Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Інститут комп’ютерних систем

Кафедра інформаційних систем

Хрущов Артур Ігорович

Рябий Андрій Ігорович

студенти групи AI-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 6](#_Toc137028404)

[1.1 Особливості використання мобільних технологій для ведення справ та особистої продуктивності 6](#_Toc137028405)

[1.2 Огляд додатків для ведення заміток 7](#_Toc137028406)

[1.2.1 Додаток Remember The Milk 8](#_Toc137028407)

[1.2.2 Додаток ColorNote 9](#_Toc137028408)

[1.3 Формування вимог до основних функцій мобільного додатку 11](#_Toc137028409)

[1.4 Огляд інформаційних технологій для розробки мобільного додатку 11](#_Toc137028410)

[1.4.1 Fragment 11](#_Toc137028411)

[1.4.2 Архітектурні компоненти Android 13](#_Toc137028412)

[1.5 Висновки до першого розділу 14](#_Toc137028413)

[2 проєктування мобільного додатку для ведення заміток 16](#_Toc137028414)

[2.1 Мета та задачі мобільного додатку 16](#_Toc137028415)

[2.2 Визначення функціональних вимог до мобільного додатку 17](#_Toc137028416)

[2.3 Формування користувацьких історій мобільного додатку 17](#_Toc137028417)

[2.4 Визначення нефункціональних вимог до мобільного додатку 19](#_Toc137028418)

[2.5 Ідентифікація архетипу мобільного додатку 19](#_Toc137028419)

[2.6 Проектування навігаційного графу мобільного додатку 20](#_Toc137028420)

[2.7 Проектування користувацького інтерфейсу мобільного додатку 20](#_Toc137028421)

[2.8 Висновки до другого розділу 21](#_Toc137028422)

[3 ПРОГРАМНА РЕАЛІЗАЦІЯ мобільного додатку для ведення заміток 22](#_Toc137028423)

[3.1 Структура серверного програмного проєкту 22](#_Toc137028424)

[3.2 Діаграма класів веб-ресурсу 23](#_Toc137028425)

[3.3 Керування вихідним кодом веб додатку 23](#_Toc137028426)

[3.4 Функціональне тестування розробленого веб-ресурсу 34](#_Toc137028427)

[3.5 Інструкція користувача веб-ресурсу 34](#_Toc137028428)

[3.6 Висновки до третього розділу 34](#_Toc137028430)

[ВИСНОВКИ 35](#_Toc137028431)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 36](#_Toc137028432)

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, Fragment та архітектура MVVM.

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 Fragment and the MVVM architecture.

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 проєктування мобільного додатку заміток;
* виконати програмну реалізацію спроєктованого мобільного додатку.

# Огляд систем-аналогів та технологій їх розробки

## 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.

## Огляд додатків для ведення заміток

Для розробки мобільного додатку 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.

### 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].

|  |  |  |
| --- | --- | --- |
| Зображення, що містить текст, знімок екрана, програмне забезпечення, Операційна система  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, програмне забезпечення, Операційна система  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма  Автоматично згенерований опис |

Рисунок 1.2 – Знімки екрану додатку Remember The Milk

Основний функціонал додатку:

* можливість створювати списки справ;
* можливість отримати нагадування через електрону 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.

### 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;
* віджет 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, електроної пошти чи твітеру;
* можливість кольорової організації нотаток.

|  |  |  |
| --- | --- | --- |
| Зображення, що містить текст, знімок екрана, Шрифт  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, число, програмне забезпечення  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма  Автоматично згенерований опис |

Рисунок 1.3 – Знімки екрану додатку ColorNote

## Формування вимог до основних функцій мобільного додатку

Аналіз додатків для ведення справ та нотаток дозволив виділити їх переваги та недоліки та сформувати 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 паролю, відбитка пальця або інших методів аутентифікації. Це гарантує, що тільки авторизовані користувачі можуть отримати доступ до приватних даних.

## Огляд інформаційних технологій для розробки мобільного додатку

### Fragment

Фрагмент представляє повторно використовувану частину інтерфейсу користувача програми. Фрагмент визначає та керує своїм власним макетом, має власний життєвий цикл і може обробляти власні події введення. Фрагменти не можуть жити самі по собі. Вони повинні бути розміщені в діяльності або іншому фрагменті на платформі Android. Ось деякі переваги та недоліки використання Фрагмент:

Розглянемо переваги Fragment:

* Фрагмент має власний макет і власну поведінку з власними зворотними викликами життєвого циклу;
* Ви можете об’єднати кілька фрагментів в одній дії, щоб створити багатопанельний інтерфейс користувача;
* Ви можете додавати або видаляти фрагменти в дії під час її виконання;
* Фрагмент може реалізовувати поведінку, яка не має компонента інтерфейсу користувача;
* Життєвий цикл фрагмента тісно пов’язаний із життєвим циклом його хост-активності, що означає, що коли активність призупиняється, усі наявні в активності фрагменти також зупиняються.

Розглянемо недоліки Fragment:

* Оновлення бібліотеки підтримки іноді призводять до того, що фрагменти поводяться по-новому та неочікувано, особливо якщо у вас є дуже складні інтерфейси користувача, складені з вкладених фрагментів;
* Фрагментні транзакції дозволяють виконувати набір фрагментних операцій. На жаль, фіксація транзакції є асинхронною та розміщується в кінці черги обробника основного потоку. Це може залишити вашу програму в невідомому стані під час отримання кількох подій кліків або під час змін конфігурації;
* Фрагменти створюють, зв’язують і налаштовують представлення, вони містять багато коду, пов’язаного з представленнями. Фактично це означає, що бізнес-логіка не відокремлена від коду перегляду, що ускладнює написання модульних тестів на основі фрагментів;

Незважаючи на свої недоліки, фрагменти навчили нас безцінним урокам, які тепер ми можемо повторно застосувати під час написання програм:

* Інтерфейс однієї дії: немає необхідності використовувати одну дію для кожного екрана. Ми можемо розділити нашу програму на окремі віджети та збирати їх, як нам заманеться. Це полегшує анімацію та життєвий цикл. Ми можемо розділити наші віджети на код перегляду та код контролера.
* Backstack не є специфічним поняттям діяльності; ви можете реалізувати backstack в межах діяльності.
* Немає потреби в нових API; все, що нам було потрібно, було з самого початку: дії, перегляди та розповсюджувачі макетів.

### 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.

## 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 застосунку, 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. Для розробки користувацького інтерфейсу було вирішено обрати Fragment. У якості технології для розробки архітектури додатку були обрані архітектурні компоненти. Вони допомагають забезпечити структурованість, модульність, перевикористовуваність та тестованість коду.

# проєктування мобільного додатку для ведення заміток

## 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.

Задачі мобільного додатку для ведення заміток можуть включати:

* створення нових заміток – користувачі повинні мати можливість швидко створювати нові записи заміток з використанням тексту;
* 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 пріоритетністю, датою створення, чи датою закінчення;
* синхронізація та резервне копіювання – важливо, щоб додаток забезпечував можливість синхронізувати замітки між різними пристроями користувача і забезпечував регулярне резервне копіювання для запобігання втраті даних;

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.

## 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.

## 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 опцію «Створити нову замітку(+)»;
* користувач вводить назву нотатки та текст нотатки;
* 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 нотатку;
* 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 у списку;
* він 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 «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 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 «Пріоритет» або подібний елемент;
* користувач вибирає пріоритетний рівень для замітки (високий, середній, низький);
* після збереження, замітка отримує позначення пріоритету, що допомагає користувачеві виділити найважливіші завдання.

## Визначення нефункціональних вимог до мобільного додатку

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

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

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

NFR3 Оперативна пам'ять (RAM) – мінімальний обсяг оперативної пам'яті, необхідний для плавної роботи додатку 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 – 1920x1080, забезпечує належний візуальний досвід користувача.

NFR6 Інтернет-підключення – додаток вимагає наявності активного Інтернет-підключення для синхронізації заміток, резервного копіювання тощо.

## Ідентифікація архетипу мобільного додатку

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

## Проектування навігаційного графу мобільного додатку

Навігаційний граф (Navigation Graph) – це компонент 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.

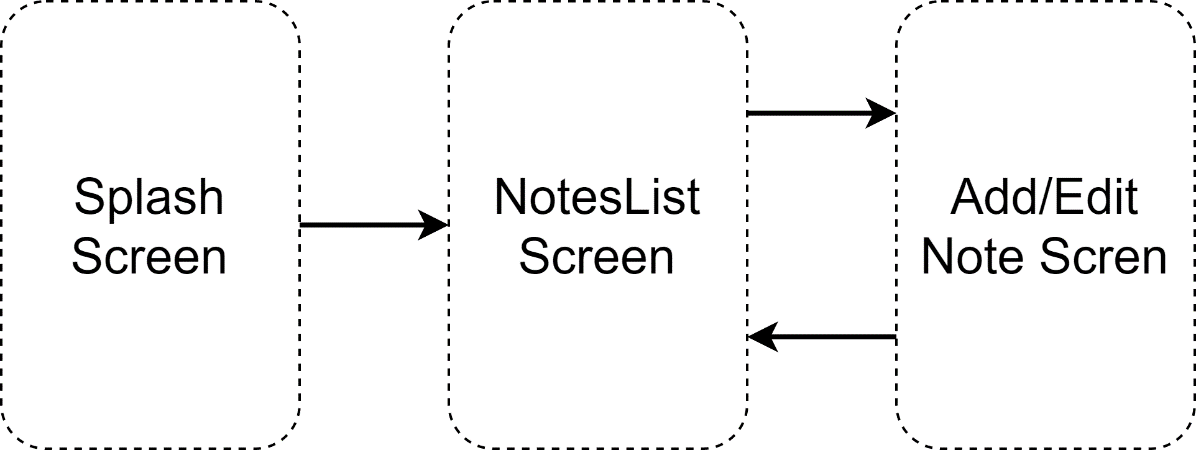


Рисунок 2.2 – Навігаційний граф мобільного додатку

## Проектування користувацького інтерфейсу мобільного додатку

На основі вимог до мобільного застосунку, які включають функціональні та нефункціональні вимоги, можна приступити до проектування користувацького інтерфейсу веб-додатку. Це вимагає розробки макетів ключових вікон мобільного застосунку, визначення сценаріїв взаємодії користувачів з цими вікнами, складання стислого опису кожного макету та визначення їх ролі в досягненні визначеної мети в цій кваліфікаційній роботі.

## Висновки до другого розділу

У другому розділі цієї курсової роботи було визначено мету та задачі мобільного додатку, функціональних вимог до мобільного додатку, сформованно користувацькі історії, визначенно нефункціональні вимоги до мобільного додатку, ідентифіковано архетипу мобільного додатку,спроектовано навігаційний граф мобільного додатку, користувацькиї інтерфейс мобільного додатку

# ПРОГРАМНА РЕАЛІЗАЦІЯ мобільного додатку для ведення заміток

## Структура серверного програмного проєкту

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 проєкту 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 (*NotesFragment.kt*) – відповідає за відображення списку задач, їх додавання, видалення та редагування;
* контролер для створення задач (*AddNoteFragment.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:

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

## 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.

## 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.

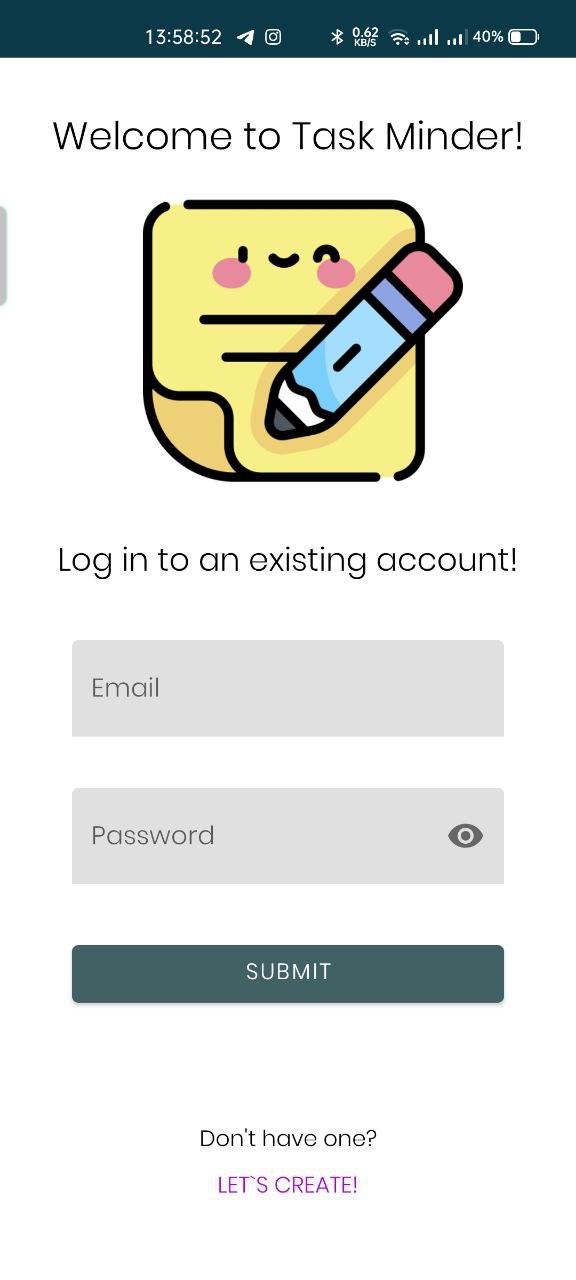


Рисунок 3.2 – Екранна форма логіну

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 еmail та пароля.

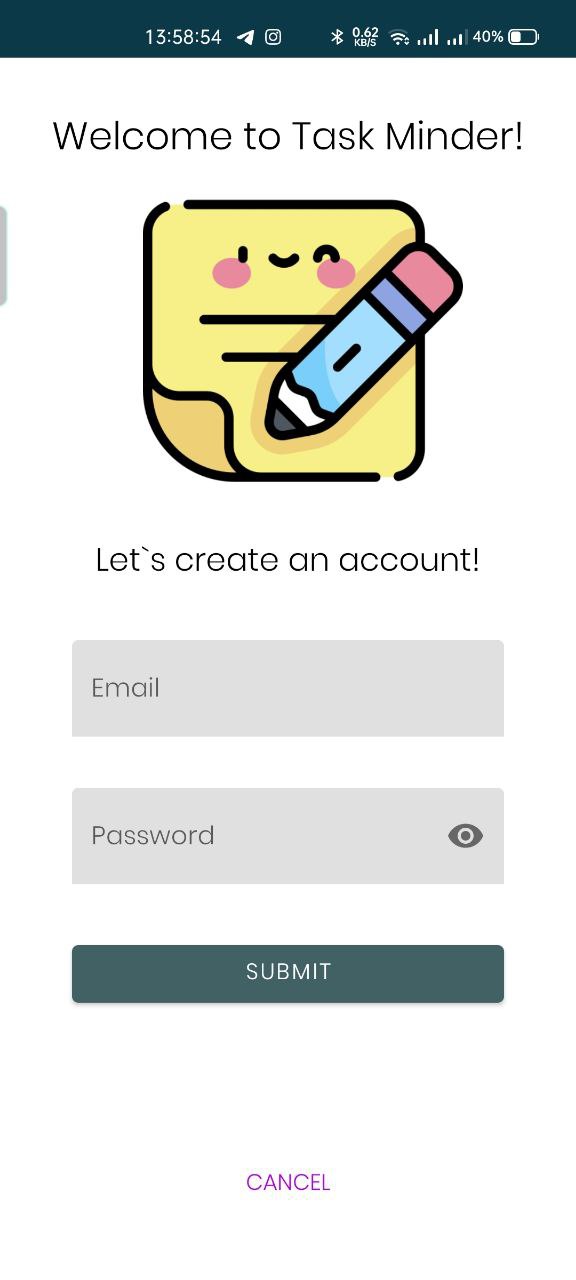


Рисунок 3.3 – Екранна форма створення нового акаунту

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, навігаційне меню та кнопку "Фільтру", щоб відфільтрувати задачі.

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 та надає користувачам можливість обрати дати та пріоритет.

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.

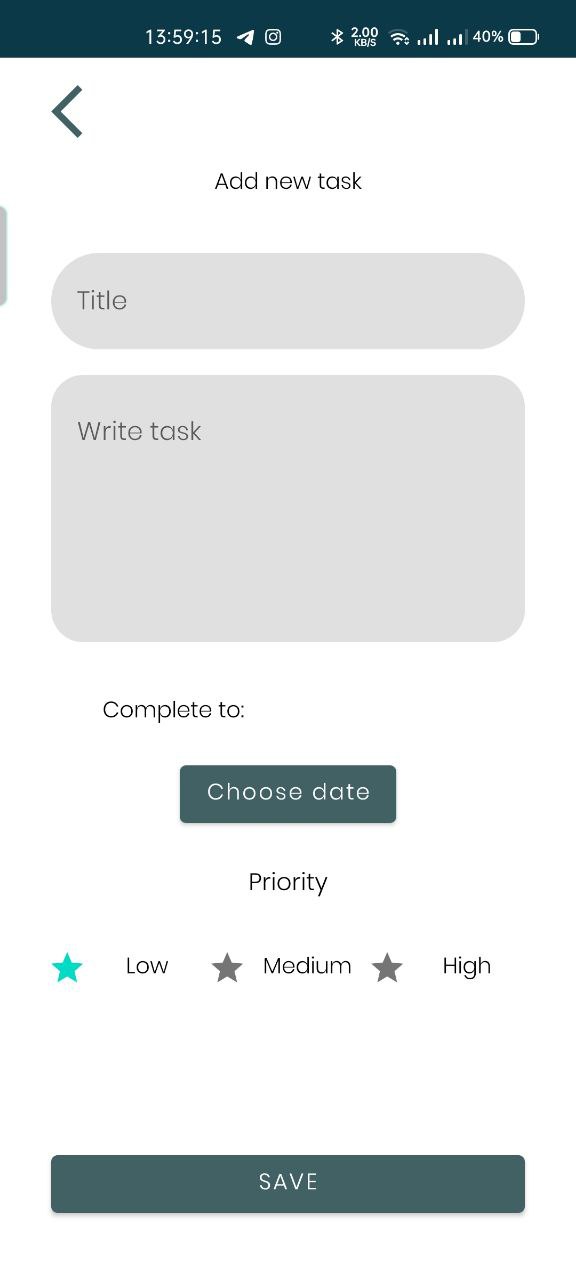


Рисунок 3.5 – Екранна форма додавання нової задачі

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 налаштувати інтерфейс та вийти з додатку

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



Рисунок 3.6 – Екранна форма налаштування

## Вихідний код веб-ресурсу

Вихідний код програмних класів розробленого мобільного застосунку наведений на гитхабі даної роботи.

## 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 проєкт на платформі Android з використанням Firebase, який був розроблений в другому розділі даної роботи.

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

Окремо було розглянуто питання використання системи контроля 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

У даній курсовій роботі був розроблений мобільного додатку для ведення задач на базі фреймворку Android.

Такий додаток дозволяє користувачам переглядати задачі, створювати нові задачі, редагувати та видаляти. Таким чином, мета, поставлена перед даною курсовою роботою досягнута в повному обсязі. Для досягнення мети були вирішені наступні задачі.

У першому розділі даної курсової роботи була детально розглянута предметна область створення мобільного додатку з використанням фреймворку Android. Були визначені основні завдання, які повинні бути вирішені в процесі розробки мобільного додатку. Був проведений аналіз існуючих аналогів та визначені ключові вимоги до створення власного мобільного додатку. У якості технологій для розробки клієнтської частини був обраний Android, для розробки серверної частини – Firebase.

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

В третьому розділі роботи було проведено програмну реалізацію мобільного додатку. Проведено функціональне тестування, розроблена інструкція користувача у вигляді множини знімків екрану та пояснювального тексту та 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).

# Firebase Home. URL: https://firebase.google.com/ (дата звернення 01.03.2023).