

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки» _____,

освітньо-професійної програми «Інформаційні технології проектування» _____

на тему: Мобільний додаток автоматизації презентування нерухомості з використанням ChatGPT

Здобувача групи ІТ.м-22 Шелехова Дениса Володимировича

(шифр групи)

(прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис)

Денис ШЕЛЕХОВ

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник

к.т.н., доц. Вікторія АНТИПЕНКО

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

_____ (підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Шелехову Денису Володимировичу

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи Мобільний додаток автоматизації презентування нерухомості з використанням ChatGPT

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом кваліфікаційної роботи «__» __ грудня __ 2023 р.

3 Вхідні дані до кваліфікаційної роботи завдання ріелтора, презентація нерухомості _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі, методи дослідження, проектування мобільного додатку автоматизації презентування нерухомості з використанням ChatGPT, розробка мобільного додатку, тестування мобільного додатку.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації) актуальність, мета та задачі дослідження, огляд існуючих аналогів, результати проведеного аналізу аналогів, функціональні вимоги, засоби реалізації, структурно-функціональне моделювання, моделювання варіантів використання, фізична модель бази даних, архітектура мобільного додатку, програмна реалізація мобільного додатку, демонстрація роботи програмного продукту, тестування, апробація, висновок.

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Підготовка специфікації	20.09.2023	
2	Проектування мобільного додатку	20.11.2023	
3	Розробка мобільного додатку автоматизації презентування нерухомості з використанням ChatGPT	16.11.2023	
4	Тестування мобільного додатку	01.12.2023	

Магістрант _____ Денис ШЕЛЕХОВ

Керівник роботи _____ к.т.н., доц. Вікторія АНТИПЕНКО

АНОТАЦІЯ

Темою кваліфікаційної роботи магістра є «Мобільний додаток автоматизації презентування нерухомості з використанням ChatGPT».

Пояснювальна записка складається зі вступу, 4 розділів, висновку, списку використаних джерел із 55 найменувань, 2 додатків. Загальний обсяг роботи – 97 сторінок, у тому числі 45 сторінок основного тексту, 7 сторінок списку використаних джерел, 45 сторінок додатків.

Актуальність роботи полягає в зменшенні часу ріелторам на створення презентацій нерухомості завдяки використанню запропонованого мобільного додатку з інтеграцією штучного інтелекту, а саме ChatGPT. Вибір напрямлення на розробку даного програмного продукту було обрано через розповсюдженість застосування та портативність смартфонів.

Мета роботи: розробка мобільного додатку для автоматизації процесу створення презентацій нерухомості з використанням ChatGPT. Для цього проведений аналіз останніх публікацій, близьких до теми, та аналогічних програмних продуктів. Визначено задачі, функціональні вимоги, методи дослідження та засоби реалізації. Здійснено моделювання роботи даного мобільного додатку та побудовано його архітектурута. Розроблено запропонований програмний продукт. Проведено тестування роботи даного мобільного додатку методом «чорного ящика».

Ключові слова: **МОБІЛЬНИЙ ДОДАТОК, АВТОМАТИЗАЦІЯ, ПРЕЗЕНТАЦІЯ, РІЕЛТОР, ANDROID STUDIO, JAVA, SQLITE, ШТУЧНИЙ ІНТЕЛЕКТ, CHATGPT.**

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Огляд останніх досліджень і публікацій	8
1.2 Аналіз програмних продуктів-аналогів	9
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	15
2.1 Мета та задачі дослідження.....	15
2.2 Методи дослідження.....	16
2.3 Вибір технологій.....	17
3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ	19
3.1 Діаграми нотації IDEF0	19
3.2 Діаграма Use Case	23
3.3 Проєктування моделі бази даних	25
4 МОБІЛЬНИЙ ДОДАТОК АВТОМАТИЗАЦІЇ ПРЕЗЕНТУВАННЯ НЕРУХОМОСТІ З ВИКОРИСТАННЯМ СНАТGPT	27
4.1 Архітектура мобільного додатку	27
4.2 Реалізація мобільного додатку.....	28
4.3 Демонстрація роботи мобільного додатку	34
4.4 Тестування мобільного додатку.....	42
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46
ДОДАТОК А.....	53
ДОДАТОК Б	66

ВСТУП

Сфера нерухомості є досить динамічною сьогодні. А ефективна комунікація є запорукою успіху. Тому в сучасному світі саме презентації нерухомості є ключовим елементом при взаємодії ріелтора з потенційними клієнтами. За їхньої допомоги краще демонструвати та роз'яснювати останнім унікальні переваги придбання тієї чи іншої власності. Звісно традиційні методи презентування виконують своє призначення. Однак поява технологій штучного інтелекту відкриває нові можливості для інноваційних підходів. Наприклад, застосування останніх дозволяє зменшити витрату часу на створення самих презентацій. Водночас забезпечуючи їх якісним контентом. У даній роботі розглядається розробка мобільного додатку, готового автоматизувати процес створення презентації нерухомості, використовуючи можливості ChatGPT для генерації осмисленого тексту.

Актуальність. У сучасній сфері нерухомості час має важливе значення. А захоплюючі презентації можуть укласти або розірвати угоду. Тому потреба в інноваціях є першорядною. Професійні ріелтори за день можуть взаємодіяти з великою кількістю клієнтів і не мати часу на розробку персоналізованих презентацій для кожного об'єкту нерухомості. Ураховуючи все це, вони дійсно можуть отримати значну користь, використовуючи інструмент, який автоматизує процес створення презентацій. Запропонований мобільний додаток відповідає цій потребі. Тому його розробка є актуальною. Крім того дане нове рішення інтегрує ChatGPT для створення персоналізованих презентацій нерухомості.

Тема. Мобільний додаток автоматизації презентування нерухомості з використанням ChatGPT.

Тому **мета даної роботи** – розробка мобільного додатку для автоматизації процесу створення презентацій нерухомості з використанням ChatGPT. Включення останнього забезпечує елемент розмови, надаючи користувачам інтерактивний та персоналізований

досвід. Застосування даної розробки дозволить зменшити витрату часу на створення самих презентацій і водночас забезпечувати їх якісним контентом.

Для досягнення мети цього проєкту необхідно виконати такі задачі:

- дослідити предметну область;
- провести аналіз останніх публікацій;
- дослідити існуючі програмні продукти з аналогічним функціоналом;
- розробити комплексний дизайн і структуру, а також визначити функціональні вимоги до представленого мобільного додатку;
- здійснити моделювання роботи даного програмного продукту;
- розробити мобільний додаток автоматизації презентування нерухомості та додати інтеграцію з ChatGPT і його функціональними можливостями для генерації необхідного тексту;
- провести тестування отриманого програмного продукту.

Об’єкт дослідження. Процес автоматизації презентування нерухомості з використанням ChatGPT.

Предмет дослідження. Мобільний додаток автоматизації презентування нерухомості з використанням ChatGPT.

Практичне значення. Запропонований мобільний додаток є перспективним для використання фахівцями сфери нерухомості. Це ефективний, зручний та привабливий інструмент. Він автоматизує процес створення презентацій нерухомості. Також використовує потужність ChatGPT для забезпечення потенційних клієнтів персоналізованим досвідом. Застосування даного мобільного додатка дозволить зменшити витрату часу на створення самих презентацій. Водночас забезпечує останні якісним контентом.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

В останні роки спостерігається помітна тенденція інтеграції передових технологій у різні галузі. І сектор нерухомості не є винятком.

На даний момент смартфон є пристроєм із найбільшою кількістю користувачів по всьому світу [1]. А, отже, й актуальність застосування мобільних додатків незаперечна [2-4]. Далі розглянемо специфіку їхнього використання у певній області.

У сфері нерухомості мобільні додатки застосовують здебільшого для проведення менеджменту агентами нерухомості або для пошуку інвесторами вигідних об'єктів інвестицій [5-12]. Є й більш спеціалізовані, наприклад, геоінформаційні системи (GIS). Це програмні інструменти, які використовуються для зберігання, візуалізації, оцінки та інтерпретації географічних даних. Можливості статистичного аналізу GIS підвищують ефективність і точність оцінки землі та нерухомості [13,14].

Далі були опрацьовані останні дослідження та публікації, які присвячені аналізу ChatGPT [15] як потужної мовної моделі для генерації тексту. А також розглянуті можливості існуючих інструментів для швидкого створення презентацій від компанії OpenAI [16].

У публікаціях [17,18] проводиться аналіз відкритого штучного інтелекту (ШІ) ChatGPT. А також його функцій і сфер використання. Зміни парадигми в доступі до інформації, які спричинив ChatGPT, можуть бути корисними для багатьох галузей. Наприклад, таких, як освіта, наука, журналістика, масові комунікації, інформаційні технології (ІТ), роздрібна торгівля та ін.. Цей чат-бот може генерувати тексти майже для будь-якого напрямку.

Оскільки сучасна тенденція – це впровадження моделей штучного інтелекту до різних інформаційних систем, то мобільні додатки також отримали поштовх у цій

сфері [19,20]. Однією з найбільших галузей їхнього використання зі ШІ є медицина [21-25].

У роботах [26,27] досліджуються інструменти, в основі яких лежить штучний інтелект. За їх допомогою можна створювати презентації. Однак жодних із них не надає можливості впливу на кінцевий згенерований результат.

Проводячи пошук публікацій по даній темі, можна помітити, що більшість з них досліджує можливість залучення штучного інтелекту в основному для автоматизації процесів освітнього напрямку. Якщо розглядати використання ШІ в сфері нерухомості, то, в основному, це аналіз великої кількості даних для прогнозування цін, проведення маркетингу, тощо. Тобто для застосування великими компаніями. Тому можна зробити висновок, що наразі існує необхідність у створенні для ріелторів спеціалізованого мобільного додатку автоматизації презентування нерухомості їхнім клієнтам із використанням ChatGPT.

1.2 Аналіз програмних продуктів-аналогів

Після нещодавньої популярності штучного інтелекту, багато компаній намагаються додати певні інструменти взаємодії з ШІ [28] до своїх додатків, або створювати нові, де він лежить в основі. Тому далі розглянуто такі програмні засоби, як Canva [29] та Gamma App [30]. Це два існуючі продукти-аналоги.

Canva – це платформа у вигляді мобільного або web-додатку для розробки графічного дизайну. І створення презентацій – це тільки одна з багатьох її можливостей. Далі розглянемо функціонал Canva. У першу чергу, це повноцінний редактор презентацій із можливістю вибору шаблону дизайну та додаванням тексту, медіа матеріалів тощо (рис. 1.1). Взаємодія зі штучним інтелектом відбувається за допомогою багатьох інструментів. Наприклад, це генерація тексту за ключовими словами або

зображення та музики за описом (рис. 1.2). Однак розроблювати всю презентацію потрібно споживачу самостійно. Надані на платформі інструменти можуть тільки допомогти, якщо вміти ними користуватися.

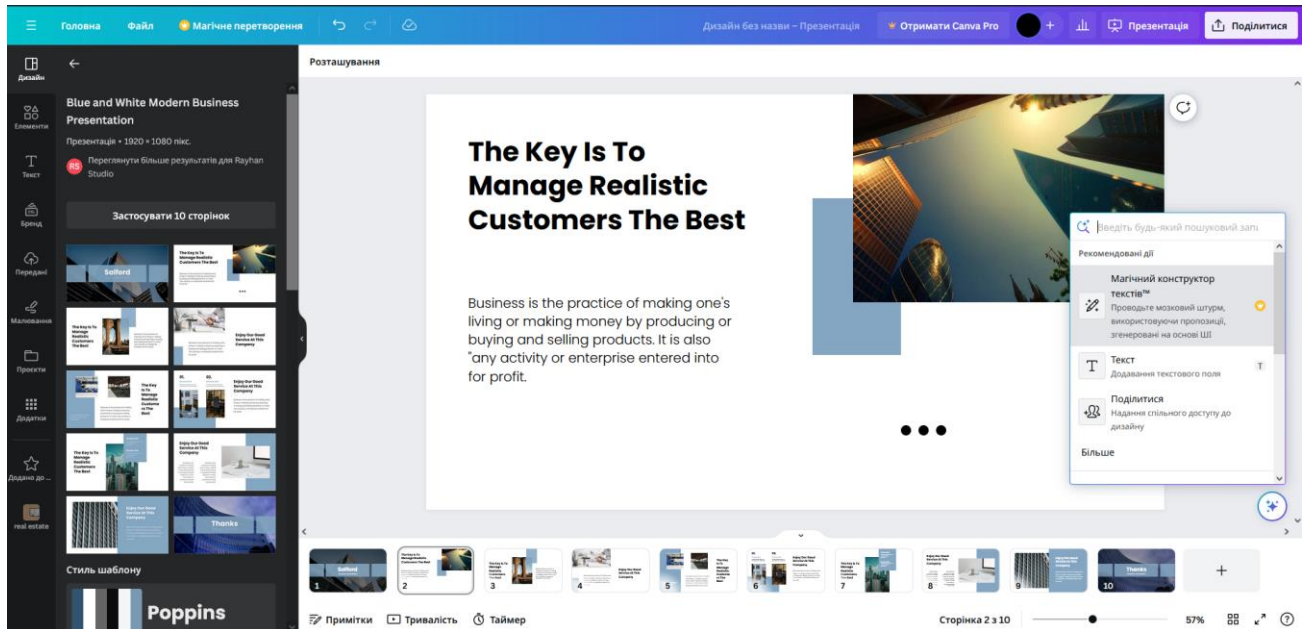


Рисунок 1.1 – Редактор презентації Canva

Джерело: [29]

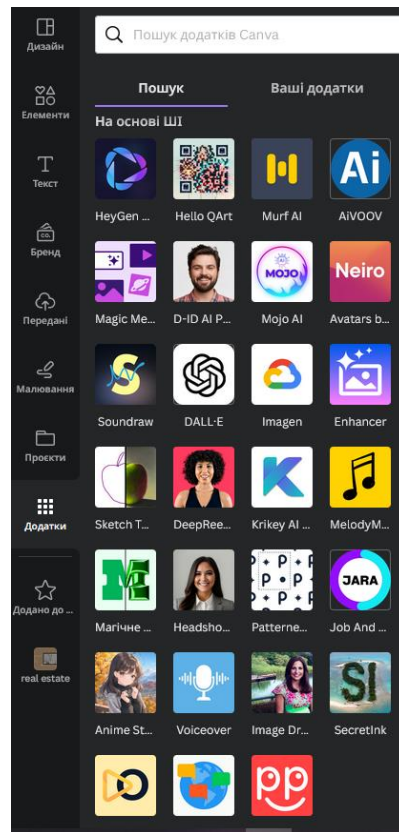


Рисунок 1.2 – Список додатків на основі штучного інтелекту Canva
Джерело: [29]

Gamma App – це сервіс також для генерації презентацій. Крім того можна створювати документи та дизайн web-сторінок (рис. 1.3). Це робиться за допомогою штучного інтелекту Gamma. ШІ генерує презентацію на обрану користувачем тематику. Доступною є можливість редагування кінцевого результату як власноруч, так і за допомогою команд до ШІ (рис. 1.4). Також сервіс дозволяє генерувати зображення (рис. 1.5). Gamma App доступний тільки у вигляді web-додатку.

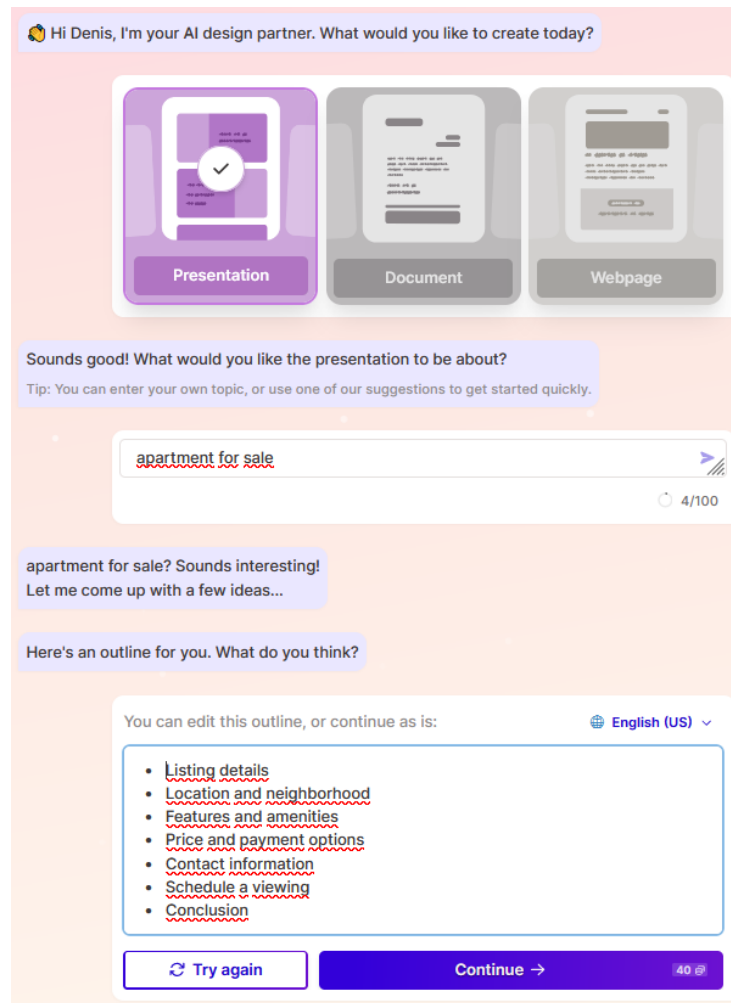


Рисунок 1.3 – Процес створення презентації в Gamma App

Джерело: [30]

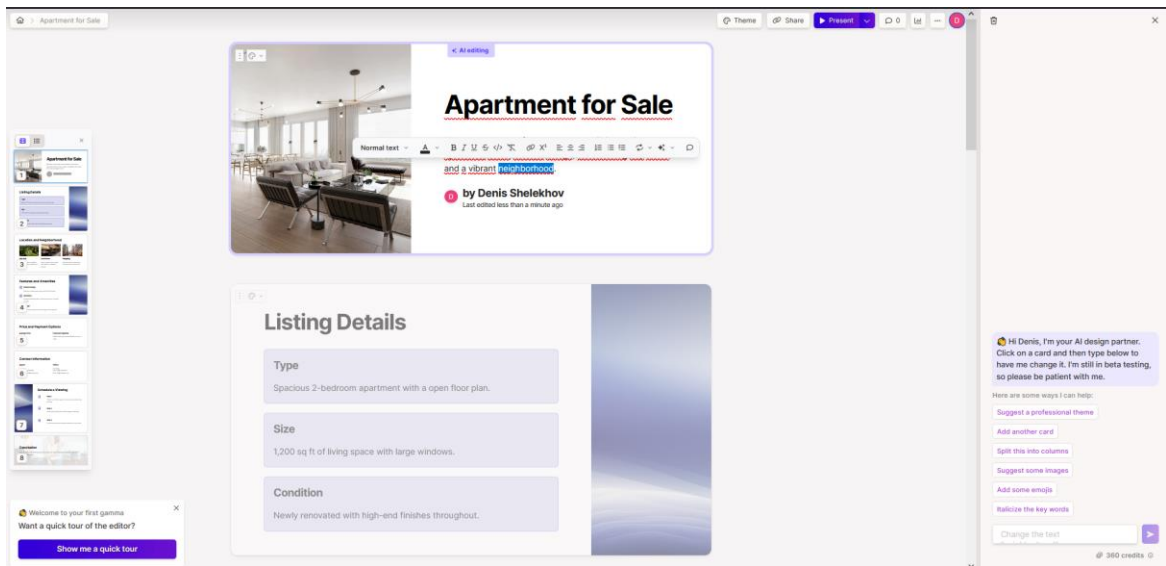


Рисунок 1.4 – Редагування презентації власноруч або через чат із штучним інтелектом
Джерело: [30]

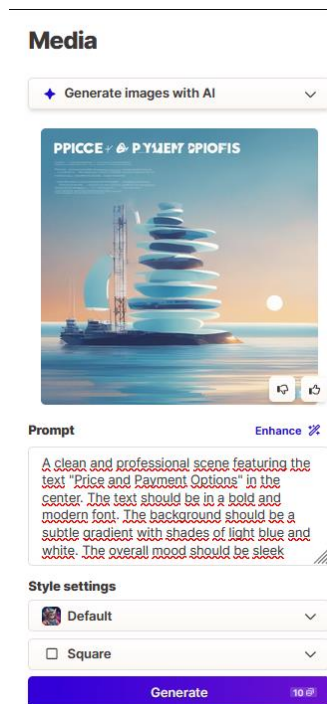


Рисунок 1.5 – Генерація зображень у Gamma App
Джерело: [30]

Результати проведеного порівняльного аналізу двох вказаних вище програмних продуктів представлені в таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця існуючих продуктів-аналогів

Характеристика/Програмний продукт	«Canva»	«Gamma»
Генерація презентації за допомогою ШІ	-	+
Вибір шаблону презентації	+	-
Зміна стилю презентації	-	+
Можливість редагування презентацій	+	+
Можливість генерації тексту	+	+
Можливість генерації зображень	-	+
Збереження презентації	+	+
Наявність мобільної версії	+	-

Джерело: побудовано автором

Проаналізувавши дані з таблиці 1.1, можна виділити переваги кожного продукту, які треба врахувати у власній розробці, та недоліки, які варто подолати.

Кожного дня дійсно розроблюються нові рішення з використанням штучного інтелекту в багатьох галузях. Проте це все ще досить молодий напрямок. Тому він ще не охоплює всі сфери, де такі рішення необхідні. А, отже, дана робота є доречною та нагальною сьогодні.

Перевагою цього дослідження є охоплення сфери продажу нерухомості, використання й фіксація саме на розробці презентацій на мобільному пристрої. Для генерації якісного тексту буде використовуватися ChatGPT, як найбільш потужний штучний інтелект для обробки природної мови. Також для досягнення найменшого ступеню входу, а також для зменшення затрат часу на створення презентації, інструментарій буде обмежений до мінімально необхідного рівня.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Метою роботи є розробка мобільного додатку для автоматизації процесу створення презентацій нерухомості з використанням ChatGPT. Він орієнтований на ріелторів. Є професіонали, які займаються продажем великої кількості нерухомості та не мають часу, або бажання його витратити, на власноручне створення презентацій. Застосовуючи даний мобільний додаток, користувач матиме можливість значно пришвидшити цей процес за допомогою шаблонного заповнення та генерації осмисленого та доцільного тексту за допомогою взаємодії через API [31] з ChatGPT.

Запропонований програмний продукт повинен мати наступний функціонал:

- вибір необхідного шаблону;
- заповнення короткої форми для подальшої генерації презентації;
- перегляд згенерованої презентації;
- редагування тексту;
- додавання зображень;
- регенерація тексту за допомогою ChatGPT;
- вибір стилю шаблону;
- завантаження презентації в PDF [32] форматі.

Для досягнення мети даного проєкту необхідно виконати наступні задачі:

- дослідити предметну область застосування мобільних додатків для створення презентацій, рівень використання штучного інтелекту та сферу нерухомості;
- провести аналіз останніх публікацій на тему штучного інтелекту та сфер його використання;
- дослідити існуючі програмні продукти з створення презентацій із застосуванням штучного інтелекту;

- розробити комплексний дизайн і структуру, а також визначити функціональні вимоги до представленого мобільного додатку з автоматизації створення презентацій з нерухомості з використанням ChatGPT;
- здійснити моделювання роботи мобільного додатку з автоматизації створення презентацій з нерухомості з використанням ChatGPT;
- розробити мобільний додаток автоматизації презентування нерухомості та додати інтеграцію з ChatGPT і його функціональними можливостями для генерації необхідного тексту;
- провести тестування отриманого мобільного додатку з автоматизації створення презентацій з нерухомості з використанням ChatGPT.

Планування виконання робіт даного проєкту представлено у додатку А.

2.2 Методи дослідження

При розробці запропонованого мобільного додатку, спрямованого на досягнення конкретних цілей, вирішальним є системний підхід із залученням різноманітних методів дослідження. Використання останніх є досить важливим для даного проєкту. Нижче описано дослідницькі методи, які було застосовано:

1. Теоретичний метод [33]:

- проведення детального аналізу предметної області використання даного мобільного додатку та існуючих продуктів аналогів для розуміння потреб споживачів, галузеві тенденції та конкретні вимоги;
- дослідження особливостей та можливостей в рамках розробки мобільного додатку, наголошуючи на таких міркуваннях, як досвід користувача (UX) [34] та дизайн інтерфейсу користувача (UI) [35];

- аналіз переваг та недоліки різних технологій для розробки мобільних додатків, зосереджуючись на таких аспектах, як ефективність та швидкодія.

2. Метод моделювання [36]:

- розробка діаграм IDEF0 [37] для ілюстрації загальної структури мобільного додатку та процесу автоматизації презентування нерухомості;

- створення діаграми варіантів використання (UseCase) [38] для візуального представлення взаємодії між користувачами та мобільним додатком, окреслюючи різні варіанти застосування та сценарії.

3. Системно-функціональний метод [39]:

- здійснення дослідження та опису компонентів, залежностей і взаємодії в роботі мобільного додатку;

- використання отриманих знань для розробки архітектури мобільного додатку, забезпечуючи повну інтеграцію особливостей і функціональних можливостей.

2.3 Вибір технологій

Для реальної реалізації запропонованого мобільного додатку було обрано такі технології:

1. Android Studio [40]. Використовується як основне інтегроване середовище розробки (IDE [41]) програм для операційної системи та платформи мобільних телефонів та планшетних комп'ютерів Android [42].

2. Java [43]. Мова програмування, яка застосовується для кодування логіки та функціональності мобільного додатку, забезпечуючи сумісність із платформою Android.

3. SQLite [44]. Легкий вбудований реляційний базовий менеджер даних, який не вимагає окремого сервера. Використовується для зберігання баз даних на мобільних пристроях та вбудовується в багато програм для доступу до даних.

4. ChatGPT. Використовується для створення контекстуально доцільного тексту презентацій в запропонованому мобільному додатку.

Завдяки застосуванню вищезазначених дослідницьких методів і технологій процес розробки систематично управляється, гарантуючи, що мобільний додаток відповідає потребам споживачів, галузевим тенденціям і технологічним можливостям. Використання Android Studio, Java і ChatGPT підвищує ефективність і результативність процесу розробки. Це зрештою призводить до створення якісного та зручного мобільного додатку.

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ

Етап проєктування та моделювання при створенні даного мобільного додатку є незамінним, оскільки він закладає основу для успішного та ефективного процесу розробки. Він дозволяє концептуалізувати архітектуру програмного продукту, інтерфейс споживача та функціональні можливості до написання будь-якого коду. Завдяки комплексному моделюванню критичні аспекти, такі як взаємодія з користувачем, потік інформації та взаємодія компонентів мобільного додатку, ретельно плануються, забезпечуючи добре організований та інтуїтивно зрозумілий кінцевий продукт. Вкладаючи час у фазу проєктування та моделювання, потенційні проблеми та вимоги можна виявити та вирішити на ранній стадії. Це дозволить зменшити ймовірність переробки та покращити загальну якість і задоволеність користувачів мобільним додатком.

3.1 Діаграми нотації IDEF0

IDEF0 (Integration Definition for Function Modeling) – методологія функціонального моделювання, яка використовується в системній інженерії для зображення функцій, дій і взаємодій у складній системі. Вона розроблена Міжнародною радою з системної інженерії (INCOSE) [45]. IDEF0 є частиною сімейства мов моделювання IDEF (Integrated Definition) [46]. Дана методологія є доцільною для реалізації цього проєкту.

Основна мета IDEF0 – забезпечити чітке та структуроване представлення функцій, які виконує система, а також взаємозв'язків та залежностей між ними. Ця методологія особливо цінна на ранніх стадіях розробки. Оскільки при концептуалізації системи,

розуміння її функціональності має вирішальне значення. Це важливо для ефективного проектування та прийняття рішень.

У IDEF0 стрілки відіграють вирішальну роль у представленні потоку вхідних і вихідних даних між функціями. Кожна з них має певне значення та напрямок. Вони використовуються для ілюстрації зв'язків і залежностей між різними функціями в системі. Функції та значення, пов'язані зі стрілками на діаграмах IDEF0 для даного проекту, такі:

- стрілки зліва – представляють вхідні дані для функції, демонструючи, що вона отримує від джерела або функції з лівого боку;
- стрілки вправо – представляють вихідні дані функції, демонструючи, її результати роботи, які передаються цільовому або функції з правого боку;
- стрілки знизу – позначають механізми, які впливають на функцію;
- стрілки зверху – позначають елементи управління, які обмежують функцію.

Процес «Автоматизація презентування нерухомості з використанням ChatGPT» містить такі дані:

- вхідні дані – запит на створення презентації, фото та зображення.
- вихідні дані – pdf файл з презентацією.
- управління – список шаблонів, список питань до шаблону, правила складання запиту до ChatGPT, алгоритм генерації презентації, список стилей презентації, права доступу до сховища даних смартфона, алгоритм заповнення pdf файлу.
- механізми – користувач, смартфон, база даних, мобільний додаток, ChatGPT.

На рисунку 3.1 продемонстрована діаграма IDEF0 процесу автоматизованого створення презентацій нерухомості з використанням ChatGPT, який забезпечує розроблений мобільний додаток.



Рисунок 3.1 – Контекстна діаграма нотації IDEF0

Джерело: побудовано автором

Після визначення основного процесу необхідно провести декомпозицію для кращого його розуміння. Було виділено наступні п'ять процесів:

- вибір шаблону презентації;
- заповнення форми питань до шаблону;
- генерація презентації;
- редагування презентації;
- збереження презентації на смартфоні.

Далі була розроблена діаграма першого рівня. Для представлення інформації по ній було створено таблицю входів та виходів даних (табл. 3.1).

Таблиця 3.1 – Таблиця входів та виходів даних діаграми першого рівня

Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Вибір шаблону презентації	Запит на створення нової презентації	Список шаблонів	Користувач	Обраний шаблон
Заповнення форми питань до шаблону	Обраний шаблон	Список питань до шаблону	Користувач	Заповнена форма
	Фото та зображення		Смартфон	
		Права доступу до сховища даних смартфона	База даних	
Генерація презентації	Заповнена форма	Правила складання запиту до ChatGPT	ChatGPT	Згенерована презентація
		Алгоритм генерації презентації	Мобільний додаток	
			База даних	
Редагування презентації	Згенерована презентація	Права доступу до сховища даних смартфона	Користувач	Відредагована презентація
	Фото та зображення	Список стилей презентації	Смартфон	
			ChatGPT	
Збереження презентації на смартфоні	Відредагована презентація	Права доступу до сховища даних смартфона	Користувач	Pdf файл з презентацією
	Згенерована презентація	Алгоритм заповнення pdf файлу	Смартфон	

Джерело: побудовано автором

На рисунку 3.2 представлена діаграма першого рівня.

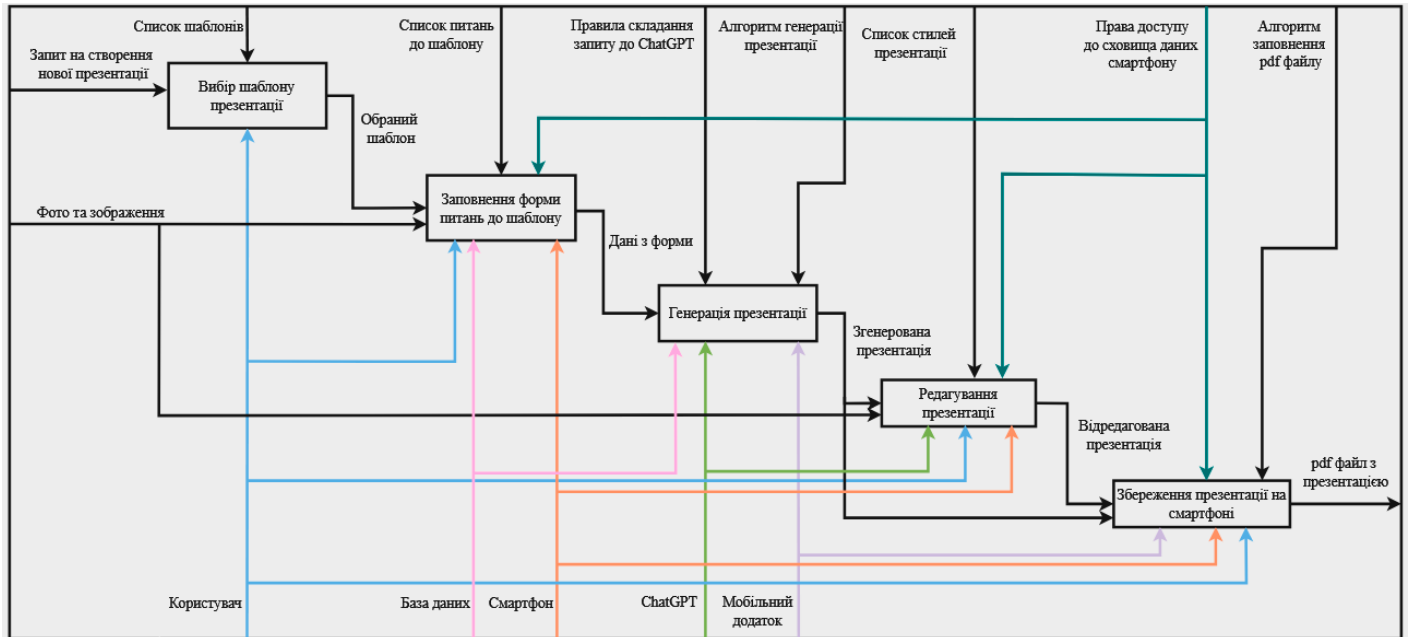


Рисунок 3.2 – Діаграма першого рівня

Джерело: побудовано автором

3.2 Діаграма Use Case

Розробка діаграми варіантів використання (Use Case) також має важливе значення на ранніх етапах розробки системи. Оскільки вона забезпечує чітке та стисле уявлення про те, як клієнти взаємодіють із програмним продуктом. Ілюструючи різні варіанти використання та їхні зв'язки, ця діаграма допомагає зацікавленим сторонам, зокрема розробникам і дизайнерам, отримати цілісне розуміння функціональних можливостей системи з точки зору споживача. Дана діаграма служить потужним інструментом для аналізу вимог. Вона допомагає ідентифікувати та документувати потреби користувачів

і функції системи. Діаграма Use Case також допомагає візуалізувати потенційні сценарії, винятки та межі в роботі програмного продукту.

У розробленій діаграмі варіантів використання було визначено один артефакт – сховище даних смартфону та трьох акторів – користувач, база даних та ChatGPT. Нижче надано такі сценарії використання:

- вибір шаблону;
- заповнення форми (включає в себе відповідь на питання, можливе додавання зображень);
- генерація презентації;
- редагування презентації (можливості: зміна стилю шаблону, регенерація тексту, редагування тексту);
- збереження презентації на смартфон.

На основі визначених вище сценаріїв була розроблена Use Case діаграма, яка представлена на рисунку 3.3.

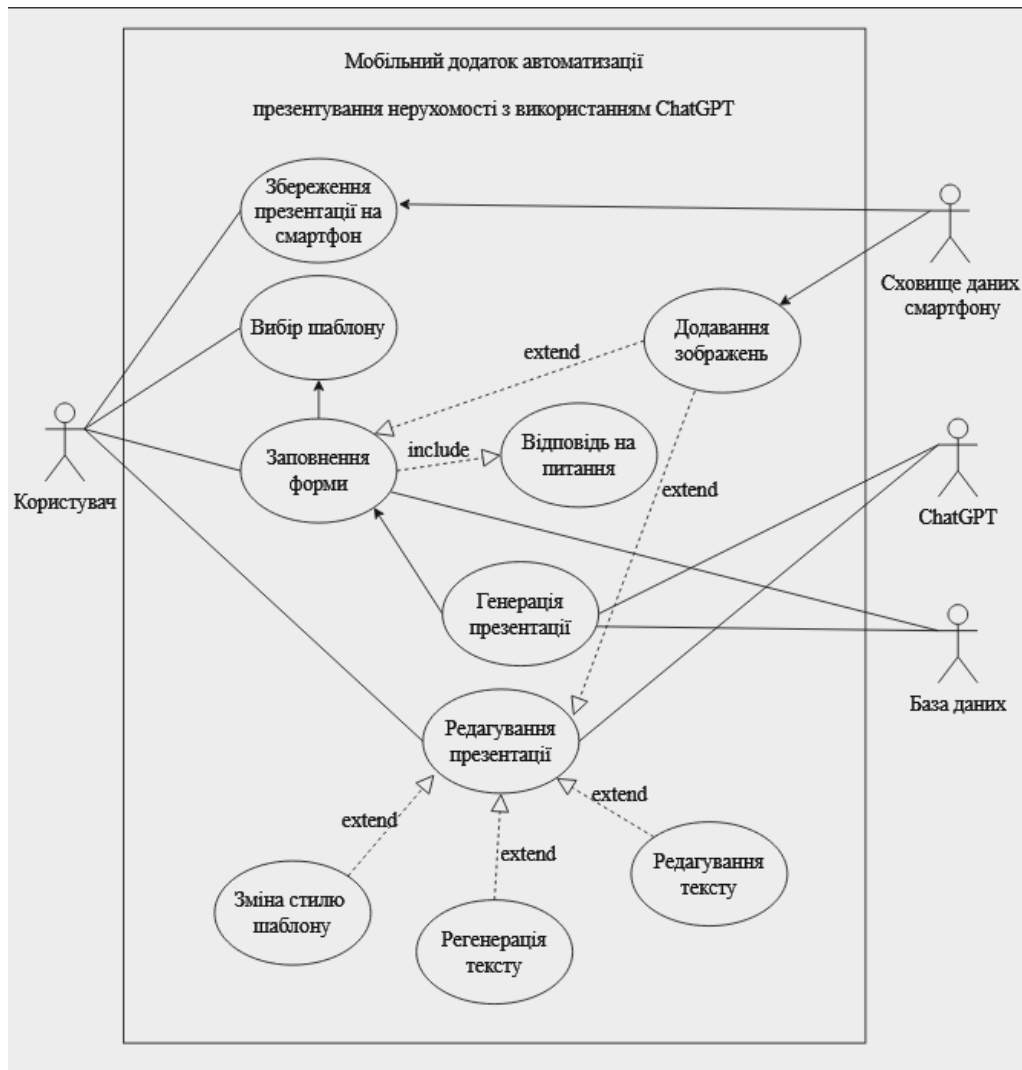


Рисунок 3.3 – Діаграма Use Case

Джерело: побудовано автором

3.3 Проєктування моделі бази даних

Модель бази даних – це структуроване представлення організації та зв'язків даних у системі бази даних (БД). Він охоплює загальний дизайн, схему та архітектуру БД, що визначає, як дані зберігаються, організовуються та доступні. Модель зазвичай включає таблиці, які представляють сутності, атрибути. Вони визначають властивості цих

сутностей, і зв'язки, які встановлюються між різними сутностями. Модель може використовувати різні концепції системи управління базами даних. Наприклад, нормалізацію, індексування та обмеження для підвищення продуктивності та зручності обслуговування. Загальна мета полягає в тому, щоб забезпечити надійну основу для управління даними, що відповідає конкретним потребам і вимогам додатків або систем, які використовують базу даних [47].

БД розроблюваного мобільного додатку має одну сутність – питання. Її атрибутами є шаблон презентації, блок питань, питання, текст (відповідь), кількість, необхідність тексту, необхідність кількості, необхідність прапорця, необхідність зображення.

На рисунку 3.4 представлена фізична модель бази даних.

Questions			
PK	id	INTEGER	NOT NULL
	template	VARCHAR	NOT NULL
	block	VARCHAR	NOT NULL
	title	VARCHAR	NOT NULL
	text	VARCHAR	NOT NULL
	checkbox	BOOLEAN	NOT NULL
	quantity	INTEGER	NOT NULL
	image	BLOB	NOTNULL
	is_text	BOOLEAN	NOT NULL
	is_checkbox	BOOLEAN	NOT NULL
	is_quantity	BOOLEAN	NOT NULL
	is_image	BOOLEAN	NOT NULL

Рисунок 3.4 – Фізична модель бази даних

Джерело: побудовано автором

4 МОБІЛЬНИЙ ДОДАТОК АВТОМАТИЗАЦІЇ ПРЕЗЕНТУВАННЯ НЕРУХОМОСТІ З ВИКОРИСТАННЯМ CHATGPT

4.1 Архітектура мобільного додатку

У процесі розробки даного мобільного додатку була створена його архітектурна модель (рис. 4.1). Вона складається з двох частин – xml файлів та Java класів. Перші – це візуальна частина мобільного додатку: елементи інтерфейсу та їх розміщення. Java класи – це логіка взаємодії елементів інтерфейсу з клієнтом, із іншими елементами інтерфейсу та зі сторонніми об'єктами, такими як база даних, ChatGPT та смартфон.

Архітектура мобільного додатку автоматизації презентування нерухомості з використанням ChatGPT зображена на рисунку 4.1

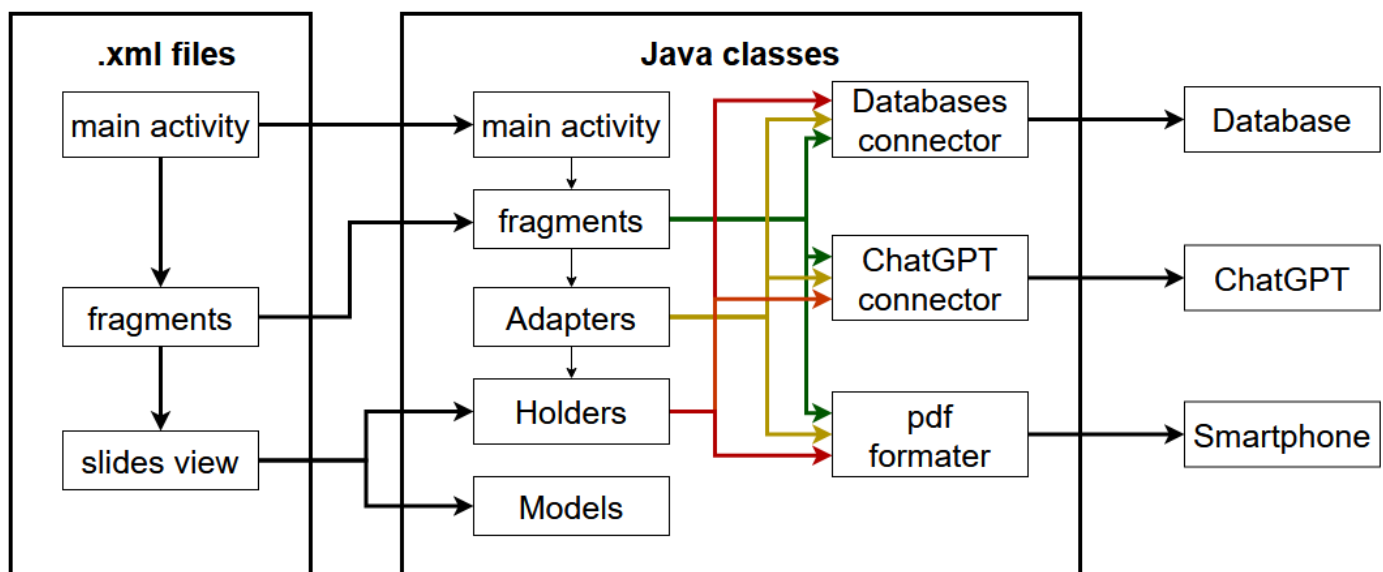


Рисунок 4.1 – Архітектура мобільного додатку

Джерело: побудовано автором

4.2 Реалізація мобільного додатку

Для відображення інтерфейсу були обрані такі елементи, як фрагменти [48]. Вони можуть бути використані повторно під час роботи даного програмного продукту. На рисунку 4.2 представлена головна Activity [49] запропонованого мобільного додатку. На ній подалі будуть розміщені фрагменти. Повний лістинг коду файлів реалізації представлений в додатку Б.

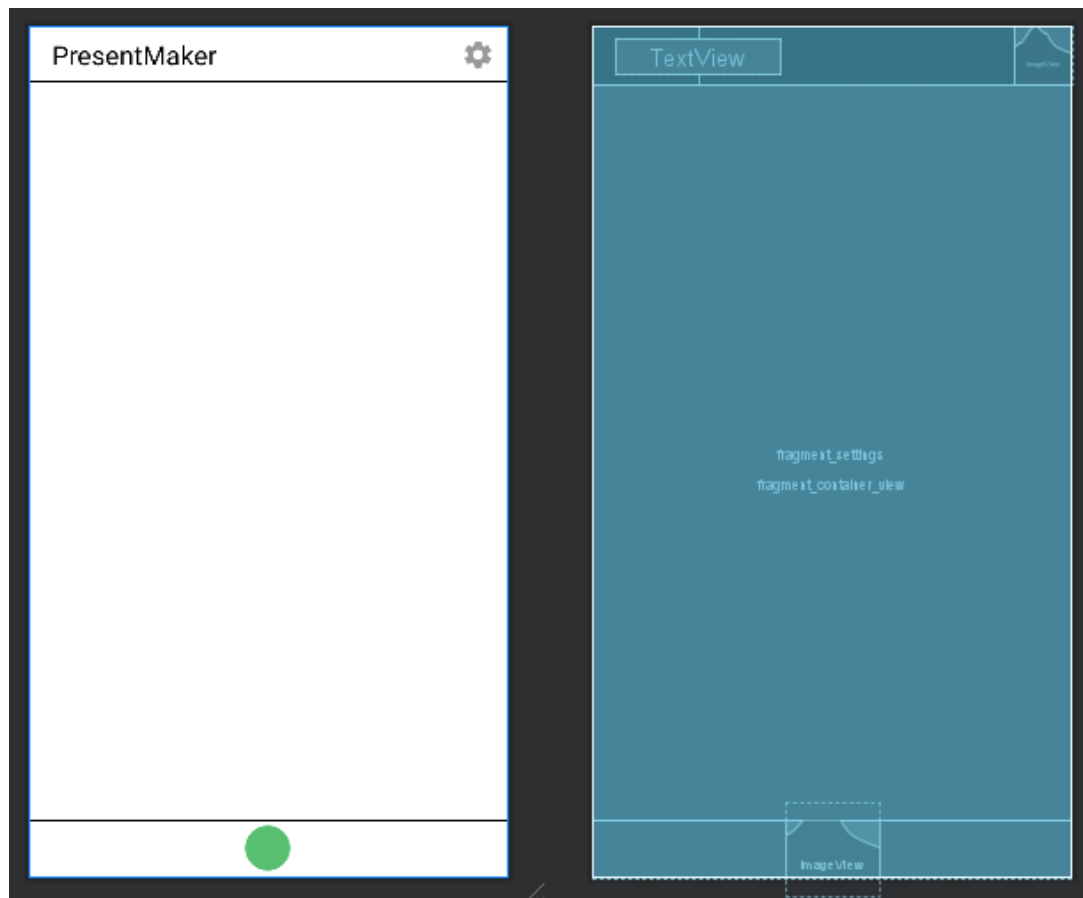


Рисунок 4.2 – Візуальне зображення дизайну activity_main.xml

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.3 показано один із фрагментів мобільного додатку. А саме фрагмент перегляду шаблону перед його вибором. На ньому можна побачити елемент для відображення динамічного списку RecyclerView [50].

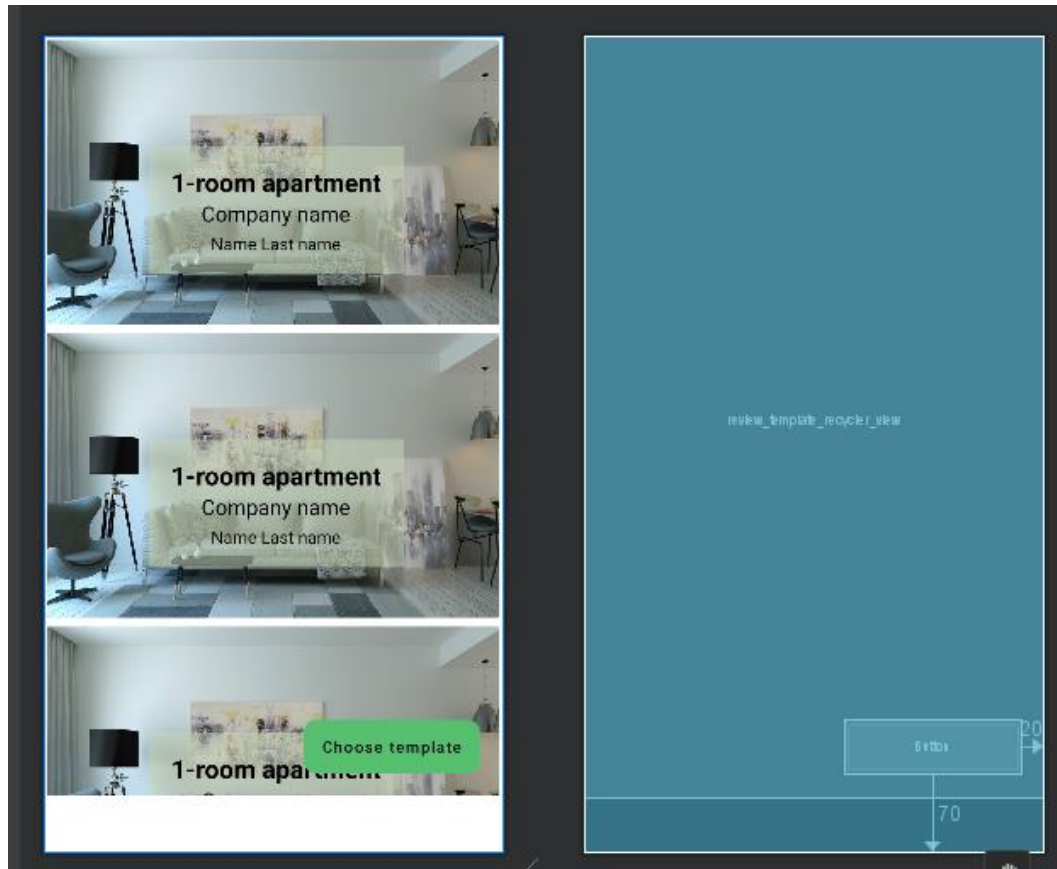


Рисунок 4.3 – Візуальне зображення дизайну review_template_fragment.xml
Джерело: побудовано автором (знімок з екрану)

Кожен слайд презентацій представляє собою елемент CardView [51]. Він використовується для відображення інформації в картках, які мають узгоджений вигляд в усьому мобільному додатку. На рисунку 4.4 зображений файл для головного слайду шаблону квартири.

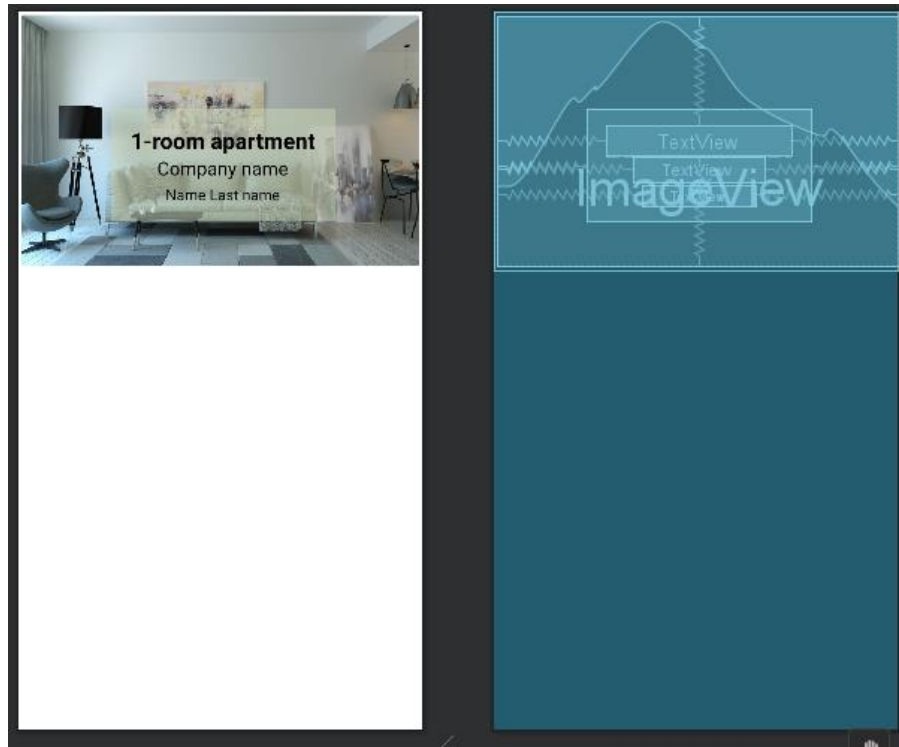


Рисунок 4.4 – Візуальне зображення дизайну apartment_template_1_main_slide.xml

Джерело: побудовано автором (знімок з екрану)

Для зміни теми на світлу або темну були написані файли ресурсів лістинг коду світлої теми представлений нижче:

```
<style name="Theme.ProjectMaker"
parent="Theme.MaterialComponents.DayNight.NoActionBar">
    <item name="colorPrimary">@color/green_100</item>
    <item name="colorPrimaryVariant">@color/grey_100</item>
    <item name="colorOnPrimary">@color/black_100</item>
    <item name="colorSecondary">@color/green_200</item>
    <item name="colorSecondaryVariant">@color/black_100</item>
    <item name="colorOnSecondary">@color/black</item>
    <item name="android:textColor">@color/black</item>
    <item name="android:textAllCaps">>false</item>
    <item name="android:statusBarColor"
tools:targetApi="l"?attr/colorPrimaryVariant</item>
    <!-- Customize your theme here. -->
</style>
```

Для зв'язку з базою даних було створено клас DatabaseHelper поширений від SQLiteOpenHelper [52]. У ньому прописані всі необхідні для роботи мобільного додатку запити (вибір, додавання, зміна).

Нижче представлений лістинг методу створення бази даних:

```
@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    String homeTemplateQuestionsTable = "DROP TABLE IF EXISTS " +
TABLE_NAME;
    sqLiteDatabase.execSQL(homeTemplateQuestionsTable);
    homeTemplateQuestionsTable = "CREATE TABLE " + TABLE_NAME + "(" +
        "id INTEGER PRIMARY KEY AUTOINCREMENT, " +
        "template VARCHAR, " +
        "block VARCHAR, " +
        "title VARCHAR, " +
        "text VARCHAR, " +
        "checkbox BOOLEAN, " +
        "quantity INTEGER, " +
        "image BLOB, " +
        "is_text BOOLEAN, " +
        "is_checkbox BOOLEAN, " +
        "is_quantity BOOLEAN, " +
        "is_image BOOLEAN)";
    sqLiteDatabase.execSQL(homeTemplateQuestionsTable);
}
```

Для взаємодії з ChatGPT був написаний клас ChatGptApi. Він будує та посилає запити. Однак також і приймає відповіді та форматує їх у потрібному для використання формі.

Нижче приведений фрагмент коду, що будує запит до ChatGPT.

```
JSONObject jsonBody = new JSONObject();
try {
    jsonBody.put("model", "gpt-3.5-turbo-instruct");
    jsonBody.put("prompt", question);
    jsonBody.put("max_tokens", 100);
    jsonBody.put("temperature", 0);
} catch (JSONException e) {
    throw new RuntimeException(e);
}
RequestBody body = RequestBody.create(jsonBody.toString(), JSON);
Request request = new Request.Builder()
    .url("https://api.openai.com/v1/completions")
    .header("Authorization", "Bearer sk-
pzGIZ5X8I6N5ifYg6udAT3BlbkFJZac7ALlCufn6gZCrLGpw")
```

```

        .post(body)
        .build();
    CountdownLatch countdownLatch = new CountdownLatch(1);
    OkHttpClient client = new OkHttpClient();

```

Для опису вигляду слайду та його метаданих був розроблений абстрактний клас `TemplateBasesHolder`. Він поширений від `ViewHolder` [53]. Код класу нижче:

```

public abstract class TemplateBasesHolder<T> extends
RecyclerView.ViewHolder {

    protected TemplateBasesHolder(View itemView) {
        super(itemView);
    }
    public abstract void bind(T type);
}

```

Уже на його основі описані холдери для кожного шаблону слайду. Також для них була написана модель для тримання їх даних.

Нижче приведений приклад такого класу, а саме модель головного слайду:

```

public class MainSlide {
    private Bitmap image;
    private String title, company, pib;
    public MainSlide(Bitmap image, String title, String company, String
pib) {
        this.image = image;
        this.title = title;
        this.company = company;
        this.pib = pib;}
    public Bitmap getImage() {return image;}
    public void setImage(Bitmap image) {this.image = image;}
    public String getTitle() {return title;}
    public void setTitle(String title) {this.title = title;}
    public String getCompany() {return company;}
    public void setCompany(String company) {this.company = company; }
    public String getPib() {return pib;}
    public void setPib(String pib) {this.pib = pib;}
}

```

Для кожного фрагменту необхідно прописати логіку. Тому для цього були створені спеціальні класи, які походять від класу `Fragment`. Нижче продемонстрований один з таких класів:

```

public class HomeFragment extends Fragment {
    RecyclerView mRecyclerView;

```



```

public HomeFragment() {
    super(R.layout.home_fragment);
}
@Override
public void onCreateView(@NonNull View view, @Nullable Bundle
savedInstanceState) {
    super.onCreateView(view, savedInstanceState);
    mRecyclerView = view.findViewById(R.id.list_view);
    TemplateAdapter adapter = new TemplateAdapter(view.getContext(),
getAllTemplates());
    mRecyclerView.setAdapter(adapter);
}
private ArrayList<Template> getAllTemplates() {
    ArrayList<Template> arrayList = new ArrayList<>();
    arrayList.add(new Template(R.drawable.apartment_template));
    arrayList.add(new Template(R.drawable.house_template));
    arrayList.add(new Template(R.drawable.office_template));
    return arrayList;
}
}

```

Для використання RecyclerView необхідно поширити Adapter [54] зі своєю логікою для коректної з ним роботи. Нижче приведений модуль в якому прописана логіка заповнення RecyclerView в адаптері:

```

@Override
public void onBindViewHolder(@NonNull Holder holder, final int
position) {
    Template template = arrayList.get(position);
    holder.templateButton.setClipToOutline(true);
    holder.templateButton.setImageResource(template.getImage());
    holder.templateButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            template_number = position;
            EditPresentationFragment.lastSlide = 0;
            ReplaceFragment.replaceFragment();
        }
    });
}
}

```

4.3 Демонстрація роботи мобільного додатку

При запуску даного мобільного додатку користувача зустрічає фрагмент з вибором шаблону. Для його вибору потрібно натиснути на нього. Також можна зверху побачити навігаційне меню з налаштуваннями. А саме зі зміною теми мобільного додатку. Знизу знаходиться кнопка для повернення до вибору шаблону. Вона, як і верхнє навігаційне меню, буде доступна користувачу протягом усієї роботи з даним програмним продуктом (рис. 4.5).



Рисунок 4.5 – Головне вікно мобільного додатку
Джерело: побудовано автором (знімок з екрану)

При натисканні на іконку налаштувань відкриється вікно з можливістю обрати світлу або темну тему за допомогою перемикача. Повернутись назад можна, натиснувши кнопку знизу. Це перший варіант. Також це можна зробити, натиснувши кнопку зліва від вікна (рис. 4.6).

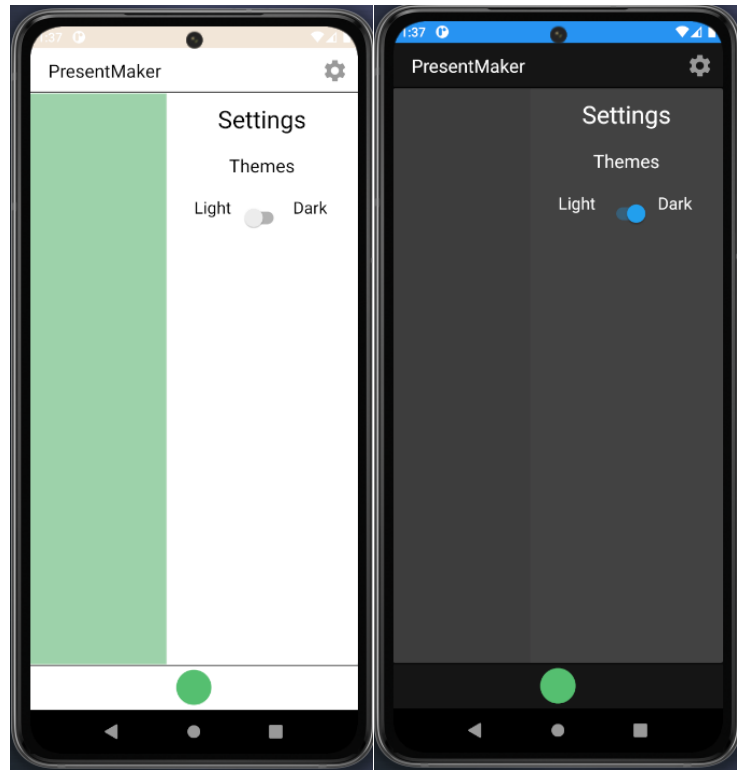


Рисунок 4.6 – Вікно налаштувань із увімкненою світлою татемною темами відповідно

Джерело: побудовано автором (знімок з екрану)

Після вибору шаблону відкриється вікно з попереднім оглядом можливих слайдів-шаблонів у ньому. Обрати потрібний можна, натиснувши на кнопку «Choose template» (рис. 4.7).



Рисунок 4.7 – Сторінка Workspace

Джерело: побудовано автором (знімок з екрану)

Погодившись із обраним шаблоном, можна переглянути список питань, на основі яких буде відбуватися генерація. Їхня форма складається з декількох блоків. Між питаннями можна переключатися за допомогою кнопок у нижній частині. Їхня нумерація представлена знизу екрану. Питання складаються з тексту. Також є певний елемент для відповіді. Це може бути поле для вводу тексту або числового значення, прапорця, вибір зображення з галереї та будь-яке поєднання цих полів. Відповіді на питання зберігаються для спрощення подальших створень презентацій за цим шаблоном (рис. 4.8).

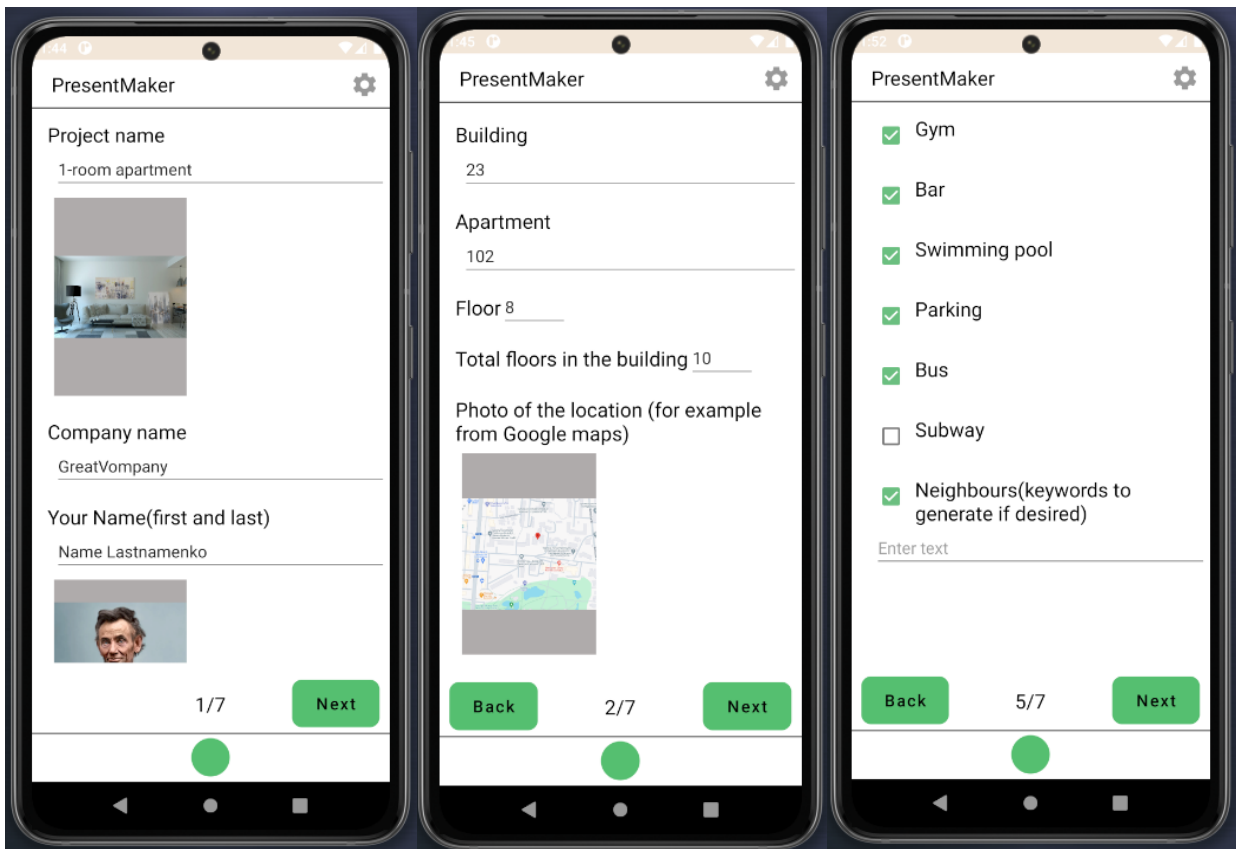


Рисунок 4.8 – Вікна з формою запитань

Джерело: побудовано автором (знімок з екрану)

Після надання відповідей на всі питання буде згенерована презентація. Відкриється певне вікно, зображене на рисунку 4.9. Тут можна побачити головний слайд презентації та три кнопки: повернення до питань, збереження презентації (якщо користувач не бажає переглядати результат генерації, то він може одразу її зберегти в папку Downloads на смартфоні у форматі pdf і редагування презентації).

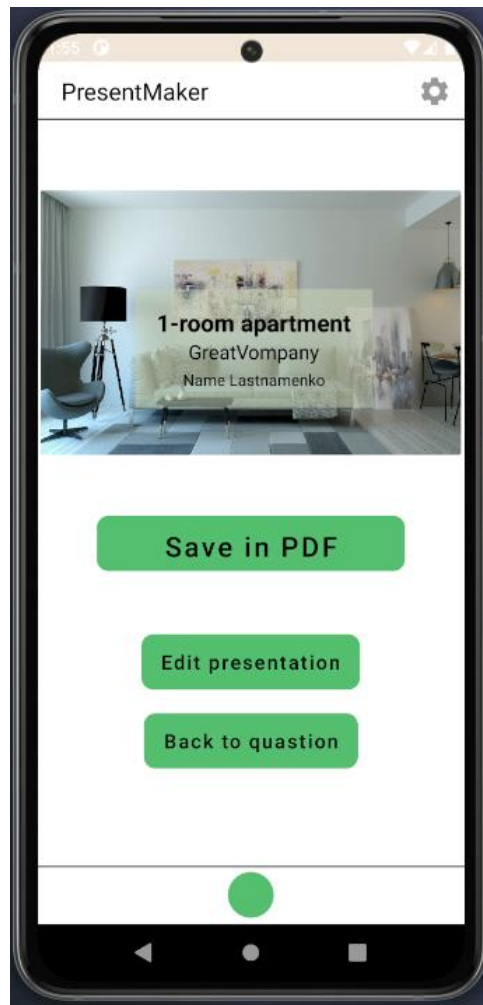


Рисунок 4.9 – Вікно з готовою презентацією

Джерело: побудовано автором (знімок з екрану)

Після натискання на кнопку редагування відкривається вікно зображене на рисунку 4.10. На ньому можна побачити наступне:

- зверху кнопки повернення назад та закінчення редагування;
- нижче – можливість вибору стилю презентації;
- власне слайд, при натисканні на текстовий елемент відкриється вікно з редагуванням тексту;
- нижче слайду кнопки для переходу між слайдами;
- кнопку зміни зображення та додавання або видалення слайду.

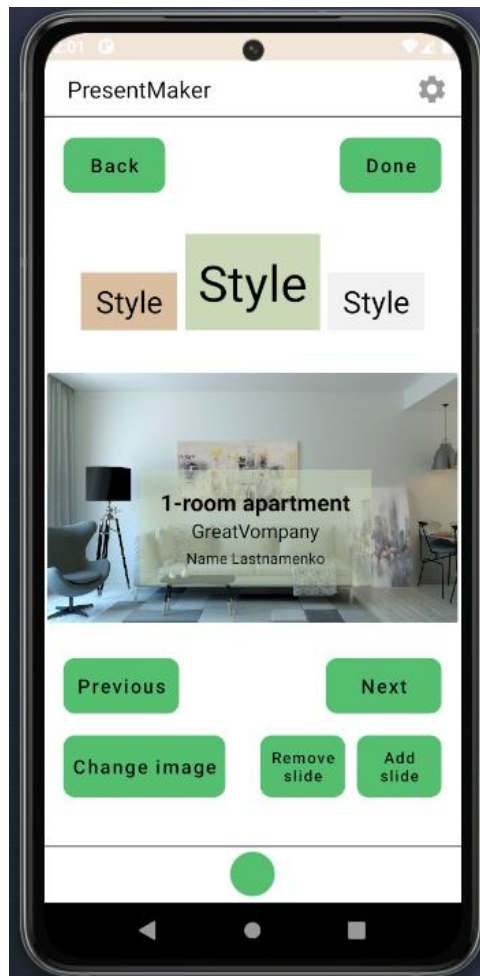


Рисунок 4.10 – Вікно редагування презентації

Джерело: побудовано автором (знімок з екрану)

При натисканні на текст відбувається перехід на вікно з його редагуванням. Або регенерацією. При натисканні на кнопку зверху відбудеться перехід на попереднє вікно, на кнопку з «галочкою» – текст із відповідного поля перенесеться. Він відобразиться на слайді. При натисканні на кнопку посередині ChatGPT згенерує текст на основі введеного. А кнопка з «хрестиком» поверне його попередню версію – варіанту, який був до генерації. Також можна натиснути на інший текст на слайді для його редагування (рис. 4.11).

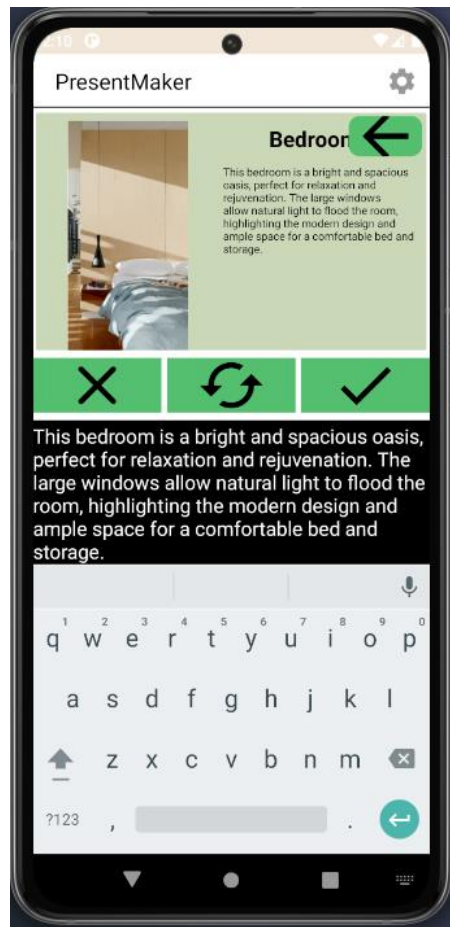


Рисунок 4.11 – Вікно редагування тексту

Джерело: побудовано автором (знімок з екрану)

Якщо натиснути на кнопку «Add slide», то відкриється список слайдів шаблону (рис. 4.12). Далі треба обрати один із запропонованих варіантів. При натисканні на необхідний слайд він буде доданий до презентації і відкриється в вікні редагування.



Рисунок 4.12 – Вікно з вибором слайду

Джерело: побудовано автором (знімок з екрану)

Після завершення редагування презентації треба натиснути кнопку збереження. Далі можна відкрити сформований файл у форматі pdf будь яким відповідним читачем (рис. 4.13).

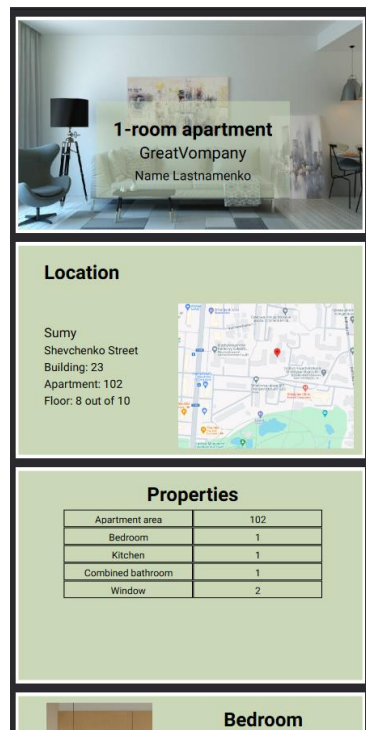


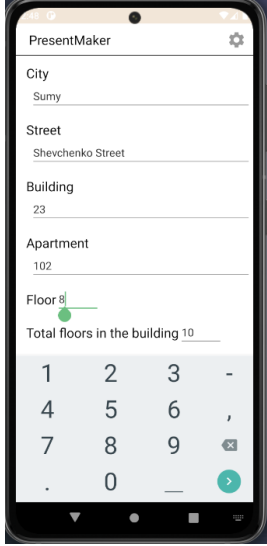

Рисунок 4.13 – Відкрита презентація у pdf форматі стороннім читачем файлів
Джерело: побудовано автором (знімок з екрану)

4.4 Тестування мобільного додатку

Ретельне тестування мобільного додатка є вирішальним кроком у забезпеченні його функціональності, зручності використання та загальної якості. Воно допомагає виявити та виправити помилки. Тому потрібно провести функціональне тестування. Варто перевірити чи будуть дії користувача викликати очікуваний результат.

Тестування функціональності форм у мобільному додатку було здійснено за допомогою методу «чорного ящика». Він передбачає перевірку роботи програмного продукту без доступу до його коду [55]. Деталі та результати тестування продемонстровані у таблиці 4.1.

Таблиця 4.1 – Результати тестування даного мобільного додатку

№	Назва тест кейсу	Очікуваний результат	Фактичний результат	0/1
1	<p>Перевірка коректності вводу даних на відповіді до питань (у поля з числовими значеннями введення інших символів).</p>	<p>У дані поля неможливо ввести не число.</p>	 <p>Відкривається клавіатура з цифрами, при введенні не числових символів нічого не відбувається</p>	1
2	<p>Замість зображення прикріплення іншого типу файлу.</p>	<p>Відсутність можливості вибору іншого файлу.</p>	 <p>у галереї відображаються тільки зображення</p>	1

Продовження табл. 4.1

№	Назва тест кейсу	Очікуваний результат	Фактичний результат	0/1
3	Відсутність доступу до мережі Інтернет	Помилка відповіді ChatGPT.		1
4	Пропуск поля для введення.	Генерація презентації продовжиться.	При залишенні поля пустим на згенерованій презентації також не буде тексту.	1
5	Натискання на кнопки переходів між екранами.	Перехід відбувається успішно.	Усі кнопки навігації працюють у штатному режимі.	1

Джерело: побудовано автором

У результаті проведеного тестування, серйозних помилок чи недоліків не було знайдено. Функціонал мобільного додатку працює належним чином.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено концептуалізацію та моделювання мобільного додатку автоматизації презентування нерухомості, використовуючи можливості ChatGPT для додавання змістовних описів тієї чи іншої власності.

Завдяки глибокому аналізу предметної області та продуктів-аналогів були визначені необхідні для розробки проєкту засоби реалізації та функціонал даного мобільного додатку.

Використання передових технологій, таких як ChatGPT, не лише виділяє цей мобільний додаток, але й забезпечує її перспективність та адаптивність до нових вимог сфери нерухомості.

Проведення моделювання за допомогою IDEF0 та UseCase, забезпечує достатню структурованість для розуміння програмного продукту та управління ним. Застосування запропонованого мобільного додатку не тільки спрощує процес створення презентації, але й доповнює його використанням новітніх технологій штучного інтелекту. Це відповідає необхідності в постійній еволюції технологій в будь-якій сфері, у даному випадку сфері нерухомості. Застосування даної розробки дозволить зменшити витрату часу на створення самих презентацій і водночас забезпечувати їх якісним контентом.

Було розроблено мобільний додаток автоматизації презентування нерухомості, з використанням ChatGPT. Використано такі інструменти розробки як Android Studio, Java, SQLite та ChatGPT. Були реалізовані всі функціональні вимоги до даного мобільного додатку, які визначені в теоретичній частині кваліфікаційної роботи. Проведене тестування розробленого програмного продукту за допомогою методу «чорного ящика». Значних проблем у роботі мобільного додатку не виявлено.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. How many smartphones are in the world? BankMyCell. URL: <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world> (date of access: 15.11.2023).
2. Мартинюк Н. Актуальність сучасних мобільних додатків. XII Всеукраїнська науково-практична конференція «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРОФЕСІЙНІЙ ДІЯЛЬНОСТІ», м. Рівне, 13 жовт. 2019 р. С. 65–66. URL: <https://iktmvi.rshu.edu.ua/files/konf/ZB-ITVPD-2019.pdf#page=65> (дата звернення: 10.11.2023).
3. Weather on the go: an assessment of smartphone mobile weather application use among college students / M. D. Phan et al. Bulletin of the american meteorological society. 2018. Vol. 99, no. 11. P. 2245–2257. URL: <https://doi.org/10.1175/bams-d-18-0020.1> (date of access: 10.11.2023).
4. Ichanska N. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки / N. Ichanska, S. Ulko // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2020. – Т. 1 (59). – С. 74-78. – doi:<https://doi.org/10.26906/SUNZ.2020.1.074>. (дата звернення: 10.11.2023)
5. Norea Vlad Giurgiu, Gabriela Prostean, Andra Diaconescu. Software application development for real estate activities. Buletinul științific al universității politehnica timisoara – seria inginerie și management. 2020. P. 19–25. URL: <https://www.ceeol.com/search/article-detail?id=913826> (date of access: 12.11.2023).
6. Benjamin Leong E.-Jenn. Mobile based real time property/land for sale viewing using google map and communication tool for buyer and client/real estate agent. 2021. 267 p. URL: <http://eprints.utar.edu.my/id/eprint/4419> (date of access: 12.11.2023).

7. Mta Wickramasinghe, Shidn de Silva, D. Gunasekera. An android application to manage house rental and maintenance in sri lanka. International journal of research in computing. P. 32–41. URL: <http://ir.kdu.ac.lk/handle/345/5771> (date of access: 12.11.2023).
8. Sallow A. B., Hussain S. R. Multi-Agent system for supporting and managing real estate marketing. Academic journal of nawroz university. 2020. Vol. 9, no. 3. P. 54. URL: <https://doi.org/10.25007/ajnu.v9n3a703> (date of access: 15.12.2023).
9. Real estate master / A. Nagaraja Rao et al. International journal of research in engineering, science and management. 2019. URL: https://www.ijresm.com/Vol.2_2019/Vol2_Iss11_November19/IJRESM_V2_I11_13.pdf (date of access: 13.11.2023).
10. Алещенко М.в. Розробка мобільного застосунку для галузі нерухомості України. XXI Всеукраїнська науково-технічна конференція молодих вчених, аспірантів та студентів «СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ», м. Одеса, 22–23 квіт. 2021 р. С. 105–106. URL: <https://card-file.ontu.edu.ua/server/api/core/bitstreams/344b01ec-03e1-467b-8ceb-7de22d9c36f3/content> (дата звернення: 13.11.2023).
11. Saeed Ul Karim Siddiqui, Shayla Islam, Kasthuri Subaramaniam. Android application development for property rental services. Proceedings of mechanical engineering research day, August 2022, Kuala Lumpur. P. 147–148. URL: https://www3.utm.edu.my/care/proceedings/merd22/pdf/04%20Computer%20Modeling%20and%20Simulation/070_p147_148.pdf (date of access: 14.11.2023).
12. Athina Kouklinou. Usability heuristics for real estate renting apps : masterThesis. Athens, 2021. 74 p. URL: <https://repository.ihu.edu.gr/xmlui/handle/11544/30005> (date of access: 14.11.2023).
13. Christian Osita Ifediora. Ssensing the use of smart phones-based apps, software and geographic information system (gis) in real estate practice. International journal of development and economic sustainability. 2022. Vol. 10. P. 26–40. URL: <https://tudr.org/id/eprint/304/> (date of access: 15.11.2023).

14. A. C. Aydinoglu, R. Bovkir. Developing a mobile application for smart real estate information. The international archives of the photogrammetry, remote sensing and spatial information sciences. 2020. XLIV-4/W3-2020. P. 89–94. URL: <https://isprs-archives.copernicus.org/articles/XLIV-4-W3-2020/89/2020/isprs-archives-XLIV-4-W3-2020-89-2020.html> (date of access: 15.11.2023).
15. OpenAI. ChatGPT. URL: <https://chat.openai.com/> (date of access: 11.11.2023).
16. OpenAI. URL: <https://openai.com/> (date of access: 11.11.2023).
17. Abid Haleem Mohd Javaid Ravi Pratap Singh. An era of ChatGPT as a significant futuristic support tool: a study on features, abilities, and challenges. BenchCouncil transactions on benchmarks, standards and evaluations. 2022.
18. Dinesh Kalla, Nathan Smith. Study and Analysis of Chat GPT and its Impact on Different Fields of Study. International Journal of Innovative Science and Research Technology Volume 8, Issue 3. 2023.
19. AI benchmark: running deep neural networks on android smartphones / Andrey Ignatov et al. CVF Open Access. URL: https://openaccess.thecvf.com/content_ECCVW_2018/papers/11133/Ignatov_AI_Benchmark_Running_Deep_Neural_Networks_on_Android_Smartphones_ECCVW_2018_paper.pdf (date of access: 10.11.2023).
20. Comparison and benchmarking of AI models and frameworks on mobile devices / Chunjie Luo et al. ArXiv. 2020. URL: <https://arxiv.org/abs/2005.05085> (date of access: 10.11.2023).
21. Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence / R. Rajalakshmi et al. Eye. 2018. Vol. 32, no. 6. P. 1138–1144. URL: <https://doi.org/10.1038/s41433-018-0064-9> (date of access: 15.12.2023).
22. Diagnostic accuracy of community-based diabetic retinopathy screening with an offline artificial intelligence system on a smartphone / S. Natarajan et al. JAMA ophthalmology. 2019. Vol. 137, no. 10. P. 1182. URL: <https://doi.org/10.1001/jamaophthalmol.2019.2923> (date of access: 15.12.2023).

23. Wearable artificial intelligence biosensor networks / Y. Zhang et al. *Biosensors and bioelectronics*. 2022. P. 114825. URL: <https://doi.org/10.1016/j.bios.2022.114825> (date of access: 15.12.2023).
24. Building an artificial intelligence-powered medical image recognition smartphone application: what medical practitioners need to know / A. P. Susanto et al. *Informatics in medicine unlocked*. 2022. Vol. 32. P. 101017. URL: <https://doi.org/10.1016/j.imu.2022.101017> (date of access: 15.12.2023).
25. Zachi I Attia, David M Harmon, Elijah R Behr, Paul A Friedman, Application of artificial intelligence to the electrocardiogram, *European Heart Journal*, Volume 42, Issue 46, 7 December 2021, Pages 4717–4730, <https://doi.org/10.1093/eurheartj/ehab649> (date of access: 15.12.2023).
26. Толочко С. В. Методика використання сервісів зі штучним інтелектом для репрезентації результатів наукових досліджень. *Ricerche scientifiche e metodi della loro realizzazione*, м. Bologna, 29 верес. 2023 р.
27. Годунова А. В. Використання сервісів зі штучним інтелектом для оптимізації дослідницької та наукової діяльності. *Управління якістю освіти в умовах воєнного стану та повоєнного відновлення України*, м. Київ, 23 берез. 2023 р.
28. B.J Copeland. Artificial intelligence. *Britannica*. URL: <https://www.britannica.com/technology/artificial-intelligence> (date of access: 28.11.2023).
29. Canva. URL: <https://www.canva.com> (date of access: 14.11.2023).
30. Gamma app: generate AI presentations, webpages & docs. URL: <https://gamma.app> (date of access: 14.11.2023).
31. OpenAI. API reference. URL: <https://platform.openai.com/docs/api-reference> (date of access: 11.11.2023).
32. Ознайомлення з Adobe PDF. *Adobe Help Center*. URL: <https://helpx.adobe.com/ua/incopy/using/pdf.html> (дата звернення: 18.11.2023).

33. Методи теоретичних досліджень. Pidru4niki. URL: https://pidru4niki.com/1373112055109/pedagogika/metodi_teoretichnih_doslidzen (дата звернення: 15.11.2023).
34. What is user experience (UX) design? – updated 2023. The Interaction Design Foundation. URL: <https://www.interaction-design.org/literature/topics/ux-design> (date of access: 19.11.2023).
35. Churchville F. What is user interface (UI)? Definition from search app architecture. App Architecture. URL: <https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI> (date of access: 18.11.2023).
36. Сутність методу моделювання як науковий процес пізнання. Особливості наукового пізнання. Ужгородський національний університет. URL: <https://www.uzhnu.edu.ua/en/infocentre/get/25551> (дата звернення: 15.11.2023).
37. Методологія IDEF0, основні елементи діаграми IDEF0. URL: https://stud.com.ua/87184/ekonomika/metodologiya_idef0 (дата звернення: 19.11.2023).
38. Daly N. What is a use case & how to write one | wriike. Blog Wriike. URL: <https://www.wriike.com/blog/what-is-a-use-case/> (date of access: 20.11.2023).
39. Притченко Р. С. Системно-функціональний метод дослідження державно-правових явищ. Інституційний репозитарій Одеського національного університету імені І.І. Мечникова: Home. URL: <http://dspace.onu.edu.ua:8080/bitstream/123456789/29634/1/30-32.pdf> (дата звернення: 15.11.2023).
40. Android Studio. Flamingo. Version 2022.2.1 Patch 2. Google, 2022.
41. What is IDE or integrated development environments? | veracode. Veracode. URL: <https://www.veracode.com/security/integrated-development-environment> (date of access: 21.11.2023).
42. What is Android | Android. Android. URL: <https://www.android.com/what-is-android/> (date of access: 21.11.2023).
43. Oracle cloud applications and cloud platform. URL: <https://www.oracle.com/java/technologies/downloads/> (date of access: 21.11.2023).

44. SQLite Home Page. URL: <https://www.sqlite.org/index.html> (date of access: 18.11.2023).
45. International council on systems engineering website. INCOSE. URL: <https://www.incose.org/> (date of access: 23.11.2023).
46. Maxym Z. IDEF. Maxym Zosym. URL: <https://www.maxzosim.com/idef/> (date of access: 23.11.2023).
47. В. Шаряк. Методи дослідження системних характеристик моделей бази даних. CORE – Aggregating the world’s open access research papers. URL: <https://core.ac.uk/download/pdf/161263548.pdf> (дата звернення: 20.11.2023).
48. Fragments. Android Developers. URL: <https://developer.android.com/guide/fragments> (date of access: 28.11.2023).
49. Introduction to activities. Android Developers. URL: <https://developer.android.com/guide/components/activities/intro-activities> (date of access: 28.11.2023).
50. Create dynamic lists with RecyclerView. Android Developers. URL: <https://developer.android.com/develop/ui/views/layout/recyclerview> (date of access: 28.11.2023).
51. Create a card-based layout. Android Developers. URL: <https://developer.android.com/develop/ui/views/layout/cardview> (date of access: 29.11.2023).
52. SQLiteOpenHelper. Android Developers. URL: <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper> (date of access: 29.11.2023).
53. RecyclerView.ViewHolder. Android Developers. URL: <https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView.ViewHolder> (date of access: 30.11.2023).
54. RecyclerView.Adapter. Android Developers. URL: <https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView.Adapter> (date of access: 01.12.2023).

55. Singureanu С. Тестування чорних скриньок - процес, інструменти, контрольний список та інше. ZAPTEST. URL: <https://www.zaptest.com/uk/тестування-чорної-скриньки-що-це-так> (дата звернення: 01.12.2023).

ДОДАТОК А

Планування робіт

У сфері нерухомості, яка постійно розвивається, попит на застосування інноваційних та ефективних рішень продовжує зростати. А використання штучного інтелекту відкриває можливість зробити революцію у створенні презентацій. Технологічний прогрес дійсно продовжує змінювати ділову практику. Тому використання ШІ має великий потенціал для підвищення ефективності, точності та візуальної привабливості презентацій нерухомості. Це є досить корисним сьогодні. Застосовуючи можливості штучного інтелекту в генерації природної мови, фахівці з нерухомості тепер можуть створювати свої презентації з більшою швидкістю, пропонуючи клієнтам і зацікавленим сторонам більш персоналізований досвід.

Деталізація мети проєкту методом SMART. Для успішності та конкурентоспроможності проєкту потрібно правильно визначити його мету за допомогою SMART-методу. Формулювання SMART-мети проєкту є таким: «Розробка мобільного додатку автоматизації презентування нерухомості з використанням ChatGPT для зменшення часу рієлторам на створення відповідних презентацій із мінімальним порогом входу до розуміння даного мобільного додатку до 15 грудня 2023 року.

Результати деталізації методом SMART розміщені у таблиці А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Створення мобільного додатку для автоматизації процесу створення презентацій для продажу нерухомості з використанням ChatGPT.
-------------------------	---

Продовження табл. А.1

Measurable (вимірювана)	Результатом роботи проєкту є створена презентація в форматі pdf.
Achievable (досяжна)	Реалізація додатку здійснюється за допомогою середовища розробки Android Studio мовою програмування Java, генерація відбувається за допомогою нейромережі ChatGPT. Враховуючи, що є чітко поставлені задачі та доступні ресурсні можливості, розробник достатньо кваліфікований для їх виконання.
Relevant (реалістична)	Зменшення часу на створення презентацій з мінімальним порогом входу до розуміння даного мобільного додатку.
Time-framed (обмежена у часі)	Проєкт повинен бути виконаний згідно з календарним планом – до 15 грудня 2023 року.

Джерело: побудовано автором

Планування змісту структури робіт. Основним інструментом для планування змісту структури робіт служить структура розподілу робіт (WBS) – це ієрархічна декомпозиція проєкту на менші, більш керовані компоненти, що забезпечує систематичний і організований підхід до планування та управління проєктом. Він розбиває обсяг проєкту на окремі робочі пакети або результати, забезпечуючи чітке та візуальне представлення структури проєкту. WBS зазвичай створюється на початку фази планування проєкту і служить основою для визначення завдань, оцінки ресурсів, розподілу обов’язків і розробки реалістичного графіка проєкту.

WBS організовує роботу над проєктом у деревоподібну структуру, в якому верхній рівень представляє загальний проєкт, а наступні рівні розбивають проєкт на менші, більш керовані компоненти. Кожен рівень WBS представляє різний рівень деталізації,

полегшуючи детальне розуміння обсягу проєкту та гарантуючи, що вся необхідна робота визначена. Використання WBS допомагає у розподілі ресурсів і моніторингу прогресу, а також служить основою для створення інших документів управління проєктом, таких як графік проєкту та бюджет. Загалом WBS є потужним інструментом, який допомагає у плануванні, виконанні та контролі проєкту, надаючи повне та структуроване уявлення про обсяг проєкту та розбивку робіт.

На рисунку А.1 приведена WBS-структура даного проєкту.

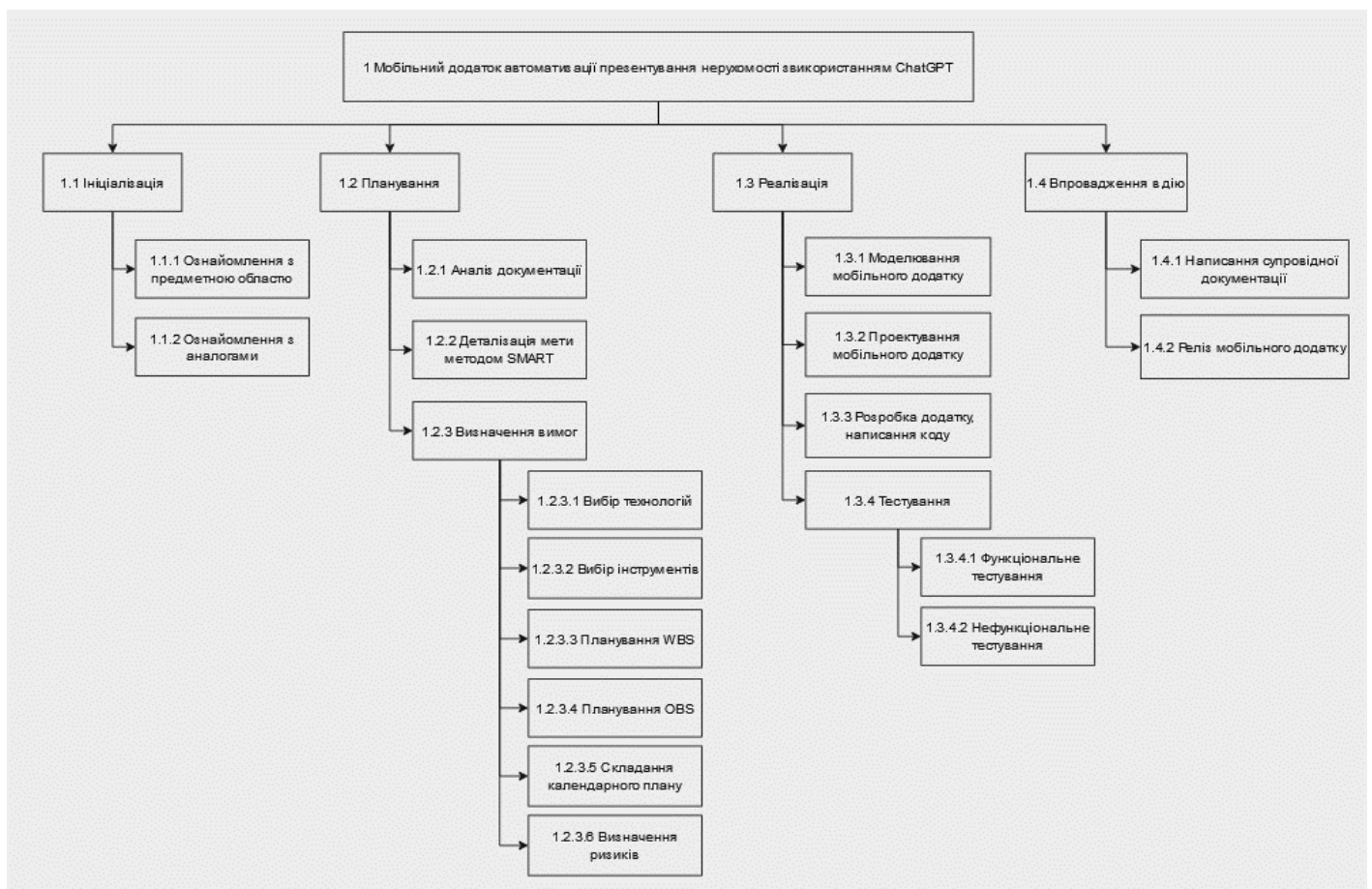


Рисунок А.1 – WBS-структура проєкту

Джерело: побудовано автором

Планування структури організації, для впровадження готового проєкту (OBS). Після того, побудови WBS структури проєкту наступним етапом є розробка

структури розподілу організації (OBS) — це ієрархічне представлення структури організації в контексті конкретного проекту. На відміну від структури розподілу робіт (WBS), яка зосереджується на розбивці проекту на робочі пакети, OBS займається ілюстрацією організаційної ієрархії та визначенням відносин і обов'язків окремих осіб або груп у проекті. OBS має вирішальне значення для управління проектами, оскільки допомагає розподіляти ролі та обов'язки, встановлювати лінії звітності та забезпечувати ефективне спілкування та координацію між членами команди.

OBS зазвичай відображає організаційну схему компанії, але пристосовується до потреб конкретного проекту. Він визначає ключових зацікавлених сторін проекту, таких як керівники проектів, керівники груп та окремі члени команди, і ілюструє, як вони вписуються в загальну структуру проекту. Такий розподіл дозволяє чітко зрозуміти, хто відповідає за кожен аспект проекту, і гарантує, що кожен знає про свої ролі та обов'язки. OBS є важливим інструментом для керівників проектів для ефективного управління ресурсами, розподілу обов'язків і сприяння співпраці в рамках організаційної структури, що сприяє успішному виконанню проекту. Учасники проекту визначені в таблиці А.2. Організаційна структура проекту зображена на рисунку А.2.

Таблиця А.2 – Виконавці проекту

Роль	Ім'я	проектна роль
Керівник проекту	Антипенко В.П.	Відповідає за дотримання термінів, формує завдання на розробку проекту
Менеджер проекту	Шелехов Д.В.	Відповідає за дотримання термінів, розподіл ресурсів та завдань між учасниками проекту.
Розробник	Шелехов Д.В.	Виконує проектування та розробку функціоналу мобільного додатку
Тестувальник	Шелехов Д.В.	Відповідає за тестування мобільного додатку

Джерело: побудовано автором

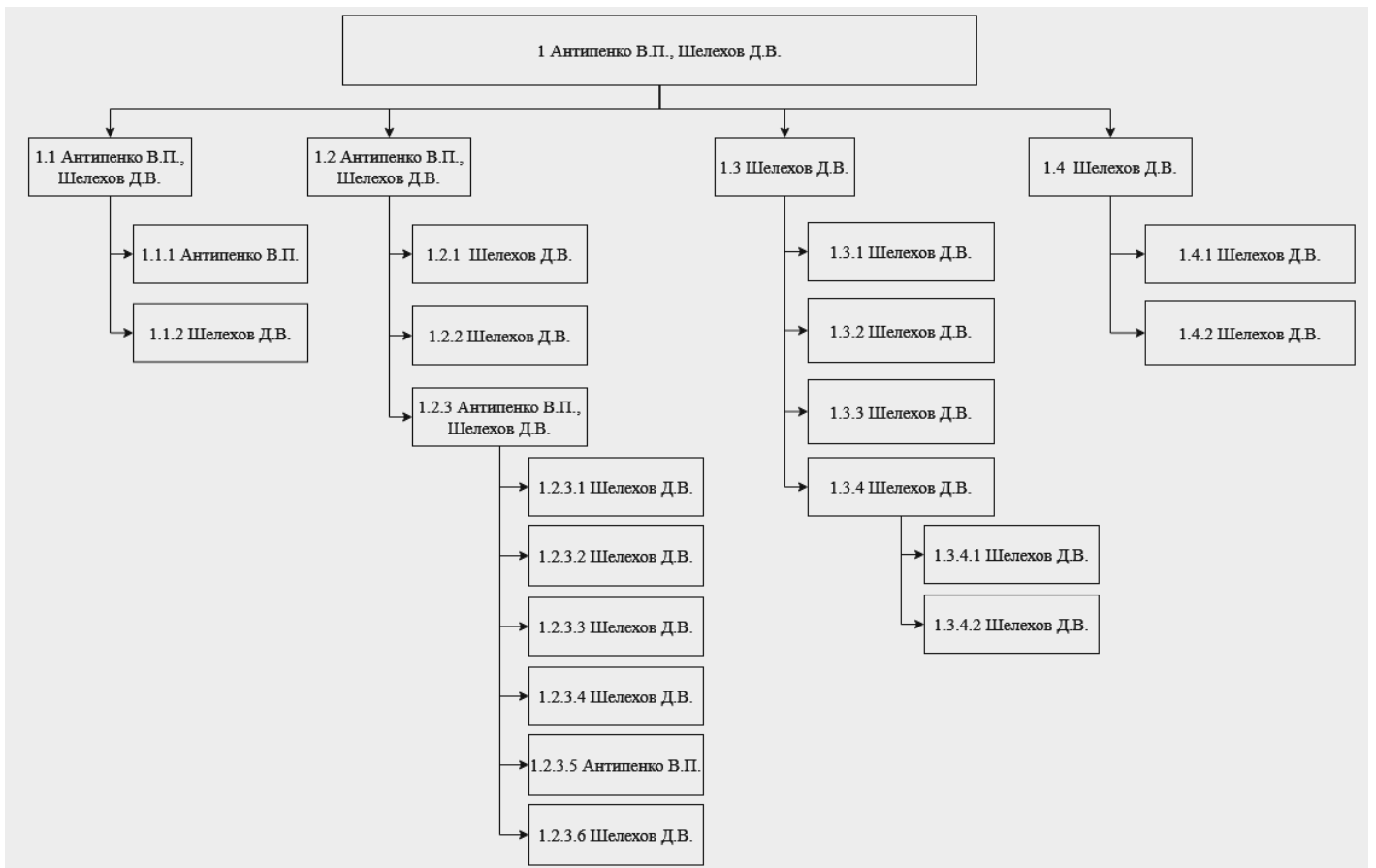


Рисунок А.2 – OBS-структура проекту

Джерело: побудовано автором

Діаграма Ганта. Наступним кроком буде розробка діаграми Ганта – це візуальний інструмент управління проектом, який використовується для ілюстрації розкладу проекту в часі. Ця діаграма забезпечує графічне представлення завдань або дій на часовій шкалі. Горизонтальна вісь діаграми відображає тривалість проекту, часто розбитого на дні, тижні або місяці, тоді як вертикальна вісь перераховує різні завдання або дії, які складають проект. Кожне завдання представлено на діаграмі горизонтальною смугою, довжина якої відповідає тривалості завдання. Діаграми Ганта мають кольорове кодування, щоб позначити різні залежності завдань, етапи та загальний хід проекту.

Діаграми Ганта пропонують керівникам проектів і членам команди візуальний та інтуїтивно зрозумілий спосіб зрозуміти розклад проекту, залежності та розподіл

ресурсів. Вони дозволяють ефективно планувати, відстежувати та повідомляти про часові рамки проекту, допомагаючи визначити критичні дії та потенційні затримки. Крім того, діаграми Ганта дають змогу зацікавленим сторонам швидко зрозуміти послідовність завдань, етапи проекту та кінцеві терміни, сприяючи прозорості та співпраці в команді проекту. Цей широко використовуваний інструмент управління проектами допомагає координувати та контролювати складні проекти, забезпечуючи повний огляд прогресу проекту та сприяючи прийняттю обґрунтованих рішень протягом життєвого циклу проекту. Діаграма Ганта представлена на рисунку А.3. Список робіт для побудови діаграми Ганта для кращого розгляду продемонстрований на рисунку А.4

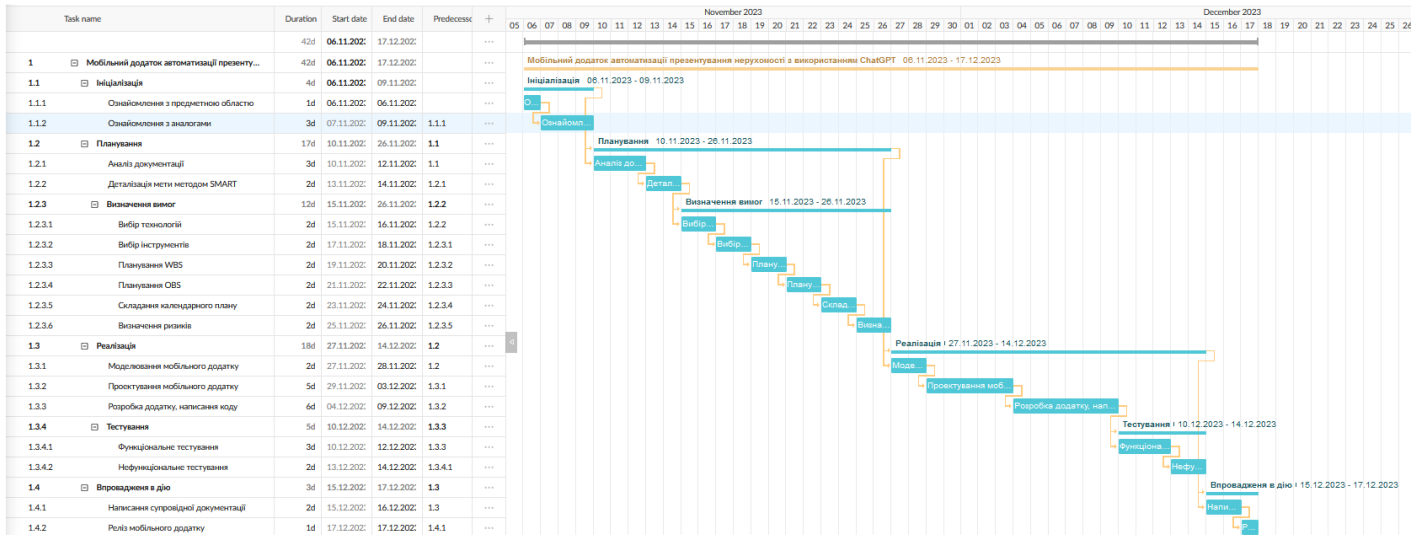


Рисунок А.3 – Діаграма Ганта проекту

Джерело: побудовано автором

Task name	Duration	Start date	End date	Predecessor
	42d	06.11.2021	17.12.2021	
1 ☐ Мобільний додаток автоматизації презенту...	42d	06.11.2021	17.12.2021	
1.1 ☐ Ініціалізація	4d	06.11.2021	09.11.2021	
1.1.1 Ознайомлення з предметною областю	1d	06.11.2021	06.11.2021	
1.1.2 Ознайомлення з аналогами	3d	07.11.2021	09.11.2021	1.1.1
1.2 ☐ Планування	17d	10.11.2021	26.11.2021	1.1
1.2.1 Аналіз документації	3d	10.11.2021	12.11.2021	1.1
1.2.2 Деталізація мети методом SMART	2d	13.11.2021	14.11.2021	1.2.1
1.2.3 ☐ Визначення вимог	12d	15.11.2021	26.11.2021	1.2.2
1.2.3.1 Вибір технологій	2d	15.11.2021	16.11.2021	1.2.2
1.2.3.2 Вибір інструментів	2d	17.11.2021	18.11.2021	1.2.3.1
1.2.3.3 Планування WBS	2d	19.11.2021	20.11.2021	1.2.3.2
1.2.3.4 Планування OBS	2d	21.11.2021	22.11.2021	1.2.3.3
1.2.3.5 Складання календарного плану	2d	23.11.2021	24.11.2021	1.2.3.4
1.2.3.6 Визначення ризиків	2d	25.11.2021	26.11.2021	1.2.3.5
1.3 ☐ Реалізація	18d	27.11.2021	14.12.2021	1.2
1.3.1 Моделювання мобільного додатку	2d	27.11.2021	28.11.2021	1.2
1.3.2 Проектування мобільного додатку	5d	29.11.2021	03.12.2021	1.3.1
1.3.3 Розробка додатку, написання коду	6d	04.12.2021	09.12.2021	1.3.2
1.3.4 ☐ Тестування	5d	10.12.2021	14.12.2021	1.3.3
1.3.4.1 Функціональне тестування	3d	10.12.2021	12.12.2021	1.3.3
1.3.4.2 Нефункціональне тестування	2d	13.12.2021	14.12.2021	1.3.4.1
1.4 ☐ Впровадження в дію	3d	15.12.2021	17.12.2021	1.3
1.4.1 Написання супровідної документації	2d	15.12.2021	16.12.2021	1.3
1.4.2 Реліз мобільного додатку	1d	17.12.2021	17.12.2021	1.4.1

Рисунок А.4 – Список робіт для побудови діаграми Ганта

Джерело: побудовано автором

Управління ризиками. У контексті управління проєктами та прийняття загальних рішень «ризик» відноситься до потенціалу невизначеності або мінливості. Це може призвести до несприятливих результатів або відхилень від очікуваних результатів. Ризики – це події або умови, які, якщо відбуваються, можуть вплинути на досягнення цілей проєкту. Вони можуть мати як позитивні (можливості), так і негативні (загрози)

наслідки. Управління ризиками передбачає їх виявлення, оцінку, визначення пріоритетів і розробку стратегій для пом'якшення або отримання прибутку від цих невизначеностей, спрямованих на підвищення ймовірності успіху проєкту.

Ідентифікація ризиків є важливим кроком у процесі управління ризиками. Цей процес включає систематичне дослідження та визнання потенційних невизначеностей або подій, які можуть вплинути на успіх проєкту. Під час ідентифікації зацікавлені сторони активно прагнуть виявити та задокументувати всі можливі ризики, які можуть виникнути протягом реалізації проєкту. Це є досить важливим завданням. Її метою є створення всеосяжного переліку потенційних ризиків, який забезпечує основу для подальшого аналізу останніх і розробку ефективних стратегій реагування на них для пом'якшення або використання цих невизначеностей.

У таблиці А.3 надано перелік ризиків даного проєкту. Результати їх оцінки надано у таблиці А.4. Таблиця А.5 представляє шкалу для класифікації ризиків за величиною впливу на проєкт та ймовірністю їх виникнення.

Таблиця А.3 – Ризики проєкту.

№ ризику	Назва (опис) ризику
1	Проблеми сумісності з різними пристроями
2	Проблеми з продуктивністю додатку
3	Невиконання календарного плану
4	Недостатність навичок розробників
5	Неправильна оцінка в масштабі проєкту
6	Порушення прав інтелектуальної власності
7	Некоректне тестування
8	Помилки проєктування

Продовження табл. А.3

№ ризику	Назва (опис) ризику
9	Відсутність доступу до мережі Інтернет
10	Відключення світла

Джерело: побудовано автором

Таблиця А.4 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу.

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив	Ранг
1	Проблеми сумісності з різними пристроями	0,7	0,2	0,14
2	Проблеми з продуктивністю додатку	0,5	0,2	0,1
3	Невиконання календарного плану	0,5	0,2	0,1
4	Недостатність навичок розробників	0,1	0,4	0,04
5	Неправильна оцінка в масштабі проєкту	0,5	0,1	0,05
6	Порушення прав інтелектуальної власності	0,3	0,1	0,03
7	Некоректне тестування	0,1	0,1	0,01
8	Помилки проєктування	0,3	0,2	0,06
9	Відсутність доступу до мережі Інтернет	0,7	0,2	0,14
10	Відключення світла	0,3	0,2	0,06

Джерело: побудовано автором

Таблиця А.5 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Джерело: побудовано автором

Для зниження негативного впливу ризиків на проєкт потрібно виконати планування реагування на них. Для цього проводиться оцінка наслідків впливу на проєкт і розробка відповідних заходів. Аналіз виконується за показниками, які описані в таблиці А.4. У результаті планування заходів реагування на ризики проєкту було отримано матрицю ймовірності виникнення та впливу ризиків (рис. А.5). Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

Ймовірність виникнення ризику	Вплив ризику				
	0,05	0,1	0,2	0,4	0,8
0,9	0,045	0,09	0,18	0,36	0,72
0,7	0,035	0,07	0,14 R1,R9	0,28	0,56
0,5	0,025	0,05 R5	0,10 R2,R3	0,20	0,40
0,3	0,015	0,03 R6	0,06 R8, R10	0,12	0,24
0,1	0,005	0,01	0,02 R7	0,04 R4	0,08

Рисунок А.5 – Матриця ймовірності та впливу

Джерело: побудовано автором

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці А.6. У таблиці А.7 описано ризики та стратегії реагування на кожен з них.

Таблиця А.6 – Шкала оцінювання за рівнем ризику

Оцінка	Тип	Межі	Ризики
1	Прийнятні	$0,005 \leq R \leq 0,05$	4,5,6,7
2	Виправдані	$0,05 \leq R \leq 0,14$	1,2,3,8,9,10
3	Недопустимі	$0,14 \leq R \leq 0,72$	

Джерело: побудовано автором

Таблиця А.7 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_1	Відкритий	Проблеми сумісності з різними пристроями	Висока	Середній	0,14	Зменшення	1. Провести ретельне тестування на різних пристроях і операційних системах. 2. Впровадити принципи адаптивного дизайну.	Оптимізувати роботу додатку на основних типах смартфонів
RS_2	Відкритий	Проблеми з продуктивністю додатку	Середня	Середній	0,10	Зменшення	1. Проведіть тестування продуктивності. 2. Розглянути можливість масштабування з самого початку розробки.	Зайнятися оптимізацією коду
RS_3	Відкритий	Невиконання календарного плану	Середня	Середній	0,10	Зменшення	1. Провести комплексний етап планування проєкту. 2. Додати запасний час для непередбачених проблем.	Регулярно переглядати та за потреби коригувати графіки проєкту.
RS_4	Відкритий	Недостатність навичок розробників	Низька	Високий	0,04	Зменшення	Провести точний аналіз здібностей виконавчого складу.	Надати час на навчання розробників
RS_5	Відкритий	Неправильна оцінка в масштабі проєкту	Середня	Низький	0,05	Зменшення	Чітко визначити обсяг проєкту.	Запровадити процес контролю змін, щоб оцінити та затвердити будь-які зміни обсягу.

Продовження табл. А.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_6	Відкритий	Порушення прав інтелектуальної власності	Низька	Низький	0,03	Передача	Провести ретельні перевірки прав інтелектуальної власності.	Зверніться до юридичних експертів, щоб переконатися в дотриманні законів про авторські права
RS_7	Відкритий	Некоректне тестування	Низька	Середній	0,02	Зменшення	Запровадити комплексний план тестування, включаючи модульне тестування, інтеграційне тестування та тестування прийнятності користувача	Виділити додатковий час на тестування
RS_8	Відкритий	Помилки проєктування	Низька	Середній	0,06	Зменшення	Детально опрацювати функціональні можливості додатку.	Здійснювати проміжний контроль результатів в ході виконання проєкту.
RS_9	Відкритий	Відсутність доступу до мережі Інтернет	Середня	Середній	0,14	Ухилення	Розробка локальних мереж або засобів зв'язку, можливості роботи офлайн та синхронізація даних/або роздача мобільного інтернету.	Швидкий перехід на інші доступні мережі.
RS_10	Відкритий	Відключення світла	Середня	Середній	0,06	Ухилення	Встановлення генераторів або резервних джерел живлення, автоматичне резервне копіювання даних.	Активация резервного живлення або перенесення роботи на альтернативні джерела живлення.

Джерело: побудовано автором

ДОДАТОК Б

Лістинг файлів мобільного додатку

Лістинг Activity

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragment_container_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="50dp">
    </androidx.fragment.app.FragmentContainerView>

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragment_settings"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginVertical="50dp"/>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_alignParentBottom="true"
        android:background="@drawable/main_border"
    >

    <ImageView
        android:id="@+id/home_view"
        android:layout_width="80dp"
        android:layout_height="80dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/ic_home_page_foreground" />
```

```

</androidx.constraintlayout.widget.ConstraintLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="@drawable/main_top_border">

    <TextView
        android:id="@+id/choose_template"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:text="PresentMaker"
        android:layout_marginStart="20dp"
        android:textSize="60px"
        />

    <ImageView
        android:id="@+id/settings_view"
        android:layout_width="50dp"
        android:layout_height="50dp"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_alignParentEnd="true"
        app:srcCompat="@drawable/ic_settings_foreground" />

</RelativeLayout>

</RelativeLayout>

```

Лістинг фрагменту перегляду шаблону презентації

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools">

    <RelativeLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        >

        <androidx.recyclerview.widget.RecyclerView

```

```

        android:id="@+id/review_template_recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"

app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
    tools:listitem="@layout/apartment_template_1_main_slide"
    android:layout_marginBottom="50dp"
/>

<Button
    android:id="@+id/review_template_done_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:background="@drawable/review_template_done_button_style"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="70dp"
    android:text="Choose template"
    android:textAlignment="center"
/>

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Лістинг головного слайду шаблону

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="0dp"
    app:cardUseCompatPadding="true"
    app:contentPadding="0dp">

    <RelativeLayout
        android:id="@+id/layout"
        android:layout_width="1080px"
        android:layout_height="670px"

```

```

android:background="@color/apartment_presentation_color_style_1"
    >

    <ImageView
        android:id="@+id/image"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/apartment_template_main_image" />

    <RelativeLayout
        android:layout_width="600px"
        android:layout_height="300px"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="250px"

android:background="@color/apartment_presentation_color_style_1"
    android:alpha="0.5"
    />

    <TextView
        android:id="@+id/apartment_template_1_main_slide_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1-room apartment"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"
        android:textColor="@color/black"
        android:textSize="60px"
        android:textStyle="bold"

    />

    <TextView

android:id="@+id/apartment_template_1_main_slide_company_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Company name"
        android:layout_centerHorizontal="true"

android:layout_below="@id/apartment_template_1_main_slide_title"
        android:textColor="#000000"
        android:textSize="50px"
    />

    <TextView

```

```

        android:id="@+id/apartment_template_1_main_slide_pib"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:layout_below="@+id/apartment_template_1_main_slide_company_name"

        android:layout_centerHorizontal="true"
        android:layout_marginTop="10px"
        android:text="Name Last name"
        android:textColor="#000000"
        android:textSize="40px" />

</RelativeLayout>

</androidx.cardview.widget.CardView>

```

Лістинг модулю взаємодії з базою даних

```

package com.projectmaker;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;

import com.projectmaker.model.Question;

import java.util.ArrayList;

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DB_NAME = "project_maker_db";
    private static final int DB_VERSION = 3;

    private static final String TABLE_NAME = "questions";

    public DatabaseHelper(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {

        String homeTemplateQuestionsTable = "DROP TABLE IF EXISTS " +
        TABLE_NAME;
    }

```

```

        sqLiteDatabase.execSQL(homeTemplateQuestionsTable);

        homeTemplateQuestionsTable = "CREATE TABLE " + TABLE_NAME +
"(" +
        "id INTEGER PRIMARY KEY AUTOINCREMENT, " +
        "template VARCHAR, " +
        "block VARCHAR, " +
        "title VARCHAR, " +
        "text VARCHAR, " +
        "checkbox BOOLEAN, " +
        "quantity INTEGER, " +
        "image BLOB, " +
        "is_text BOOLEAN, " +
        "is_checkbox BOOLEAN, " +
        "is_quantity BOOLEAN, " +
        "is_image BOOLEAN)";

        sqLiteDatabase.execSQL(homeTemplateQuestionsTable);

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int
i1) {

        String homeTemplateQuestionsTable = "DROP TABLE IF EXISTS " +
TABLE_NAME;

        sqLiteDatabase.execSQL(homeTemplateQuestionsTable);

        onCreate(sqLiteDatabase);

    }

    public ArrayList<String> selectBlocks(String template){

        SQLiteDatabase db = getWritableDatabase();
        Cursor c = db.rawQuery("SELECT DISTINCT(block) FROM " +
TABLE_NAME + " WHERE template = ?", new String[]{template});
        ArrayList<String> blocks = new ArrayList<>();
        if (c.moveToFirst()){
            do {
                blocks.add(c.getString(0));
            } while(c.moveToNext());
        }
        c.close();
        db.close();
        return blocks;
    }

```

```

    }

    public String selectQuestionText(String template, String title){

        SQLiteDatabase db = getWritableDatabase();
        String[] args = new String[]{template, title};
        Cursor c = db.rawQuery("SELECT text FROM " + TABLE_NAME + "
WHERE template = ? AND title = ?", args);
        String text = "";

        if (c.moveToFirst()){
            do {
                text = c.getString(0);
            } while(c.moveToNext());
        }
        c.close();
        db.close();
        return text;
    }

    public int selectQuestionQuantity(String template, String title){

        SQLiteDatabase db = getWritableDatabase();
        String[] args = new String[]{template, title};
        Cursor c = db.rawQuery("SELECT quantity FROM " + TABLE_NAME +
" WHERE template = ? AND title = ?", args);
        int quantity = 0;

        if (c.moveToFirst()){
            do {
                quantity = c.getInt(0);
            } while(c.moveToNext());
        }
        c.close();
        db.close();
        return quantity;
    }

    public ArrayList<Question> selectQuestionInBlock(String template,
String block){

        SQLiteDatabase db = getWritableDatabase();
        Cursor c = db.rawQuery("SELECT * FROM " + TABLE_NAME + "
WHERE template = '"+ template +"' AND block = '" + block + "'",
null);
        ArrayList<Question> templateQuestions = new ArrayList<>();

        if (c.moveToFirst()){
            do {

```



```

        Question question = new Question(
            c.getInt(0),
            c.getString(1),
            c.getString(2),
            c.getString(3),
            c.getString(4),
            (boolean) ((c.getInt(5)) == 1),
            c.getInt(6),
            c.getBlob(7),
            (boolean) ((c.getInt(8)) == 1),
            (boolean) ((c.getInt(9)) == 1),
            (boolean) ((c.getInt(10)) == 1),
            (boolean) ((c.getInt(11)) == 1)
        );
        templateQuestions.add(question);
    } while(c.moveToNext());
}
c.close();
db.close();
return templateQuestions;
}

public ArrayList<Question> selectTemplate(String template){

    SQLiteDatabase db = getWritableDatabase();
    Cursor c = db.rawQuery("SELECT * FROM " + TABLE_NAME + "
WHERE template = '"+ template + "'", null);
    ArrayList<Question> templateQuestions = new ArrayList<>();

    if (c.moveToFirst()){
        do {
            Question question = new Question(
                c.getInt(0),
                c.getString(1),
                c.getString(2),
                c.getString(3),
                c.getString(4),
                (boolean) ((c.getInt(5)) == 1),
                c.getInt(6),
                c.getBlob(7),
                (boolean) ((c.getInt(8)) == 1),
                (boolean) ((c.getInt(9)) == 1),
                (boolean) ((c.getInt(10)) == 1),
                (boolean) ((c.getInt(11)) == 1)
            );
            templateQuestions.add(question);
        } while(c.moveToNext());
    }
    c.close();
}

```

```

        db.close();
        return templateQuestions;
    }

    public boolean updateText(String template, String title, String
text){

        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();

        values.put("text", text);
        String[] args = new String[]{title, template};

        db.update(TABLE_NAME, values, " title = ? AND template = ?",
args);
        db.close();

        return true;
    }

    public boolean updateQuantity(String template, String title, int
quantity){

        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();

        values.put("quantity", quantity);
        String[] args = new String[]{title, template};

        db.update(TABLE_NAME, values, " title = ? AND template = ?",
args);
        db.close();

        return true;
    }

    public void updateImage( String template, String title, byte[]
image) throws SQLException {

        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();

        values.put("image", image);
        String[] args = new String[]{title, template};

        db.update(TABLE_NAME, values, " title = ? AND template = ?",
args);
        db.close();
    }

```

```

    }

    public void updateCheckbox( String template, String title,boolean
checkbox) throws SQLiteException {

        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();

        values.put("checkbox", checkbox);
        String[] args = new String[]{title, template};

        db.update(TABLE_NAME, values, " title = ? AND template = ?",
args);
        db.close();

    }

    public boolean addRow(String template, String block, String
title, String text, boolean checkbox, int quantity, byte[] image,
boolean isText, boolean isChecked, boolean isQuantity, boolean
isImage){

        SQLiteDatabase db = getWritableDatabase();

        ContentValues contentValues = new ContentValues();

        contentValues.put("template", template);
        contentValues.put("block", block);
        contentValues.put("title", title);
        contentValues.put("text", text);
        contentValues.put("checkbox", checkbox);
        contentValues.put("quantity", quantity);
        contentValues.put("image", image);
        contentValues.put("is_text", isText);
        contentValues.put("is_checkbox", isChecked);
        contentValues.put("is_quantity", isQuantity);
        contentValues.put("is_image", isImage);

        db.insert(TABLE_NAME, null, contentValues);
        db.close();
        return true;

    }

    public boolean dropTable(){

        SQLiteDatabase db = getWritableDatabase();

        db.delete(TABLE_NAME, null, null);

```

```

        db.close();
        return true;
    }
}

```

Лістинг модулю взаємодії з ChatGPT

```

package com.projectmaker;

import android.app.Activity;
import android.content.Context;
import android.os.Handler;

import androidx.annotation.NonNull;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.TimeUnit;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class ChatGptApi {

    public static final MediaType JSON =
    MediaType.get("application/json; charset=utf-8");

    Activity activity;
    private String answer;

    public ChatGptApi(Activity activity) {
        this.activity = activity;
    }

    public String callAPI(String question) {

        System.out.println(question);
    }
}

```

```

JSONObject jsonBody = new JSONObject();
try {
    jsonBody.put("model", "gpt-3.5-turbo-instruct");
    jsonBody.put("prompt", question);
    jsonBody.put("max_tokens", 100);
    jsonBody.put("temperature", 0);
} catch (JSONException e) {
    throw new RuntimeException(e);
}
RequestBody body = RequestBody.create(jsonBody.toString(),
JSON);
Request request = new Request.Builder()
    .url("https://api.openai.com/v1/completions")
    .header("Authorization", "Bearer sk-
pzGIZ5X8I6N5ifYg6udAT3BlbkFJZac7ALlCufn6gZCrLGpw")
    .post(body)
    .build();

CountDownLatch countDownLatch = new CountDownLatch(1);

OkHttpClient client = new OkHttpClient();

client.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(@NonNull Call call, @NonNull
IOException e) {

        answer = "Failed to load response in call";
        countDownLatch.countDown();

    }

    @Override
    public void onResponse(@NonNull Call call, @NonNull
Response response) throws IOException {

        if(response.isSuccessful()){

            JSONObject jsonObject;
            try {
                jsonObject = new
JSONObject(response.body().string());
                JSONArray jsonArray =
jsonObject.getJSONArray("choices");
                String result =
jsonArray.getJSONObject(0).getString("text");

                int i = 1;

```

```

        while (true){
            if (result.charAt(0) == '\n') {
                result = result.substring(1);
            } else {
                answer = result;
                break;
            }
        }

    } catch (JSONException e) {
        throw new RuntimeException(e);
    }

    }else{

        answer = "Failed to load response";

    }

    response.body().close();

    countdownLatch.countDown();
}
});

try {
    countdownLatch.await();
} catch (InterruptedException e) {
    throw new RuntimeException(e);
}

System.out.println(answer);
return answer;

}

}

```

Лістинг модулю холдера головного слайду

```

package com.projectmaker.templateHolders.apartmentHolder;

import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

```

```

import com.projectmaker.MainActivity;
import com.projectmaker.R;
import com.projectmaker.fragment.ChooseSlideFragment;
import com.projectmaker.fragment.EditPresentationFragment;
import com.projectmaker.fragment.EditTextFragment;
import com.projectmaker.templateModel.apartmentTemplate.MainSlide;

public class ApartmentTemplatelMainSlideViewHolder extends
com.projectmaker.templateHolders.TemplateBasesHolder<MainSlide> {

    private ImageView image;
    private TextView title, company, pib;

    private View view;
    boolean generate;

    public ApartmentTemplatelMainSlideViewHolder(View itemView,
boolean generate) {
        super(itemView);
        view = itemView;
        this.generate = generate;
    }

    @Override
    public void bind(MainSlide mainSlide) {

        image = itemView.findViewById(R.id.image);
        image.setImageBitmap(mainSlide.getImage());
        for(int i = EditPresentationFragment.image.size()-1; i >=0;
i--) {
            EditPresentationFragment.image.remove(i);
        }
        EditPresentationFragment.image.add(image);
        for(int i = ChooseSlideFragment.imageList.size()-1; i >=0; i-
-) {
            ChooseSlideFragment.imageList.remove(i);
        }
        ChooseSlideFragment.imageList.add(image);
        title =
itemView.findViewById(R.id.apartment_template_1_main_slide_title);
        title.setText(mainSlide.getTitle());
        company =
itemView.findViewById(R.id.apartment_template_1_main_slide_company_na
me);
        company.setText(mainSlide.getCompany());
        pib =
itemView.findViewById(R.id.apartment_template_1_main_slide_pib);
        pib.setText(mainSlide.getPib());
    }
}

```

```

if(generate){

    title.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            moveToEditTextFragment(title);
        }
    });

    company.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            moveToEditTextFragment(company);
        }
    });

    pib.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            moveToEditTextFragment(pib);
        }
    });
}

}

private void moveToEditTextFragment(Textview textView){

    generate = false;
    EditTextFragment.slide = getView();
    EditTextFragment.text = textView;

    MainActivity.fragmentManager.beginTransaction()
        .replace(R.id.fragment_container_view,
EditTextFragment.class, null)
        .setReorderingAllowed(true)
        .commit();

}

public View getView() {
    return view;
}

}

```


Лістинг моделі головного слайду

```
package com.projectmaker.templateModel.apartmentTemplate;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

public class MainSlide {

    private Bitmap image;

    private String title, company, pib;

    public MainSlide(Bitmap image, String title, String company,
String pib) {
        this.image = image;
        this.title = title;
        this.company = company;
        this.pib = pib;
    }

    public Bitmap getImage() {
        return image;
    }

    public void setImage(Bitmap image) {
        this.image = image;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getCompany() {
        return company;
    }

    public void setCompany(String company) {
        this.company = company;
    }

    public String getPib() {
        return pib;
    }
}
```

```

    public void setPib(String pib) {
        this.pib = pib;
    }
}

```

Лістинг фрагменту домашньої сторінки

```

package com.projectmaker.fragment;

import android.os.Bundle;
import android.view.View;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.RecyclerView;

import com.projectmaker.R;
import com.projectmaker.adapter.TemplateAdapter;
import com.projectmaker.model.Template;

import java.util.ArrayList;

public class HomeFragment extends Fragment {
    RecyclerView mRecyclerView;

    public HomeFragment() {
        super(R.layout.home_fragment);
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

        super.onViewCreated(view, savedInstanceState);

        mRecyclerView = view.findViewById(R.id.list_view);

        TemplateAdapter adapter = new
TemplateAdapter(view.getContext(), getAllTemplates());
        mRecyclerView.setAdapter(adapter);
    }

    private ArrayList<Template> getAllTemplates() {

        ArrayList<Template> arrayList = new ArrayList<>();

```

```

        arrayList.add(new Template(R.drawable.apartment_template));
        arrayList.add(new Template(R.drawable.house_template));
        arrayList.add(new Template(R.drawable.office_template));

        return arrayList;
    }
}

```

Лістинг адаптера домашньої сторінки

```

package com.projectmaker.adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.RecyclerView;

import com.projectmaker.fragment.EditPresentationFragment;
import com.projectmaker.MainActivity;
import com.projectmaker.R;
import com.projectmaker.fragment.ReviewTemplateFragment;
import com.projectmaker.model.Template;

import java.util.ArrayList;

public class TemplateAdapter extends
RecyclerView.Adapter<TemplateAdapter.Holder>{

    public static int template_number;

    private Context context;
    private ArrayList<Template> arrayList;

    public TemplateAdapter(Context context, ArrayList<Template>
arrayList) {

        this.context = context;
        this.arrayList = arrayList;
    }
}

```

```

    }

    @NonNull
    @Override
    public Holder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {

        View view =
LayoutInflater.from(context).inflate(R.layout.template_button,
parent, false);
        return new Holder(view);

    }

    @Override
    public void onBindViewHolder(@NonNull Holder holder, final int
position) {

        Template template = arrayList.get(position);

        holder.templateButton.setClipToOutline(true);
        holder.templateButton.setImageResource(template.getImage());
        holder.templateButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

                template_number = position;
                EditPresentationFragment.lastSlide = 0;

                ReplaceFragment.replaceFragment();

            }
        });
    }

    @Override
    public int getItemCount() {
        return arrayList.size();
    }

    class Holder extends RecyclerView.ViewHolder{

        ImageView templateButton;

        public Holder(@NonNull View itemView) {

            super(itemView);

```

```

        templateButton =
itemView.findViewById(R.id.template_button_example);

    }

}

private static class ReplaceFragment extends AppCompatActivity{

    public static void replaceFragment(){

        MainActivity.fragmentManager.beginTransaction()
            .replace(R.id.fragment_container_view,
ReviewTemplateFragment.class, null)
            .setReorderingAllowed(true)
            .commit();

    }

}

}

```

Лістинг фрагменту редагування презентації

```

public class EditPresentationFragment extends Fragment {

    private final int GALLERY_REQ_CODE = 1000;

    private CustomViewPager page;
    private Button nextSlideButton, prevSlideButton, doneButton,
backButton, changeImageButton, addSlide, removeSlide;

    TextView mainStyle, style1, style2;

    public static ArrayList<ImageView> image = new ArrayList<>();

    private boolean thisFragment = false;

    public static int lastSlide = 0;

    public EditPresentationFragment() {
        super(R.layout.edit_presentation_fragment);
    }

    @Override

```

```

    public void onCreateView(@NonNull View view, @Nullable Bundle
savedInstanceState) {

        super.onCreateView(view, savedInstanceState);

        page = view.findViewById(R.id.viewPage);

        nextSlideButton =
view.findViewById(R.id.edit_presentation_fragment_next_button);
        prevSlideButton =
view.findViewById(R.id.edit_presentation_fragment_prev_button);

        thisFragment = true;

        doneButton =
view.findViewById(R.id.edit_presentation_fragment_done_button);
        doneButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                thisFragment = false;
                SavePresentationFragment.withSaving = true;

                MainActivity.fragmentManager.beginTransaction()
                    .replace(R.id.fragment_container_view,
SavePresentationFragment.class, null)
                    .setReorderingAllowed(true)
                    .commit();

            }
        });

        backButton =
view.findViewById(R.id.edit_presentation_fragment_back_button);
        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                thisFragment = false;
                SavePresentationFragment.withSaving = false;

                MainActivity.fragmentManager.beginTransaction()
                    .replace(R.id.fragment_container_view,
SavePresentationFragment.class, null)
                    .setReorderingAllowed(true)
                    .commit();

            }
        });
    }
}

```

```

    });

    changeImageButton =
view.findViewById(R.id.edit_presentation_fragment_change_image);

    changeImageButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(thisFragment) {
                Intent iGallery = new Intent(Intent.ACTION_PICK);

iGallery.setData(MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                startActivityForResult(iGallery,
GALLERY_REQ_CODE);
            }
        }
    });

    mainStyle = view.findViewById(R.id.main_style);
    mainStyle.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            for(int i = 0; i < MainActivity.slideList.size();
i++) {

MainActivity.slideList.get(i).findViewById(R.id.layout).setBackground
Color(getResources().getColor(R.color.apartment_presentation_color_st
yle_1));
            }
        }
    });

    style1 = view.findViewById(R.id.style_1);
    style1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            for(int i = 0; i < MainActivity.slideList.size();
i++) {

MainActivity.slideList.get(i).findViewById(R.id.layout).setBackground
Color(getResources().getColor(R.color.apartment_presentation_color_st
yle_2));
            }
        }
    });

    style2 = view.findViewById(R.id.style_2);
    style2.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View view) {
            for(int i = 0; i < MainActivity.slideList.size();
i++) {

MainActivity.slideList.get(i).findViewById(R.id.layout).setBackground
Color(getResources().getColor(R.color.apartment_presentation_color_st
yle_3));
                }
            }
        });

        addSlide =
view.findViewById(R.id.edit_presentation_fragment_add_slide);

        addSlide.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ChooseSlideFragment.slideNumber = lastSlide;

                MainActivity.fragmentManager.beginTransaction()
                    .replace(R.id.fragment_container_view,
ChooseSlideFragment.class, null)
                    .setReorderingAllowed(true)
                    .commit();

            }
        });

        removeSlide =
view.findViewById(R.id.edit_presentation_fragment_remove_slide);

        removeSlide.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                MainActivity.slideList.remove(lastSlide);
                image.remove(lastSlide);

                MainActivity.fragmentManager.beginTransaction()
                    .remove(EditPresentationFragment.this)
                    .replace(R.id.fragment_container_view,
EditPresentationFragment.class, null)
                    .setReorderingAllowed(true)
                    .commit();

            }
        });

```



```

        EditPresentationAdapter editPresentationAdapter = new
EditPresentationAdapter(getContext(), MainActivity.slideList);
        page.setAdapter(editPresentationAdapter);
        page.setCurrentItem(lastSlide);

        nextSlideButton.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {

                if (page.getCurrentItem() <
MainActivity.slideList.size() - 1) {
                    page.setCurrentItem(page.getCurrentItem() + 1);
                    lastSlide = page.getCurrentItem();

                    if(image.get(lastSlide) == null){
                        changeImageButton.setVisibility(View.GONE);
                    } else {

changeImageButton.setVisibility(View.VISIBLE);
                        }
                    }

                });

        prevSlideButton.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {

                if (page.getCurrentItem() > 0) {
                    page.setCurrentItem(page.getCurrentItem() - 1);
                    lastSlide = page.getCurrentItem();

                    if(image.get(lastSlide) == null){
                        changeImageButton.setVisibility(View.GONE);
                    } else {

changeImageButton.setVisibility(View.VISIBLE);
                        }
                    }

                });
        }
    }
}

```

```

    public void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(resultCode==RESULT_OK){
            if(requestCode==GALLERY_REQ_CODE){

                System.out.println(lastSlide);
                Uri imageUri = data.getData();
                image.get(lastSlide).setImageURI(imageUri);

            }
        }

    }
}

```

Лістинг моделі питання

```

package com.projectmaker.model;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

import java.util.BitSet;

public class Question {

    private int id;
    private String template;
    private String block;
    private String title;
    private String text;
    private boolean questionCheckbox;
    private int quantity;
    private byte[] image;
    private boolean isText;
    private boolean isQuantity;

    private boolean isCheckbox;
    private boolean isImage;

    public Question(int id, String template, String block, String
title, String text, boolean questionCheckbox, int quantity, byte[]

```

```
image, boolean isText, boolean isCheckbox, boolean isQuantity,
boolean isImage) {
    this.id = id;
    this.template = template;
    this.block = block;
    this.title = title;
    this.text = text;
    this.questionCheckbox = questionCheckbox;
    this.quantity = quantity;
    this.image = image;
    this.isText = isText;
    this.isQuantity = isQuantity;
    this.isCheckbox = isCheckbox;
    this.isImage = isImage;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getTemplate() {
    return template;
}

public void setTemplate(String template) {
    this.template = template;
}

public String getBlock() {
    return block;
}

public void setBlock(String block) {
    this.block = block;
}

public boolean isText() {
    return isText;
}

public void setText(boolean text) {
    isText = text;
}

public String getTitle() {
```

```
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public String getText() {
        return text;
    }

    public void setText(String text) {
        this.text = text;
    }

    public boolean isQuantity() {
        return isQuantity;
    }

    public void setQuantity(boolean quantity) {
        isQuantity = quantity;
    }

    public boolean isQuestionCheckbox() {
        return questionCheckbox;
    }

    public void setQuestionCheckbox(boolean questionCheckbox) {
        this.questionCheckbox = questionCheckbox;
    }

    public boolean isImage() {
        return isImage;
    }

    public void setImage(boolean image) {
        isImage = image;
    }

    public Bitmap getImage() {
        return BitmapFactory.decodeByteArray(image, 0, image.length);
    }
}
```

```

    }

    public void setImage(byte[] image) {
        this.image = image;
    }

    public boolean isChecked() {
        return isChecked;
    }

    public void setCheckbox(boolean checkbox) {
        isChecked = checkbox;
    }
}

```

Лістинг модулю збереження презентації

```

package com.projectmaker.fragment;

import static android.Manifest.permission.READ_EXTERNAL_STORAGE;
import static android.Manifest.permission.WRITE_EXTERNAL_STORAGE;

import android.content.Context;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.pdf.PdfDocument;
import android.hardware.display.DisplayManager;
import android.os.Bundle;
import android.os.Environment;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.Display;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;

```

```

import com.projectmaker.DatabaseHelper;
import com.projectmaker.MainActivity;
import com.projectmaker.R;
import com.projectmaker.adapter.TemplateAdapter;
import com.projectmaker.model.Question;
import com.projectmaker.templateModel.apartmentTemplate.ReviewsSlide;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;

public class SavePresentationFragment extends Fragment {

    private static final int PERMISSION_REQUEST_CODE = 200;

    public static boolean withSaving = false;

    View mView;
    Button saveInPDF, backToQuastion, editPresentation;

    String[] permissionsList = {READ_EXTERNAL_STORAGE,
WRITE_EXTERNAL_STORAGE};

    public SavePresentationFragment() {

        super(R.layout.save_presentation_fragment);

    }

    @Override
    public void onCreateView(@NonNull View view, @Nullable Bundle
savedInstanceState) {

        super.onCreateView(view, savedInstanceState);

        mView = view;

        saveInPDF =
view.findViewById(R.id.save_presentation_fragment_save_in_pdf_button)
;
        backToQuastion =
view.findViewById(R.id.save_presentation_fragment_back_to_quastion_bu
tton);
        editPresentation =
view.findViewById(R.id.save_presentation_fragment_edit_presentation_b
utton);

```



```

    });

}

private boolean checkPermission() {

    // checking of permissions.
    int permission1 =
ContextCompat.checkSelfPermission(getContext(),
WRITE_EXTERNAL_STORAGE);
    int permission2 =
ContextCompat.checkSelfPermission(getContext(),
READ_EXTERNAL_STORAGE);
    return permission1 == PackageManager.PERMISSION_GRANTED &&
permission2 == PackageManager.PERMISSION_GRANTED;
}

public void convertXmlToPdf() {

    int widthPixels = 1080;
    int heightPixels = 670;

    ArrayList<View> viewArrayList = new ArrayList<>();

    // Inflate the XML layout file
    if (withSaving) {

        // Create a new PdfDocument instance
        PdfDocument document = new PdfDocument();

        for (int i = 0; i < MainActivity.slideList.size(); i++) {

MainActivity.slideList.get(i).measure(View.MeasureSpec.makeMeasureSpec(
widthPixels, View.MeasureSpec.EXACTLY),
View.MeasureSpec.makeMeasureSpec(heightPixels,
View.MeasureSpec.EXACTLY));
            Log.d("mylog", "Width Now " +
MainActivity.slideList.get(i).getMeasuredWidth());
            MainActivity.slideList.get(i).layout(0, 0,
widthPixels, heightPixels);

            // Obtain the width and height of the view
            int viewWidth =
MainActivity.slideList.get(i).getMeasuredWidth();
            int viewHeight =
MainActivity.slideList.get(i).getMeasuredHeight();

```



```

        // Create a PageInfo object specifying the page
attributes
        PdfDocument.PageInfo pageInfo = new
PdfDocument.PageInfo.Builder(viewWidth, viewHeight, i + 1).create();

        // Start a new page
PdfDocument.Page page = document.startPage(pageInfo);

        // Get the Canvas object to draw on the page
Canvas canvas = page.getCanvas();

        // Create a Paint object for styling the view
Paint paint = new Paint();
paint.setColor(Color.WHITE);

        // Draw the view on the canvas
MainActivity.slideList.get(i).draw(canvas);

        // Finish the page
document.finishPage(page);

    }

    // Specify the path and filename of the output PDF file
File downloadsDir =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_D
OWNLOADS);

    DatabaseHelper db = new DatabaseHelper(getContext());
    ArrayList<Question> questions =
db.selectQuestionInBlock(QuestionsFormFragment.template, "Main
Block");

    String fileName = questions.get(0).getText() + ".pdf";
    File filePath = new File(downloadsDir, fileName);

    try {
        // Save the document to a file
        FileOutputStream fos = new
FileOutputStream(filePath);
        document.writeTo(fos);
        document.close();
        fos.close();
        // PDF conversion successful
        Toast.makeText(getContext(), "Saving Successful",
Toast.LENGTH_LONG).show();
    } catch (IOException e) {
        e.printStackTrace();
        // Error occurred while converting to PDF
    }
}
}}}

```