

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут комп'ютерних систем
Кафедра інформаційних систем

Боровик С.О./Кірпіков Ю.Д./Фокін М.С.
студенти групи ЛАІ-215/АІ-213/АІ-213

ДИСЦИПЛІНА
Об'єктно-орієнтоване програмування

КУРСОВА РОБОТА
Розробка мобільного додатку для ведення заміток

Спеціальність:
122 Комп'ютерні науки

Освітня програма:
Комп'ютерні науки

Керівник:
Годовиченко Микола Анатолійович,
кандидат технічних наук, доцент

Одеса – 2023

ЗМІСТ

Анотація	4
Вступ	5
1 Огляд систем-аналогів та технологій їх розробки.....	5
2 проєктування мобільного додатку для ведення заміток.....	16
3 ПРОГРАМНА РЕАЛІЗАЦІЯ мобільного додатку для ведення заміток.....	28
ВИСНОВКИ.....	46
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47

АНОТАЦІЯ

Курсова робота присвячена розробці мобільного додатку для створення заміток. Додаток призначений для зручного створення та управління користувальницькими замітками на мобільних пристроях. Основною метою роботи є створення простого, ефективного та інтуїтивно зрозумілого інструменту, який дозволяє користувачам легко створювати, редагувати, видаляти та організовувати свої замітки на їх мобільних пристроях. В ході розробки додатку будуть використані сучасні технології та практики розробки мобільних додатків, зокрема мова програмування Kotlin для розробки на платформі Android, бібліотеки Authentication та Realtime Database.

ABSTRACT

The course work is devoted to the development of a mobile application for creating notes. The application is designed to conveniently create and manage user notes on mobile devices. The main goal of the work is to create a simple, efficient and intuitive tool that allows users to easily create, edit, delete and organize their notes on their mobile devices. The application will be developed using modern technologies and practices of mobile application development, including the Kotlin programming language for Android development, the Authentication and the Realtime Database libraries.

ВСТУП

У сучасному світі, де мобільні пристрої стали невід'ємною частиною нашого повсякденного життя, зростає потреба в ефективних та зручних інструментах для організації наших справ, ідей та нагадувань. Одним з найпоширеніших і водночас простих засобів є замітки. Вони дозволяють нам швидко зафіксувати важливу інформацію, зберегти цінні думки та спростити наше щоденне життя.

З урахуванням цієї потреби, було вирішено зосередитися на розробці мобільного додатку для заміток, який дозволить користувачам зручно створювати, організовувати та управляти своїми замітками прямо зі своїх мобільних пристроїв. Мета роботи полягає у створенні інтуїтивно зрозумілого інструменту, який буде доступний на платформі Android, і забезпечуватиме зручний і швидкий доступ до заміток незалежно від місця та часу.

Розробка такого додатку вимагає використання сучасних технологій, програмування та архітектурних принципів, що дозволяють створити стійкий, ефективний і надійний продукт. Додаток повинен мати інтуїтивний інтерфейс, що дозволить користувачам швидко засвоїти його функціонал і легко використовувати його у повсякденному житті.

Враховуючи викладені обставини, актуальною є тема розробки мобільного додатку для створення заміток. Такий додаток дозволяв би створювати та переглядати замітки зі збереженням їх на локальному сховищі, щоб мати можливість перегляді їх при наступних запусках додатку.

Метою курсової роботи є розробка мобільного додатку для управління замітками.

Для досягнення цієї мети необхідно виконати наступні задачі:

- здійснити огляд інших додатків для створення користувацьких заміток;
- проаналізувати та обрати сучасні технології для створення мобільного додатку;

- провести проектування мобільного додатку заміток;
- виконати програмну реалізацію спроектованого мобільного додатку.

1 Огляд СИСТЕМ-АНАЛОГІВ ТА ТЕХНОЛОГІЙ ЇХ РОЗРОБКИ

1.1 Особливості використання мобільних технологій для ведення справ та особистої продуктивності

Мобільні технології відіграють значну роль у поліпшенні особистої продуктивності та веденні справ. Вони надають користувачам широкі можливості доступу до інформації, організації задач і спрощення повсякденних обов'язків. Ось декілька особливостей використання мобільних технологій у цьому контексті:

- мобільний доступ – завдяки мобільним пристроям, ми можемо мати постійний доступ до наших завдань, розкладу, заміток та інших важливих даних незалежно від місця перебування. Це дозволяє нам бути організованими і ефективними навіть у рухливому режимі;
- синхронізація – багато мобільних додатків для ведення справ підтримують синхронізацію даних через хмарні сервіси. Це означає, що ми можемо оновлювати, редагувати або видаляти задачі на одному пристрої і автоматично синхронізувати зміни на всіх інших підключених пристроях. Це забезпечує консистентність даних і уникнення втрати інформації;
- нагадування та сповіщення – мобільні пристрої дозволяють нам встановлювати нагадування та отримувати сповіщення про надходження важливих подій або завдань. Це допомагає нам не пропустити терміни, зустрічі або інші важливі події, що сприяє підвищенню продуктивності та плануванню часу;
- можливості організації – мобільні додатки для ведення справ надають широкі можливості організації завдань, проектів і заміток. Ми можемо

створювати категорії, мітки, теги або групи для легкого сортування і пошуку необхідної інформації. Також, можна використовувати функції пріоритетування, календаря і дедлайнів для кращого керування своїм часом та завданнями;

- мобільні сповіщення – однією з переваг мобільних технологій є можливість отримувати сповіщення навіть без потреби активного відкриття додатку. Це дозволяє нам отримувати нагадування, повідомлення або важливі оновлення безпосередньо на екрані свого мобільного пристрою, що сприяє оперативності і швидкій реакції на події;
- можливості синтезу голосу та розпізнавання письма – деякі мобільні додатки для ведення справ мають функції синтезу голосу та розпізнавання письма. Це дозволяє нам записувати аудіонотатки або набирати текст, використовуючи голосові команди. Це особливо корисно, коли ми знаходимося в русі або не можемо вводити текст руками.

Використання мобільних технологій для ведення справ та особистої продуктивності дозволяє нам бути організованими, керувати часом і завданнями ефективніше, а також забезпечує постійний доступ до нашої важливої інформації. Важливо вибирати додатки, які найкраще відповідають нашим потребам та вимогам, забезпечують зручність використання та надійність.

1.2 Огляд додатків для ведення заміток

Для розробки мобільного додатку веб-ресурсу продуктової крамниці необхідно з'ясувати, які функції повинні бути доступні в ньому. Для цього потрібно дослідити та проаналізувати інші додатки, які також дозволяють вести замітки. Це дозволить встановити особливості роботи цих додатків та з'ясувати позитивні та негативні моменти їх використання. Також цей аналіз допоможе краще зрозуміти потреби користувачів в цій області. Після проведення пошуку в

Інтернеті, були знайдені найбільш популярні та відомі додатки, які будуть використовуватися як аналоги в даній роботі:

- додаток Remember The Milk [3];
- додаток ColorNote [4];

Далі потрібно провести аналіз цих додатків з метою встановлення особливостей їх роботи та з'ясування позитивних та негативних моментів їх використання. Це допоможе краще зрозуміти, які функції повинні бути доступні мобільному додатку для створення заміток, та відповісти на потреби користувачів в цій області.

1.2.1 Додаток Remember The Milk

Додаток Remember The Milk є зручним додатком для ведення справ для зайнятих людей. Ви ніколи не забудете придбати молоко (або що-небудь ще) ще раз (рис. 1.2) [3].

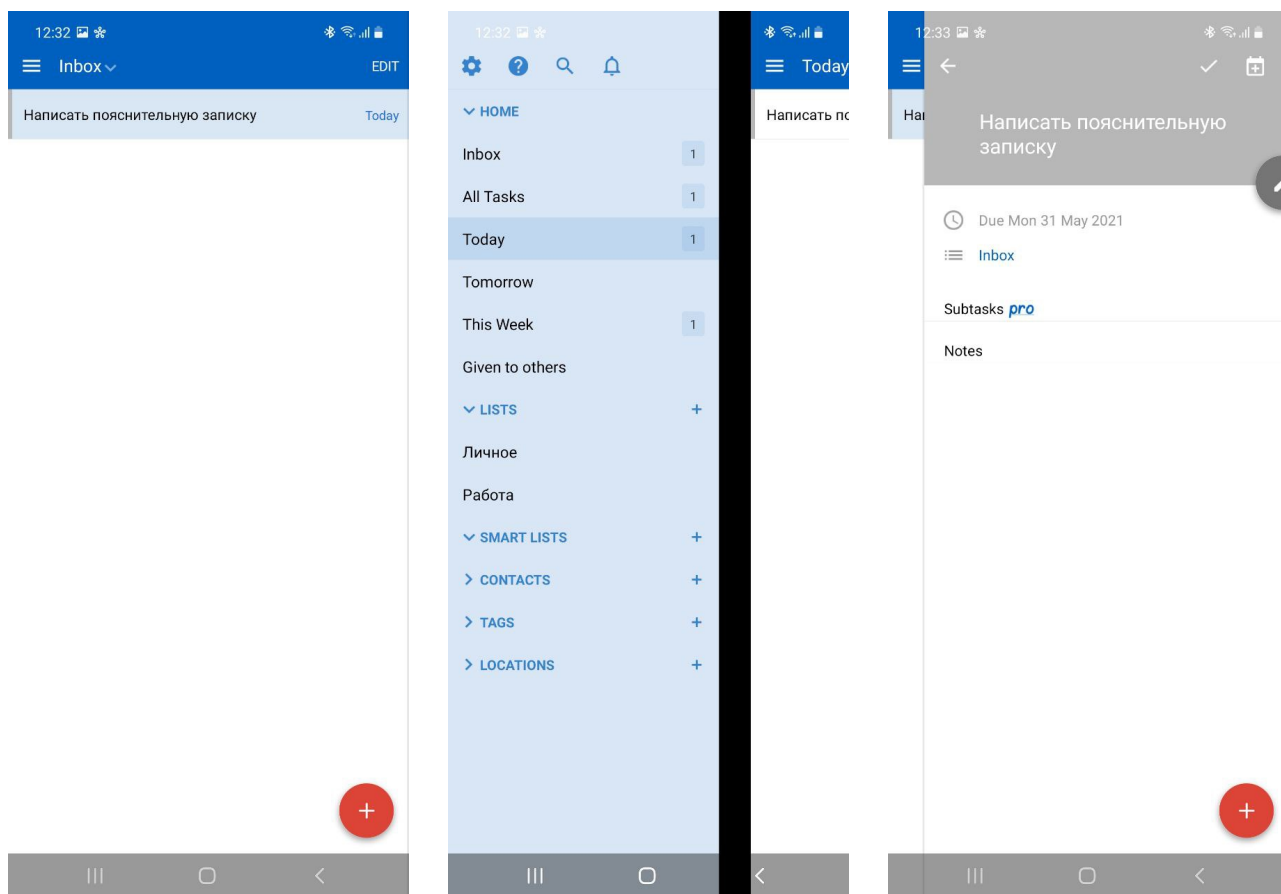


Рисунок 1.2 – Знімки екрану додатку Remember The Milk

Основний функціонал додатку:

- можливість створювати списки справ;
- можливість отримати нагадування через електронну пошту, мобільний пристрій чи популярні месенджери;
- можливість ділитися своїми списками з іншими людьми;
- синхронізація справ на всіх пристроях;
- можливість додавати пріоритети, терміни закінчення, кількість повторів справи, мітки тощо;
- пошук справ та нотаток за текстом, збереження справ та нотаток в розумних списках;

- можливість бачити справи, які фізично розташовані найближче та у найбільш ефективний спосіб.

Платна версія додатку має додатковий функціонал:

- підзадачі – можна розбивати задачі на менші частини;
- можливість ділитися задачами без обмежень;
- додатковий параметр кольору для задач;
- нагадування про необхідність виконання задачі через мобільний пристрій;
- синхронізація з Microsoft Outlook;
- безлімітне сховище для даних.

1.2.2 Додаток ColorNote

ColorNote – це простий блокнот. Він надасть можливість легкого і простого користування блокнотом при написанні заміток, нагадувань, email, повідомлень, переліків справ і покупок (рис 1.3) [4]. З ColorNote створення заміток простіше, ніж з будь-якими іншими блокнотами і органайзерами. Розглянемо можливості додатку:

- організація заміток за кольором;
- віджети стікерів;
- списки справ і покупок;
- організація розкладу в календарі;
- блокування замітки паролем;

- захищене резервне копіювання заміток на SD карту – перед вивантаженням замітки будуть зашифровані за стандартом AES, який використовується банками для захисту даних клієнтів;

- підтримка онлайн-синхронізації і резервного копіювання;

- синхронізація заміток між телефоном і планшетом;

- нагадування в статус-барі;

- нагадування про нотатки;

- можливість ділитися нотатками за допомогою SMS, електронної пошти чи твітеру;

- можливість кольорової організації нотаток.

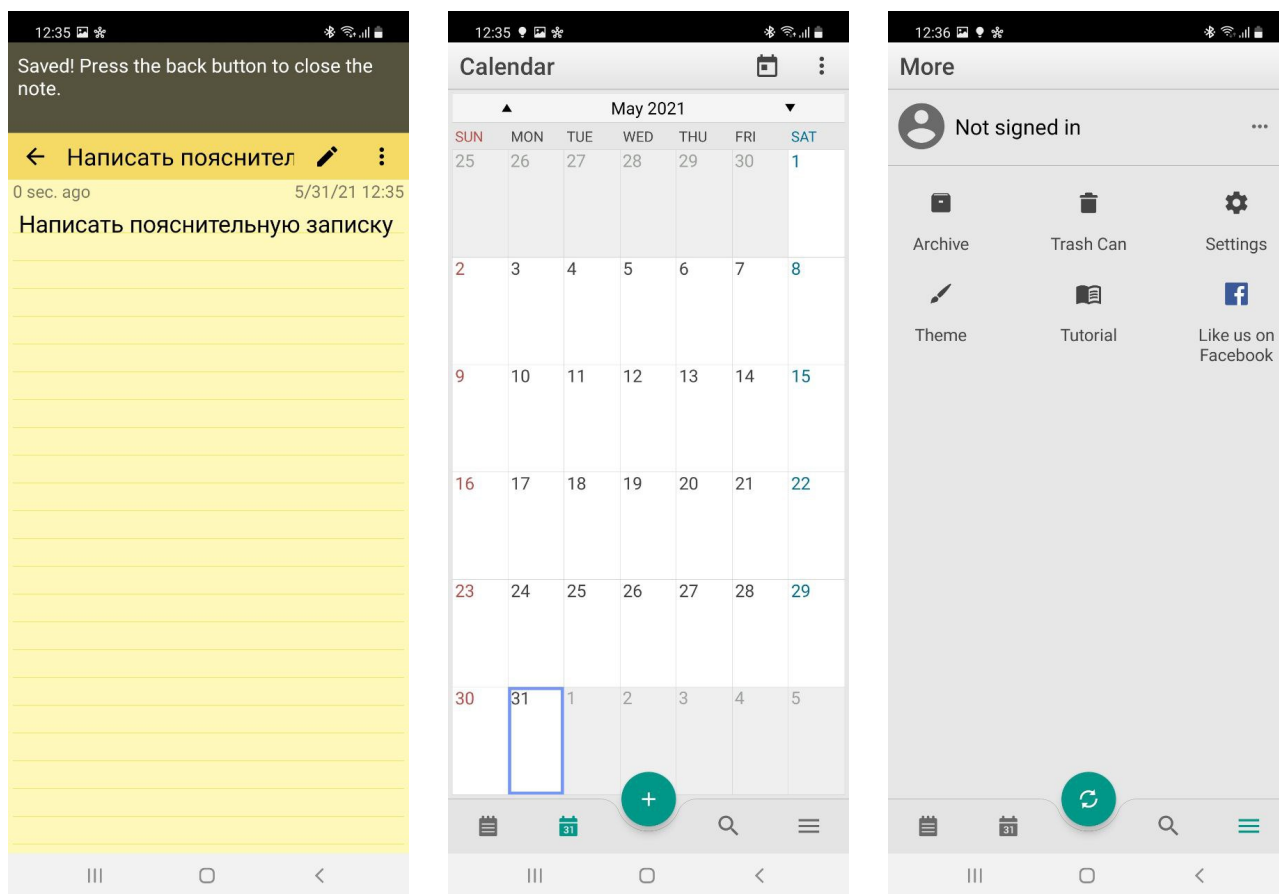


Рисунок 1.3 – Знімки екрану додатку ColorNote

1.3 Формування вимог до основних функцій мобільного додатку

Аналіз додатків для ведення справ та нотаток дозволив виділити їх переваги та недоліки та сформулювати вимоги до власного мобільного додатку.

Основний функціонал мобільного додатку для ведення заміток може включати наступні можливості:

- створення заміток – користувач може створювати нові замітки, вводячи текстовий контент. Додаток повинен забезпечувати простий та зручний спосіб створення нових заміток;
- редагування та форматування – користувач може редагувати вміст заміток, включаючи видалення, редагування або переміщення тексту. Додаток може також надавати можливість форматування тексту, таке як жирний, курсив, маркери списку, нумерація тощо, для зручності організації і вигляду заміток;

- категорії та мітки – додаток може дозволяти користувачу створювати категорії або мітки для заміток, що допомагає організувати їх за певними темами, проектами або контекстами. Це дозволяє швидко знаходити потрібну інформацію і забезпечує більшу структурованість;
- пошук та фільтрація – користувач може шукати замітки за ключовими словами або фільтрувати їх за певними критеріями, такими як дата створення, пріоритет, категорія тощо. Це допомагає знаходити необхідні замітки швидко і ефективно;
- безпека даних – з метою забезпечення конфіденційності та безпеки, додаток може пропонувати захист заміток за допомогою пароллю, відбитка пальця або інших методів аутентифікації. Це гарантує, що тільки авторизовані користувачі можуть отримати доступ до приватних даних.

1.4 Огляд інформаційних технологій для розробки мобільного додатку

1.4.1 Фреймворк Jetpack Compose

Jetpack Compose є сучасним фреймворком для розробки інтерфейсу користувача (UI) для мобільних додатків на платформі Android. Ось деякі переваги та недоліки використання Jetpack Compose:

Розглянемо переваги фреймворку Jetpack Compose:

- декларативний підхід – Jetpack Compose використовує декларативний підхід до побудови UI, що означає, що ви описуєте, як повинен виглядати ваш UI, а не як його побудувати крок за кроком. Це спрощує процес розробки та розуміння коду;

- легше утримувати – завдяки декларативному підходу Jetpack Compose дозволяє писати більш зрозумілий та компактний код. Це полегшує розуміння та утримання кодової бази, особливо великих проєктів;
- живий попередній перегляд – Jetpack Compose надає можливість відразу бачити результати змін у реальному часі без необхідності компіляції або запуску додатка. Це прискорює процес розробки та дозволяє швидко перевірити вигляд та поведінку UI;
- легка навігація – Jetpack Compose надає простий та зрозумілий API для навігації між екранами додатка. Він спрощує реалізацію функціоналу переходу між фрагментами або активностями, забезпечуючи зручну інтеграцію з іншими компонентами додатка;
- багата функціональність – Jetpack Compose має велику кількість вбудованих компонентів та функцій, які полегшують створення різноманітних UI-елементів, включаючи кнопки, тексти, списки, анімації тощо. Він також має добре підтримувану документацію та активну спільноту, що сприяє розробці та розширенню функціональності.

Розглянемо недоліки фреймворку Jetpack Compose:

- іматурність – Jetpack Compose - це нова технологія, яка все ще знаходиться на етапі активного розвитку. Це означає, що вона може мати деякі недоліки та обмеження, а також потребує постійного оновлення для вирішення проблем та додавання нового функціоналу;
- потреба в оновленні – використання Jetpack Compose вимагає оновлення вашого проєкту на новішу версію Android та використання сучасних компонентів, таких як AndroidX. Це може бути вимогливим і вимагати додаткових зусиль для перенесення існуючих проєктів на Jetpack Compose;
- навчання – Jetpack Compose відрізняється від традиційного підходу до розробки Android-інтерфейсу, що означає, що розробникам може

знадобитися деякий час для вивчення нових концепцій та API. Це може бути викликом для команд, які вже працюють зі старішими підходами;

Незважаючи на деякі недоліки – Jetpack Compose швидко розвивається і вже стає все більш популярним фреймворком для розробки інтерфейсу користувача на платформі Android. Він пропонує простоту, ефективність та розширюваність, що робить його привабливим вибором для розробників.

1.4.2 Архітектурні компоненти Android

Використання архітектурних компонентів в Android є важливим аспектом при розробці додатків, оскільки вони допомагають забезпечити структурованість, модульність, перевикористовуваність та тестованість коду. Ось деякі ключові переваги використання архітектурних компонентів.

Масштабованість: Архітектурні компоненти, такі як ViewModel, LiveData і Room, надають підхід, який сприяє масштабуваності додатка. Вони допомагають розділити бізнес-логіку від інтерфейсу користувача і забезпечують зручний спосіб управління даними та станом додатка.

Розподілення обов'язків: Архітектурні компоненти, такі як Model-View-ViewModel (MVVM) або Model-View-Presenter (MVP), допомагають чітко розподілити обов'язки між компонентами додатка. Це полегшує розуміння та утримання кодової бази, а також сприяє зручній співпраці в командному середовищі.

Тестованість: Архітектурні компоненти підтримують тестування додатків, оскільки вони розділяють бізнес-логіку від інтерфейсу користувача. Це дозволяє розробникам легко писати автоматизовані тести для перевірки функціональності та надійності додатка.

Легка утримуваність: Архітектурні компоненти пропонують структуровану організацію коду, що спрощує утримання проекту. Це дозволяє розробникам швидко зорієнтуватися в коді, вносити зміни та вдосконалювати додаток з мінімальними зусиллями.

Перевикористання: Використання архітектурних компонентів сприяє перевикористанню коду. Вони дозволяють виділити загальну функціональність у відокремлені компоненти, які можна використовувати в різних частинах додатка або навіть в інших проектах.

Спільність розробки: Архітектурні компоненти забезпечують зручну інтеграцію та спільну роботу в командному середовищі. Це полегшує спільне управління кодовою базою та співпрацю між розробниками.

Загалом, використання архітектурних компонентів в Android рекомендується для покращення якості, швидкості розробки та легкості утримання додатків. Вони допомагають розділити обов'язки, спростити тестування та забезпечити модульність, що робить їх важливою складовою частиною процесу розробки Android-додатків.

1.5 Висновки до першого розділу

В першому розділі курсової роботи був проведений огляд додатків для створення заміток та технологій їх розробки.

Було розглянуто питання впливу мобільних технологій на задачу ведення справ та особистої продуктивності. Було визначено, що мобільні технології відіграють значну роль у поліпшенні особистої продуктивності та веденні справ. Вони надають користувачам широкі можливості доступу до інформації, організації задач і спрощення повсякденних обов'язків.

Далі був проведений огляд додатків-аналогів, який дозволив визначити основні переваги та недоліки існуючих мобільних додатків для ведення заміток, встановити їх основні функції та можливості. На базі проведеного огляду були визначені основні вимоги до власного веб-ресурсу, а також наданий детальний опис цих вимог.

Був проведений огляд інформаційних технологій для розробки мобільного додатку для ведення заміток. У якості мобільної операційної системи виступає ОС Android. У якості фреймворку для розробки користувацького інтерфейсу було

вирішено обрати фреймворк Jetpack Compose. Залучення цього фреймворку дозволяє використати декларативний підхід до створення користувацького інтерфейсу, що суттєво полегшує процес його створення. У якості технології для розробки архітектури додатку були обрані архітектурні компоненти. Вони допомагають забезпечити структурованість, модульність, перевикористовуваність та тестованість коду.

2 ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВЕДЕННЯ ЗАМІТОК

2.1 Мета та задачі мобільного додатку

Мета мобільного додатку Android для ведення заміток полягає в наданні користувачам зручного та організованого інструменту для зберігання, організації

та керування їх замітками. Головною метою додатку є полегшення процесу ведення особистих або професійних записів, щоб користувачі могли легко зберігати важливу інформацію та нагадування.

Задачі мобільного додатку для ведення заміток можуть включати:

- створення нових заміток – користувачі повинні мати можливість швидко створювати нові записи заміток з використанням тексту, фотографій або голосових нотаток;
- організація заміток – додаток повинен надати можливість створення категорій або міток для заміток, щоб користувачі могли легко групувати та організовувати свої записи;
- редагування та видалення заміток – користувачам необхідно мати можливість змінювати вміст заміток, редагувати або видаляти їх в разі потреби;
- пошук та фільтрація заміток – додаток повинен забезпечувати швидкий доступ до заміток за допомогою пошуку або застосування фільтрів за категоріями, тегами чи ключовими словами;
- синхронізація та резервне копіювання – важливо, щоб додаток забезпечував можливість синхронізувати замітки між різними пристроями користувача і забезпечував регулярне резервне копіювання для запобігання втраті даних;
- сповіщення та нагадування – додаток може надавати можливість встановлення нагадувань та сповіщень для заміток, щоб користувачі не пропустили важливі події або завдання;

Призначення мобільного додатку для ведення заміток полягає в полегшенні організації та керування інформацією для користувачів. Він надає зручний спосіб зберігання і використання заміток, дозволяючи легко доступатися до них, редагувати, організовувати та забезпечувати їх безпеку. Мобільний додаток для ведення заміток сприяє покращенню особистої продуктивності, організації ідей,

плануванню завдань та важливих подій, а також виконанню рутинних завдань швидко та ефективно.

2.2 Визначення функціональних вимог до мобільного додатку

Визначення функціональних вимог є важливим етапом у процесі створення мобільного додатку для ведення заміток. Функціональні вимоги визначають, які конкретні функції та можливості повинен мати додаток. Вони встановлюють чітку спрямованість розробки, допомагають уникнути неоднозначностей та непорозумінь.

Крім того, визначення функціональних вимог дозволяє зосередитися на потребах та вимогах користувачів. Вони допомагають врахувати, які функції та можливості будуть найбільш корисними для користувачів додатку.

Також, функціональні вимоги слугують основою для комунікації між розробниками, дизайнерами та іншими учасниками проекту. Вони допомагають зрозуміти, що саме потрібно реалізувати та які очікувані результати.

Встановлення функціональних вимог дозволяє визначити обсяг роботи та потребу в ресурсах для реалізації додатку. Це допомагає планувати час, бюджет та ресурси проекту ефективно.

Єдиним актором мобільного додатку є актор «користувач»:

- користувач – користувач мобільного додатку. Має доступ до всіх функцій додатку та несе всю відповідальність за його роботу.

З метою визначення користувацьких історій та нефункціональних вимог до веб-ресурсу, було розроблено діаграму сценаріїв використання мобільного додатку (рис. 2.1). Діаграма сценаріїв UML (Unified Modeling Language) - це графічний інструмент для опису функціональної взаємодії між користувачами та системою. Вона складається з акторів, сценаріїв та взаємодії між ними.

Ця діаграма містить основних акторів системи та описує сценарії їх взаємодії з системою. Вона допомагає проаналізувати залежності між акторами та

можливими варіантами використання системи. Також ця діаграма швидко демонструє основні функції системи для розробників.

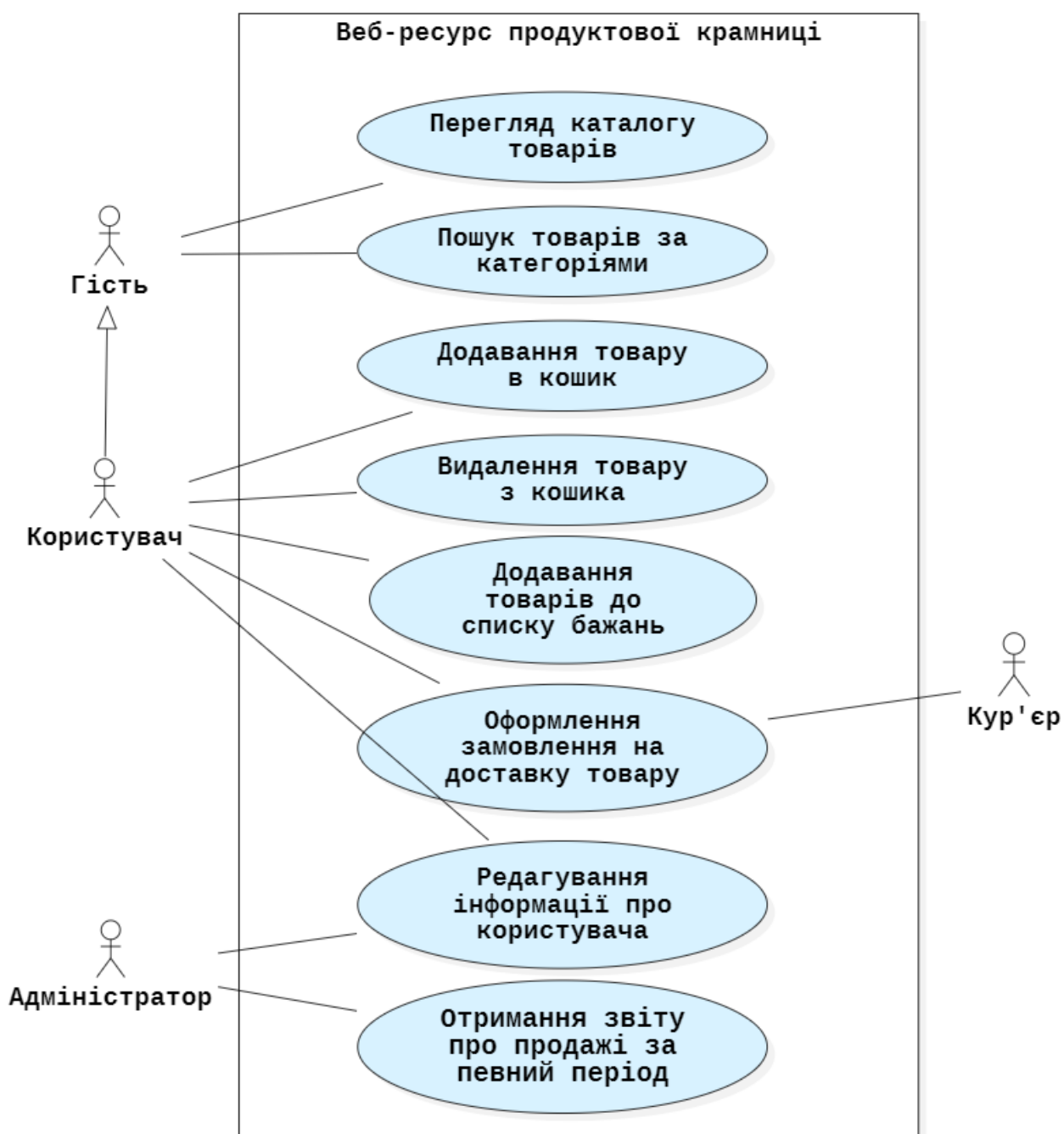


Рисунок 2.1 – Діаграма варіантів використання мобільного додатку

2.3 Формування користувацьких історій мобільного додатку

Проектування діаграми прецедентів дозволяє визначити такі користувацькі історії до мобільного додатку для ведення заміток.

US1 Як користувач, я хочу створити нову замітку, щоб зафіксувати важливу інформацію.

Сценарій користувацької історії виглядає таким чином:

- користувач відкриває додаток для ведення заміток;
- він натискає кнопку «Створити нову замітку»;
- користувач вводить текстове повідомлення або нотатку;
- після завершення введення, користувач натискає кнопку «Зберегти»;
- нова замітка зберігається в додатку і з'являється у списку заміток;

US2 Як користувач, я хочу редагувати вміст замітки, щоб вносити зміни або доповнення до інформації.

Сценарій користувацької історії виглядає таким чином:

- користувач відкриває конкретну замітку;
- він натискає на кнопку «Редагувати» або вибирає опцію редагування;
- користувач вносить необхідні зміни або доповнення до тексту замітки;
- після завершення редагування, користувач натискає кнопку «Зберегти» або подібний елемент, щоб зберегти оновлену версію замітки.

US3 Як користувач, я хочу видаляти замітки, які більше не потрібні, для підтримки чистоти та організованості списку заміток.

Сценарій користувацької історії виглядає таким чином:

- користувач відкриває конкретну замітку або виділяє багато заміток у списку;
- він натискає кнопку «Видалити» або вибирає опцію видалення;
- користувач підтверджує видалення замітки або заміток;

- замітки видаляються з додатку та зникають зі списку заміток.

US4 Як користувач, я хочу синхронізувати свої замітки з обліковим записом, щоб мати доступ до них з різних пристроїв.

Сценарій користувацької історії виглядає таким чином:

- користувач відкриває налаштування облікового запису в додатку;
- він натискає на опцію «Синхронізація» або «Увійти в обліковий запис»;
- користувач вводить свої облікові дані або реєструється новим користувачем;
- після успішної аутентифікації, замітки автоматично синхронізуються з обліковим записом, що дозволяє доступ до них з інших пристроїв.

US5 Як користувач, я хочу додавати зображення до заміток, щоб доповнювати текст ілюстраціями або фотографіями.

Сценарій користувацької історії виглядає таким чином:

- користувач відкриває конкретну замітку або створює нову;
- Він натискає на кнопку «Додати зображення» або вибирає опцію для додавання медіафайлу;
- користувач вибирає зображення зі свого пристрою або робить фотографію в додатку;
- після завершення, зображення додається до замітки і відображається разом з текстом.

US6 Як користувач, я хочу встановити пріоритети для заміток, щоб виділити найважливіші завдання.

Сценарій користувацької історії виглядає таким чином:

- користувач відкриває конкретну замітку або створює нову;
- він натискає на опцію «Встановити пріоритет» або подібний елемент;

- користувач вибирає пріоритетний рівень для замітки (наприклад, високий, середній, низький);
- після збереження, замітка отримує позначення пріоритету, що допомагає користувачеві виділити найважливіші завдання.

2.4 Визначення нефункціональних вимог до мобільного додатку

Після визначення функціональних вимог, необхідно сформулювати нефункціональні вимоги, які висуваються до мобільного додатку для ведення заміток.

NFR1 Версія операційної системи – додаток може підтримувати певний діапазон версій, наприклад, Android 8.0 або новіше.

NFR2 Процесор – мінімальний процесор з відповідною архітектурою (наприклад, ARM або x86), який підтримується операційною системою.

NFR3 Оперативна пам'ять (RAM) – мінімальний обсяг оперативної пам'яті, необхідний для плавної роботи додатку. Зазвичай, це може бути від 1 ГБ і більше.

NFR4 Внутрішня пам'ять – додаток може вимагати певний обсяг внутрішньої пам'яті для збереження даних заміток і налаштувань. Це може бути від кількох мегабайт до кількох гігабайт.

NFR5 Роздільна здатність екрану – мінімальна роздільна здатність екрану, яка забезпечує належний візуальний досвід користувача.

NFR6 Інтернет-підключення – додаток може вимагати наявності активного Інтернет-підключення для синхронізації заміток, резервного копіювання, відправки сповіщень тощо.

2.5 Ідентифікація архетипу мобільного додатку

Інформаційна система відноситься до архетипу Mobile Application (MA) – класичний додаток, який запускається на мобільному пристрою під управлінням операційної системи Android.

2.6 Проектування навігаційного графу мобільного додатку

Навігаційний граф (Navigation Graph) – це компонент фреймворку Android Jetpack, який використовується для візуалізації і управління навігацією між екранами (фрагментами) в додатку Android. Він визначає структуру навігації та взаємозв'язки між різними екранами (рис. 2.2).

За допомогою навігаційного графа розробник може легко організувати навігацію між різними екранами в додатку. Він дозволяє встановлювати точки входу (destination) для кожного екрану, визначати перехідні анімації, задавати параметри переходів (наприклад, передачу даних), а також встановлювати специфічні правила навігації, такі як повернення до попереднього екрану (back navigation) або виконання визначених дій перед показом нового екрану.

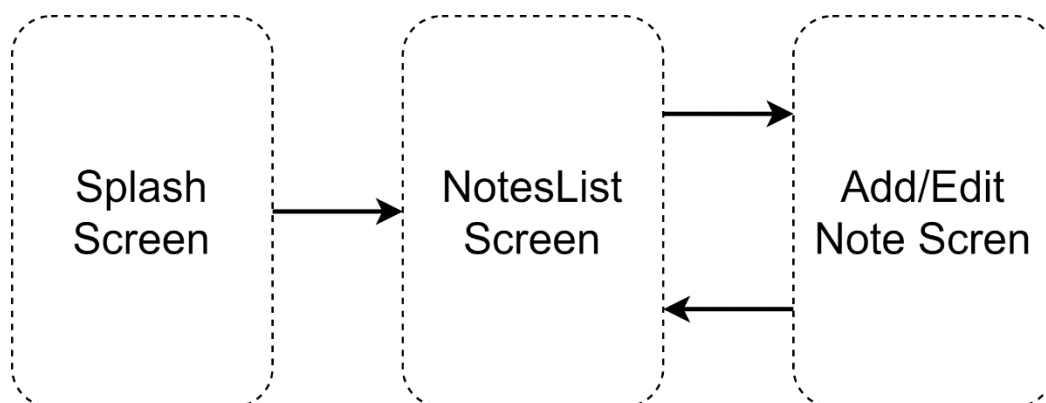


Рисунок 2.2 – Навігаційний граф мобільного додатку

2.7 Проектування користувацького інтерфейсу мобільного додатку

На основі вимог до веб-ресурсу, які включають функціональні та нефункціональні вимоги, можна приступити до проектування користувацького

інтерфейсу веб-додатку. Це вимагає розробки макетів ключових вікон веб-сайту, визначення сценаріїв взаємодії користувачів з цими вікнами, складання стислого опису кожного макету та визначення їх ролі в досягненні визначеної мети в цій кваліфікаційній роботі.

На рисунку 2.4 зображений макет головного вікна веб-ресурсу для продуктової крамниці. Головне вікно веб-ресурсу продуктової крамниці містить різні елементи. Елемент шапки включає логотип, пошукову строку, кнопку входу/реєстрації, кнопку кошика. Меню розміщується під шапкою і містить посилання на головні розділи веб-ресурсу, такі як «Продукти», «Акції», «Доставка та оплата», «Контакти» тощо.

Банер розміщується вище або нижче меню і містить зображення з рекламою акційних товарів або послуг. Категорії продуктів можуть бути розміщені на головній сторінці і містити візуальні зображення і назви категорій продуктів, таких як «Овочі та фрукти», «М'ясо та птиця», «Кондитерські вироби» тощо.

Блоки з акційними пропозиціями можуть бути розміщені в боковому меню або на головній сторінці, щоб привернути увагу клієнтів до акційних товарів. Найбільш популярні товари можуть бути розміщені на головній сторінці, щоб привернути увагу клієнтів до найбільш продаваних товарів. Відгуки клієнтів можуть бути розміщені на головній сторінці для стимулювання покупців залишати відгуки про товари та послуги. Кнопка «Показати більше» може бути розміщена на головній сторінці, щоб дозволити користувачам переглядати більше товарів безпосередньо на сторінці.

Футер містить посилання на додаткові розділи веб-ресурсу, контактні дані, логотипи платіжних систем тощо.

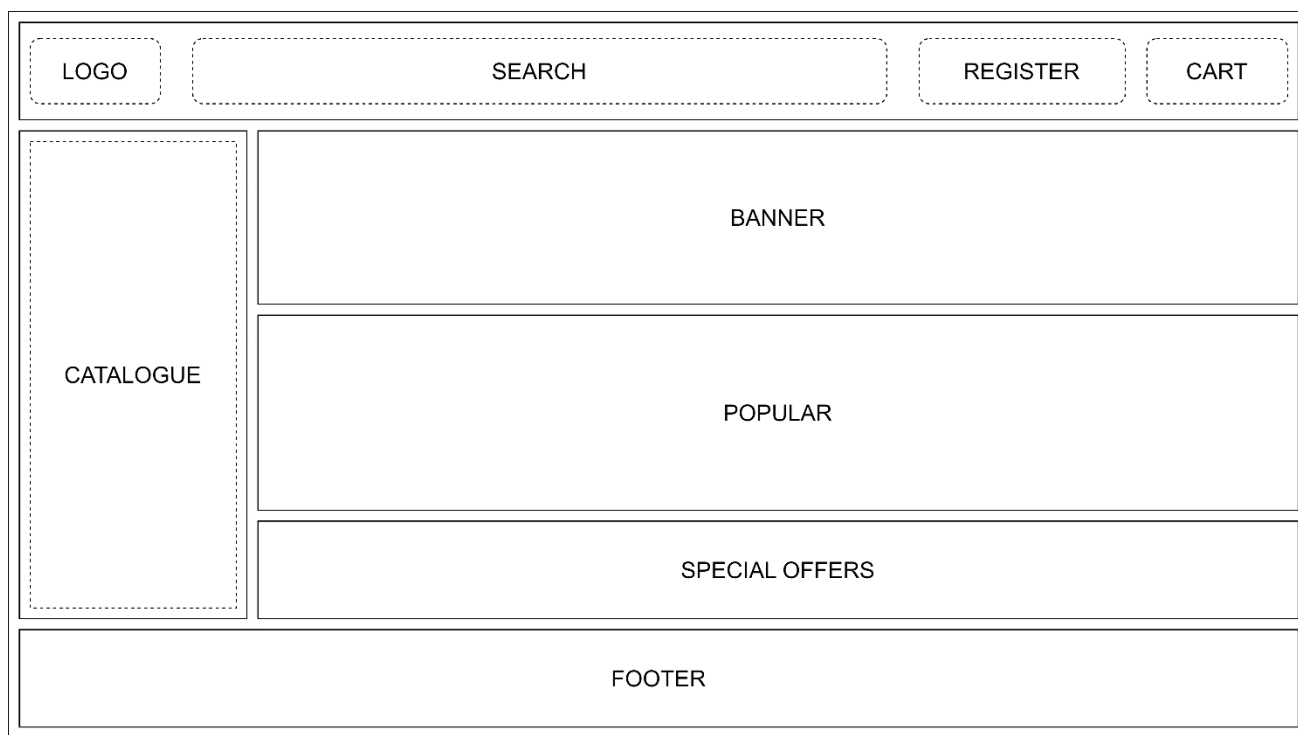


Рисунок 2.4 – Макет сторінки головного вікна веб-ресурсу

На рисунку 2.5 зображений макет вікна товару веб-ресурсу продуктової крамниці. Даний макет містить назву товару, зображення товару, ціну, опис товару, що містить характеристики, переваги, рецензії від інших покупців тощо. Також в макеті міститься кнопка "Додати в кошик", що дозволяє користувачам додавати товар до свого кошика.

Крім того, у вікні знаходиться посилання на розділ, де можна знайти інші товари того ж бренду або категорії. Також на сторінці знаходиться посилання на обрані товари користувача (список бажань або кошик)

Макет повинен мати привабливий дизайн і забезпечувати користувачам легку навігацію і доступ до усієї необхідної інформації про товар. Також, повинен бути доданий функціонал рекомендацій на основі попередніх покупок, щоб користувачі могли знайти товари, які їм можуть сподобатися.

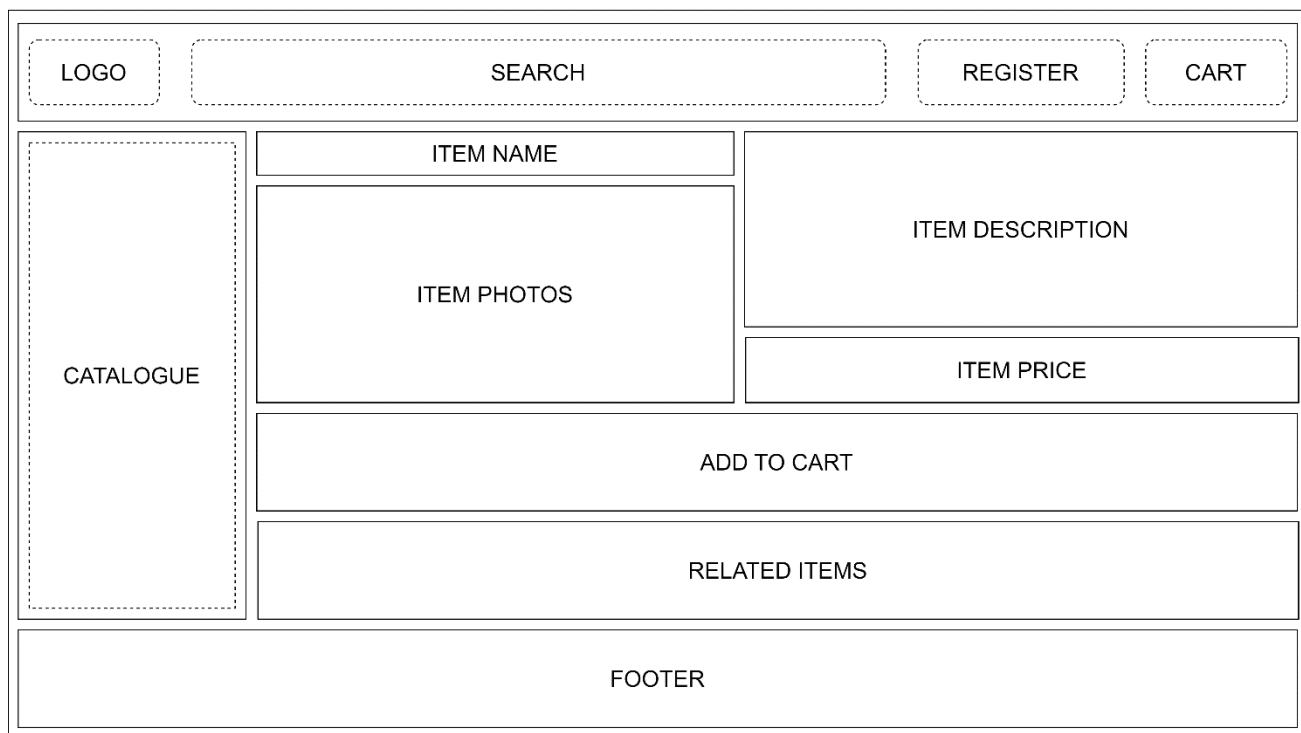


Рисунок 2.5 – Макет сторінки товару веб-ресурсу

На рисунку 2.6 зображений макет вікна кошику товарів веб-ресурсу продуктової крамниці. Макет містить список товарів, які клієнт додав до свого кошика. Для кожного товару вказується його назва, зображення, ціна, кількість, а також загальна вартість товару (ціна на одиницю товару помножена на кількість). Також у макеті є різні кнопки та поля, наприклад:

- кнопка "Видалити товар" – дозволяє видалити товар з кошика;
- кнопка "Оформити замовлення" – дозволяє перейти до оформлення замовлення на покупку товарів з кошика;
- поле для введення промо-коду – дозволяє ввести промо-код для отримання знижки на покупку товарів;
- поле для введення купону на безкоштовну доставку – дозволяє ввести купон на безкоштовну доставку товарів;

- інформація про загальну вартість товарів у кошику – показує загальну вартість всіх товарів у кошику;
- інформація про стан замовлення – показує статус замовлення, наприклад, "в обробці", "відправлено", "доставлено".

Макет кошика повинен бути зручним та інтуїтивно зрозумілим для користувачів, щоб вони могли легко знайти необхідну інформацію та керувати своїм замовленням.

LOGO	SEARCH		REGISTER	CART
CART ITEM	QNT	REMOVE	PRICE	
CART ITEM	QNT	REMOVE	PRICE	
CART ITEM	QNT	REMOVE	PRICE	
CART ITEM	QNT	REMOVE	PRICE	
PROMO-CODE			SUMMARY	
CHECKOUT				

Рисунок 2.6 – Макет вікна кошику товарів веб-ресурсу

2.8 Висновки до другого розділу

У другому розділі цієї курсової роботи було проведено проектування веб-ресурсу продуктової крамниці на базі фреймворку Angular.

У процесі проектування була визначена мета сайту, його потенційна аудиторія та основні можливості. На основі цих даних були встановлені основні вимоги до веб-ресурсу, включаючи функціональні та нефункціональні. Також була розроблена діаграма сценаріїв використання веб-ресурсу. Крім того, була

розроблена діаграма інформаційних потоків та визначені вхідні та вихідні інформаційні потоки системи.

З метою проектування інтерфейсу користувача були розроблені макети ключових вікон веб-ресурсу, а також діаграма станів системи. Крім того, була розроблена діаграма логічного уявлення системи, а також діаграма розгортання веб-ресурсу. Для побудови веб-ресурсу була визначена схема даних системи, а також ключові технології, які були використані в процесі проектування та реалізації веб-ресурсу.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВЕДЕННЯ ЗАМІТОК

3.1 Структура серверного програмного проекту

Пакет програмного проекту для мобільного застосунку може містити різні компоненти. Визначено ці компоненти та визначимо, які класи реалізують ці компоненти.

Першим компонентом програми є контролери. Компоненти програми є основними будівельними блоками програми Android. Кожен компонент є точкою входу, через яку система або користувач може увійти у вашу програму. В рамках серверного проекту були створені наступні контролери:

- Точка входу для взаємодії з користувачем. (Activities) – він являє собою єдиний екран з інтерфейсом користувача;
- Точка входу загального призначення для підтримки роботи програми у фоновому режимі з усіх причин (Services) – Це компонент, який працює у

фоновому режимі для виконання тривалих операцій або виконання роботи для віддалених процесів;

- контролер для продуктів (*ToDoAdapter.kt*) – відповідає за відображення списку задач, їх додавання, видалення та редагування;

- контролер для створення задач (*AddTodoFragment.kt*) – відповідає за обробку замовлення, введення адреси доставки та підтвердження оплати;

- контролер для аутентифікації та авторизації користувачів (*SignInFragment.kt* and *SignUpFragment.kt*) – відповідає за відображення сторінок входу, реєстрації та інші функції, що пов'язані з аутентифікацією та авторизацією користувачів.

Компонент сервісів відповідає за логіку бізнесу та операції, які пов'язані з даними. Сервіси можуть включати методи, які повертають дані з бази даних, змінюють їх та зберігають зміни. В рамках серверного проєкту були створені наступні сервіси:

- *HomeFragment.kt* – сервіс, що забезпечує роботу з задачами, їх створення, оновлення, видалення та отримання даних про задачи;

Також, в проєкті містяться такі компоненти:

- моделі – представляють сутності, які використовуються в системі. Моделі можуть містити поля, які відображають дані з бази даних, та методи, які допомагають взаємодіяти з цими даними;

- *ToDoData.kt* – сервіс, що забезпечує роботу з задачами, їх створення, оновлення, видалення та отримання даних про задачи;

3.2 Функціональне тестування розробленого мобільного застосунку

Функціональне тестування мобільного застосунку є дуже важливим процесом, оскільки воно дозволяє перевірити правильність функціонування різних функцій та можливостей цього ресурсу перед його введенням в експлуатацію.

Функціональне тестування дозволяє перевірити, що всі функції та можливості мобільного застосунку працюють правильно, як очікувалося, що забезпечує коректну роботу мобільного застосунку та надійність для його використання.

Також функціональне тестування дозволяє виявити та виправити помилки, що можуть негативно впливати на досвід користувачів, підвищуючи їх задоволеність та вірогідність повернення до застосунку в майбутньому.

3.3 Інструкція користувача мобільного застосунку

Для забезпечення успішного користувацького досвіду використання розробленого мобільного застосунку необхідно скласти інструкцію користувача, яка включає в себе знімки екранів та пояснювальний текст до кожного знімка. Це допоможе користувачам легко зорієнтуватись у функціоналі мобільного застосунку та ефективно використовувати його.

На рисунку 3.2 зображено вікно створення нового акаунту. Дизайн сторінки виконаний в світлих тонах з використанням контрастних елементів, що робить її зручною для використання.

На сторінці присутня форма реєстрації, яка містить поля для введення основної інформації, такої як адреса електронної пошти та пароль. Також є кнопка "Зареєструватися", яка відправляє заповнену форму на сервер для обробки. Кнопка входу знаходиться під формою реєстрації та дозволяє користувачам, які вже зареєструвалися, увійти на сайт.

Ця сторінка є важливою для користувачів, оскільки дозволяє їм зареєструватися та отримати доступ до додатку. Дизайн сторінки є зручним та легким для використання, що сприяє покращенню користувацького досвіду.

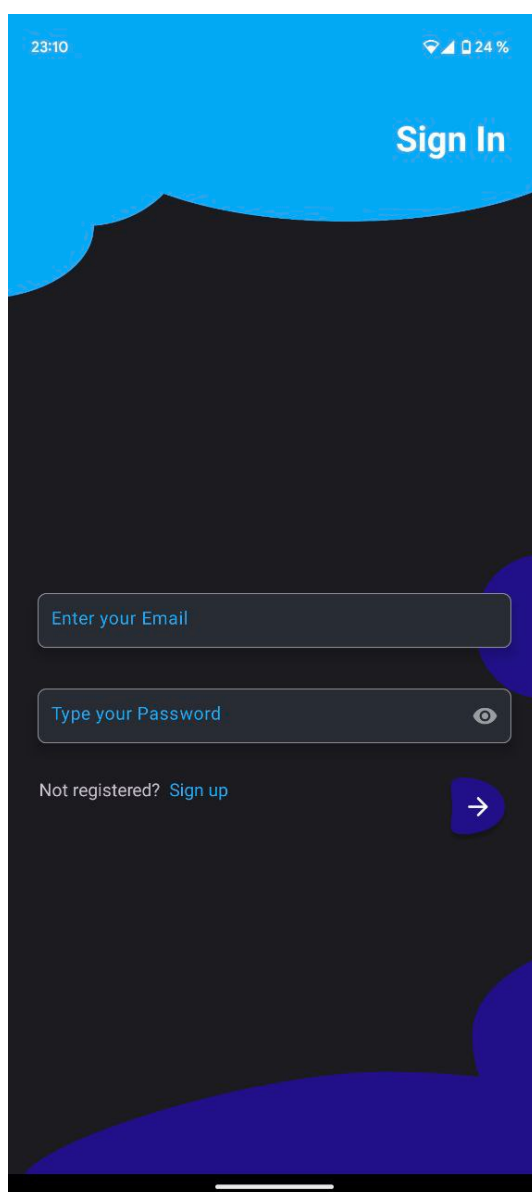


Рисунок 3.2 – Екранна форма логіну

На рисунку 3.3 зображене вікно логіну мобільно застосунку. Вона містить форму для введення інформації користувача та кнопку "Підтвердити", щоб здійснити вхід у додаток.

На сторінці присутня форма входу, яка містить поля для email та пароля.

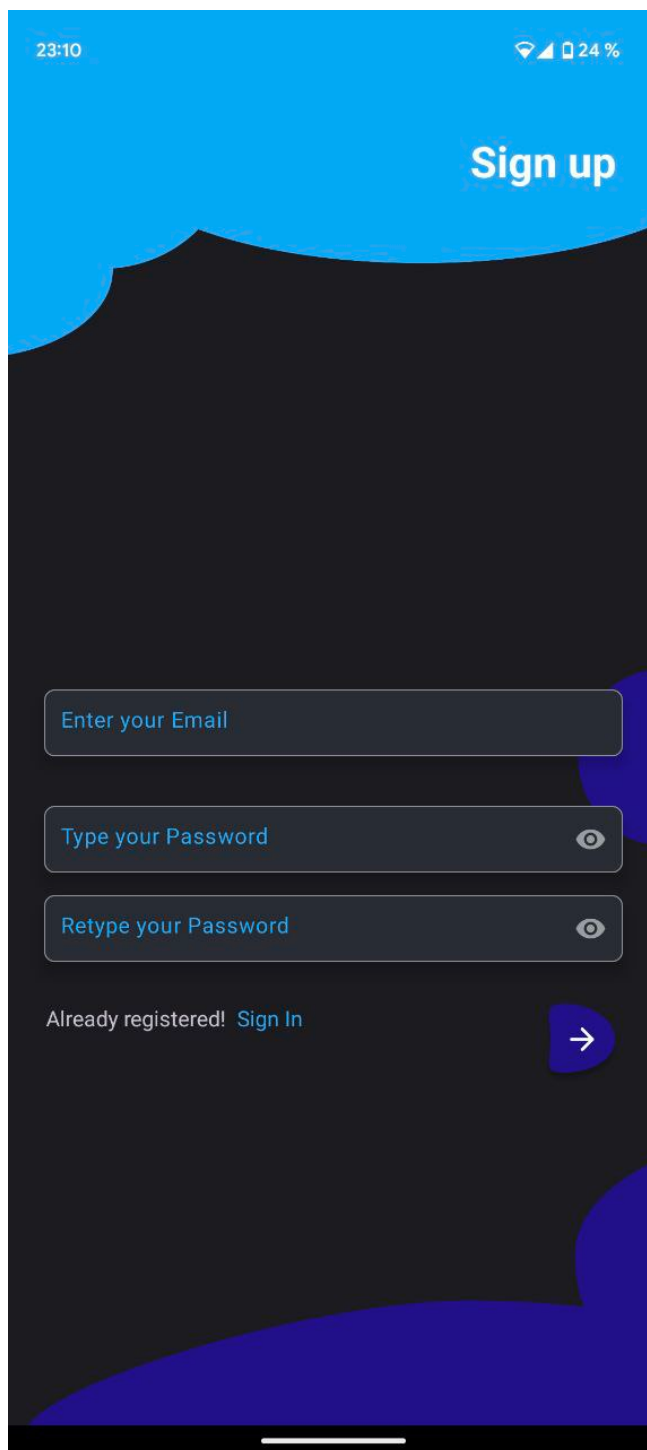


Рисунок 3.3 – Екранна форма створення нового акаунту

На рисунку 3.4 зображена екранна форма головного вікна мобільного застосунку. Вона містить інформацію про задачі та надає користувачам можливість знайти потрібну інформацію та здійснити необхідні дії.

Шапка додатку містить, кнопку виходу з додатку та налаштування.

Ця сторінка є важливою для користувачів, оскільки вона надає інформацію про задачі та дозволяє здійснити необхідні дії.

Дизайн сторінки є зручним та легким для використання, що сприяє покращенню користувацького досвіду.

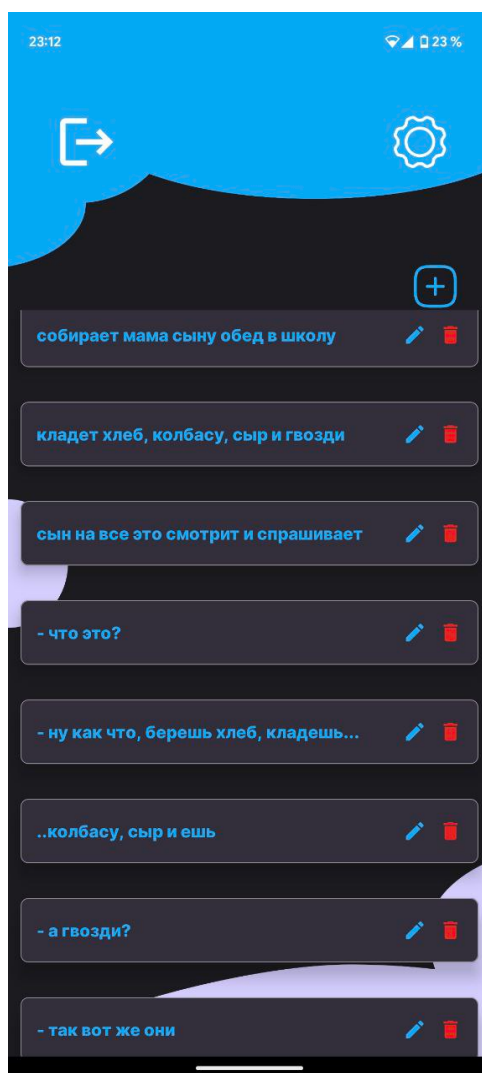


Рисунок 3.4 – Екранна форма головного вікна

На рисунку 3.5 зображена екранна форма додавання нової задачі. Вона містить поля для вводу необхідних компонентів задач.

Ця сторінка є важливою для користувачів, оскільки вона надає можливість створювати нові задачі

Дизайн сторінки є зручним та легким для використання, що сприяє покращенню користувацького досвіду.

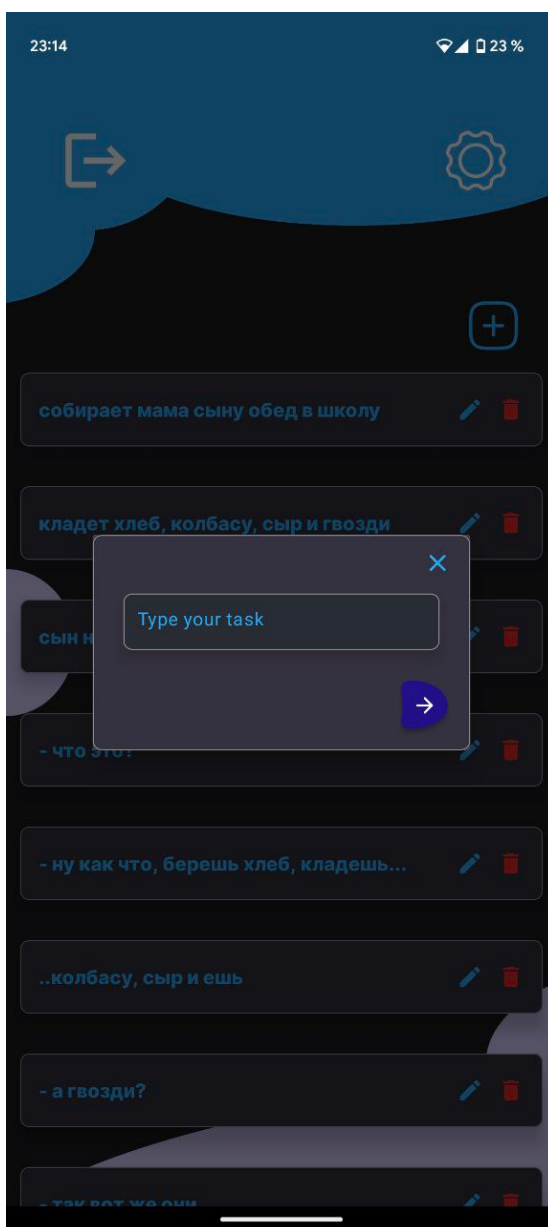


Рисунок 3.5 – Екранна форма додавання нової задачі

На рисунку 3.6 зображена екранна форма налаштування. Вона містить кнопки, такі як: зміна теми, мови та виходу з системи.

Ця сторінка є важливою для користувачів, оскільки вона надає можливість змінити тему інтерфейсу

Дизайн сторінки є зручним та легким для використання, що сприяє покращенню користувацького досвіду.

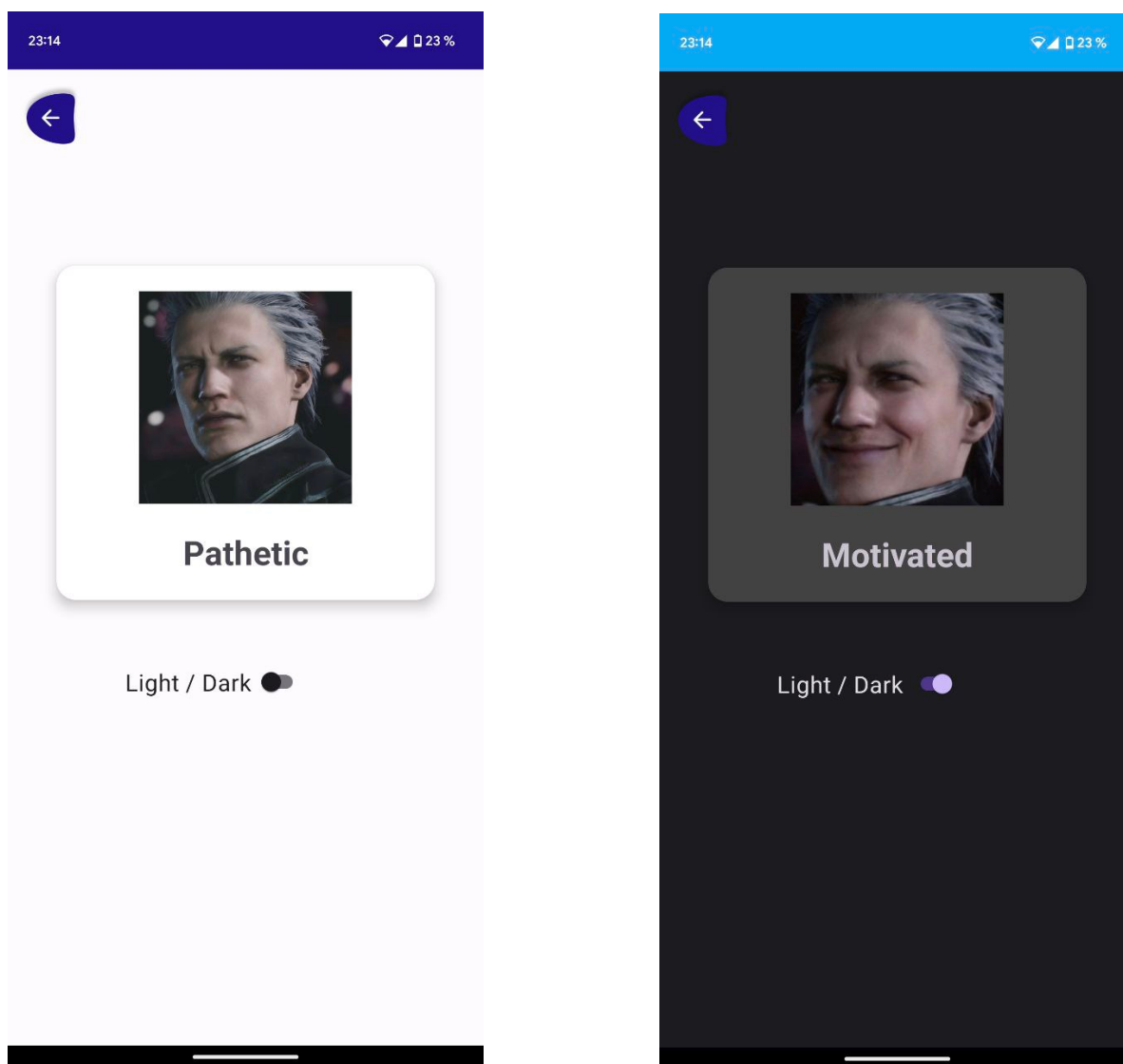


Рисунок 3.6 – Екранна форма налаштування

3.4 Вихідний код веб-ресурсу

Вихідний код програмних класів розробленого мобільного застосунку наведений на [гітхабі](#) даної роботи.

3.5 Висновки до третього розділу

В даному розділі курсової роботи було проведено програмну реалізацію мобільного застосунку на базі Fragment. Для того, щоб провести програмну реалізацію, був використаний проект на платформі Android з використанням Firebase, який був розроблений в другому розділі даної роботи.

Перш за все, була розглянута структура серверного програмного проекту. Були визначені основні програмні компоненти, з яких складається програмний проект мобільного застосунку, а також наданий перелік основних класів, які відносяться до цих компонентів.

Окремо було розглянуто питання використання системи контролю версій для упорядкування розробки мобільно застосунку.

Далі було проведено функціональне тестування розробленого мобільного додатку. Був розроблений протокол тестування. Було визначено, що всі тест-кейси були пройдені, а отже фактична поведінка мобільного додатку співпадає з очікуваною поведінкою, визначеною у функціональних вимогах.

Також була розроблена інструкція користувача у вигляді множини знімків екрану та пояснювального тексту, який визначає функціонал додатку у даному вікна та можливі дії користувача. Крім того, наданий вихідний код розробленого мобільного додатку.

ВИСНОВКИ

В даній курсовій роботі був успішно розроблений мобільний додаток на основі фреймворку Android для ведення задач. Цей додаток надає користувачам можливість переглядати, створювати, редагувати та видаляти задачі. Таким чином, було повністю досягнуто поставленої мети курсової роботи. Для досягнення цієї мети були успішно вирішені наступні завдання.

У першому розділі була детально розглянута предметна область створення мобільного додатку з використанням фреймворку Android. Були визначені основні завдання, які потрібно було виконати під час розробки. Був проведений аналіз існуючих аналогів та визначені ключові вимоги до створення власного мобільного додатку. Для розробки клієнтської частини був вибраний фреймворк Android, а для серверної частини – Firebase.

У другому розділі було проведено проектування мобільного додатку. Була визначена його мета та основні функціональні можливості. Також були встановлені основні вимоги до додатку, включаючи функціональні та нефункціональні вимоги.

У третьому розділі була реалізована програмна частина мобільного додатку. Було проведено функціональне тестування, розроблена інструкція користувача зі знімками екрану та пояснювальним текстом, а також наданий вихідний код розробленого мобільного додатку.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The importance of information technology in retail. URL: <https://www.retailcouncil.org/community/technology/the-importance-of-information-technology-in-retail/> (дата звернення 01.03.2023).
2. The Importance of Information Technology in Retail. URL: <https://www.aeologic.com/blog/the-importance-of-information-technology-in-retail/> (дата звернення 01.03.2023).
3. Fozzyshop. URL: <https://fozzyshop.ua/odesa/> (дата звернення 01.03.2023).
4. Інтернет-магазин продуктів «Таврія В». URL: <https://www.tavriav.ua/> (дата звернення 01.03.2023).
5. Онлайн-супермаркет «Сільно». URL: <https://silpo.ua/> (дата звернення 01.03.2023).
6. Angular is a platform for building mobile and desktop web applications. URL: <https://angular.io/> (дата звернення 01.03.2023).
7. Angular components overview. URL: <https://angular.io/guide/component-overview> (дата звернення 01.03.2023).
8. The Good and the Bad of Angular Development. URL: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/> (дата звернення 01.03.2023).
9. Node.js – an open-source, cross-platform JavaScript runtime environment. URL: <https://nodejs.org/en> (дата звернення 01.03.2023).
10. Django: The web framework for perfectionists with deadlines. URL: <https://www.djangoproject.com/> (дата звернення 01.03.2023).
11. Ruby on Rails — A web-app framework. URL: <https://rubyonrails.org/> (дата звернення 01.03.2023).
12. Laravel - The PHP Framework For Web Artisans. URL: <https://laravel.com/> (дата звернення 01.03.2023).
13. Spring | Home. URL: <https://spring.io/> (дата звернення 01.03.2023).
14. Express – fast, unopinionated, minimalist web framework for Node.js. URL: <https://expressjs.com/> (дата звернення 01.03.2023).

15. Welcome to Flask. URL: <https://flask.palletsprojects.com/> (дата звернення 01.03.2023).
16. ASP.NET – open-source web framework for .NET infrastructure. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата звернення 01.03.2023).
17. Java spring framework – pros, cons, common mistakes. URL: <https://bit.ly/3A1dS4w> (дата звернення 01.03.2023).
18. Key Components and Internals of Spring Boot Framework. URL: <https://bit.ly/3ojMdtb> (дата звернення 01.03.2023).
19. MySQL Home. URL: <https://www.mysql.com/> (дата звернення 01.03.2023).
20. PostgreSQL: The world's most advanced open source database. URL: <https://www.postgresql.org/> (дата звернення 01.03.2023).
21. MongoDB Atlas Database - The Database For Modern Apps. URL: <https://bit.ly/414ADjX> (дата звернення 01.03.2023).
22. Industry-leading performance and security with SQL Server 2019. URL: <https://www.microsoft.com/en-us/sql-server/sql-server-2019?rtc=1> (дата звернення 01.03.2023).
23. Database | Oracle. URL: <https://www.oracle.com/database/> (дата звернення 01.03.2023).
24. SQLite Home Page. URL: <https://sqlite.org/index.html> (дата звернення 01.03.2023).
25. Amazon RDS Features. URL: https://aws.amazon.com/rds/features/?nc1=h_ls (дата звернення 01.03.2023).
26. Firebase Home. URL: <https://firebase.google.com/> (дата звернення 01.03.2023).