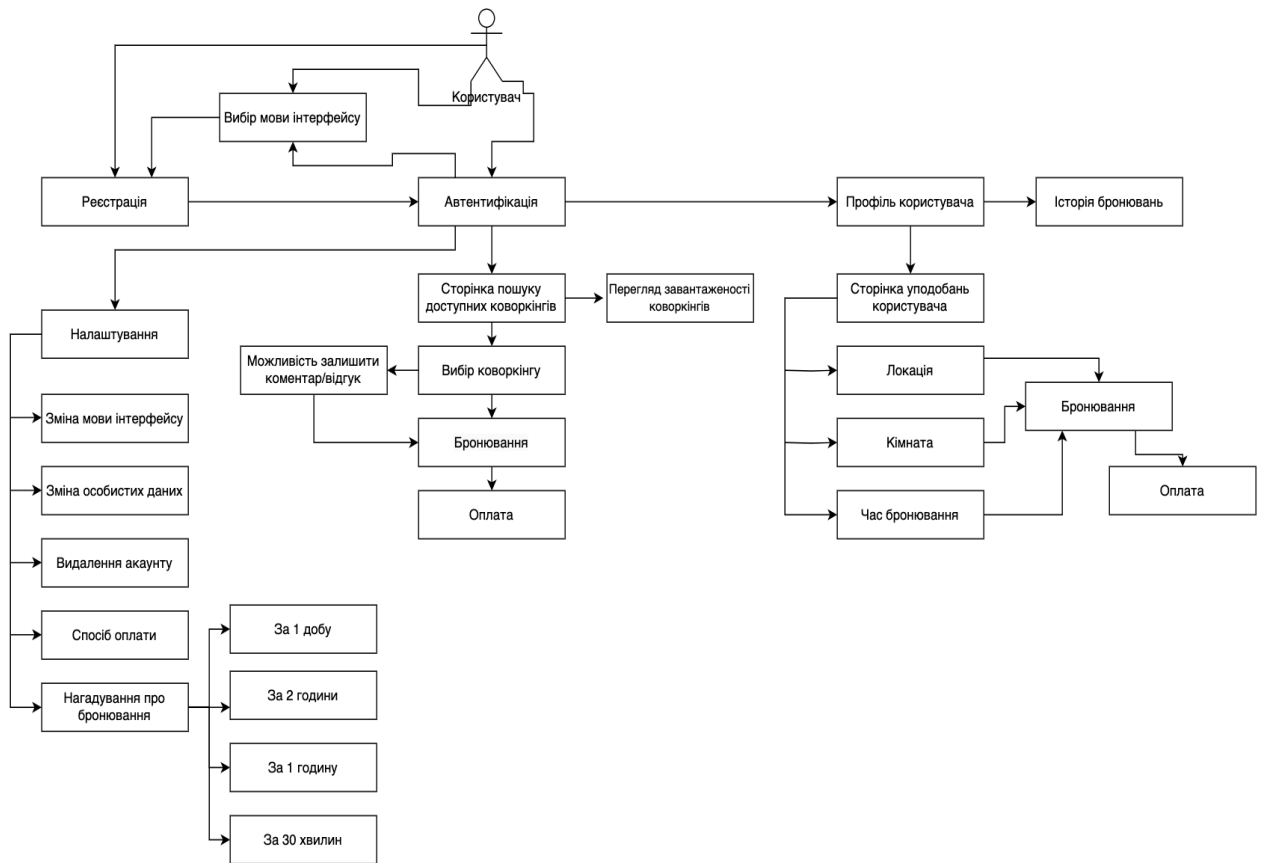


Рис. 2.4. Діаграма використання додатку Coworkly. Use Case Diagram

Ця діаграма є корисною, оскільки:

1. Дає загальне уявлення про функціональність системи.
2. Допомогає узгодити очікування користувачів і вимоги до продукту.
3. Служить основою для розробки архітектури програмного забезпечення.
4. Полегшує комунікацію між технічною командою та замовником.

РОЗДІЛ 3. ТЕХНОЛОГІЇ УПРАВЛІННЯ ПРОЄКТОМ ТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ КОРИСТУВАННЯ МЕРЕЖЕЮ КОВОРКІНГІВ «COWORKLY»

3.1 Аналіз методологій управління проєктами

Управління проєктами з розробки програмного забезпечення (ПЗ) є складною дисципліною, що вимагає ретельного планування, координації та контролю. Вибір правильної методології управління є одним з ключових, фундаментальних факторів, що визначають успішність проєкту. Він впливає на все: від швидкості розробки та здатності адаптуватися до змін до прозорості процесів та фінальної якості продукту.

Історично в ІТ-галузі домінували традиційні (каскадні) моделі, що прийшли з інженерних галузей, однак з часом вони поступилися місцем більш гнучким підходам, розробленим спеціально для хаотичної природи ІТ-проєктів. Для коректного вибору методології для проєкту «Coworkly» необхідно провести порівняльний аналіз цих двох філософій.

Традиційні (каскадні) методології, з яких найвідомішим представником цього класу є модель «Водоспаду». Цей підхід є лінійним та суворо послідовним. Весь проєкт розбивається на чіткі, окремі фази, і команда може перейти до наступної фази лише після стовідсоткового завершення попередньої [17].

Стандартний життєвий цикл моделі «Водоспад» включає наступні етапи:

1. *Аналіз вимог.* Збір та повна, вичерпна документація всіх вимог до продукту.

2. *Проектування системи.* Створення детальної архітектури та технічного дизайну майбутньої системи.

3. *Реалізація (написання коду).* Безпосередня розробка продукту згідно з затвердженою проєктною документацією.

4. *Тестування*. Верифікація готового продукту на відповідність початковим вимогам.

5. *Впровадження та підтримка*. Реліз продукту та його подальше обслуговування.

Головною перевагою такого підходу є його простота, зрозумілість та чіткий контроль над процесом. Кожен етап має визначені результати та кінцеву точку, що полегшує планування та управління.

Однак недоліки «Водоспаду» у контексті сучасної розробки ПЗ є критичними. Модель є абсолютно негнучкою та має низьку адаптивність. Вона припускає, що всі вимоги можна ідеально визначити на самому початку і вони не зміняться. Будь-які зміни в вимогах на пізніх етапах призводять до значного подорожчання та зсуву термінів, оскільки вимагають повернення на попередні етапи.

Крім того, замовник бачить готовий продукт лише на етапі тестування, часто через багато місяців після початку. Це створює високий ризик того, що кінцевий продукт не буде відповідати реальним очікуванням ринку або потребам клієнта. Ця модель може бути виправдана лише у проєктах з чітко визначеними, незмінними вимогами. Класичними прикладами є розробка ПЗ для медичного обладнання (наприклад, апарат МРТ), де вимоги до безпеки фіксовані і не можуть змінюватися ітеративно, або великі урядові проєкти (як-от система податкової звітності), де функціонал жорстко визначений законодавством та контрактом.

Гнучкі методології, які були створені у відповідь на недоліки каскадної моделі на початку 2000-х років був сформульований «Маніфест гнучкої розробки». Agile – це не одна методологія, а ціле сімейство підходів та філософія, що базуються на ітеративній та інкрементній розробці [17].

На відміну від лінійного «Водоспаду», Agile-підхід розбиває проєкт на короткі, чітко обмежені в часі цикли (ітерації або «спринти»). Наприкінці кожного циклу команда постачає замовнику працюючу, потенційно готову до випуску частину продукту (інкремент).

Ключові цінності Agile Manifesto визначають його сутність:

- Люди та взаємодія важливіші за процеси та інструменти.
- Працюючий продукт важливіший за вичерпну документацію.
- Співпраця з замовником важливіша за узгодження умов контракту.
- Готовність до змін важливіша за дотримання початкового плану.

Переваги Agile-підходу очевидні:

- Гнучкість: зміни у вимогах вітаються та можуть бути внесені на будь-якому етапі розробки, дозволяючи продукту еволюціонувати.
- Залучення замовника: клієнт є активним учасником процесу, регулярно надає зворотний зв'язок, що гарантує відповідність продукту його очікуванням.
- Швидкий результат: замовник отримує першу робочу версію продукту (MVP - Minimum Viable Product) значно раніше, що дозволяє швидше вийти на ринок.

Agile є ідеальним вибором для проєктів, де вимоги невідомі або швидко змінюються, а ринок є висококонкурентним. Прикладами таких проєктів є розробка мобільних додатків (як соціальні мережі чи ігри), створення платформ електронної комерції, де потрібно швидко реагувати на тренди ринку, та будь-які стартапи, що шукають свою бізнес-модель і потребують постійного зворотного зв'язку від реальних користувачів [17].

Ключові фреймворки та методи Agile:

1. Scrum: Найпопулярніший фреймворк для гнучкої розробки. Він не дає жорстких інструкцій «як робити», а визначає «каркас» процесу: ролі (проєктний менеджер, скрам-майстер, команда розробки), артефакти (беклог продукту, беклог спринту, інкремент) та регулярні події.

2. Kanban: Метод, орієнтований на візуалізацію робочого процесу та оптимізацію потоку завдань. Команда використовує Kanban-дошку, де завдання проходять через колонки, що відповідають етапам роботи (наприклад, «До виконання», «В роботі», «Тестування», «Готово»).

Вибір методології для проєкту «Coworkly» є найважливішим етапом в акценті управління проєктом. Для проєкту розробки мобільного додатку для мережі коворкінгів «Coworkly», який є класичним стартапом в динамічному, висококонкурентному середовищі, традиційна каскадна модель є абсолютно неприйнятною. Вимоги до такого продукту неможливо визначити повністю наперед – вони гарантовано зміняться.

Тому вибір гнучкої методології є обґрунтованим та життєво необхідним. Серед гнучких підходів, фреймворк Scrum є найбільш доцільним. Він забезпечує чітку структуру (спринти) для ітеративної розробки та змушує команду регулярно, наприкінці кожного спринту, отримувати зворотний зв'язок від зацікавлених сторін. Це дає високий рівень адаптивності, дозволяючи команді швидко реагувати на ринкові зміни та побажання користувачів.

3.2 Імплементация фреймворку Scrum. Визначення User Story та формування беклогу продукту проєкту

Як було обґрунтовано раніше, для проєкту «Coworkly» обрано гнучку методологію Agile, а саме фреймворк Scrum. Цей вибір зумовлений високим рівнем невизначеності вимог у стартап-середовищі та необхідністю швидкого отримання зворотного зв'язку від кінцевих користувачів [18].

Практична імплементация Scrum для проєкту «Coworkly» передбачає визначення ключових ролей, подій та артефактів. Далі описано гіпотетичну команду для реальної реалізації даного проєкту:

Ролі у проєкті:

1. *Проектний менеджер.* Цю роль виконує відповідно менеджер проєкту, на якого також в даному випадку припадають обов'язки власника продукту. Він відповідає за формування бачення продукту, управління Беклогом Продукту, визначення пріоритетів User Stories та приймання готового функціоналу (інкременту) наприкінці кожного спринту.

2. *Технічний лідер*. Виконує роль головного серед розробників, має крос-платформенний досвід, консультує за потребою. Спілкується з проєктним менеджером, аналізує функціонал, аналізує як краще/за допомогою чого імплементувати цей або інший функціонал.

3. *Команда розробки*. Гіпотетична команда, що складається з 4 спеціалістів (напр., Frontend розробник, Firebase/Backend розробник, UX/UI дизайнер). Команда є самоорганізованою та відповідає за технічну реалізацію функціоналу, визначеного в Беклозі Спринту, та дотримання стандартів якості.

4. *Скрам-майстер*. Роль, яку може виконувати один з членів команди. Його завдання – організовувати та налагоджувати всі Scrum-події, усувати перешкоди для команди та стежити за дотриманням принципів Scrum.

5. *QA-Тестувальник*. Контролює якість імплементації, тестує додаток на двох платформах, різними методами, знаходить баги, перевіряє різні підходи до користування додатком, після тесту відповідає і аналізує чи можна зливати новий функціонал з основною гілкою коду для додатку.

Список подій, які відповідають фреймворку Scrum [19]:

- *Спринт*. Тривалість одного циклу розробки встановлюється приблизно у 2 тижні. Протягом цього часу команда фокусується на створенні готового до демонстрації інкременту продукту.

- *Планування спринту*. На початку кожного спринту проєктний менеджер представляє пріоритезовані елементи беклогу. Команда розробки оцінює їхню складність (наприклад, у Story Points) та обирає той обсяг роботи, який зобов'язується виконати за спринт.

- *Щоденний Скрам*. Короткі 5-10 хвилинні дзвінки/зустрічі для синхронізації команди. Швидка комунікація про те, хто що робив вчора, хто що збирається робити сьогодні. Чи немає ніякого блоку від одного члена команди для іншого для виконання наступного завдання.

- *Огляд спринту*. Наприкінці спринту команда демонструє робочий функціонал (інкремент) проєктному менеджеру та іншим зацікавленим сторонам для отримання зворотного зв'язку.

- *Ретроспектива спринту.* Внутрішня зустріч команди для обговорення того, що пройшло добре, що пішло не так, і як можна покращити процес у наступному спринті.

Формування Беклогу продукту є невід’ємною частиною фреймворку та основним артефактом Scrum і являє собою динамічний, пріоритезований список усього функціоналу, який необхідний для створення «Coworkly». Основою для його формування слугують функціональні вимоги.

Для зручності управління, User Stories згруповані у більші функціональні блоки – Епіки: управління акаунтом, налаштування інтерфейсу, оплата, бронювання, пошук та персоналізація.

User Stories є ключовим артефактом у гнучких методологіях управління проектами, зокрема у використаному фреймворку Scrum. Вони являють собою короткий, неформальний опис певної функціональності, сформульований з точки зору кінцевого користувача або клієнта системи.

Головне призначення User Story полягає у тому, щоб змістити фокус з технічної реалізації на бізнес-цінність, яку ця функціональність має принести. Замість того, щоб описувати, *що* система повинна зробити, історія користувача пояснює, *для кого* ця функція призначена, *яку* його потребу вона вирішує і *яку* користь він у результаті отримує.

Найчастіше для формулювання User Stories використовується шаблон: «Як [тип користувача], я хочу [виконати дію], щоб [отримати цінність]». Такий підхід дозволяє команді розробки краще зрозуміти контекст та пріоритети завдань, а також слугує основою для подальшого обговорення, планування спринтів та формування беклогу продукту. Нижче (табл. 3.1) описано та наведено фрагмент формування історії користувача даного проекту.

Формування User Story

Код US	Епіс	Формулювання US	Acceptance Criteria	Пріоритет
1	2	3	4	5
US001(FR001: Вибір мови інтерфейсу)	Налаштування інтерфейсу	Як новий користувач, я хочу мати змогу обрати мову інтерфейсу під час реєстрації, щоб користуватися додатком комфортно з самого початку.	<ol style="list-style-type: none"> 1. Під час реєстрації доступне випадаюче меню з вибором мови (UA/EN). 2. Вибрана мова застосовується до всіх елементів інтерфейсу після реєстрації. 3. Вибір зберігається у профілі користувача. 4. При повторному вході інтерфейс відображається тією ж мовою. 	Низький (Low)
US002 (FR001)	Налаштування інтерфейсу	Як постійний користувач, я хочу мати змогу змінити мову інтерфейсу в будь-який момент, щоб користуватись додатком у зручній для мене мові.	<ol style="list-style-type: none"> 1. У налаштуваннях профілю доступна зміна мови. 2. Після зміни всі елементи інтерфейсу оновлюються миттєво. 3. Мова зберігається до наступного входу в систему. 4. Користувачу не потрібно перезапускати додаток після зміни мови. 	Низький (Low)
US003 (FR001)	Налаштування інтерфейсу	Як користувач, я хочу, щоб мова інтерфейсу відповідала мові мого пристрою за замовчуванням, щоб мені не потрібно було вручну її змінювати.	<ol style="list-style-type: none"> 1. Під час першого запуску додаток автоматично визначає мову системи. 2. Якщо мова підтримується додатком, вона встановлюється автоматично. 3. Користувач має можливість змінити мову вручну після автоматичного вибору. 	Низький (Low)
US004 (FR002: Авторизація та реєстрація користувача)	Управління акаун-том	Як новий користувач, я хочу зареєструватися за допомогою електронної пошти або соціальних мереж, щоб швидко створити обліковий запис.	<ol style="list-style-type: none"> 1. На сторінці реєстрації є опції «Зареєструватися через email», «Google», «Facebook». 2. Усі дані валідуються перед створенням облікового запису. 3. Після входу користувач отримує сповіщення про успішну авторизацію. 	Високий (High)

1	2	3	4	5
US005 (FR003: Пошук доступних коворкінгів)	Пошук	Як користувач, я хочу знайти коворкінги у своєму місті з потрібними умовами, щоб вибрати найкраще місце для роботи.	<ol style="list-style-type: none"> 1. Система має фільтри за містом, районом, ціною, вільними місцями та послугами. 2. Пошук оновлюється автоматично після зміни фільтра. 3. У результатах показано кількість вільних місць і загальну оцінку коворкінгу. 	Високий (High)
US006 (FR004: Бронювання робочого місця)	Бронювання	Як користувач, я хочу мати змогу забронювати робоче місце в коворкінгу, щоб мати гарантію наявності вільного місця.	<ol style="list-style-type: none"> 1. Користувач бачить доступні місця в реальному часі. 2. Користувач обирає дату, час та тип місця. 3. Система підтверджує бронювання та відображає його у профілі. 4. Місце стає недоступним для інших користувачів після бронювання. 	Високий (High)
US007 (FR004)	Бронювання	Як користувач, я хочу отримувати нагадування про майбутнє бронювання, щоб не забути прийти вчасно.	<ol style="list-style-type: none"> 1. За 1 годину до бронювання приходить push-сповіщення. 2. У налаштуваннях можна обрати час нагадування (30хв/1год/2год/1 день). 3. Нагадування містить адресу, час та тип бронювання. 	Середній (Medium)
US008 (FR004)	Бронювання	Як користувач, я хочу мати змогу скасувати своє бронювання, якщо мої плани змінилися.	<ol style="list-style-type: none"> 1. Скасування можливе не пізніше, ніж за 30 хв до початку. 2. Після скасування місце стає знову доступним. 3. Користувач отримує повідомлення про успішне скасування. 	Середній (Medium)
US009 (FR005: Оплата послуг)	Оплата	Як користувач, я хочу мати можливість оплатити послугу онлайн, щоб уникнути черг та готівки.	<ol style="list-style-type: none"> 1. Доступні кілька способів оплати: картка, Google Pay, Apple Pay. 2. Після оплати користувач отримує електронний чек. 3. Оплачене бронювання зберігається у розділі «Мої бронювання». 	Високий (High)

1	2	3	4	5
US010 (FR005)	Оплата	Як користувач, я хочу обрати валюту для оплати, щоб бачити актуальну вартість у зрозумілій одиниці.	<ol style="list-style-type: none"> 1. У налаштуваннях доступний вибір валюти: UAH, USD, EUR. 2. Ціни в додатку автоматично оновлюються відповідно до вибраної валюти. 3. Валюта зберігається до змін користувачем. 	Середній (Medium)
US011 (FR005)	Оплата	Як користувач, я хочу бачити історію своїх оплат, щоб контролювати витрати.	<ol style="list-style-type: none"> 1. У профілі доступний розділ «Історія оплат». 2. Для кожного бронювання відображається дата, сума та статус. 3. Дані зберігаються щонайменше 12 місяців. 	Середній (Medium)
US012 (FR006: Інтеграція з ШІ-модулем рекомендаці й)	Персона- лізація ШІ	Як постійний користувач, я хочу отримувати персональні рекомендації, щоб швидше знаходити цікаві для мене коворкінги.	<ol style="list-style-type: none"> 1. На головній сторінці є блок «Рекомендовані коворкінги». 2. Рекомендації базуються на історії бронювань та вподобаннях. 3. Користувач може оцінити рекомендацію або позначити її як нерелевантну. 	Середній (Medium)
US013 (FR007: Перегляд завантажено сті коворкінгів)	Пошук	Як користувач, я хочу бачити рівень завантаженості коворкінгів у реальному часі, щоб обрати менш переповнене місце.	<ol style="list-style-type: none"> 1. Для кожного коворкінгу відображається графік або індикатор завантаженості. 2. Інформація оновлюється не рідше ніж раз на 5 хвилин. 3. Якщо завантаженість перевищує 90%, відображається попередження. 	Високий (High)
US014 (FR008: Управління профілем користувача)	Управ- ління акаунтом	Як користувач, я хочу редагувати свій профіль та переглядати історію бронювань, щоб керувати інформацією про себе.	<ol style="list-style-type: none"> 1. У профілі є окремі секції для особистих даних, історії бронювань і платіжної інформації. 2. Зміни особистих даних зберігаються після натискання «Зберегти». 3. Історія бронювань доступна у вигляді списку з датами та статусами. 	Середній (Medium)

3.3 Планування спринтів та розробка Діаграми Ганта

Планування спринтів є ключовою подією в Scrum, що відбувається на початку кожної ітерації. Його мета – взяти пріоритезовані елементи з беклогу продукту та сформувані з них беклог спринту – детальний план роботи на найближчу ітерацію.

Для проєкту «Coworkly» було визначено тривалість спринту у 2 тижні. Виходячи з пріоритетів, визначених у табл. 3.1, весь обсяг робіт (15 User Stories) можна логічно згрупувати у 5 послідовних спринтів. Планування спринтів було розроблено за принципом MVP. Перші два спринти зосереджені на реалізації абсолютно необхідного функціоналу (пріоритет High), який дозволить запустити продукт та отримати перших користувачів. Наступні спринти додають важливий (Medium) та бажаний (Low) функціонал.

Спринт 1: «MVP: Ядро системи».

Тривалість: 2 тижні (тижні 1-2).

Мета: створення базової інфраструктури, реалізація автентифікації користувачів та надання їм можливості пошуку коворкінгів і перегляду їх завантаженості.

User Stories для спринту:

- US004 (High). Реєстрація/Авторизація через Email/Google/Facebook.
- US005 (High). Пошук та фільтрація коворкінгів (за містом, районом, ціною, послугами).
- US013 (High). Перегляд завантаженості коворкінгів у реальному часі.

Завдання спринту:

- Налаштувати проєкт Firebase (Firestore, Authentication, Cloud Functions).
- Створити UI екранів реєстрації та входу в систему (згідно з дизайном Figma).
- Реалізувати логіку валідації, створення користувача та автентифікації (включаючи соцмережі).

- Створити UI головного екрану з рядком пошуку, списком коворкінгів та кнопкою фільтрів.
- Реалізувати логіку запиту до Firestore для отримання та фільтрації списку коворкінгів.
- Реалізувати відображення індикатора завантаженості (на основі даних з БД).
- Провести тестування основного шляху користувача: реєстрація -> вхід -> пошук -> фільтрація.

Очікуваний результат. Користувач може створити акаунт, увійти в систему, побачити головний екран зі списком коворкінгів, успішно відфільтрувати їх за основними параметрами та побачити поточну завантаженість кожного простору.

Спринт 2: «Основний цикл бронювання».

Тривалість: 2 тижні (тижні 3-4)

Мета: реалізація ключової бізнес-логіки проєкту – надати користувачеві можливість забронювати обране місце та здійснити онлайн-оплату.

User Stories для спринту:

- US006 (High): Бронювання робочого місця на конкретну дату і час.
- US009 (High): Онлайн-оплата бронювання (Карта, Google/Apple Pay).

Завдання спринту:

- Створити UI екрану деталей коворкінгу та календаря бронювання.
- Реалізувати логіку перевірки доступності (наявності вільних місць) у реальному часі.
- Реалізувати процес створення запису «booking» у базі даних Firestore зі статусом «pending».
- Інтегрувати платіжний шлюз (наприклад, Fondy або Stripe) через Firebase Cloud Functions.
- Реалізувати логіку оплати через Apple Pay / Google Pay на клієнті.

- Забезпечити оновлення статусу бронювання на «confirmed» після успішної оплати.
- Налаштувати відправку електронного чека (або повідомлення про успішну оплату).
- Протестувати повний цикл: пошук -> вибір місця -> бронювання -> оплата -> підтвердження.

Очікуваний результат: Користувач може повністю завершити основний бізнес-процес: обрати коворкінг, забронювати вільне місце на певну дату і час, та успішно оплатити це бронювання онлайн. Бронювання фіксується в системі, місце стає недоступним для інших.

Спринт 3: «Покращення UX та Управління акаунтом».

Тривалість: 2 тижні (тижні 5-6)

Мета: розширити функціонал управління акаунтом, надати користувачам контроль над своїми бронюваннями (скасування, нагадування) та платежами.

User Stories для спринту:

- US014 (Medium): Управління профілем (редагування даних, історія бронювань).
- US007 (Medium): Нагадування про бронювання (push-сповіщення).
- US008 (Medium): Скасування бронювання (з дотриманням правил).
- US011 (Medium): Історія оплат.

Завдання спринту:

- Створити UI екрану профілю користувача.
- Реалізувати логіку редагування особистих даних (ім'я, пароль) та їх оновлення в базі даних.
- Створити UI для списків «Історія бронювань» та «Історія оплат».
- Реалізувати логіку скасування бронювання (напр., кнопка «Скасувати» стає неактивною за 30 хв до початку).
- Налаштувати Firebase Cloud Messaging (FCM) для надсилання push-сповіщень.

- Створити серверну логіку (Cloud Function) для відправки нагадувань (напр., за 1 годину до бронювання).
- Протестувати всі функції управління профілем та бронюваннями.

Очікуваний результат. Користувач має повноцінний особистий кабінет, де може керувати своїми даними, бачити історію транзакцій, отримувати корисні push-нагадування та самостійно скасовувати бронювання.

Спринт 4: «Імплементація процесу відгуків та персоналізація з AI».

Тривалість: 2 тижні (тижні 7-8)

Мета: впровадження ключової конкурентної переваги – системи AI-рекомендацій, а також механізму збору зворотного зв'язку від користувачів.

User Stories для спринту:

- US015 (Medium): Оцінка та коментування коворкінгів.
- US012 (Medium): Персоналізовані AI-рекомендації.

Завдання спринту:

- Створити UI для залишення відгуку та 5-зіркової оцінки.
- Реалізувати логіку збереження відгуків та рейтингу в базі даних (у колекціях reviews та coworkings).
- Налаштувати підключення до OpenAI API через захищену Firebase Cloud Function [15].
- Створити серверну логіку, яка за запитом користувача аналізує його історію бронювань та вподобань (з Firestore) та генерує запит до OpenAI для отримання текстової рекомендації.
- Створити UI-компонент для відображення AI-рекомендацій на головному екрані.
- Протестувати систему відгуків та точність і релевантність рекомендацій.

Очікуваний результат. Користувач може залишати відгуки про відвідані коворкінги та бачить на головному екрані персоналізований, корисний блок з AI-рекомендаціями, що покращує його досвід взаємодії з додатком.

Спринт 5: «Завершення та Локалізація».

Тривалість: 2 тижні (тижні 9-10).

Мета: додавання бажаного функціоналу (nice-to-have), що покращує гнучкість додатку, та проведення фінального тестування перед релізом.

User Stories для спринту:

- US001, US002, US003 (Low): Повна підтримка багатомовності (UA/EN).
- US010 (Low): Можливість вибору валюти (UAH/USD/EUR).

Завдання спринту:

- Інтегрувати бібліотеку інтернаціоналізації (напр., i18next) у React Native.
- Створити файли перекладів (JSON) для всіх текстових елементів додатку (українською, англійською та німецькою).
- Реалізувати логіку автоматичного визначення мови пристрою та ручного вибору в профілі.
- Додати UI-елемент для вибору валюти в налаштуваннях.
- Реалізувати логіку відображення цін у вибраній валюті (з конвертацією або отриманням з API [20]).
- Провести повне регресійне тестування всього функціоналу додатку.
- Підготувати фінальні збірки (builds) для публікації в App Store та Google Play.

Очікуваний результат. Додаток повністю локалізований (UA/EN), підтримує мультивалютність. Усі відомі помилки виправлено, продукт стабільний і готовий до релізу.

Особливості гнучкого планування на практиці: важливо розуміти, що представлений вище план спринтів є високорівневим прогнозом. У реальному проєкті гнучкість є ключовим фактором, і процес роботи враховує наступні практичні аспекти:

1. Гнучкість дедлайнів спринту: Scrum використовує «таймбокси» – спринт має фіксовану тривалість (у нашому випадку – 2 тижні). Це робиться для встановлення ритму роботи. Однак, на практиці команда може не встигнути виконати *всі* заплановані завдання через неочікувані збої, складніші,

ніж здавалося, проблеми інтеграції або виявлені баги. У такому випадку, за правилами Scrum, спринт не продовжується. Натомість, наприкінці 2 тижнів команда демонструє те, що встигла зробити, а незавершені завдання повертаються до беклогу продукту, де проєктний менеджер продукту повторно їх пріоритезує для наступних спринтів.

2. Декомпозиція завдань: User Story (напр., US006: «Як користувач, я хочу забронювати місце...») описує цінність для користувача, але не є технічним завданням. На етапі Планування Спринту команда розробки бере User Story і розбиває її на конкретні технічні «Завдання спринту». Саме ці завдання і стають окремими «картками» або «тікетами» на дошці Kanban. Іноколи одна User Story часто перетворюється на декілька тікетів (наприклад: «Створити UI календаря», «Налаштувати API-endpoint для бронювання», «Оновити логіку в БД», «Написати тести для бронювання»). Такий поділ на «підзавдання» робить роботу для розробників більш комфортною, прозорою та легкою для відстеження прогресу.

3. Динамічність: Початковий Беклог продукту – це лише початкове бачення. Головна перевага Agile якраз у тому, що цей список не є «сталим». Абсолютно нормально, коли впродовж реалізації якогось спринту у команди чи Product Owner з'являється ідея нової функції (якої не було в плані). Ця ідея не «ламає» поточний спринт. Натомість вона додається як нова User Story до загального беклогу. На наступному плануванні спринту проєктний менеджер може надати цій новій функції вищий пріоритет, ніж тим, що планувалися раніше. Agile вітає зміни, оскільки це означає, що проєкт адаптується до нових знань та реальних потреб ринку, а не сліпо слідує застарілому початковому плану.

Для проєкту «Coworkly» Діаграма Ганта візуалізує послідовність 10-тижневої фази активної розробки, розбитої на 5 двотижневих спринтів. Вона показує загальний таймлайн проєкту, де кожен спринт є окремим блоком робіт. Така діаграма не деталізує залежності між окремими User Stories (оскільки вони керуються командою всередині спринту), але показує, коли певний "Епік"

(наприклад, «MVP: Оплата» або «Персоналізація (AI)») планується до взяття в роботу та завершення. Діаграму Ганта наведено нижче (рис. 3.1–3.2.).

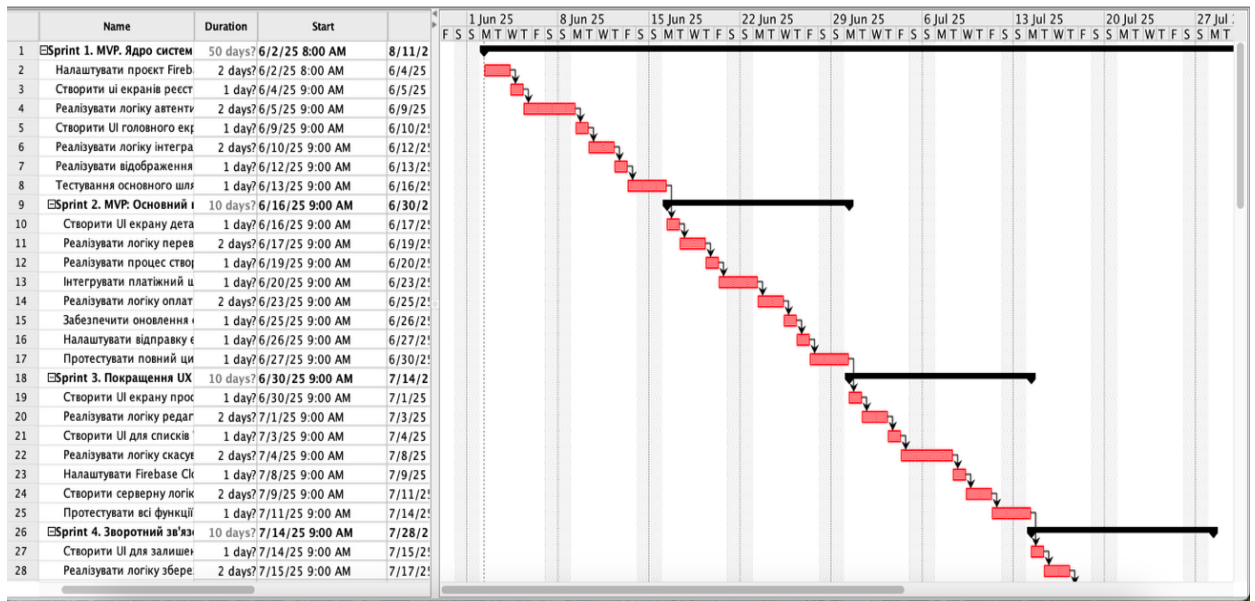


Рис 3.1. Діаграма Ганта

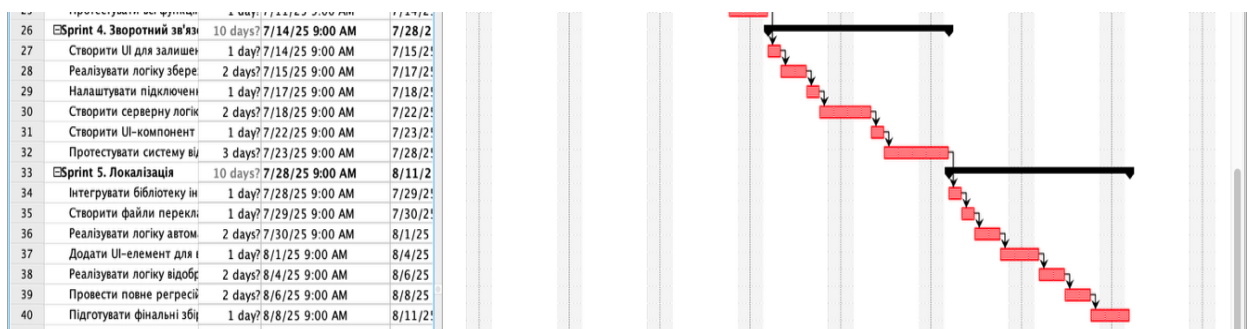


Рис 3.2. Діаграма Ганта. Продовження

Була розроблена також WBS діаграма, яка показує ієрархічне розбиття всієї роботи, яку необхідно виконати для досягнення цілей проєкту, на дрібніші операції та дії до такого рівня, на якому способи виконання цих дій цілком зрозумілі та відповідні роботи можуть бути оцінені та сплановані. Нижче представлена WBS діаграма (рис. 3.3.).

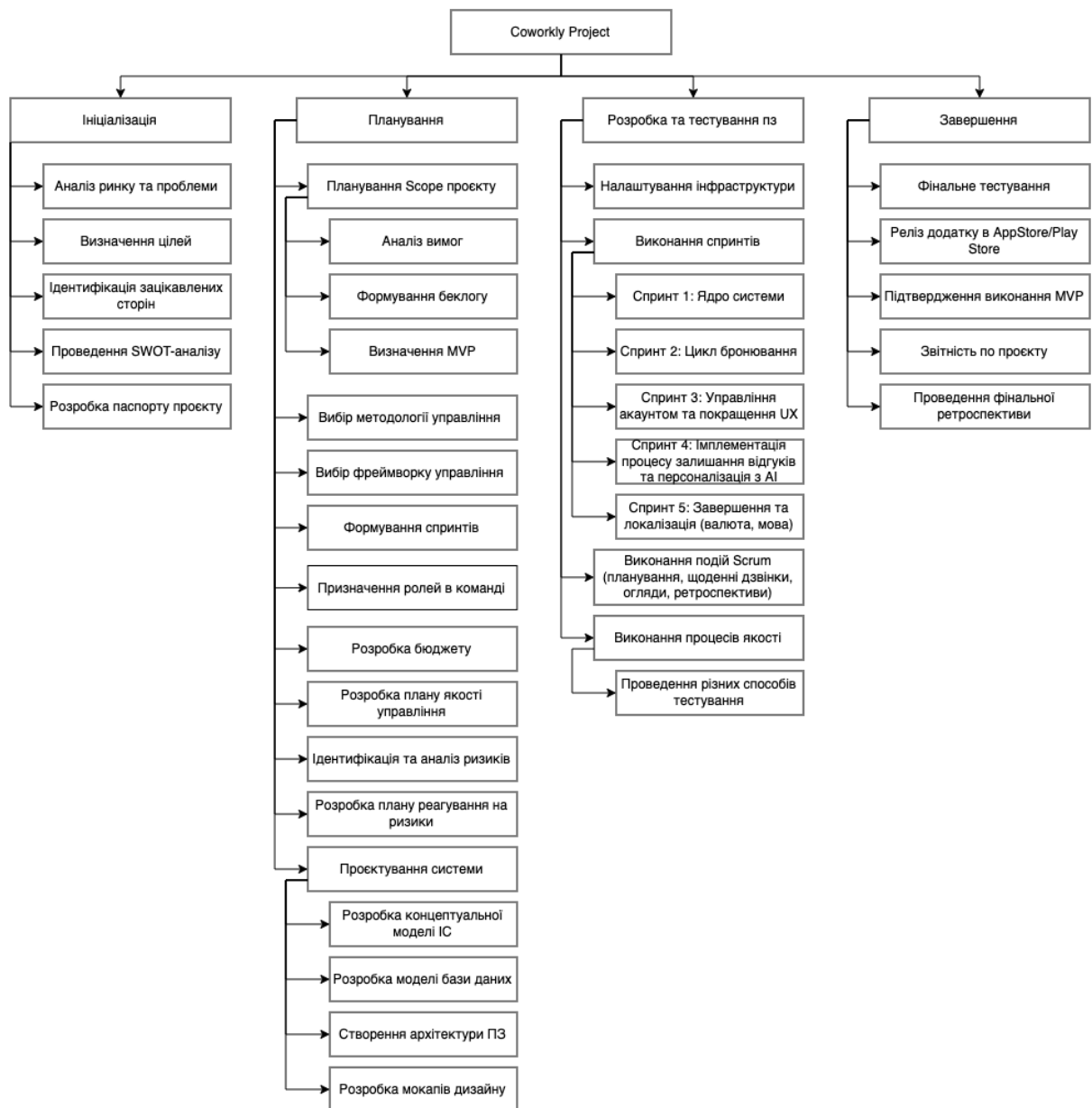


Рис. 3.3. WBS діаграма проекту «Coworkly»

На рис. 3.3. наведено вичерпну структуру декомпозиції робіт (WBS) для «Coworking Project», що слугує повною дорожньою картою для його реалізації. Проект чітко розділений на чотири ключові фази життєвого циклу: *Ініціалізація*, яка визначає цілі, проводить SWOT-аналіз і завершується розробкою паспорту проекту; найбільш об'ємна фаза *Планування*, що охоплює розробку Score, аналіз вимог, визначення MVP, вибір методології управління, формування команди, бюджетування та деталізоване технічне проектування системи, включаючи архітектуру та макети дизайну; фаза *Розробки та тестування ПЗ* сфокусована на ітераційному підході через п'ять деталізованих

спринтів, які покривають функціонал від ядра системи та бронювання до управління акаунтами, впровадження AI для персоналізації та локалізації, а також включає виконання регулярних подій Scrum; і, нарешті, фаза *Завершення*, яка передбачає фінальне тестування, підтвердження виконання MVP, реліз додатку в магазинах і підготовку підсумкової звітності та ретроспективи. Таким чином, ця діаграма ієрархічно та логічно впорядковує весь обсяг робіт, необхідний для успішного запуску коворкінг-платформи.

3.4 Управління ризиками проєкту

Управління ризиками є невід'ємною частиною управління IT-проєктами, особливо при розробці нових продуктів у конкурентному середовищі. Для проєкту «Coworkly», який залежить від зовнішніх сервісів (Firebase, OpenAI [15]) та орієнтований на висококонкурентний ринок, був розроблений детальний план управління ризиками, що охоплює вибір методології, ідентифікацію, аналіз та розробку заходів реагування, забезпечуючи успішне досягнення його цілей [21].

3.4.1 Вибір методології та структури управління

Для проєкту «Coworkly» було проведено аналіз трьох ключових стандартів управління ризиками (PMBOK, ISO 31000, MSF):

- PMBOK пропонує чітку структуру та методологію. Однак для проєкту масштабу «Coworkly» (бюджет \$40-41 тис., термін 6 міс.) ця система може виявитися занадто громіздкою та вимагатиме надлишкової формалізації.
- ISO 31000: надає більш абстрактний підхід та загальні принципи, що є гнучким, але менш специфікованим для IT-задач.
- Microsoft Solutions Framework (MSF [22]): створений спеціально для IT-проєктів і враховує специфіку розробки ПЗ. Його постійний цикл адаптації дозволяє швидко реагувати на нові ризики, що є критичним для проєкту.

Обґрунтування вибору MSF: Аналізуючи специфіку проєкту «Coworkly», MSF виявляється найбільш влучним стандартом. Проєкт передбачає значну залежність від зовнішніх API (Firebase, OpenAI), що створює специфічні технічні ризики [20]. Середній масштаб проєкту не потребує повної формалізації PMBOK, але вимагає більш структурованого підходу, ніж ISO 31000. MSF дозволить зафіксувати досвід роботи з React Native та інтеграціями, що буде цінним ресурсом для майбутнього.

Організаційна структура управління ризиками: враховуючи обмежений розмір команди, обрано дворівневу структуру:

1. *Роль 1.* Менеджер проєкту / Власник ризиків: відповідає за загальну ідентифікацію, оцінку, ведення реєстру ризиків та координацію реагування. Також відповідає за бізнес-ризики (конкуренція, фінанси).

2. *Роль 2.* Технічний лідер / Аналітик технічних ризиків: концентрується на глибокому аналізі технічних ризиків: стабільність Firebase, ліміти OpenAI API [15], архітектурні рішення, масштабованість та безпека. Відповідає за розробку технічних планів реагування та запасних стратегій.

Процес взаємодії та звітності: передбачаються щотижневі 30-хвилинні мітинги з огляду ризиків, під час яких обидва члени команди синхронізують інформацію про поточний стан ризиків, оновлюють тригери та планують заходи реагування.

3.4.2 Ідентифікація та якісна оцінка ризиків

Модель класифікації ризиків: для проєкту "Coworkly" обрано систему класифікації за типом джерела: технологічні, технічні, маркетингові, фінансові, організаційні та зовнішні ризики. Ця система найкраще підходить для IT-проєктів, оскільки дозволяє чітко структурувати ризики за їх природою.

Шкала якісного оцінювання: як шкалу оцінювання обрано розширену якісну шкалу (Н, НС, НВ, С, СН, СС, СВ, В, ВН, ВС, ВВ). Розширена шкала дозволяє забезпечити достатню деталізацію для середнього за складністю

проєкту, допомагаючи точніше диференціювати ризики за рівнем впливу та керуваності. Нижче наведено оцінку ризиків (табл. 3.2).

Таблиця 3.2

Ідентифікація та оцінка ризиків

№	Тип ризику	Ризикова подія	Сила впливу	Керуваність
1	2	3	4	5
1	Технологічні	Недоступність Firebase сервісів	ВВ	НС
2		Зміна тарифів OpenAI API	ВС	НВ
3		Досягнення лімітів OpenAI запитів	ВН	СС
4		Застарілі Firebase функцій	СВ	СН
5	Технічні	Проблеми масштабування React Native	ВС	СВ
6		Несумісність з iOS/Android версіями	ВН	СС
7		Проблеми з безпекою персональних даних	ВВ	СН
8		Низька продуктивність мобільного додатку	ВС	СВ
9	Маркетингові	Низький попит на коворкінги	ВВ	НВ
10		Сильна конкуренція (WeWork [23], Mindspace [24])	ВС	СН
11		Неточне визначення цільової аудиторії	ВН	СВ
12		Негативні відгуки в App Store/Google Play	СВ	СС
13	Фінансові	Перевищення бюджету на 20-30%	ВС	СН
14		Затримка фінансування від інвесторів	ВВ	НС
15		Збільшення вартості зовнішніх сервісів	СС	НВ
16	Організаційні	Втрата ключового розробника	ВВ	СН
17		Недостатня експертиза команди з AI	ВС	СВ
18		Конфлікти між членами команди	СН	ВС
19	Зовнішні	Зміна законодавства про захист даних	ВН	НС

3.4.3 Кількісний аналіз та ранжування ризиків

Для кількісної оцінки ризиків (деталізовано у таблиці 3.3) використовувалася шкала від 1 до 9 для оцінки впливу на час та фінанси, а також для ймовірності та частоти. Комплексний показник важливості ризику розраховувався за формулою:

$$\text{Важливість} = (\text{Вплив на час} + \text{Вплив на фінанси}) \times \text{Ймовірність} \times \text{Частота} \quad (3.1)$$

Кількісний аналіз ризиків

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота (за проект)		Важливість ризику компл. показн
		Якіс оц	Кільк оц	Якіс оц	Кільк оц	Якіс оц	Кільк оц	Якіс оц	Кільк оц	
1	2	3	4	5	6	7	8	9	10	11
1	Недоступність Firebase сервісів	ВВ	9	ВВ	9	ВН	7	НС	2	63
2	Зміна тарифів OpenAI API	СВ	6	ВС	8	СС	5	НВ	3	40
4	Застарілі Firebase функцій	СС	5	СН	4	НВ	3	НН	1	12
6	Несумісність з iOS/Android версіями	ВС	8	СВ	6	СС	5	СН	4	30
7	Проблеми з безпекою персональних даних	ВВ	9	ВВ	9	НС	2	НН	1	18
8	Низька продуктивність мобільного додатку	СВ	6	СС	5	СВ	6	СС	5	30
9	Низький попит на коворкінги	ВВ	9	ВВ	9	СН	4	НН	1	36
10	Сильна конкуренція (WeWork, Mindspace)	СС	5	ВС	8	ВС	8	НН	1	64
11	Неточне визначення цільової аудиторії	СВ	6	СВ	6	СС	5	НС	2	30
12	Негативні відгуки в App Store/Google Play	СН	4	СВ	6	СС	5	НВ	3	30
13	Перевищення бюджету на 20-30%	НС	2	ВС	8	СВ	6	НН	1	48

1	2	3	4	5	6	7	8	9	10	11
14	Затримка фінансування від інвесторів	ВВ	9	ВВ	9	НВ	3	НН	1	21
15	Збільшення вартості зовнішніх сервісів	НВ	3	СС	5	СС	5	СВ	6	25
16	Втрата ключового розробника	ВВ	9	ВВ	9	СН	4	НС	2	36
17	Недостатня експертиза команди з АІ	ВН	7	ВС	8	СВ	6	НС	2	48
18	Конфлікти між членами команди	СВ	6	СН	4	НВ	3	НВ	3	12
19	Зміна законодавства про захист даних	СС	5	СВ	6	НС	2	НН	1	12

На основі розрахованих показників важливості ризику, проведено (табл. 3.4–3.6) ранжування всіх 19 ризиків:

Таблиця 3.4

Найбільш впливові ризики (показник 48-64)

Ранг	Ризик	Показник важливості
1	Сильна конкуренція (WeWork, Mindspace)	64
2	Недоступність Firebase сервісів	63
3-4	Перевищення бюджету на 20-30%	48
3-4	Недостатня експертиза команди з АІ	48

Таблиця 3.5

Середнього впливу ризику (показник 25-40)

Ранг	Ризик	Показник важливості
1	2	3
5	Зміна тарифів OpenAI API	40
6-8	Досягнення лімітів OpenAI запитів	36

1	2	3
6-8	Низький попит на коворкінги	36
6-8	Втрата ключового розробника	36
9	Проблеми масштабування React Native	32
10-13	Несумісність з iOS/Android версіями	30
10-13	Низька продуктивність мобільного додатку	30
10-13	Неточне визначення цільової аудиторії	30
10-13	Негативні відгуки в App Store/Google Play	30
14	Збільшення вартості зовнішніх сервісів	25

Таблиця 3.6

Найменш впливові ризики (показник 12-21)

Ранг	Ризик	Показник важливості
15	Затримка фінансування від інвесторів	21
16	Проблеми з безпекою персональних даних	18
17-19	Застарілі Firebase функцій	12
17-19	Конфлікти між членами команди	12
17-19	Зміна законодавства про захист даних	12

3.4.4 Розробка протиризикових заходів (План реагування)

На основі проведеного аналізу та ранжування, особливо для ризиків з високим та середнім рівнем впливу, було розроблено план реагування (табл. 3.7). Цей план включає три типи заходів:

1. Профілактичні заходи (ПРЗ 1): дії, що вживаються до настання ризику, щоб зменшити його ймовірність або вплив.
2. Заходи при виявленні симптомів (ПРЗ 2): дії, що вживаються, коли з'являються *ранні ознаки* (тригери) ризику.
3. Заходи при настанні ризику (ПРЗ 3): конкретний план дій, якщо ризик вже відбувся (план «Б»).

Протиризикові заходи проєкту

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		профілактика		при симптомі	при проблемі
1	2	3	4	5	6
1	Сильна конкуренція (WeWork, Mindspace)	Провести детальний аналіз конкурентів, розробити унікальну пропозицію цінності, зосередитися на нішевих сегментах	Зниження завантажень, негативна динаміка реєстрацій	Агресивна маркетингова кампанія, тимчасові знижки, покращення UX	Переосмислення бізнес-моделі, пошук нових ніш, партнерства
2	Недоступність Firebase сервісів	Моніторинг стану Firebase, резервні копії, план аварійного переключення	Повільна відповідь API, помилки в логах, скарги користувачів	Активація резервної інфраструктури, повідомлення користувачам	Екстрена міграція на AWS/Azure, відновлення з backup
3	Перевищення бюджету на 20-30%	Детальний бюджет з резервом 15%, етапи контролю витрат	Витрати перевищують план на 10%, затримки завдань	Заморожування нових функцій, перегляд пріоритетів	Пошук додаткового фінансування, скорочення до MVP
4	Недостатня експертиза команди з AI	Навчання команди з AI/ML, залучення консультанта, база знань	Повільна розробка AI-функцій, низька якість рекомендацій	Зовнішній експерт на проєктній основі, інтенсивне навчання	Найом AI-розробника або аутсорсинг AI-частини
5	Зміна тарифів OpenAI API	Відстеження ценової політики, 20% резерв у бюджеті	Оголошення про підвищення цін, зростання витрат	Оптимізація використання API, пошук альтернатив	Міграція на Anthropic/Google, розробка власних алгоритмів
6	Досягнення лімітів OpenAI запитів	Моніторинг використання API, встановлення лімітів в додатку	Наближення до 80% від ліміту, уповільнення відповідей	Тимчасове обмеження функцій, оптимізація запитів	Екстрене підвищення тарифного плану, кешування результатів
7	Низький попит на коворкінги	Дослідження ринку, тестування гіпотез з MVP	Низькі показники реєстрації, відсутність активності користувачів	Перегляд маркетингової стратегії, фокус-групи	Зміна бізнес-моделі, розширення на суміжні сегменти

Продовження табл. 3.7

1	2	3	4	5	6
8	Втрата ключового розробника	Документування процесів, knowledge sharing, підготовка заміни	Конфлікти в команді, незадоволеність, пошук роботи	Переговори про утримання, розподіл обов'язків	Екстрений найм заміни, передача знань, фрілансер
9	Проблеми масштабування React Native	Code review, використання best practices, тестування навантаження	Уповільнення додатку, збої при збільшенні користувачів	Оптимізація коду, розбиття на модулі, кешування	Рефакторинг архітектури, перехід на native розробку
10	Несумісність з iOS/Android версіями	Тестування на різних версіях ОС, використання стабільних API	Скарги на роботу в певних версіях ОС	Швидкий патч для проблемних версій	Підтримка обмеженої кількості версій ОС
11	Низька продуктивність мобільного додатку	Профілювання коду, оптимізація з самого початку	Повільне завантаження, скарги користувачів	Аудит продуктивності, оптимізація критичних частин	Повне переписування проблемних модулів
12	Неточне визначення цільової аудиторії	Проведення інтерв'ю з потенційними користувачами, створення persona	Низький рівень зацікавленості від користувачів	A/B тестування різних підходів, додаткові дослідження	Кардинальний перегляд цільової аудиторії
13	Негативні відгуки в App Store/Google Play	Якісне тестування, система зворотного зв'язку	Перші негативні відгуки, зниження рейтингу	Швидке реагування на відгуки, випуск патчів	Масштабна кампанія з покращення репутації
14	Збільшення вартості зовнішніх сервісів	Моніторинг цін сервісів, пошук альтернатив	Повідомлення про підвищення тарифів	Переговори з постачальниками, оптимізація використання	Міграція на дешевші альтернативи
1	2	3	4	5	6
15	Затримка фінансування від інвесторів	Робота з декількома інвесторами, альтернативні джерела	Затягування переговорів, зміна умов	Активний пошук інших інвесторів, краудфандинг	Скорочення команди, фокус на самофінансуванні
16	Проблеми з безпекою персональних даних	Аудит безпеки, шифрування даних, дотримання GDPR	Підозрілі спроби доступу, скарги на приватність	Посилення системи безпеки, аудит доступів	Залучення експертів з кібербезпеки, повідомлення користувачів

17	Застарілі Firebase функцій	Регулярні оновлення, відстеження roadmap Firebase	Попередження про застарілість функцій	Планове оновлення до нових API	Термінова міграція, можлива зміна архітектури
18	Конфлікти між членами команди	Тімбілдінг, чіткі ролі в команді	Негативні висловлювання, напруженість	One-on-one зустрічі, медіація конфлікту	Реструктуризація команди, заміна конфліктних учасників
19	Зміна законодавств а про захист даних	Відстеження законодавчих змін, юридичні консультації	Оголошення нових вимог, обговорення в ЗМІ	Аудит відповідності, план адаптації	Термінове приведення у відповідність, юридична підтримка

3.5 Розробка бюджету проєкту

Бюджет проєкту є невід’ємною та найважливішою частиною будь-якого проєкту. Розробка бюджету є ключовим процесом в управлінні проєктом, що визначає фінансові рамки, необхідні для успішної реалізації всіх запланованих робіт. Для даного проєкту була обрана модель фінансування Fixed price for MVP. Тобто, зідяна фіксована сума від інвесторів для виконання мінімуму для проєкту, який вже можна запустити. Процес імплементації такої моделі становить такі кроки: замовник (інвестор) та команда домовляються не про весь проєкт, а лише про перший великий реліз. Обсяг цього MVP чітко визначений у беклозі продукту та плані спринтів. Фіксується ціна на цей MVP. Команда зобов’язується доставити визначений MVP за надану суму та термін, який в даному проєкті становить 5 спринтів та резервний час, наприклад – місяць. Як було зазначено в описанні беклогу, спринтів раніше, зазвичай не виходить завжди слідувати дедлайнам, особливо в розробці додатку (наприклад, захворів головний девелопер, або взяв відпустку, неочікувані проблеми з бекендом, деякі баги потребують більше часу на вирішення). Багато речей може піти не так, враховуючи ті ж стандартні баги, які будуть потребувати переносу на наступний спринт, або будуть змінюватися виходячи з пріоритетності. Або колаборація з адміністраціями інших коворкінгів, які

бажають взяти участь в даному проєкті, Перемовини можуть зайняти більше часу, ніж описано з самого початку. (наприклад, захворів головний девелопер, або взяв відпустку, неочікувані проблеми з бекендом, деякі баги потребують більше часу на вирішення)

Головною перевагою цієї моделі для інвестора становить прогнозованість: інвестор точно знає, скільки він заплатить і що він за це отримає. Для команди перевагою буде внутрішня гнучкість, команда має повну гнучкість всередині цих 10 тижнів. Вона може використовувати Scrum, проводити спринти, змінити пріоритети всередині спринтів, не змінюючи загальної вартості контракту. Нижче наведено (табл. 3.8) з потенційним бюджетним розрахунком за методом оцінки знизу-вгору, де вартість розраховується на основі декомпозиції робіт та вартості залучених спеціалістів, а також прямих операційних витрат.

Таблиця 3.8

Бюджет проєкту

Категорія витрат	Орієнтовна вартість
Проектний менеджер	\$6000
UX/UI-дизайн	\$3000
Back-end розробка	\$12000
Front-end (мобільна розробка)	\$9000
Розробка та впровадження персоналізованих сервісів ШІ	\$1500
Тестування	\$6000
Хостинг та технічна інфраструктура	\$1000
Реклама та маркетинг на старті	\$3000
Резерв на непередбачувані витрати	\$5000
Всього:	\$46000 - \$50000

3.6 Розробка плану управління якістю в проєкті

Успіх проєкту з розробки програмного забезпечення визначається не лише дотриманням термінів та бюджету, але й, у першу чергу, якістю кінцевого продукту. Для системної інтеграції управління якістю в життєвий цикл проєкту використовуються міжнародні стандарти, розроблені Міжнародною організацією зі стандартизації (ISO). Досвід, отриманий під час проходження

практики в ТОВ «Центр сучасної сертифікації «АЛЬТЕРНАТИВА», дозволив поглиблено дослідити цю тему [25].

Хоча «Альтернатива» спеціалізується на сертифікації *систем* управління (як-от ISO 9001), принципи та підходи, закладені в стандартах ISO, є універсальними. Для завдань дипломної роботи, пов'язаної з управлінням ІТ-проєктом, найбільш релевантними є наступні стандарти.

ISO 21500:2012 – Настанови з управління проєктами (Guidance on project management). Цей стандарт надає загальні високорівневі настанови щодо концепцій та процесів управління проєктами. Він є описовим, а не приписним, тобто він пояснює, *що* потрібно робити в управлінні проєктами, але не диктує, *як* саме. ISO 21500 структурує управління проєктами через 39 процесів, згрупованих у 5 груп (Ініціація, Планування, Виконання, Контроль, Закриття) та 10 предметних областей (Інтеграція, Зацікавлені сторони, Обсяг, Ресурси, Час, Вартість, Ризики, Якість, Закупівлі, Комунікації). Цей стандарт є корисним для формування загальної структури та розуміння життєвого циклу проєкту [26].

ISO 10006:2017 – Системи управління якістю. Настанови щодо управління якістю в проєктах. Цей стандарт є прямим доповненням до відомого ISO 9001, але сфокусованим виключно на проєктній діяльності. Він надає детальні вказівки щодо того, як застосовувати принципи управління якістю в проєктному середовищі. ISO 10006 допомагає керівникам проєктів інтегрувати процеси забезпечення якості (Quality Assurance) та контролю якості (Quality Control) на всіх етапах проєкту, від планування до закриття. Він підкреслює важливість чіткого визначення вимог до якості, планування заходів з контролю та постійного моніторингу для запобігання дефектам [27].

Серія ISO/IEC 25000 (SQuaRE) – Вимоги до якості та оцінка програмного продукту (Systems and software Quality Requirements and Evaluation)

Якщо ISO 21500 та 10006 фокусуються на якості *процесів* управління, то серія стандартів ISO/IEC 25000 (відома як SQuaRE) фокусується виключно на

якості самого *програмного продукту*. Це робить її найбільш критично важливим стандартом для проєкту «Coworkly».

Ця серія стандартів надає єдину термінологію та набір моделей для оцінки якості ПЗ. Ключовим у цій серії є стандарт ISO/IEC 25010:2011 «Моделі якості систем та програмного забезпечення».

ISO/IEC 25010 визначає дві основні моделі якості:

1. Модель якості продукту. Ця модель описує вісім характеристик, які визначають якість програмного продукту [28]. Нижче наведено, як нефункціональні вимоги проєкту «Coworkly» співвідносяться з цією моделлю:

- *Функціональна придатність*. Наскільки повно і коректно додаток реалізує заявлені функції. Це включає коректну роботу пошуку, бронювання та оплати, а також вимогу NFR007 (Підтримка багатомовності та мультивалютності), що забезпечує коректне відображення інтерфейсу (UA/EN) та цін (UAH, USD, EUR).
- *Продуктивність*. Як швидко додаток реагує та скільки ресурсів споживає. Ця характеристика прямо покривається вимогами NFR001 (обробка запитів не більше 2 секунд) та NFR002 (підтримка горизонтального масштабування для обслуговування великої кількості користувачів).
- *Сумісність*. Здатність додатку працювати в своєму середовищі разом з іншими системами. Для «Coworkly» це означає коректну взаємодію з платіжними шлюзами (Apple Pay, Google Pay) та, наприклад, системними календарями користувача для додавання бронювань.
- *Юзабіліті*. Наскільки легко та приємно користувачам взаємодіяти з додатком. Ця характеристика частково покривається вимогою NFR005 (Доступність для людей з обмеженими можливостями: контрастність, озвучування, масштабування тексту).

- *Надійність.* Здатність системи працювати без збоїв. Це прямо відображено у вимозі NFR006 (забезпечення безперебійної роботи з доступністю не менше 99,9% на місяць).
- *Безпека.* Захищеність даних. Ця вимога детально описана в NFR003 (відповідність стандартам безпеки OWASP Mobile Top 10, шифрування, захист особистих даних).
- *Супроводжуваність.* Характеристика, що не була вказана в NFR, але є критичною для команди розробки. Вона означає, наскільки легко вносити зміни в код, виправляти помилки або додавати нові функції. Для «Coworkly» це означає дотримання чистих принципів архітектури (напр., Clean Architecture) та високе покриття коду тестами.
- *Портативність:* Легкість, з якою додаток можна перенести на різні платформи. Це прямо покрито вимогою NFR004 (Продукт повинен працювати на Android та iOS з однаковим функціоналом).

2. Модель якості у використанні. Ця модель оцінює якість не з технічної, а з точки зору кінцевого користувача «Coworkly», який взаємодіє з системою у реальному контексті для досягнення своєї мети. Вона включає:

- *Ефективність,* а саме чи досягає користувач своїх цілей. Наприклад, користувач відкрив додаток, знайшов коворкінг «WorkHub» на 14:00, оплатив і отримав підтвердження бронювання. Ціль досягнуто.
- *Результативність,* а саме скільки зусиль (часу, кліків) користувач на це витратив. Наприклад, чи зміг користувач забронювати місце за 3 кліки, чи йому довелося проходити 10 екранів, повторно вводити дані та розбиратися в незрозумілому календарі.
- *Задоволеність,* а саме суб'єктивне сприйняття користувачем процесу. Наприклад, чи був процес бронювання приємним та зрозумілим? Чи почувався користувач впевнено, що його бронювання надійно збережено? Чи не викликав інтерфейс роздратування?

- Відсутність ризику, а саме чи захищений користувач від економічних, фінансових ризиків або ризиків для здоров'я. Наприклад, мати впевненість у тому, що платіжні дані в безпеці (NFR003), що бронювання не «злетить» (NFR006), і що додаток не призведе до фінансової втрати (наприклад, подвійне списання).
- Покриття контексту, а саме наскільки додаток є корисним у всіх запланованих сценаріях. Наприклад, чи може користувач однаково легко забронювати місце за тиждень наперед з дому, і так само швидко знайти та забронювати найближче вільне місце, стоячи посеред вулиці і шукаючи, де попрацювати «прямо зараз».

Проаналізовані стандарти формують комплексну теоретичну базу для управління проектом «Coworkly». ISO 21500 та ISO 10006 надають структуру для управління проектними процесами, тоді як ISO/IEC 25010 надає конкретну, вимірювану модель для визначення та контролю якості кінцевого продукту – мобільного додатку.

Для розробки плану з управління якістю в проекті використовується два фундаментальних види діяльності, а саме забезпечення та контроль якості [29].

Забезпечення якості – це процес-орієнтований підхід. Його мета – запобігти появі дефектів шляхом побудови правильних та ефективних процесів розробки. QA відповідає на питання: «Чи правильно ми робимо роботу?».

Контроль якості – це продукт-орієнтований підхід. Його мета - виявити дефекти у вже створеному продукті. Відповідає на питання: «Чи відповідає результат нашим вимогам?».

Для даного проекту не можна покладатися лише на один із цих підходів. Без QA команда буде постійно створювати нові помилки, а без QC ці помилки потраплятимуть до кінцевого користувача. Тому план якості включає обидва напрямки. Наступний крок це призначення ролей та відповідальності за якість.

Успішне впровадження плану якості починається з чіткого розподілу відповідальності в межах Scrum-команди:

- Проєктний менеджер (на якого також припали обов'язки власника продукту), який відповідає за бізнес-якість. Він гарантує, що проєкт рухається у правильному напрямку, визначаючи якість через чіткі критерії до кожної user story. Він єдиний, хто приймає рішення, чи є функціонал готовим під час перегляду спринту.

- Команда розробки, яка відповідає за технічну якість. Команда колективно зобов'язана створювати якісний продукт, дотримуючись визначення готовності, писати unit-тести, проводити code review та не допускати накопичення технічного боргу.

- Scrum-майстер, який відповідає за якість процесу. Він забезпечує дотримання практик Scrum, усуває перешкоди та допомагає команді дотримуватись стандартів якості, визначених у цьому плані.

Як зазначалося вище, забезпечення якості – це процеси, спрямовані на запобігання дефектам. Для проєкту «Coworkly» це життєво необхідно, оскільки виправляти помилки у вже опублікованому мобільному додатку (особливо пов'язані з оплатами та бронюваннями) набагато дорожче, ніж запобігти їм на етапі написання коду.

Основні інструменти QA у проєкті:

- Definition of Done [30] – це єдиний і найважливіший стандарт якості, який застосовується до кожної User Story. Він є формальною угодою всередині команди про те, що означає «завершити» роботу. Для проєкту «Coworkly» DoD включає:

1. Код написаний відповідно до стайлгайду (забезпечує супроводжуваність).

2. Код пройшов обов'язкове Code Review (огляд коду) іншим членом команди (запобігає логічним помилкам).

3. Написані та успішно пройдені Unit-тести (гарантує стабільність окремих функцій).

4. Функціонал пройшов мануальне тестування QA-спеціалістом (перевірка з точки зору користувача).

5. Успішно пройдені всі критерії прийняття (Acceptance Criteria), визначені проєктним менеджером (відповідність бізнес-вимогам).

6. Відсутні нові критичні або блокуючі баги.

- *Code Review*: впровадження обов'язкового огляду коду для кожної нової функції перед її злиттям в основну гілку розробки. Це не лише виявляє помилки, але й забезпечує обмін знаннями та покращує супроводжуваність коду в довгостроковій перспективі.

- *Планування спринту (в контексті QA)*: включення завдань з якості (наприклад, «Написати автотести для модуля оплати», «Налаштувати CI/CD») у беклог спринту на рівні з функціональними задачами, а не як щось вторинне.

На відміну від забезпечення якості, контроль якості – це процеси, спрямовані на виявлення дефектів у вже створеному продукті. Це класичне тестування. Якщо QA – це «побудувати правильно», то QC – це «перевірити, чи ми справді побудували правильно».

Для проєкту «Coworkly» QC включатиме кілька рівнів тестування, щоб гарантувати стабільність на всіх етапах:

- *Unit-тестування*. Розробники пишуть тести, що перевіряють працездатність окремих малих «юнітів» коду (наприклад, чи правильно функція валідації email, чи коректно розраховується вартість бронювання).

- *Інтеграційне тестування*. Перевірка того, як різні модулі системи працюють разом (наприклад, чи коректно модуль «Бронювання» взаємодіє з модулем «Оплата» та базою даних Firebase).

- *Системне тестування*. тестування повних сценаріїв користувача з точки зору «чорної скриньки» (наприклад, повний шлях: відкриття додатку → пошук → вибір коворкінгу → оплата → отримання підтвердження).

- *Регресійне тестування.* Проведення автоматизованих або мануальних тестів перед кожним релізом, щоб переконатися, що новий функціонал не зламав ті функції, які працювали раніше.

Наступний крок це сформувавши ключові метрики якості (KPIs) для проєкту. План якості не має сенсу, якщо його результати не можна виміряти. Для вимірювання успішності плану використовуються наступні метрики, що безпосередньо базуються на моделі ISO/IEC 25010:

- Метрики продуктивності (NFR001) → час відповіді сервера, ціль – 95% запитів обробляються швидше ніж за 2 секунди.

- Метрики надійності (NFR006) → uptime системи (час безвідмовної роботи), ціль – 99.9%).

- Метрики супроводжуваності → покриття коду тестами, ціль – не менше 80%.

- Метрики процесу → час «життя» багу, середній час від виявлення критичного багу до його виправлення (ціль: < 24 годин).

Таким чином, наданий комплексний план, що поєднує заходи забезпечення якості та контролю якості, дозволить системно керувати якістю проєкту «Coworkly».

РОЗДІЛ 4. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ КОРИСТУВАННЯ МЕРЕЖЕЮ КОВОРКІНГІВ «COWORKLY»

4.1 Опис структури програмного забезпечення та його архітектури

Програмне забезпечення платформи для бронювання коворкінг-просторів реалізується з використанням Google Firebase – сучасної хмарної платформи, яка надає повний стек для розробки, масштабування та обслуговування веб- і мобільних додатків [31; 24]. Такий підхід дозволяє швидко розгорнути проєкт, зменшити витрати на підтримку серверної інфраструктури та забезпечити високу надійність, масштабованість і безпеку.

Архітектура системи має три логічні рівні: клієнтський інтерфейс (frontend), хмарні серверні функції (Cloud Functions) та база даних (Firebase Realtime Database / Firestore).

Клієнтський рівень [33; 34]. Цей рівень реалізовано за допомогою фреймворку React Native для мобільного додатку. Користувацький інтерфейс включає наступні можливості:

1. Створення облікового запису, вхід через email або соціальні мережі.
2. Вибір мови інтерфейсу (українська, англійська).
3. Фільтрація та перегляд доступних коворкінгів.
4. Бронювання робочих місць на інтерактивному календарі.
5. Онлайн-оплата.
6. Оцінювання та коментування.
7. Перегляд рекомендацій, заснованих на попередньому досвіді користувача.

Frontend безпосередньо взаємодіє з Firebase через офіційні SDK, що спрощує інтеграцію та зменшує кількість посередників між клієнтом і базою даних [35].

Хмарний логічний рівень. Замість традиційного серверного середовища в цьому проєкті використано Firebase Cloud Functions, які виконують роль безсерверної обробки логіки, а саме:

1. Обробка запитів на бронювання, тобто перевірка доступності, створення запису в базі, генерація квитанції.

2. Обробка платежів, тобто інтеграція з платіжними сервісами (наприклад, Stripe або Fondy).

3. Автентифікація та безпека, тобто валідація токенів доступу, контроль прав доступу.

4. Тригери на зміну даних, тобто автоматичне оновлення статистики, рейтингу, історії тощо.

5. Виклик AI-інтерфейсів OpenAI, тобто передача даних для генерації персоналізованих рекомендацій.

Cloud Functions масштабуються автоматично відповідно до навантаження, не потребують ручного адміністрування і мають високий рівень безпеки, оскільки виконуються у захищеному середовищі Google Cloud.

Рівень бази даних. У якості сховища даних використовується Firebase Firestore [36] – документно-орієнтована хмарна база даних, що підтримує реальновчасну синхронізацію даних між клієнтом і сервером. Її переваги:

1. Висока швидкість доступу.
2. Автоматична реплікація і масштабування.
3. Підтримка офлайн-режиму на клієнтах.
4. Гнучка система правил безпеки доступу.

У базі зберігаються:

1. Дані користувачів (акаунти, налаштування).
2. Інформація про коворкінги (локація, доступність, опис).
3. Бронювання, історія, платежі.
4. Відгуки та рейтинги.
5. Збережені рекомендації користувачу.

Інтеграція штучного інтелекту. Особливістю даного проєкту є інтеграція зі штучним інтелектом, зокрема за допомогою OpenAI API. Ця інтеграція реалізована через виклики Cloud Functions [31], які:

1. Аналізують історію бронювань, час користування, вподобання.
2. На основі аналізу генерують персоналізовані тексти рекомендацій щодо нових коворкінгів.
3. Додатково – обробляють відгуки користувачів, здійснюючи їх класифікацію (позитивний, негативний, нейтральний) та тематичний аналіз.

Це дозволяє реалізувати інтелектуальні функції, які не просто фіксують дані, а й надають цінні підказки користувачу – наприклад, рекомендуючи менш завантажені простори або простори з високою оцінкою у визначений час.

Безпека та обробка даних.

1. Аутентифікація реалізується через Firebase Authentication (підтримка email/password, Google, Facebook) [37; 38].
2. Доступ до даних регулюється через Firebase Security Rules [39].
3. Усі дані передаються по зашифрованому каналу (HTTPS).
4. Для зберігання платіжної інформації використовуються зовнішні сертифіковані платформи, що відповідають стандартам PCI DSS.

Важливість формування інформаційної структури IT-проєкту розробки інтернет-сервісу визначається стрімким розвитком технологій і зростанням вимог користувачів до простоти використання та ефективності. У світлі постійних змін в інтернет-середовищі оптимальна організація інформації стає важливою складовою успішної реалізації проєкту. Забезпечення легкого доступу та зрозумілого інтерфейсу стає необхідним для залучення та утримання користувачів. Інформаційна структура відображає спрощення та оптимізацію взаємодії між користувачами та сервісом, що є критично важливим у конкурентному цифровому середовищі [40].

На рис. 4.1. наведено модель архітектури інформаційної системи в графічному вигляді.

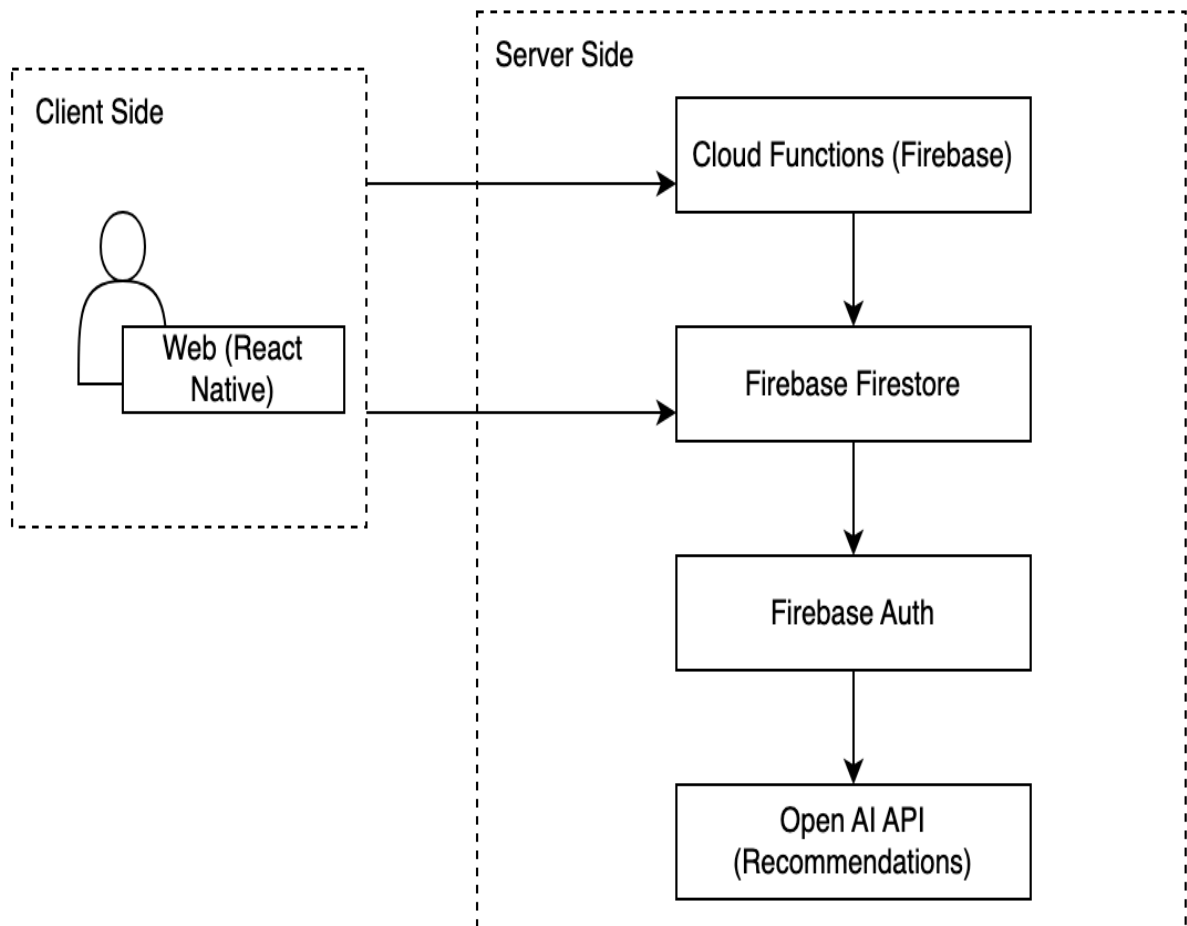


Рис. 4.1. Модель архітектури в графічному вигляді

4.2 Розробка концептуальної та логічної моделей бази даних проєкту

У межах роботи з розробки мобільного додатку «Coworkly» була створена концептуальна модель бази даних [41] наведена нижче (рис. 4.2), яка виступає основою для формалізації структури системи. На поданій діаграмі можна побачити ключові сутності, що беруть участь у функціонуванні сервісу. Серед них – користувач, коворкінг-простір, робоче місце, бронювання, відгук та улюблене. Кожна з цих сутностей має власний набір атрибутів, що описують її характеристики: наприклад, користувач має ім'я, прізвище, email, номер телефону та пароль, а коворкінг включає назву, адресу, кількість місць, опис і графік роботи.

Зв'язки між об'єктами чітко визначають логіку взаємодії: один користувач може залишити багато відгуків, але кожен відгук пов'язаний лише з одним користувачем; кожне робоче місце належить до конкретного коворкінгу, а бронювання поєднує користувача, коворкінг і робоче місце в певному часовому проміжку. Улюблене дозволяє користувачам зберігати вподобані коворкінги.



Рис 4.2. Концептуальна модель бази даних

Логічна модель бази даних [42] показана далі (рис. 4.3) є наступним етапом після створення концептуальної моделі інформаційної системи. Вона представляє собою більш формалізовану та деталізовану структуру даних, яка вже наближена до реалізації в системі управління базами даних (СУБД), але ще не містить конкретних технічних рішень щодо фізичного зберігання.

Її основна мета – описати логіку організації даних, структуру таблиць, атрибути (поля), типи даних, а також ключі та обмеження, які забезпечують цілісність і узгодженість інформації. Логічна модель бази даних демонструє, яким чином дані з концептуальної моделі будуть організовані у вигляді таблиць, які між ними існують зв'язки (один до одного, один до багатьох,

багато до багатьох) та які поля відіграють роль первинних або зовнішніх ключів. Це дозволяє глибше зануритися у технічні аспекти побудови інформаційної системи, передбачити можливі проблеми при реалізації запитів, оптимізації структури бази та забезпечення її масштабованості.

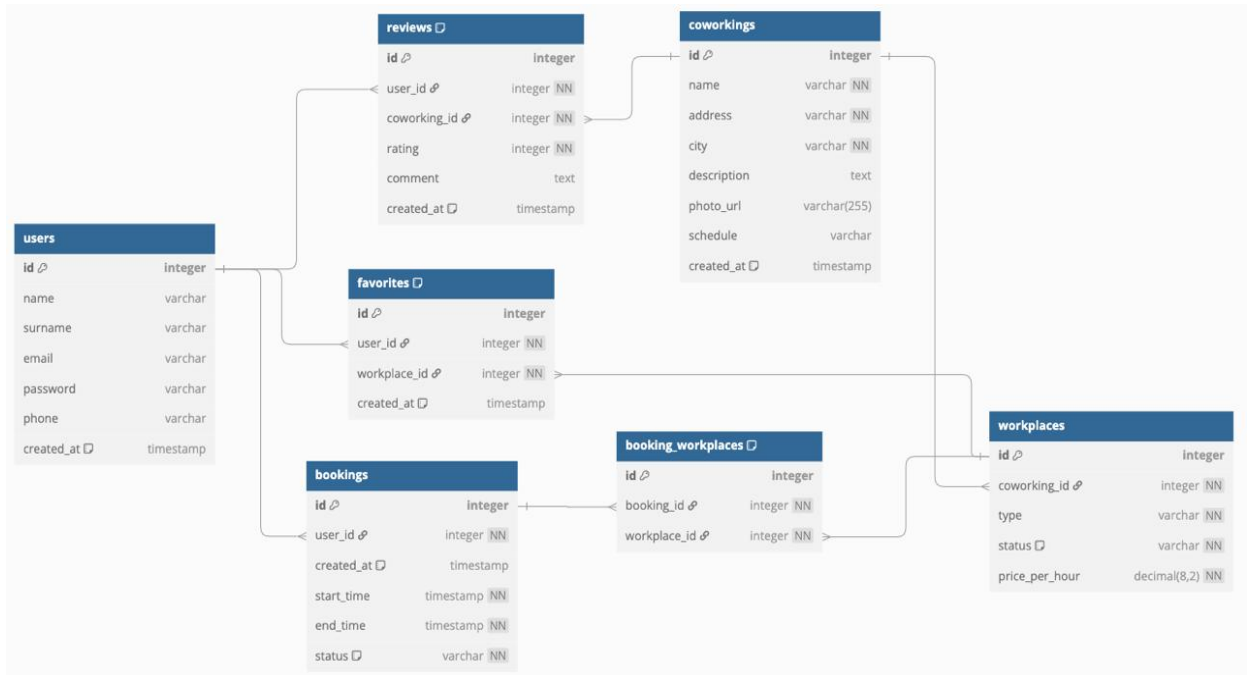


Рис. 4.3. Логічна модель бази даних

Важливість логічної моделі полягає передусім у тому, що вона виступає технічним фундаментом для створення реальної бази даних. Вона дозволяє розробникам спроектувати зручну та ефективну структуру, яка забезпечить швидкий доступ до інформації, її цілісність, захист і послідовність при оновленнях. Також ця модель є критично важливою для подальшого написання запитів, створення API [20], формування інтерфейсів взаємодії з даними, а також – у випадку інтеграції з іншими системами.

На основі логічної моделі можна побачити, які саме таблиці буде створено в базі даних, які атрибути вони матимуть, які поля використовуються для встановлення зв'язків між таблицями, як реалізуються зв'язки «один до багатьох» або «багато до багатьох», та як забезпечується унікальність даних. Наприклад, у системі коворкінгу це дозволяє точно відстежити, хто саме

забронював робоче місце, в якому коворкінгу, на який час, і яким був статус цього бронювання.

Формування концептуальної та логічної баз даних є критично важливим етапом у процесі розробки веб-платформи для автоматизованої перевірки та оцінювання письмових тестів. Це дозволяє структуровано зберігати всю необхідну інформацію для ефективної роботи системи. Правильна концептуалізація даних допомагає забезпечити зручний доступ до них та швидкий пошук необхідної інформації. Логічна база даних дозволяє забезпечити коректність і несуперечливість даних, що важливо для точності і надійності процесу перевірки та оцінювання тестів. Це також полегшує масштабування та модифікацію системи в майбутньому [43].

4.3 Розробка інтерфейсів програмного забезпечення

У процесі створення інтерфейсу мобільного застосунку для мережі коворкінгів Coworkly було дотримано сучасні підходи до розробки UX/UI-дизайну [44], орієнтуючись насамперед на зручність і зрозумілість для кінцевого користувача. Незважаючи на те, що в досвіді немає професійної дизайнерської освіти, метою було створити інтерфейс, який би забезпечував інтуїтивну навігацію, простоту взаємодії та позитивний користувацький досвід.

В основі концепції дизайну було застосовано принцип інтуїтивного інтерфейсу, який передбачає логічну побудову всіх елементів інтерфейсу. Основні функції, такі як пошук коворкінгів, бронювання робочих місць, перегляд профілю та онлайн-оплата, були розміщені так, щоб користувач міг легко їх знайти та використовувати без необхідності додаткових інструкцій.

Крім того, використовувались принципи мінімалізму та адаптивності, що є надзвичайно актуальними у сучасному мобільному дизайні. В інтерфейсі були використані прості кольорові рішення, чітка типографіка та достатній

простір між елементами, що покращує зчитуваність інформації та взаємодію з додатком.

Для створення прототипів було використано хмарний інструмент Figma [45; 46], який є одним із найпопулярніших сервісів для розробки інтерфейсів. Його перевагами є зручність у використанні, інтуїтивний інтерфейс, багатофункціональність та можливість роботи безпосередньо в браузері. Також важливою перевагою Figma є підтримка командної роботи: всі правки та оновлення можуть синхронізуватися в режимі реального часу. Це особливо цінно в умовах колективної розробки або при отриманні зворотного зв'язку від керівника чи команди.

Підхід до проєктування був орієнтований на користувача, що означає створення інтерфейсу з урахуванням потреб, очікувань та поведінки потенційних користувачів. Було детально продумано сценарії взаємодії: від реєстрації в додатку до завершення бронювання. Кожен крок був організований логічно, а процеси максимально спрощені, щоб уникнути перевантаження інформацією чи зайвих дій.

Особливу увагу було приділено візуальній структурі: контент згруповано у зрозумілі блоки, основні дії підкреслені кольоровими акцентами (наприклад, кнопки бронювання виділені яскравим кольором).

Таким чином, вдалося створити інтерфейс, що відповідає сучасним стандартам, орієнтований на зручність користувача та готовий для подальшої реалізації у вигляді повноцінного програмного продукту.

Нижче буде послідовно описано фрагменти вікон додатку з розробленими інтерфейсами (рис. 4.4.-4.9.).

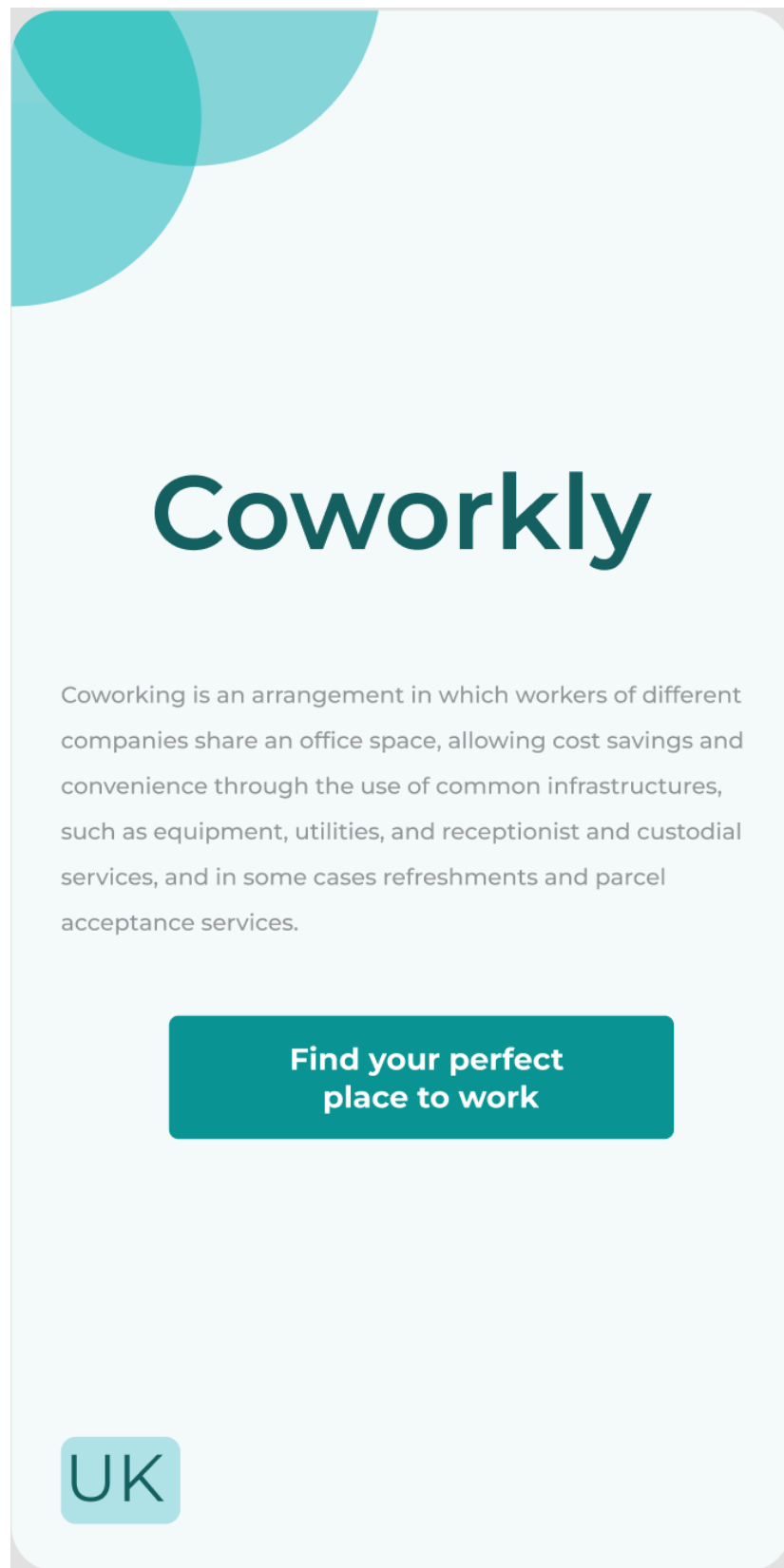


Рис. 4.4. Вікно після завантаження додатку

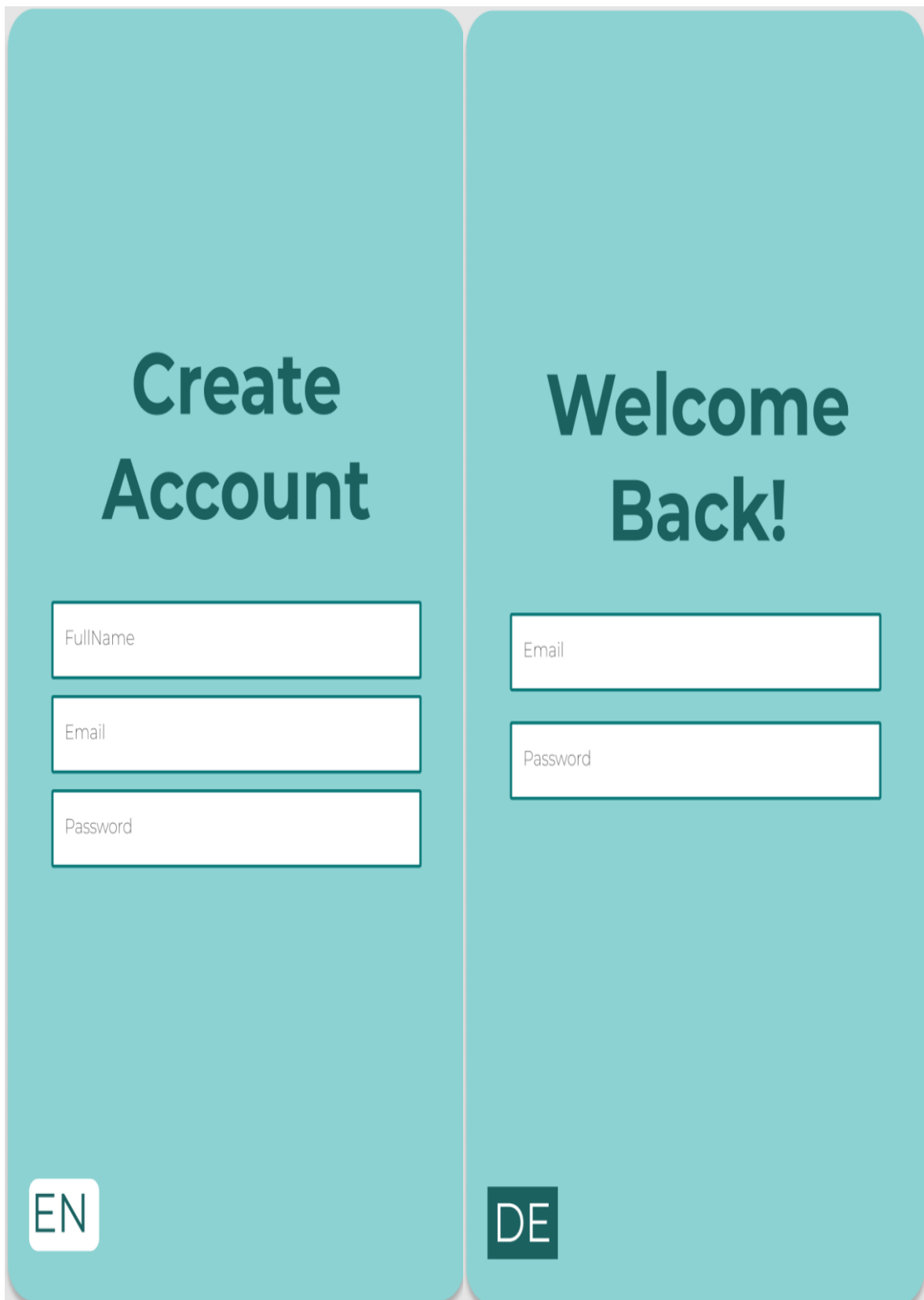


Рис. 4.5. Вікна реєстрації та автентифікації користувача

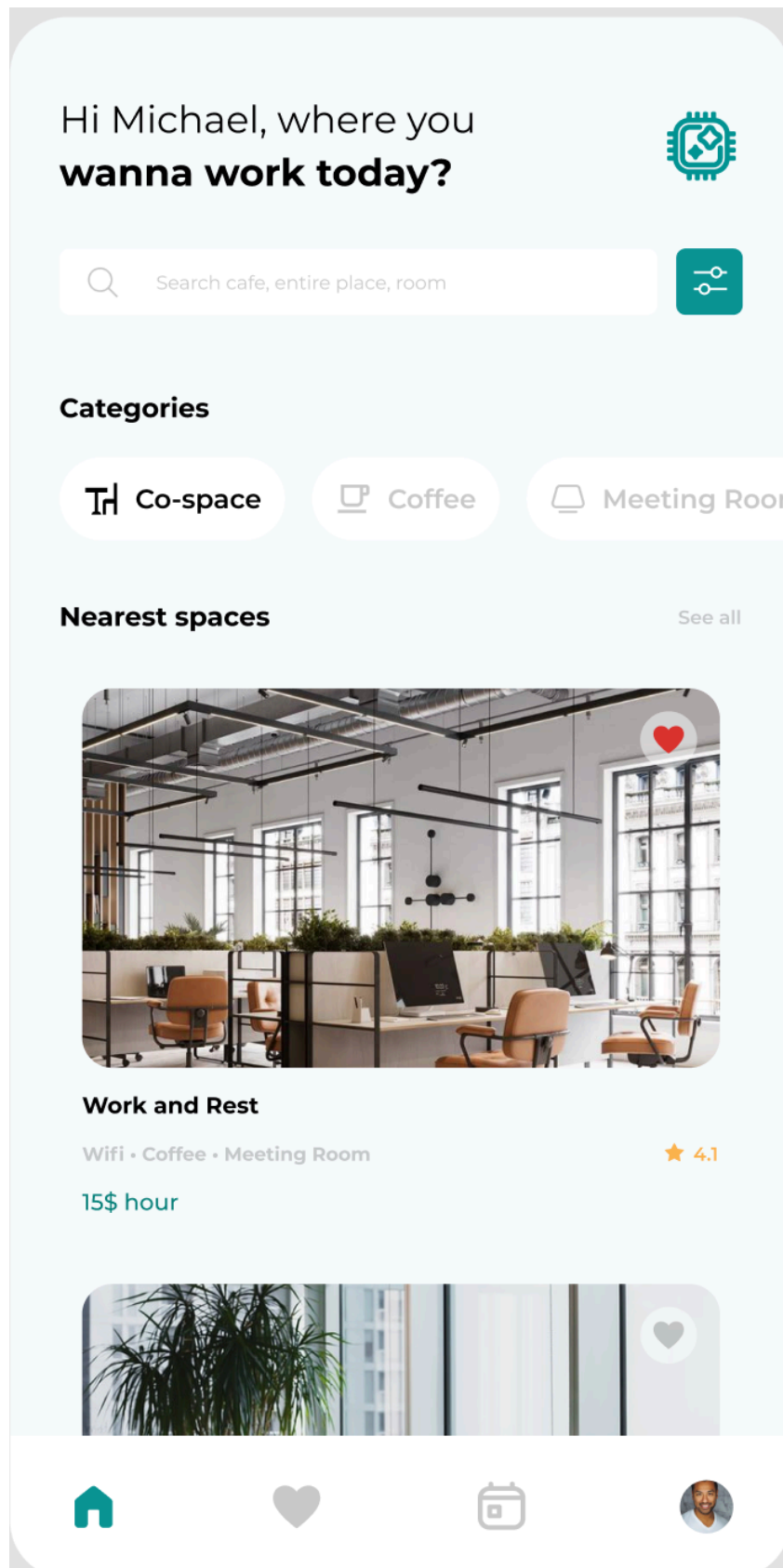


Рис. 4.6. Вікно зі списком і можливою фільтрацією доступних найближчих коворкінгів

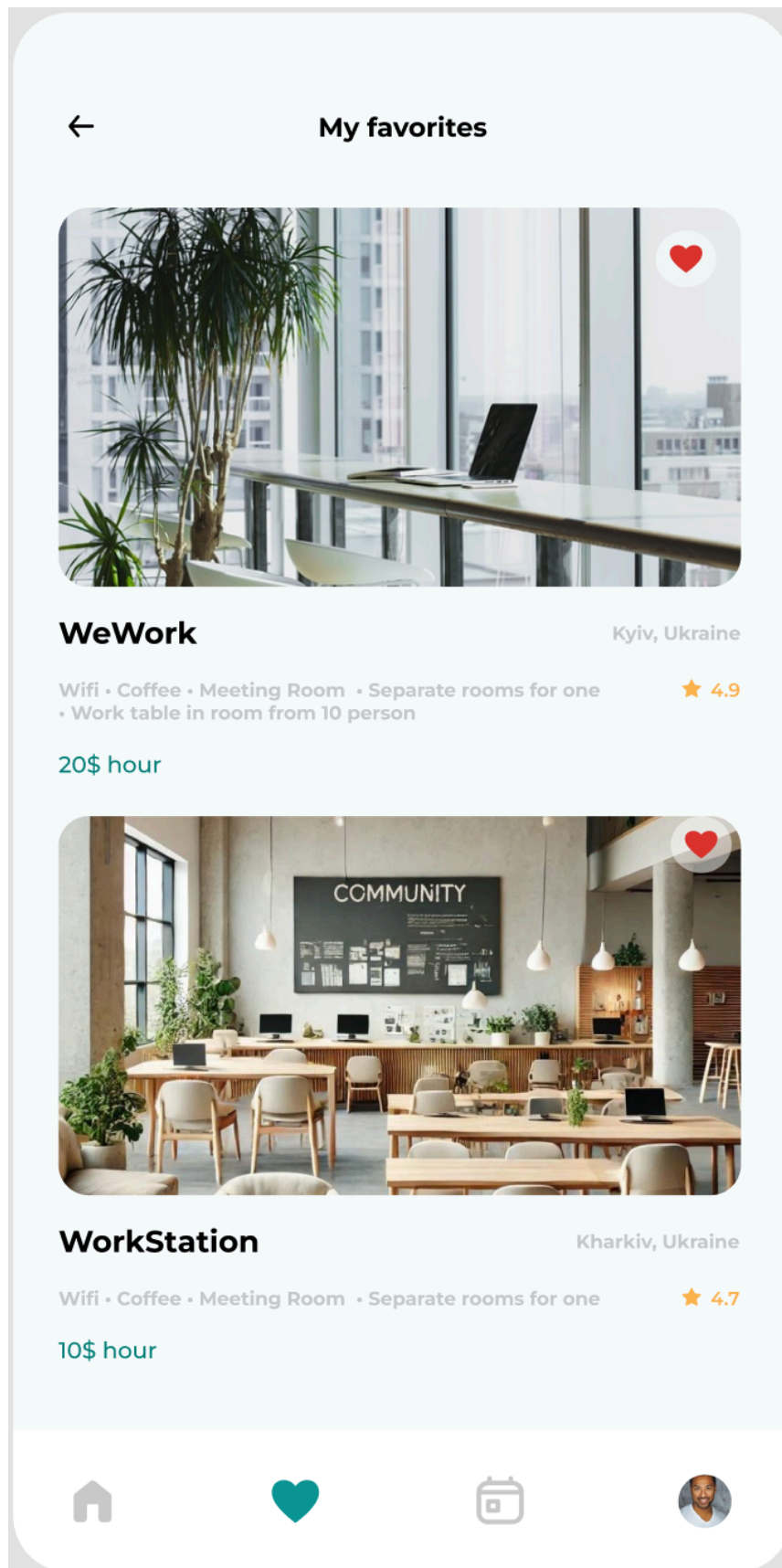


Рис. 4.7. Вікно «Улюблене», яке включає в себе коворкінгі, які сподобались користувачу і він їх зберіг за допомогою лайка

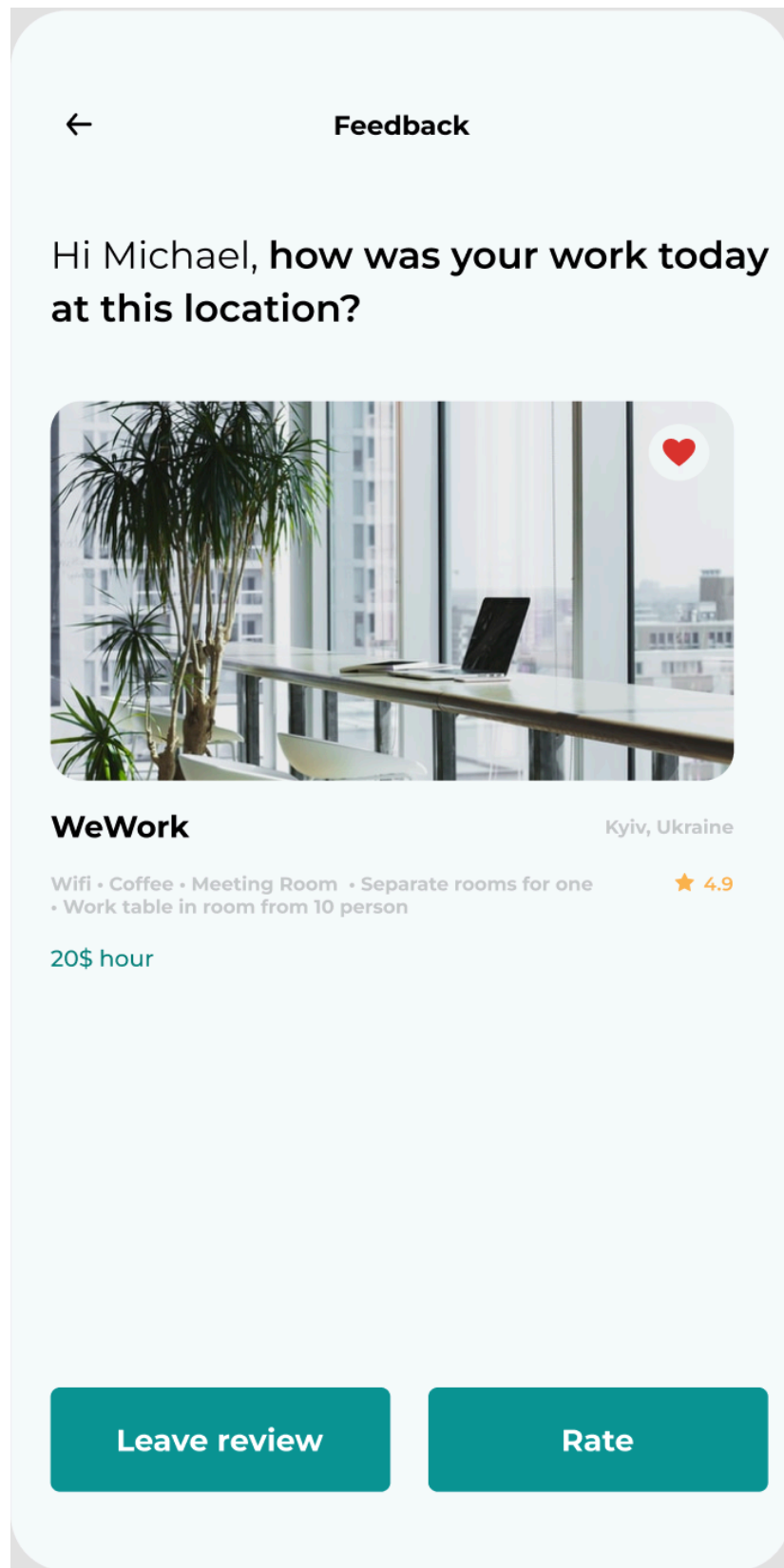
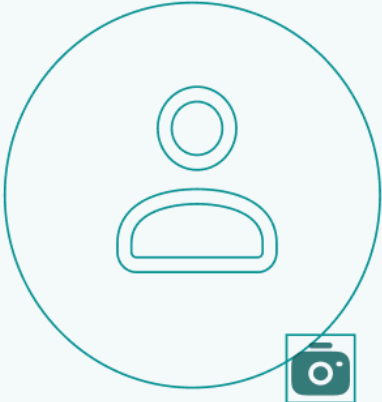



Рис. 4.8. Вікно яке з'являється після пройденого часу броні, питаючи у користувача як пройшла його робота на цій локації. У користувача є можливість оцінити коворкінг, та залишити текстовий коментар

Edit Profile





Name


Email

Password

Date of Birth

Country/Region

City


 

Рис. 4.9. Вікно редагування профілю користувача

4.4 Формування переліку пропозицій щодо успішного функціонування сервісу продукту

Успішне функціонування мобільного додатку «Coworkly» після фази MVP вимагає не лише технічної стабільності, але й постійної адаптації до динамічних вимог ринку та стратегічного розвитку. На основі проведеного дослідження сформовано перелік ключових пропозицій для забезпечення довгострокової життєздатності та масштабування сервісу.

Пропозиції щодо розширення функціоналу, що спрямовані на посилення унікальної торгової пропозиції та усунення ключових проблем, які не були повністю покриті в межах MVP.

1. Поглиблення інтеграції ШІ та персоналізації.

Впровадження прогностичної аналітики попиту, а саме використання машинного навчання для прогнозування завантаженості коворкінгів не лише на основі історії, але й на основі зовнішніх факторів (погода, державні свята, події в місті). Це дозволить партнерам-коворкінгам динамічно змінювати ціни.

Створення віртуального консультанта (Чат-бот), а саме розробка AI-бота на базі LLM, інтегрованого в додаток, для миттєвих відповідей на типові запитання користувачів про коворкінг, замість звернення до адміністратора.

2. Розвиток B2B-сервісу та командної роботи:

Модуль управління командами, тобто надання корпоративним клієнтам можливості додавати співробітників, встановлювати щомісячні ліміти витрат для кожного члена команди та генерувати єдину фінансову звітність.

Впровадження системи перепусток: інтеграція з локальними системами коворкінгів (QR-коди, NFC) для автоматичного доступу до заброньованого місця або приміщення.

3. Соціальний та нетворкінг-модуль, створення внутрішньої спільноти, тобто надання користувачам можливості переглядати профілі інших користувачів (з їхньої згоди, які наприклад мають бронювання в той же самий час в тій кімнаті), які працюють у тому ж коворкінгу, для налагодження

професійних зв'язків та спільної роботи. Можливо інтеграція чату між користувачами теж буде корисною.

4. Пропозиції щодо організаційної стійкості та управління, які стосуються підтримки процесів, розроблених у раніше, для забезпечення безперервної роботи та мінімізації ризиків.

Наприклад, створення служби підтримки 24/7 для швидкого реагування на проблеми з бронюванням або оплатою, що критично для сервісу, який залежить від реального часу.

Також, автоматизація тестування та масштабування, яка представляю собою перехід від переважно мануального до автоматизованого регресійного тестування, що забезпечить швидку та надійну доставку нових функцій у кожному спринті.

Моніторинг продуктивності, тобто впровадження інструментів APM для постійного відстеження показників, таких як час відповіді сервера та його надійність 99.9%.

5. Пропозиції щодо монетизації та масштабування, що спрямовані на використання аналітичних даних та фінансову складову проєкту.

Наприклад, аналітичні звіти для партнерів, такі як продаж власникам коворкінгів преміум-підписки на доступ до детальної аналітики: демографія користувачів, пікові години попиту в районі, порівняння цін з конкурентами. Рекламні послуги.

Географічне масштабування, яке можливо після успішного пілотного запуску на українському ринку, розгляд плану експансії на ринки Східної Європи (Польща, Румунія), де ринок коворкінгів активно зростає, і де гібридна модель праці вже добре прийнята.

ВИСНОВКИ

У процесі виконання магістерської кваліфікаційної роботи, яка була спрямована на розробку мобільного додатку для мережі коворкінгів, було виконано такі ключові завдання:

- проведено аналіз ринкових умов, впливу зацікавлених сторін на проєкт та їхніх очікувань. Проведено SWOT-аналіз, який допоміг ідентифікувати сильні сторони проєкту, можливі ризики та напрямки їхньої мінімізації;
- визначено функціональні та нефункціональні вимоги до додатку, що дозволило структурувати завдання та визначити пріоритети у розробці через формування User Stories;
- розроблено ієрархічну структуру робіт (WBS) та календарний план (Діаграма Ганта), що включає ініціацію, планування, реалізацію, тестування та завершення. Це дозволило створити чіткий та структурований план реалізації проєкту тривалістю у 6 місяців.

Під час виконання проєкту було обґрунтовано та впроваджено підхід Agile з фреймворком Scrum, що дозволило розбити розробку на короткі ітерації (5 спринтів), кожен з яких завершується тестуванням і наданням результатів для аналізу та зворотного зв'язку. Такий підхід забезпечив гнучкість, адаптивність до змін вимог ринку та прозорість процесу розробки.

Ключові результати проєкту включають:

- розробку комплексної управлінської документації, що включає бюджет проєкту, план управління якістю згідно зі стандартами ISO та стратегію реагування на ризики, що забезпечує фінансову та організаційну стійкість стартапу;
- оптимізацію використання ресурсів завдяки створенню гібридної моделі фінансування в рамках методології Scrum, що дозволяє інвесторам контролювати бюджет, а команді – зберігати гнучкість у технічних рішеннях;
- розробку технічної архітектури на базі React Native та Firebase, а також дизайну інтерфейсів, які включають інтеграцію штучного інтелекту для

надання персоналізованих рекомендацій користувачам, що є ключовою конкурентною перевагою.

Проект охоплює всі аспекти управління життєвим циклом розробки: від визначення мети та побудови дерева цілей до оцінки результатів і пропозицій щодо подальшого функціонування продукту. Розроблена система базується на сучасних технологіях управління проєктами та програмного забезпечення.

Отже, результати виконаної роботи демонструють ефективність використаних методів та підходів в управлінні ІТ-проєктами. Мобільний додаток «Coworkly» слугуватиме важливим інструментом для цифровізації ринку нерухомості, об'єднуючи зусилля розробників, бізнесу та користувачів. Ця робота може бути основою для подальших досліджень у сфері впровадження штучного інтелекту в продукти сервісної економіки.

Таким чином, результати кваліфікаційної роботи підтверджують високу ефективність інтегрованого підходу до управління ІТ-проєктами, що поєднує адаптивність гнучких методологій із міжнародними стандартами якості. Створений проєкт мобільного додатку «Coworkly» є готовим до реалізації інструментом, який не лише вирішує актуальну проблему пошуку робочого простору, але й демонструє потенціал застосування новітніх технологій штучного інтелекту в продуктах сервісної економіки.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Аналіз розміру та частки ринку коворкінг-просторів Європи - тенденції та прогнози зростання (2025-2030 рр.). Режим доступу:
<https://www.mordorintelligence.com/industry-reports/coworking-spaces-in-europe>
2. Hub-and-spoke model. Режим доступу:
<https://www.the-data-economist.com/blog-eng/the-hub-spoke-model-a-roadmap-to-a-data-inspired-organization>
3. Офіційний сайт платформи “Croissant”. Режим доступу:
<https://www.getcroissant.com/>
4. Інформація про “Croissant”. Режим доступу:
<https://liveitup-laspalmas.com/community/discounts/join-getcroissant-with-e15-20-discount/>
5. On-Demand strategy. How does it work? Режим доступу:
<https://www.wework.com/ideas/workspace-solutions/flexible-products/what-is-wework-on-demand-and-how-does-it-work>
6. Pay as you go service. Режим доступу:
<https://dealhub.io/glossary/pay-as-you-go-pricing/>
7. Stakeholders: Definition, Types, and Examples. Режим доступу:
<https://www.investopedia.com/terms/s/stakeholder.asp>
8. Інтеграція Apple та Google Pay в мобільний додаток. Режим доступу:
<https://medium.com/@queenXynok/how-to-integrate-applepay-googlepay-into-your-react-native-application-5f21db63515f>
9. ROI in Project Management. Режим доступу:
<https://www.float.com/resources/calculate-project-roi>
10. What is KPI in Project Management. Режим доступу:
https://www.projectmanagement.com/wikis/345150/key-performance-indicators#_
11. Що таке SWOT аналіз. Режим доступу:
<https://www.investopedia.com/terms/s/swot.asp>
12. What is a unique selling proposition? USP examples and definition. Режим доступу: <https://www.zendesk.de/blog/unique-selling-proposition/>

13. Ziuziun V., Kolomiets A., Kosenko D (2025). Utilizing swot analysis in designing an information system for real-time coverage of sports events. <https://doi.org/10.70286/eoss-21.04.2025>

14. Морозов В.В., Коломієць Г.С. Методи розробки концепцій ІТ-проектів. Методичні вказівки, 2024.

15. OpenAI Platform. Режим доступу: <https://openai.com/api/>

16. Що таке OBS. Режим доступу: <https://www.projectmanagertemplate.com/post/what-is-an-organizational-breakdown-structure-obs>

17. Інформація та порівняння методологій управління проектами. Водоспадна модель VS Agile. // Електронний ресурс. Режим доступу: <https://www.geeksforgeeks.org/software-engineering/agile-vs-waterfall/>

18. Все про Scrum. // Електронний ресурс. Режим доступу: <https://www.scrum.org/resources/what-scrum-module>

19. Scrum Events. // Електронний ресурс. Режим доступу: <https://www.scrum.org/resources/introduction-scrum-events>

20. What is an API (Application Programming Interface)?. Режим доступу: <https://aws.amazon.com/what-is/api/>

21. Методи управління ризиками в ІТ проектах [Текст]: методичні вказівки до виконання практичних, лабораторних робіт та самостійної роботи для студентів освітньої програми «Управління проектами» спеціальності 122 «Комп'ютерні науки» для денної і заочної форми навчання / Тімінський О.Г., Коломієць А.С. – К. : КНУ імені Тараса Шевченка, 2021. – 40 с.

22. MSF Methodology. Режим доступу: <https://www.msf.org/speakingout/methodology>

23. Аналіз WeWork. Режим доступу: <https://www.wework.com/ideas/workspace-solutions/how-robin-ai-scaled-globally-with-wework>

24. Mindspace becoming AI Hothouse. Режим доступу:

https://www.mindspace.me/pr/2017-11-27_global-co-working-provider-mindspace-becomes-a-hothouse-for-ai-research/

25. Офіційний вебсайт ТОВ «АЛЬТЕРНАТИВА» // Електронний ресурс. Режим доступу: <https://alternatyva.org.ua/>

26. ISO 21500:2012 Інформація. // Електронний ресурс. Режим доступу: <https://www.iso.org/obp/ui/#iso:std:iso:21500:ed-1:v1:en>

27. ISO 10006:2017 Інформація. // Електронний ресурс. Режим доступу: <https://www.iso.org/obp/ui/en/#iso:std:iso:10006:ed-3:v1:en>

28. Інформація про Модель якості продукту (Product Quality Model) Оновлена версія. // Електронний ресурс. Режим доступу: <https://quality.arc42.org/articles/iso-25010-update-2023>

29. Quality Assurance VS Quality Control. Режим доступу: <https://asq.org/quality-resources/quality-assurance-vs-control?srsId=AfmBOoo5lsmWcEhsZAsNWzm6axl0J75KkfMCSdmtM3cPyqZ8dHMJiQOe>

30. Definition of Done. Режим доступу: <https://www.atlassian.com/agile/project-management/definition-of-done>

31. Cloud Functions for Firebase. (2023). Режим доступу: <https://developers.google.com/assistant/console/firebase-services>

32. GOOGLE. Firebase. Documentação do FirebaseRealtimeDatabase. (2024). Режим доступу: <https://firebase.google.com/docs/database>.

33. Moroney, L. (2017). The Firebase Realtime Database. In: The Definitive Guide to Firebase. Apress, Berkeley, CA. Режим доступу: https://doi.org/10.1007/978-1-4842-2943-9_3

34. What Is React Native? Beginner's Guide + FAQ. (2025). Режим доступу: <https://www.coursera.org/articles/what-is-react-native>

35. React Native Firebase. (2025). Режим доступу: <https://rnfirebase.io>

36. Bernardino, S.; De Jesus Rodrigues De Barros, L. (2025). Firebase and its usage in android and ios applications. Revista Interface Tecnológica, Taquaritinga, SP, 21(1), 279–287, 2025. <https://doi.org/10.31510/infa.v21i1.1978>.

37. ReleaseNotes Firebase. (2025). Режим доступу:
<https://firebase.google.com/support/releases>
38. Firebase Studio. (2025). Режим доступу:
<https://firebase.google.com/docs/studio>
39. Ralenko, V, & Stoiev, Ye. (2025). Using Firebase Studio in developing React applications. Scientific and Practical Journal «Materials of Scientific Conferences of the Petro Mohyla Black Sea National University». Режим доступу:
<https://doi.org/10.34132/mspc2025.01.14.14>
40. Ziuziun V., Petrenko N. AI-Enhanced System Design for Agile Sprint Management and Velocity Prediction. 2025 IEEE 5th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 2025, pp. 1-6. <https://doi.org/10.1109/SIST61657.2025.11139157>
41. Ziuziun V., Bredikhin D. Rozrobka informacijnoi' struktury projektu stvorennja veb-platfomy dlja pidvyshhennja ekologichnoi' kul'tury naselennja [Development of the information structure of the project for creating a web platform to improve the ecological culture of the population]. Collection of Scientific Papers of Admiral Makarov National University of Shipbuilding, 2025, 1(499), 126-136. [https://doi.org/10.15589/znp2025.1\(499\).18](https://doi.org/10.15589/znp2025.1(499).18)
42. Ziuziun, V., & Petrenko, N. Formation of conceptual and logical databases for the project of creating a web platform for automated verification and evaluation of written tests. XXI International scientific and practical conference «Project management in the development of society», 2024, Kyiv, KNUBA, 32-37. URL:https://library.knuba.edu.ua/books/zbirniki/Upravlinnya_proektami_u_rozvitku_suspilstvaa_XXI.pdf#page=32
43. Ziuziun V., Petrenko N. Formation of conceptual and logical databases for the project of creating a web platform for automated verification and evaluation of written tests. XXI Міжнародна науково-практична конференція «Управління проєктами у розвитку суспільства». Тема: «Управління проєктами післявоєнного відновлення України», м. Київ, 24 травня 2024 року. – Київ, КНУБА. – 2024. – С. 32-37. Режим доступу:

https://library.knuba.edu.ua/books/zbirniki/Upravlinnya_proektami_u_rozvitku_suspilstvaa_XXI.pdf#page=32

44. Rahmadani, F., & Susetyo, Y. A. (2025). The UI/UX Design of a Mobile Based Catering Application Using the Activity Centered Design Method. *International Journal Software Engineering and Computer Science (IJSECS)*, 5(1), 29-39. <https://doi.org/10.35870/ijsecs.v5i1.3396>

45. Figma: The Collaborative Interface Design Tool. (2025). Режим доступу: <https://www.figma.com>

46. Green, T., & Brandon, K. (2024). *Microinteractions in Figma. UX Design with Figma. Design Thinking.* Apress, Berkeley, CA. Режим доступу: https://doi.org/10.1007/979-8-8688-0324-6_8

47. Ziuziun, V., Demchuk, G. (2025). Informational architecture and UI/UX design of a mobile application for coworking spaces, 102-110. Section: Information Technology, Cyber Security and Computer Engineering. <https://doi.org/10.70286/eoss-24.11.2025.002>

48. Інститут проектного менеджменту (США) // Режим доступу: <https://www.pmi.org/>

49. Розробка діаграм. Режим доступу: <https://app.diagrams.net/>

50. Аналіз і створення функціональних та нефункціональних вимог додатку. Конструювання програмного забезпечення. Режим доступу: https://raw.githack.com/Aranaur/aranaur.rbind.io/main/lectures/soft_construction/slides/2023/.01.html#/title-slide

51. Kolomiiets A., Miroshnychenko I., Ziuziun V., Datsenko N., Kmytiuk T. Development of Project Management Models for Information Systems to Improve Website SEO Metrics. XI International Scientific Conference "Information Technology and Implementation" (IT&I 2024). CEUR Workshop Proceedings, Vol-3909, 334-345. Режим доступу: https://ceur-ws.org/Vol-3909/Paper_27.pdf

The image shows a mobile application interface for editing a user profile. It features several sections, each with a title and a corresponding input field or dropdown menu. At the bottom, there is a prominent teal button labeled 'Save changes'.

- Working Calendar:** A text input field containing 'Google Calendar'.
- Booking History:** A text input field containing 'WeWork, Prague, 20/01/2023' and a teal icon with a gear and a plus sign.
- Registration Date:** A dropdown menu showing '23/05/2020' with a downward arrow.
- Telephone number:** A text input field containing '+38036372920'.
- Language:** A dropdown menu showing 'English' with a downward arrow.
- AI Recommendations:** A dropdown menu showing 'Based on your favorites' with a downward arrow.
- Payment method:** A text input field containing 'Apple Pay'.

Save changes

Рис. А.1. Продовження вікна для редагування профілю користувача

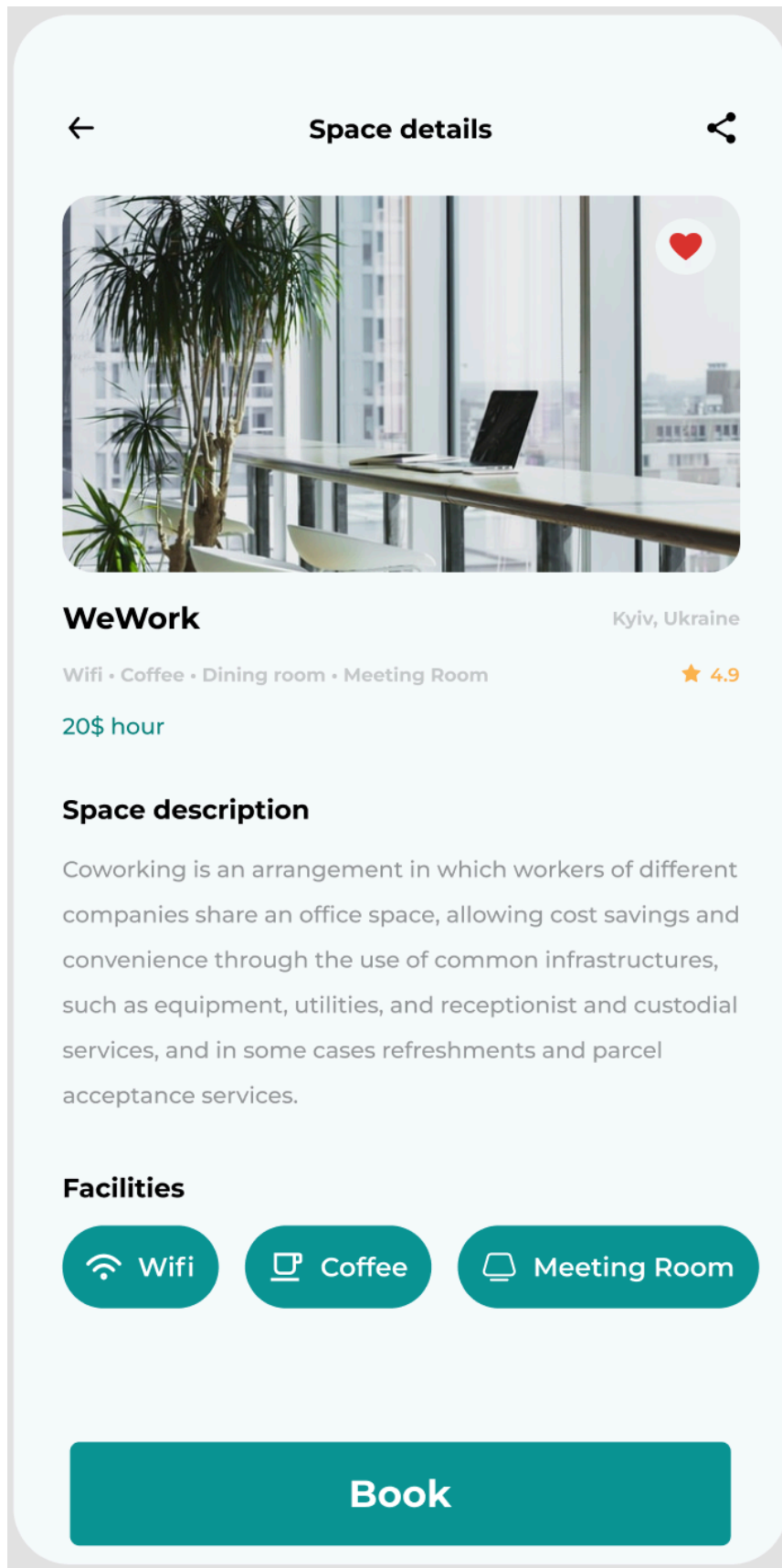


Рис. А.2. Вікно для детального опису коворкінгу який обирає користувач

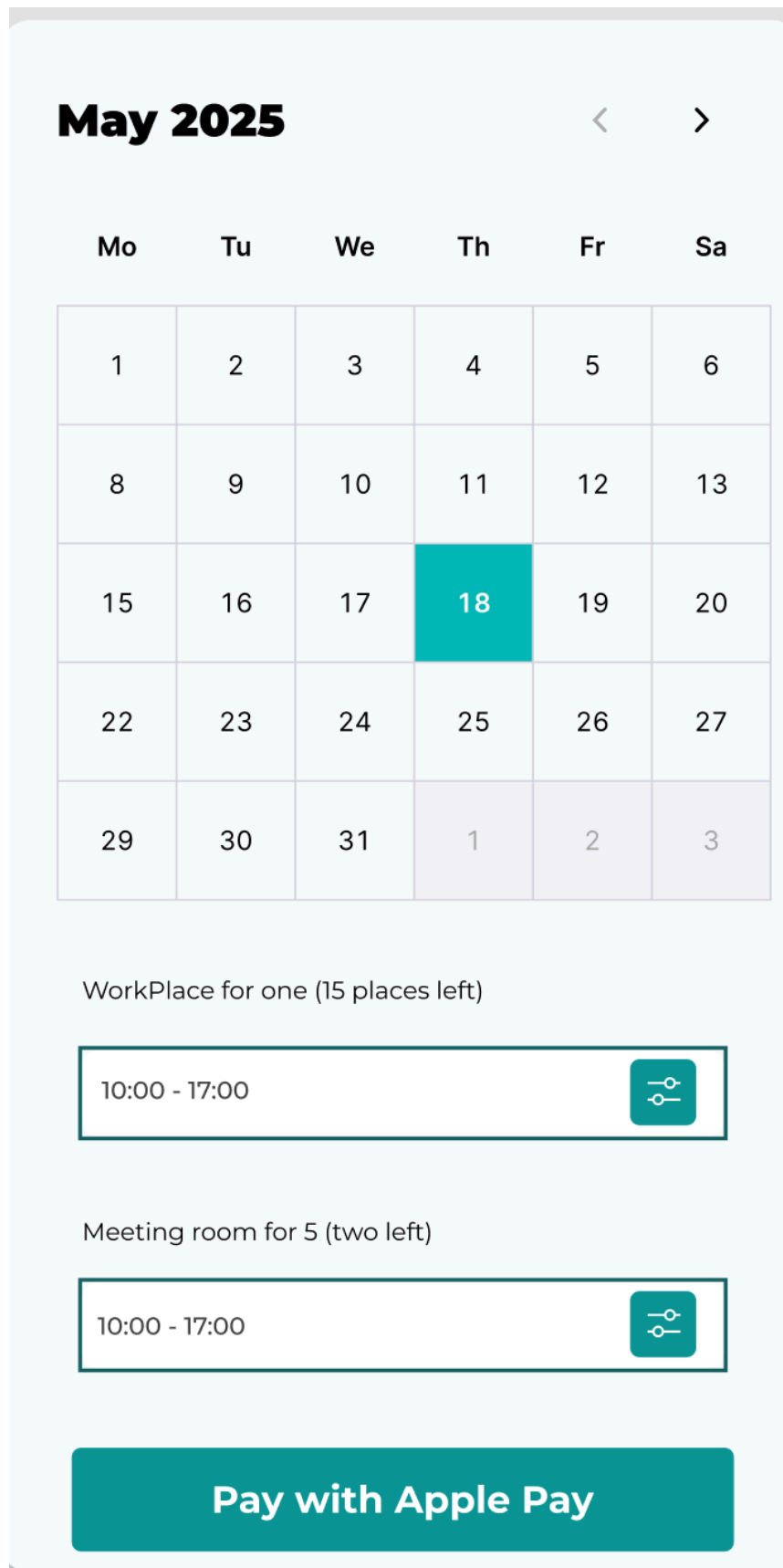


Рис. А.3. Вікно, яке з'являється після кнопки бронювання з календарем і доступними годинами в коворкінгу

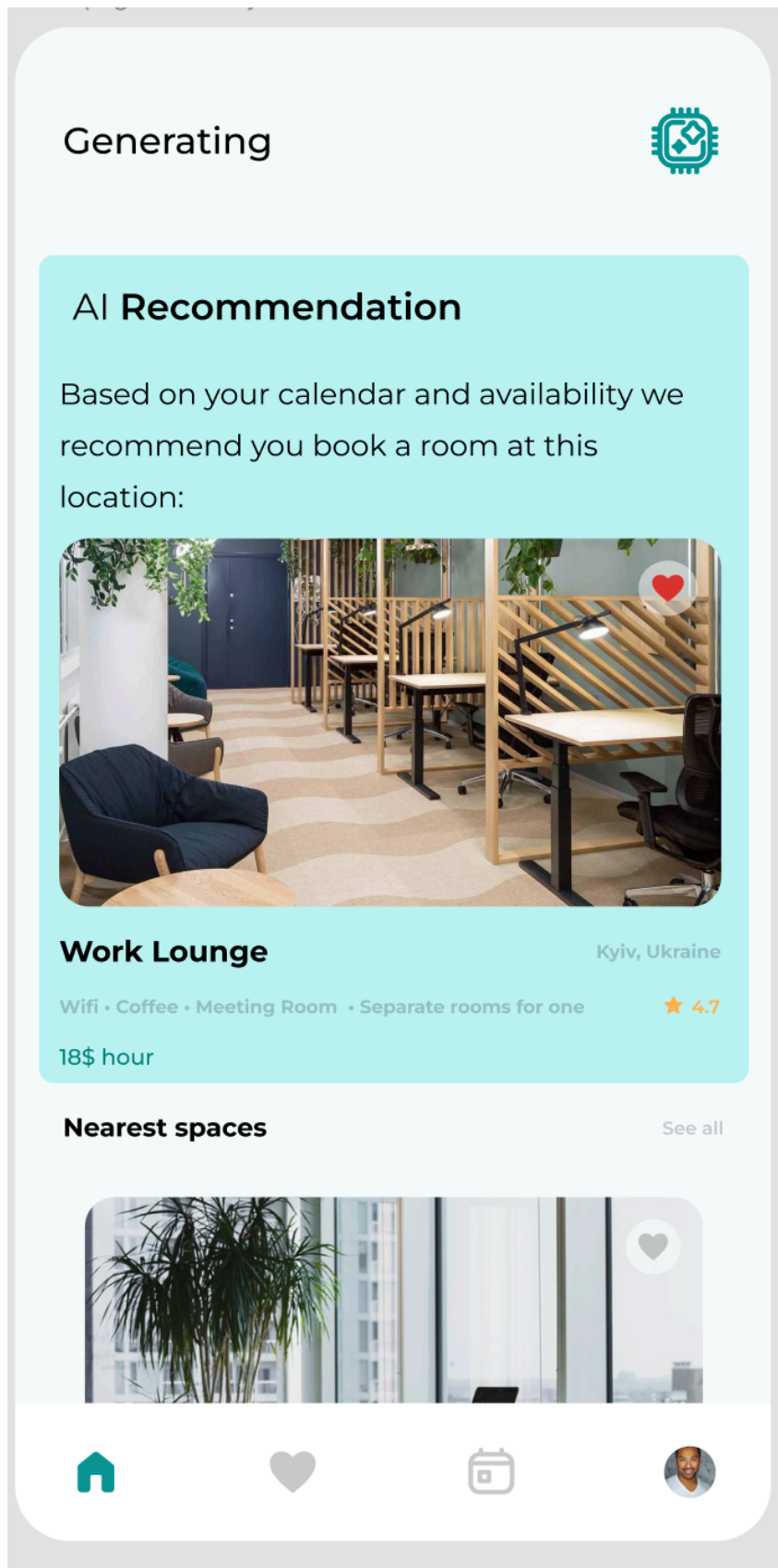


Рис. А.4. Вікно, яке з'являється коли включена функція персоналізованих рекомендацій від штучного інтелекту