

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «РОЗРОБКА МОБІЛЬНОГО ДОДАТКА ДЛЯ  
АНАЛІЗУ БІЗНЕС-ДАНИХ»

Виконав: студент 4 курсу, групи 6.1219-2пi  
спеціальності 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)

освітньої програми програмна інженерія  
(назва освітньої програми)

О.В. Левченко

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,  
доцент, к.ф.-м.н. Горбенко В.І.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної  
математики, професор, д.т.н. Гребенюк С.М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри програмної  
інженерії, к.ф.-м.н., доцент

\_\_\_\_\_ Лісняк А.О.  
(підпис)

“ 07 ” 02 2023 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Левченку Олексію Васильовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка мобільного додатка для аналізу бізнес-даних

керівник роботи Горбенко Віталій Іванович, к.ф.-м.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 26 » січня 2023 року № 102-с

2. Строк подання студентом роботи 07.06.2023

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Реалізація системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 07.02.2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	17.02.2023	
2.	Аналіз предметної області	24.02.2023	
3.	Збір вихідних даних.	03.03.2023	
4.	Обробка методичних та теоретичних джерел.	17.03.2023	
5.	Вибір засобів реалізації.	24.03.2023	
6.	Розробка першого розділу.	14.04.2023	
7.	Розробка другого розділу.	02.05.2023	
8.	Проектування системи.	15.05.2023	
9.	Розробка третього розділу.	22.05.2023	
10.	Тестування.	26.05.2023	
11.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	01.06.2023	
12.	Захист кваліфікаційної роботи.	22.06.2023	

Студент \_\_\_\_\_  
(підпис)

О.В. Левченко \_\_\_\_\_  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

В.І. Горбенко \_\_\_\_\_  
(ініціали та прізвище)

### Нормоконтроль пройдено

Нормоконтролер \_\_\_\_\_  
(підпис)

А.В. Столярова \_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка мобільного додатка для аналізу бізнес-даних»: 52 с., 17 рис., 10 джерел, 1 додаток.

ANDROID, JAVA, SQLITE, МОБІЛЬНИЙ ДОДАТОК.

Об'єкт дослідження – мобільний додаток для аналізу бізнес-даних.

Мета роботи: розробка мобільного додатку для аналізу бізнес-даних.

Метод дослідження – проаналізувати предметні області, вивчення та узагальнення

У кваліфікаційній роботі досліджено предметну область, засоби реалізації, обрано найкращий засіб реалізації розроблено та протестовано мобільний додаток. Було спроектовано архітектуру системи. Користувацька частина була розроблена за допомогою фреймворку MPAndroidChart.

## SUMMARY

Bachelor's qualifying paper «Development of a Mobile Application for Business Data Analysis»: 52 pages, 17 figures, 10 references, 1 supplement.

ANDROID, JAVA, SQLITE, MOBILE APPLICATION.

Research object is mobile application for business data analysis.

Objective of the work is development of a mobile application for business data analysis.

Research method are analysis of the subject area, study, and generalization. The subject area was investigated in the thesis, implementation tools were analyzed, the best implementation tool was chosen, and a mobile application was developed and tested. The system architecture was designed. The user interface was developed using the MPAndroidChart framework.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат.....	4
Summary.....	5
Вступ.....	8
1 Теоретичні відомості .....	10
1.1 Актуальність проблеми.....	10
1.2 Вимоги до мобільного додатку для аналізу бізнес-даних.....	11
1.3 Огляд наявних мобільних додатків для аналізу бізнес-даних.....	12
1.3.1 IBM Cognos Analytics .....	12
1.3.2 Google Analytics .....	14
1.3.3 Qlik Sense Mobile .....	16
1.4 Висновки до першого розділу .....	18
2 Проєктування мобільного додатку.....	19
2.1 Вибір мови програмування.....	19
2.2 Бібліотека для візуалізації .....	22
2.3 Середовище розробки .....	23
2.4 Вибір бази даних.....	24
2.4.1 Реляційна модель бази даних .....	24
2.4.2 Нереляційна модель бази даних .....	25
2.5 SQLite.....	26
2.6 Діаграма прецедентів .....	27
2.7 Діаграма послідовностей.....	30
2.8 Дизайн.....	30
2.8.1 Логотип мобільного додатку .....	33
2.9 Висновки до другого розділу .....	35
3 Робота мобільного додатку .....	37
3.1 Опис інструментів розробки.....	37

3.2 Тестування.....	37
Висновки .....	43
Перелік посилань.....	44
Додаток А Код програмного забезпечення.....	45

## ВСТУП

За останні десятиліття мобільні пристрої значно змінили наш спосіб життя та роботи, надаючи доступ до інформації у будь-який час та в будь-якому місці. У світі бізнесу мобільні додатки стали невід'ємною частиною робочого процесу, допомагаючи менеджерам та аналітикам здійснювати швидкий та зручний доступ до важливої інформації про бізнес-процеси та діяльність компанії.

У рамках даної дипломної роботи зосередимось на розробці мобільного додатку для аналізу бізнес даних, що дозволить користувачам легко та зручно отримувати, аналізувати та візуалізувати дані з різних джерел в одному місці. Додаток буде дозволяти користувачам отримувати різноманітну інформацію про фінансові показники, динаміку продажів, витрати та інші ключові фактори, необхідні для успішної роботи компанії.

Дана робота має на меті розробку додатку з використанням сучасних технологій та методологій програмування, щоб забезпечити користувачам якісний та зручний інструмент для аналізу бізнес даних. В результаті виконання даної роботи буде розроблено мобільний додаток, який дозволить користувачам швидко та ефективно аналізувати дані, що допоможе компанії у прийнятті важливих рішень та збільшенні ефективності її діяльності.

Ця кваліфікаційна робота присвячена розробці мобільного додатку для аналізу бізнес-даних.

Метою дослідження є створення системи, що дозволяє автоматизувати процес збору та обробки даних для аналізу в бізнес-середовищі.

Об'єктом дослідження є процес аналізу бізнес-даних, а предметом - розробка мобільного додатку.

У відповідності з метою дослідження ставляться такі завдання:

– розглянути найбільш популярні засоби для аналізу бізнес-даних та визначити найефективніші підходи;



- розробити функціональні вимоги та дизайн інтерфейсу мобільного додатку;
- спроектувати програмну структуру мобільного додатку;
- дослідити предметну область та ознайомитись з функціоналом та вимогами до розроблюваного мобільного додатку;
- реалізувати мобільний додаток для збору та обробки бізнес-даних;
- провести тестування мобільного додатку та виправити помилки та недоліки.

# 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1 Актуальність проблеми

У сучасному світі бізнес-аналітика є надзвичайно важливою складовою успішної діяльності будь-якої компанії. Одним із інструментів для збору та аналізу даних є мобільні додатки, які забезпечують швидкий та зручний доступ до бізнес-інформації.

Актуальність проблеми мобільних додатків для аналізу бізнес-даних полягає в тому, що вони дозволяють компаніям ефективніше управляти своєю діяльністю та приймати обґрунтовані рішення на основі точних даних [1]. Мобільні додатки можуть збирати, обробляти та аналізувати дані з різних джерел, включаючи соціальні мережі, веб-сайти та інші додатки. Це дозволяє компаніям здійснювати моніторинг своєї діяльності та діяльності конкурентів, а також слідкувати за тенденціями та прогнозувати майбутні події.

Однією з переваг мобільних додатків для аналізу бізнес-даних є можливість їх використання в режимі реального часу. Це дозволяє компаніям швидко реагувати на зміни в бізнес-середовищі та внесені зміни в роботу. Мобільні додатки також забезпечують більш високу мобільність та доступність інформації, що дозволяє бізнесу діяти швидше та ефективніше.

Зрештою, важливо зазначити, що мобільні додатки для аналізу бізнес-даних дозволяють збільшити точність та об'єктивність прийнятих рішень. Вони допомагають уникнути неочікуваних результатів та помилок, що можуть виникнути при ручному аналізі даних. Забезпечують автоматизований процес збору та обробки даних, що дозволяє знизити ризики помилок та збільшити ефективність роботи.

Отже, можна стверджувати, що проблема мобільних додатків для аналізу бізнес-даних є актуальною та важливою в наш час. Вони дозволяють

компаніям збільшувати ефективність своєї діяльності, приймати обґрунтовані рішення на основі точних даних та забезпечувати мобільний та швидкий доступ до необхідної інформації. Мобільні додатки стають все більш популярними в бізнес-середовищі та відкривають нові можливості для компаній у галузі аналітики та управління діяльністю.

## **1.2 Вимоги до мобільного додатку для аналізу бізнес-даних**

Для мобільного додатку, який призначений для аналізу бізнес-даних, варто врахувати наступні функції:

- відображення ключових показників ефективності бізнесу: додаток повинен мати можливість відображати графіки, таблиці та інші візуальні засоби для відображення даних, такі як звіти про продажі, відвідування сайту, конверсії тощо;

- функції збору даних: додаток повинен мати можливість збирати дані з різних джерел, таких як бази даних, CRM-системи, соціальні мережі, аналітичні інструменти, інтернет-магазини та інші;

- функції аналізу даних: додаток повинен мати можливість аналізувати дані та відображати їх у зручному для користувача форматі, наприклад, додаток може розраховувати коефіцієнти ефективності, проводити аналіз даних за різними параметрами, відслідковувати тренди тощо;

- функції співпраці: додаток повинен мати можливість дозволяти користувачам обмінюватися даними та інформацією між собою, наприклад, додаток може мати функцію спільного доступу до даних для команди, можливість обговорювати результати аналізу внутрішнього чату тощо;

- функції безпеки: додаток повинен мати високий рівень безпеки для захисту конфіденційної інформації, наприклад, додаток може мати функцію двофакторної аутентифікації, шифрування даних, контроль доступу

до даних тощо;

- функції налаштування: додаток повинен мати можливість налаштовувати показники та параметри аналізу в залежності від потреб користувача, наприклад, додаток може мати функцію налаштування граничних даних коефіцієнтів ефективності, відображення різних графіків та таблиць, вибір періоду аналізу та інші;

- функції експорту: додаток повинен мати можливість експортувати дані та результати аналізу в різних форматах, таких як Excel, PDF, CSV та інші (це дозволить користувачам поділитися результатами аналізу з колегами, зберегти дані для майбутнього використання та інші цілі);

- функції розширення: додаток повинен мати можливість підключати різні плагіни та розширення для поліпшення функціоналу та розширення можливостей додатку.

### **1.3 Огляд наявних мобільних додатків для аналізу бізнес-даних**

Для кращого розуміння подальшої задачі. Оглянемо та проаналізуємо деякі вже існуючі мобільні додатки для аналізу бізнес-даних. Для цього скористаємося Google Play Market.

Google Play Market (Google Play Store) – це офіційний магазин додатків для Android-пристроїв, який належить компанії Google. В цьому магазині можна знайти різноманітні додатки, ігри, музику, фільми та інші цифрові контенти для своєї мобільної техніки на базі Android.

#### **1.3.1 IBM Cognos Analytics**

IBM Cognos Analytics – це потужний аналітичний інструмент, який допомагає підприємствам та організаціям вирішувати складні бізнес-

завдання та приймати досліджені рішення шляхом збору, оцінки та візуалізації даних. Цей додаток надає можливість виконувати повний аналіз даних за допомогою широкого спектру функцій, включаючи створення звітів, візуалізацію даних, створення дашбордів, моделювання та прогнозування даних, а також розробку та здійснення аналітичних запитів.

Однією з ключових особливостей IBM Cognos Analytics є його здатність до взаємодії з різними джерелами даних, такими як бази даних, електронні таблиці, веб-сервери та інші системи управління даними. Це дозволяє користувачам отримувати доступ до всієї необхідної інформації в одному місці та аналізувати її в режимі реального часу. Приклад сторінок програми наведені на рисунку 1.1.

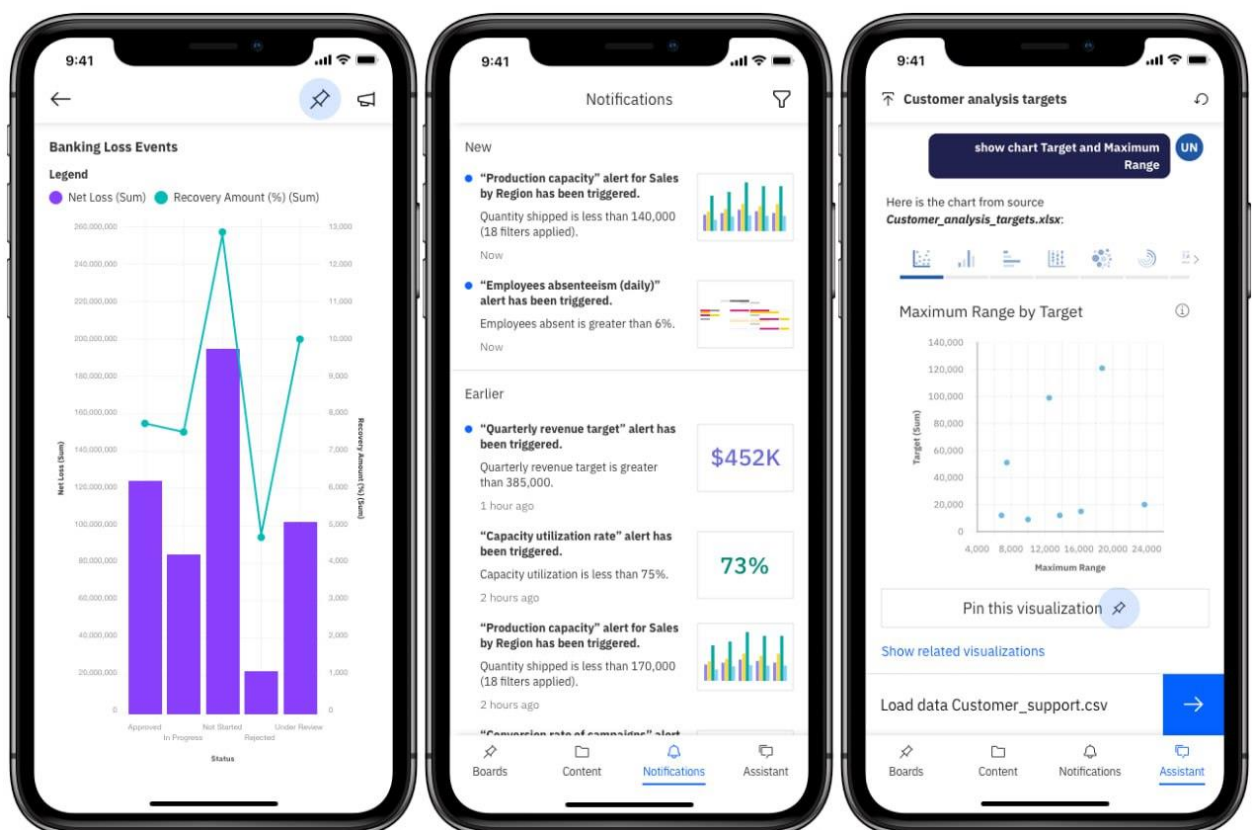


Рисунок 1.1 – Інтерфейс сторінок IBM Cognos Analytics

З технічної точки зору, IBM Cognos Analytics – це багатофункціональний аналітичний додаток, який базується на різноманітних технологіях та інструментах, щоб забезпечити користувачам можливість

збору, обробки, аналізу та візуалізації даних. Нижче наведено деякі з технічних складових IBM Cognos Analytics.

*База даних:* IBM Cognos Analytics підтримує підключення до різних джерел даних, включаючи бази даних, електронні таблиці, веб-сервери та інші системи управління даними. Ці джерела даних можуть бути оброблені та збережені в різних форматах, що дозволяє користувачам аналізувати дані з різних джерел в одному місці.

*Моделювання та прогнозування даних:* IBM Cognos Analytics надає можливість моделювання та прогнозування даних, що дозволяє користувачам розробляти прогнози та вірогідні сценарії для майбутнього розвитку бізнесу.

*Візуалізація даних:* IBM Cognos Analytics дозволяє створювати різні типи візуалізації даних, такі як графіки, діаграми, картинки, що дозволяє користувачам легко розуміти дані та взаємодіяти з ними.

*Керування даними:* IBM Cognos Analytics має різні функції керування даними, такі як збереження, обмін та керування доступом до даних. Це дозволяє користувачам легко та безпечно обробляти та зберігати дані.

*Розширення та налаштування:* IBM Cognos Analytics має відкрите API та можливості розширення, що дозволяє користувачам налаштовувати та розробляти функціонал додатку під свої потреби. Крім того, IBM Cognos Analytics має вбудовану систему керування версіями, що дозволяє користувачам зберігати та контролювати різні версії своїх аналітичних звітів та даних.

### **1.3.2 Google Analytics**

Google Analytics – найперша програма, яку ми побачимо, якщо напишемо в Google Play Market запит “для аналізу бізнес-даних”. це інструмент веб-аналітики, розроблений компанією Google, який дозволяє

веб-власникам відстежувати та аналізувати поведінку користувачів на їх сайтах.

За допомогою Google Analytics можна дізнатися, скільки людей відвідали сайт, звідки вони прибули, як довго вони перебували на сайті, які сторінки вони переглянули, які дії вони виконували (наприклад, заповнили форму або зробили покупку), та іншу корисну інформацію.

Ця інформація може бути використана веб-власниками для оптимізації свого сайту та покращення його ефективності, зокрема, для збільшення конверсії та покращення користувацького досвіду. Інтерфейс програми наведений на рисунку 1.2.

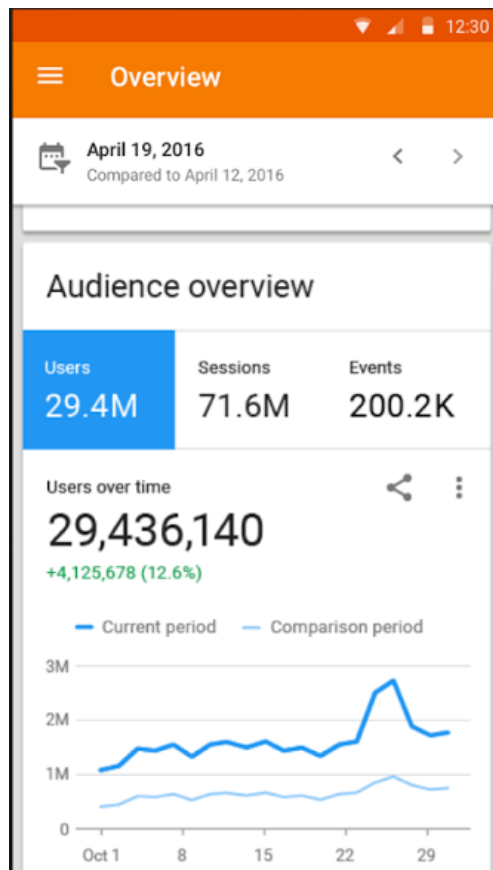


Рисунок 1.2 – Інтерфейс Google Analytics

Google Analytics працює з використанням JavaScript-коду, який вбудовується на сторінки веб-сайту. Коли користувач відвідує сторінку, браузер завантажує JavaScript-код Google Analytics і виконує його.

JavaScript-код збирає різну інформацію про користувача та його поведінку на сайті, таку як:

- інформація про браузер та операційну систему користувача;
- URL-адреса, з якої користувач прибув на сайт;
- інформація про дату та час відвідування сторінки;
- інформація про те, як користувач взаємодіє зі сторінкою (наприклад, кліки, наведення курсору тощо);
- інформація про те, які сторінки на сайті відвідує користувач та як довго він перебуває на кожній з них;
- інформація про те, як користувач повернувся на сайт у майбутньому.

Отримана інформація передається на сервер Google Analytics, де вона обробляється та зберігається. Після цього веб-власник може переглядати зібрану інформацію в аналітичному інтерфейсі Google Analytics та отримувати звіти про поведінку користувачів на своєму сайті.

Важливо зазначити, що для коректної роботи Google Analytics необхідно виконати правильну настройку інструмента та вбудувати його JavaScript-код на всі необхідні сторінки веб-сайту.

### **1.3.3 Qlik Sense Mobile**

Qlik Sense Mobile – це мобільний додаток для роботи з бізнес-аналітикою, розроблений компанією Qlik. Він дає змогу отримувати доступ до аналітичних даних з будь-якого місця та на будь-якому пристрої з iOS або Android.

Завдяки Qlik Sense Mobile користувачі можуть переглядати та аналізувати дані, ділитися даними з колегами та приймати управлінські рішення, використовуючи свої смартфони або планшети. Додаток має інтуїтивно зрозумілий інтерфейс, що дозволяє зручно переглядати дані та



проводити аналіз.

Крім того, Qlik Sense Mobile надає можливість підключення до різноманітних джерел даних, включаючи бази даних, електронні таблиці, CRM-системи та інші джерела. Додаток підтримує використання функцій аналізу даних, таких як візуалізація, фільтрація, пошук та інші. На рисунку 1.3 ми можемо побачити інтерфейс даної програми.



Рисунок 1.3 – Інтерфейс Qlik Sense Mobile

Qlik Sense Mobile є частиною платформи Qlik Sense, що дозволяє створювати складні звіти та аналітичні додатки. З його допомогою користувачі можуть переглядати та аналізувати дані, які були створені на основі Qlik Sense, а також створювати свої власні звіти та дашборди.

## 1.4 Висновки до першого розділу

На основі аналізу існуючих програм, таких як Qlik Sense Mobile, Google Analytics та IBM Cognos Analytics, було отримано цінну інформацію про те, які функції та можливості можна включити в мобільний додаток для аналізу бізнес-даних. Відомості, зібрані в першому розділі, будуть корисні при проектуванні та розробці мобільного додатку, оскільки вони надають важливі вказівки щодо того, які функції можна реалізувати, які інтерфейси користувача можуть бути ефективними та які технології можна використовувати для забезпечення оптимальної продуктивності та безпеки даних. В цілому, перший розділ є важливим етапом у розробці мобільного додатку для аналізу бізнес-даних, оскільки він надає цінну інформацію та напрямки для подальшої роботи. Весь цей аналіз є важливим для успішної розробки мобільного додатку для аналізу бізнес-даних.

## 2 ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

### 2.1 Вибір мови програмування

Існує багато мов програмування, які можна використовувати для розробки мобільних додатків. Розробка мобільних додатків стала надзвичайно популярною в останні роки, і з кожним днем з'являється все більше та більше мов програмування, які можна використовувати для цієї мети. Однак, вибір правильної мови програмування є ключовим фактором успіху в розробці мобільного додатку, тому перед початком роботи слід ретельно обміркувати всі переваги та недоліки кожної мови, щоб знайти ту, яка найкраще відповідає вашим потребам.

Один з найважливіших факторів при виборі мови програмування для розробки мобільного додатку – це платформа, на якій додаток буде запущено. Якщо ви плануєте розробляти додаток для iOS, ви можете використовувати мови, такі як Swift або Objective-C. Для розробки додатку для Android можна використовувати мови, такі як Java або Kotlin. Якщо ж ви плануєте розробити додаток для обох платформ, вам може знадобитись кросплатформова мова програмування, така як JavaScript або Dart.

Інший фактор, який слід врахувати – це тип додатка, який ви плануєте розробити. Якщо додаток вимагає обробки великої кількості даних, то мови програмування, такі як Python, можуть бути найбільш підходящими. Якщо ж ви працюєте з графікою, то вам можуть знадобитись мови, такі як C++ або Java.

Розглянемо деякі з вищеперерахованих мов.

**Java** – спочатку була офіційною мовою для розробки додатків для Android (але тепер її замінив Kotlin), і, отже, вона також є найбільш використовуваною мовою. Багато програм у магазині Play створено на Java, і це також найбільш підтримувана мова Google [2]. На додаток до всього

цього, Java має чудову онлайн-спільноту для підтримки у разі будь-яких проблем.

**Kotlin** – зараз є офіційною мовою для розробки додатків для Android, оголошеною Google у 2019 році. Kotlin – це кросплатформна мова програмування, яка може використовуватися як альтернатива Java для розробки додатків для Android [3]. Єдина значна відмінність полягає в тому, що Kotlin видаляє зайві функції Java, такі як винятки нульового покажчика. Це також усуває необхідність закінчувати кожен рядок крапкою з комою. Коротше кажучи, початківцям набагато простіше випробувати Kotlin порівняно з Java, і його також можна використовувати як «точку входу» для розробки додатків для Android.

C++ – можна використовувати для розробки додатків Android за допомогою Android Native Development Kit (NDK) [4]. Однак програму неможливо створити повністю за допомогою C++, і NDK використовується для реалізації частин програми у рідному коді C++. Це допомагає використовувати бібліотеки коду C++ для програми за потреби.

Хоча в деяких випадках C++ корисний для розробки додатків для Android, його набагато складніше налаштувати та він набагато менш гнучкий. Це також може призвести до збільшення кількості помилок через підвищену складність. Отже, краще використовувати Java порівняно з C++, оскільки вона не забезпечує достатнього приросту, щоб компенсувати необхідні зусилля.

**Javascript** – програми для Android можна створювати за допомогою HTML, CSS і JavaScript, за допомогою фреймворку Adobe PhoneGap, який підтримує Apache Cordova [5]. Фреймворк PhoneGap в основному дозволяє використовувати навички веб-розробки для створення гібридних програм, які відображаються через «WebView», але упаковані як програма.

Хоча фреймворку Adobe PhoneGap достатньо для базових завдань у сфері розробки програм для Android, він навряд чи потребує багато програмування, окрім JavaScript. І оскільки навіть для створення пристойної

програми потрібно багато працювати, краще використовувати інші мови в цьому списку.

**Dart** – це мова програмування з відкритим вихідним кодом, яка базується на фреймворку Flutter, який сьогодні набуває великої популярності завдяки своїй здатності створювати красиві та продуктивні програми для Інтернету, комп'ютера та мобільних пристроїв за менший час [6]. Ключова перевага dart полягає в тому, що він розроблений Google як клієнтська оптимізована мова для швидких програм на будь-якій платформі. Dart головним чином зосереджується на полегшенні розробки інтерфейсу користувача для розробників за допомогою таких функцій, як гаряче перезавантаження, що дозволяє розробникам миттєво бачити зміни під час роботи над програмою. Dart також відомий своєю високою продуктивністю, він компілюється в машинний код ARM і x64 для мобільних пристроїв, комп'ютерів і серверної частини. І до JavaScript для веб-програм.

За версією української IT-спільноти – Dou.ua, найпоширенішими мовами програмування для комерційного проекту є мови JavaScript, Java та Python. На рисунку 2.1 можемо побачити рейтинг [7].

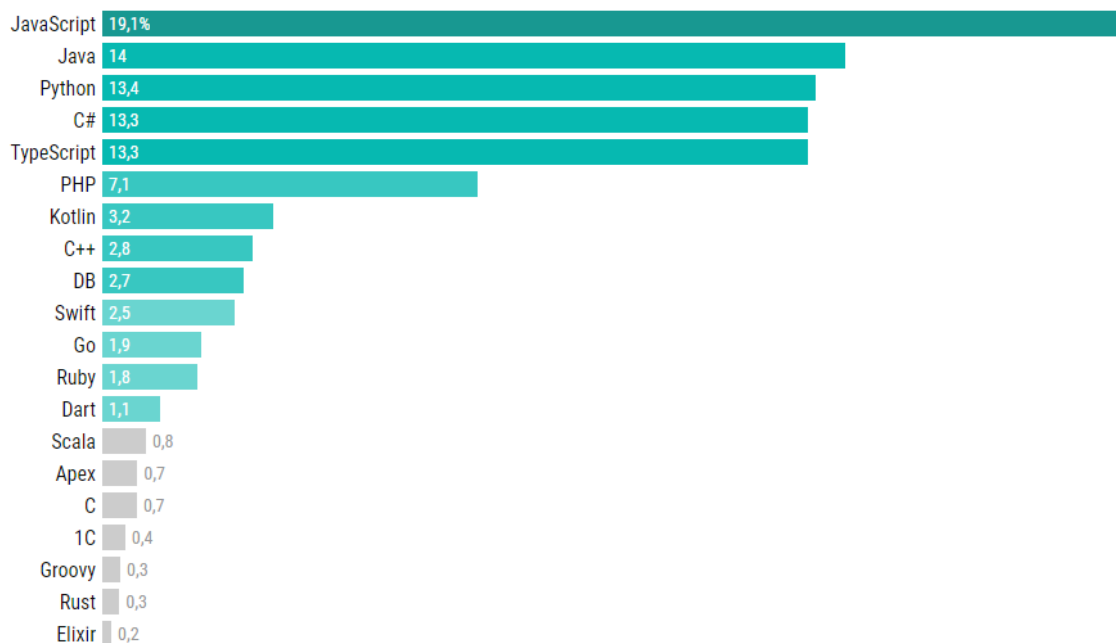


Рисунок 2.1 – Рейтинг мов програмування за використанням

Враховуючи вище зазначену інформацію, ми зупинимося на мові Java, як на найбільш оптимальному інструменті для нашого проєкту.

## 2.2 Бібліотека для візуалізації

Серед усіх бібліотек, які знадобилися б нам для візуалізації даних, я би виділив MPAndroidChart.

MPAndroidChart – це бібліотека для відображення даних у вигляді графіків на платформі Android. Вона надає широкі можливості для створення різноманітних графічних елементів, таких як лінійні графіки, стовпчасті діаграми, колові діаграми, розсіювальні графіки та багато інших.

Ця бібліотека дозволяє налаштовувати кожен аспект графіка, включаючи його вигляд, кольори, форматування міток осей та легенди. Вона також підтримує інтерактивні можливості, такі як масштабування та переміщення по графіку.

### *Переваги використання бібліотеки MPAndroidChart:*

- широкий вибір типів графіків: бібліотека надає можливість створення різноманітних графічних елементів, включаючи лінійні графіки, стовпчасті діаграми, колові діаграми, розсіювальні графіки та інші;
- налаштування графіків: бібліотека дозволяє налаштовувати кожен аспект графіка, включаючи його вигляд, кольори, форматування міток осей та легенди;
- інтерактивні можливості: бібліотека підтримує зумування та переміщення по графіку;
- документація та підтримка: бібліотека має докладну документацію та підтримку, що дозволяє розробникам швидко зрозуміти, як використовувати її функціональність;
- відкритий код: бібліотека є відкритою та безкоштовною для використання в комерційних проєктах.

### *Недоліки використання бібліотеки MPAndroidChart:*

- ресурсомісткість: створення складних графіків може вимагати значних ресурсів, що може вплинути на продуктивність додатку;
- навчання: для користування бібліотекою потрібно володіти базовими знаннями з програмування на платформі Android;
- особливості взаємодії з іншими компонентами: іноді можуть виникати проблеми з взаємодією графіків, створених з використанням бібліотеки MPAndroidChart, з іншими компонентами додатку;
- обмежена функціональність: хоча бібліотека має досить широкий вибір типів графіків, вона може мати обмежений функціонал для деяких специфічних вимог до графіків, які можуть виникнути в додатках;
- необхідність забезпечення сумісності: перед використанням бібліотеки потрібно врахувати її сумісність з версією Android, на якій працює додаток;
- обмеження на мобільних пристроях: мобільні пристрої мають обмежену площу екрана, що може становити проблему для відображення складних графіків.

## **2.3 Середовище розробки**

Важливим фундаментом майбутнього проекту є середовище розробки. Для системи Android існує нативне середовище, яке дуже добре підтримує мову Java та усі її бібліотеки – Android SDK.

Android SDK (Software Development Kit) – це набір інструментів для розробки програмного забезпечення для платформи Android. Він містить різноманітні компоненти, такі як бібліотеки, засоби розробки і документацію, що дозволяє розробникам створювати, тестувати та налагоджувати Android-додатки.

SDK містить в собі середовище розробки (IDE) Android Studio, а також

набір інструментів для будування, збірки і пакування додатків. Крім того, містить емулятор Android, який дозволяє розробникам тестувати свої додатки на різних пристроях і версіях операційної системи Android без необхідності володіти фізичними пристроями.

## **2.4 Вибір бази даних**

Обирання правильної бази даних є важливим кроком в розробці будь-якої програмної системи, що потребує зберігання та організації даних. Вибір бази даних впливає на продуктивність, масштабованість та зручність у використанні програмної системи.

### **2.4.1 Реляційна модель бази даних**

Реляційна модель бази даних є однією з найбільш поширених та ефективних моделей для зберігання даних. Вона базується на принципах математичної теорії множин та логіки першого порядку і забезпечує зручний та ефективний доступ до даних.

У реляційній моделі дані зберігаються у вигляді таблиць, кожна з яких містить набір стовпців та рядків. Кожен стовпець таблиці представляє атрибут даних, тоді як кожен рядок відображає запис, який містить значення цих атрибутів. Кожна таблиця має унікальний ідентифікатор (ключ), який дозволяє ідентифікувати кожен запис.

Одна з головних переваг реляційної моделі бази даних полягає в тому, що вона дозволяє виконувати складні запити та взаємодії між даними шляхом використання зв'язків між таблицями. Наприклад, можна виконати запит, щоб отримати всі записи в таблиці А, які мають відповідні записи в таблиці В, що відповідають певним критеріям.



Для роботи з реляційними базами даних використовуються спеціальні мови запитів, такі як SQL (Structured Query Language), який дозволяє взаємодіяти з базою даних та виконувати операції, такі як створення таблиць, додавання та видалення даних, зміна структури таблиць та інше.

Однією з основних переваг реляційної моделі баз даних є її нормалізація, яка дозволяє ефективно зберігати та управляти даними відповідно до їх структури. Це допомагає уникнути дублювання даних та забезпечує їх консистентність та інтегритет.

Однак, реляційна модель бази даних також має свої недоліки, такі як обмежена масштабованість та швидкодія при обробці великих обсягів даних. Також, вона не завжди підходить для зберігання даних, що мають складну структуру, наприклад, зображення, відео та інші мультимедійні дані.

#### **2.4.2 Нереляційна модель бази даних**

Нереляційна модель бази даних є альтернативою реляційній моделі та забезпечує зберігання даних у більш гнучкому та складному форматі. Ця модель баз даних дозволяє зберігати та управляти даними, які мають складну структуру та не можуть бути легко представлені у вигляді таблиць з реляційними зв'язками.

Нереляційні бази даних забезпечують зберігання даних у вигляді документів, графів, ключ-значення та інших форматах. Наприклад, документо-орієнтована база даних дозволяє зберігати дані у вигляді JSON-документів, а графова база даних – у вигляді графів з вершинами та зв'язками між ними.

Однією з переваг нереляційної моделі є її гнучкість та можливість швидкого змінювати структуру даних без необхідності виконання складних процедур нормалізації. Нереляційні бази даних також дозволяють більш ефективно зберігати дані, які мають складну структуру, такі як

мультимедійні дані, геодані та інші дані, які не підходять для зберігання у вигляді таблиць.

Однак, нереляційна модель має свої недоліки. Оскільки вона не має жорстких правил відносно структури даних, то можуть виникати проблеми з консистентністю та цілісністю даних. Також, вона може бути менш ефективною при виконанні деяких операцій з даними, таких як запити зі складними умовами та агрегаціями.

У певних випадках, нереляційні бази даних можуть бути кращим вибором, особливо якщо даними необхідно швидко маніпулювати та якщо вони мають складну структуру.

## 2.5 SQLite

Для розробки мобільного додатка для аналізу бізнес-даних, знадобиться база даних для зберігання, організації та опрацювання даних.

Зважаючи на те, що ми розробляємо додаток для Android, нам може підійти локальна база даних SQLite, яка є вбудованою в Android SDK та забезпечує зручний та ефективний спосіб для зберігання та опрацювання даних в мобільних додатках [8].

Використання MPAndroidChart дозволить вам відображати графіки та діаграми, використовуючи дані з вашої бази даних.

SQLite – це вбудовувана реляційна база даних, яка працює на рівні файлової системи. Вона зберігає дані у вигляді одного файлу в системі та не вимагає окремого сервера баз даних для своєї роботи. SQLite підтримує більшість стандартних операцій з базами даних, такі як SQL-запити, індексування, транзакції, тригери та багато іншого. Вона легка у використанні та має невисокі вимоги до ресурсів системи, що робить її популярним вибором для вбудовуваної бази даних в додатки та пристрої з обмеженими можливостями. SQLite є безкоштовним та доступним для

використання в комерційних проєктах.

## 2.6 Діаграма прецедентів

Діаграма прецедентів – це графічне зображення взаємодії між користувачами та системою в процесі розробки програмного забезпечення. Ця діаграма містить інформацію про функціональні вимоги до системи (прецеденти) та осіб або інших систем, які будуть взаємодіяти з системою (актори).

З допомогою діаграми прецедентів можна проаналізувати потреби користувачів та зрозуміти, які функції повинна виконувати система, щоб задовольнити їх потреби. Кожен прецедент містить назву та опис дії, яку повинна виконувати система для задоволення потреб користувачів. Актори можуть бути людьми, іншими системами або зовнішніми процесами, що взаємодіють з системою.

Акторами в даній системі будуть виступати Користувач та Сервер Google Drive.

Користувач – це основний актор, який взаємодіє з системою. Користувач авторизується у системі, вибирає файли або папки з Google Drive, завантажує дані, виконує операції над ними та зберігає результати.

Сервер Google Drive – це зовнішній системний компонент, який забезпечує доступ до файлів та папок на Google Drive. Користувач взаємодіє з сервером Google Drive, авторизується та отримує доступ до своїх даних.

Таким чином, акторська структура системи включає як мінімум два актори: користувача і сервер Google Drive. Користувач є основним актором, який здійснює взаємодію з системою, а сервер Google Drive виступає як зовнішній компонент, який забезпечує доступ до даних з Google Drive.

Рисунок 2.2 є представленням діаграми прецедентів.



Рисунок 2.2 – Діаграма прецедентів

Розглянемо детально кожен прецедент:

Авторизація:

- користувач вводить свої облікові дані Google;
- користувач надає дозвіл додатку на доступ до своїх даних на Google Drive.

Вибір файлу/папки:

- користувач переглядає свої файли та папки на Google Drive;
- користувач вибирає конкретний файл або папку для роботи.

Завантаження даних:

- користувач вибирає файл з Google Drive, з якого він хоче завантажити дані;
- додаток завантажує вміст вибраного файлу з Google Drive.

Оновлення даних:

- користувач може оновити дані у вибраному файлі на Google Drive;
- користувач може синхронізувати зміни, внесені в додатку, з

## Google Drive.

### Збереження даних:

- користувач може зберегти результати аналізу або обробки даних у файл на Google Drive;
- додаток створює новий файл або оновлює вже існуючий файл на Google Drive з результатами.

### Обробка даних:

- користувач може застосувати фільтри або умови до набору даних, щоб відібрати тільки певні записи, які задовольняють певні критерії;
- користувач може створити графіки, діаграми або інші візуальні елементи для відображення даних зрозумілим способом;
- користувач може групувати дані за певними категоріями або критеріями для подальшого аналізу.

### Фільтрація:

- користувач може застосувати фільтри або умови до набору даних, щоб відібрати тільки певні записи, які задовольняють певні критерії;
- користувач може сортувати дані за певними стовпцями або полями, щоб отримати їх у певному порядку, наприклад, за зростанням або спаданням значень.

### Перегляд звіту:

- користувач може вибрати конкретний звіт, який він бажає переглянути (це може бути список доступних звітів або можливість вибору з раніше збережених звітів);
- після вибору звіту система відображає його на екрані користувача (це може бути у вигляді таблиці, графіка, діаграми або будь-якого іншого формату, який найкраще відповідає вмісту звіту);
- залежно від функціональності системи, користувач може мати можливість взаємодіяти з звітом (наприклад, він може масштабувати графіки, фільтрувати дані або переглядати подробиці на певній частині звіту);

- користувач може мати можливість зберегти звіт для майбутнього використання або експортувати його у різні формати, такі як PDF, Excel або CSV.

Видалення звіту:

- користувач вибирає конкретний звіт, який він бажає видалити (це може бути список доступних звітів або можливість вибору з раніше збережених звітів);
- підтвердження видалення: система попросить користувача підтвердити своє рішення щодо видалення звіту (це забезпечує запобігання випадковому видаленню та забезпечує впевненість користувача в своєму виборі);
- після підтвердження видалення, система видаляє звіт зі своєї бази даних або зберігається в архіві, якщо це необхідно для збереження історії.

## 2.7 Діаграма послідовностей

UML-діаграма (Уніфікована Мова Моделювання) – це графічний інструмент, який дозволяє розробникам візуалізувати, спроектувати та аналізувати структуру та поведінку системи перед її реалізацією.

Це потужний засіб для моделювання програмного забезпечення, що допомагає команді розробників уявити систему як цілісний об'єкт та зрозуміти взаємозв'язки між її складовими частинами.

На рисунку 2.3 приклад діаграми для нашого доданку.

На рисунку видно як працює програма. Користувач входить в систему та створює звіт, шляхом імпорту даних з потрібної таблиці в Google Drive.



частіше оцінюють додатки не лише за їх функціоналом, а й за їх зовнішнім виглядом. Мінімалістичний дизайн, використання актуальних трендів у графіці та кольорах, відповідність бренду компанії – це все впливає на перше враження та загальну задоволеність від використання додатку.

Третій аспект – це адаптивність дизайну до різних пристроїв та розмірів екранів. Оскільки розміри та пропорції екранів мобільних пристроїв різняться, дизайн додатку повинен бути адаптований до різних роздільних здатностей та форматів екранів. Це забезпечить однаково комфортний досвід користувача незалежно від того, на якому пристрої він використовує додаток.

Крім того, важливо враховувати психологічні аспекти дизайну мобільних додатків. Наприклад, використання певних кольорів може викликати певні емоції у користувачів. Так, червоний колір може викликати почуття невпевненості або тривоги, тоді як зелений – спокій та впевненість. Крім того, важливо враховувати принципи доступності дизайну для людей з різними видами обмежень.

У процесі розробки дизайну мобільних додатків важливо враховувати потреби та очікування цільової аудиторії. Наприклад, додаток для підлітків може мати яскравий та молодіжний дизайн, тоді як додаток для фахівців може мати більш консервативний вигляд. Важливо враховувати також місце застосування додатку – якщо додаток призначений для використання в умовах з високим рівнем освітлення, дизайн повинен бути адаптований до таких умов.

У підсумку, ефективний дизайн мобільних додатків повинен бути простим, зрозумілим, естетичним, адаптивним та враховувати психологічні та соціальні аспекти. Такий дизайн дозволить забезпечити комфортний та приємний користувацький досвід та зробить додаток популярним серед користувачів.

Не будемо придумувати велосипед, та візьмемо перевірений часом шаблон, на рисунку 2.4 побачимо макет стартової сторінки мобільного додатку розробленого у Figma [10].



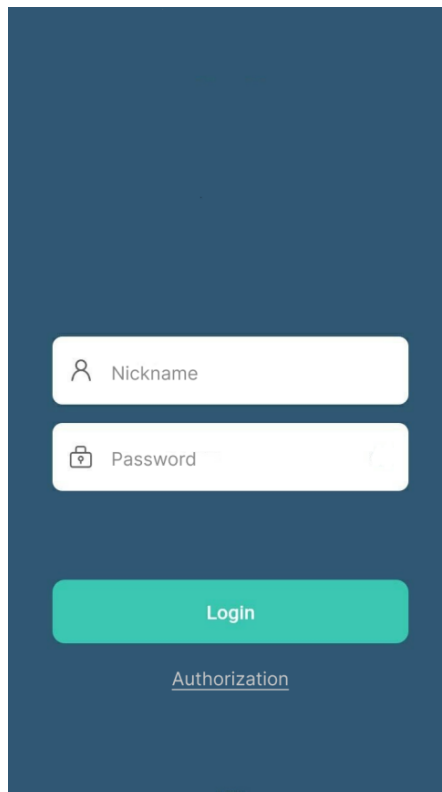


Рисунок 2.4 – Макет стартової сторінки

Figma – це веб-платформа та інструмент для дизайну і прототипування інтерфейсів користувача. Вона надає можливості для створення графічних елементів, макетів, прототипів та спільної роботи над дизайн-проектами в режимі реального часу.

### 2.8.1 Логотип мобільного додатку

Логотип є важливою складовою успіху будь-якого мобільного додатку. Він виступає символічним образом, який ідентифікує додаток та відрізняє його від інших на ринку. Важливість логотипу для мобільного додатку полягає в аспектах, що наведені нижче.

**Розпізнаваність бренду:** логотип є ключовим елементом брендингу додатку. Він створює унікальний візуальний ідентифікатор, який допомагає користувачам легко розпізнавати додаток серед інших. Сильний логотип

здатний викликати емоції, асоціації та пам'ять, що допомагає збудувати впізнаваність бренду серед аудиторії.

**Передача повідомлення:** логотип може відтворювати основні цінності та повідомлення додатку. Через використання візуальних елементів, кольорів, шрифтів та символів, логотип може передавати інформацію про характер та функціонал додатку. Він може викликати емоції, залучати увагу та зацікавлення користувачів.

**Професійний вигляд:** логотип допомагає створити враження про додаток та його рівень професійності. Якщо логотип виглядає добре спроектованим і професійним, це може позитивно вплинути на сприйняття додатку користувачами. Логотип, який викликає довіру та надійність, може стимулювати користувачів встановити та використовувати додаток.

**Маркетинговий інструмент:** логотип може бути використаний в рекламних матеріалах, прес-релізах, веб-сайтах та соціальних мережах для просування додатку. Він стає візуальним символом, що викликає інтерес та привертає увагу потенційних користувачів. Добре відомий логотип може стати розпізнаваним брендом, що сприяє популярності та визнанню додатку на ринку.

**Відображення профілю аудиторії:** логотип може передати інформацію про цільову аудиторію додатку. Наприклад, через вибір кольорів, шрифтів та стилістики, логотип може відображати характер або інтереси користувачів, що спонукає їх встановити додаток та використовувати його.

**Відрізнення від конкурентів:** у конкурентному світі мобільних додатків, логотип може допомогти виділитись серед схожих рішень та привернути увагу користувачів. Сильний та оригінальний логотип дозволяє створити унікальний образ та впізнаваність, що може бути ключовим фактором в розрізненні від конкурентів.

Усі ці аспекти підкреслюють важливість логотипу для мобільного додатку. Він створює не лише візуальне враження, а й викликає емоції, передає повідомлення та сприяє впізнаваності додатку. Належна увага до

дизайну та розробки логотипу може мати значний вплив на успіх мобільного додатку та сприяти його популярності серед користувачів.

Отже, взявши до уваги усе вище зазначене, ми розробили дизайн логотипу для бізнес-аналітики, рисунок 2.5.

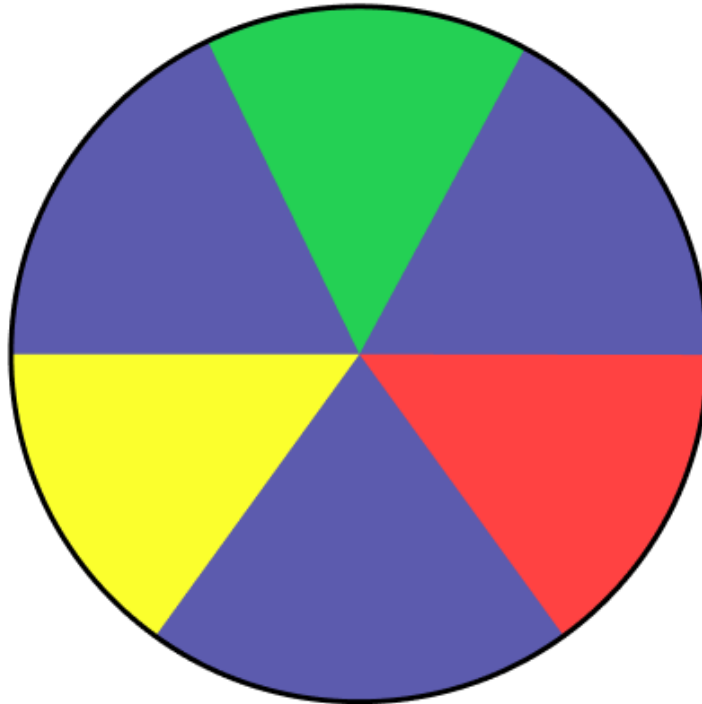


Рисунок 2.5 – Логотип додатку що розробляється

Як ми бачимо, логотип нагадує кругову діаграму, знак радіації та вид зонтика зверху. Знак радіації не так часто зустрічається в нашому житті, на відміну від зонтика або кругової діаграми, тому можемо не звертати увагу на «негативні асоціації». Кольори приглушені, щоб не дратувати зір насиченістю.

## 2.9 Висновки до другого розділу

За час виконання цього розділу ми спроектували образ системи мобільного додатку для аналізу бізнес-даних. Визначили технології з якими

будемо працювати, визначили як буде відбуватися взаємодія між користувачем та системою нашого додатку. Обрали найбільш зручну базу даних. Вирішили яка саме мова програмування нам знадобиться. Визначилися с платформою для розробки. Створили дизайн логотипу та макет сторінок у веб-застосунку Figma.

## 3 РОБОТА МОБІЛЬНОГО ДОДАТКУ

### 3.1 Опис інструментів розробки

Для реалізації додатку був використаний фреймворк Spring. Мова програмування – Java.

База даних – SQLite. Взаємодія с додатком відбувається за допомогою драйвера Java Database Connectivity.

Бібліотека візуалізації – MPAndroidChart

### 3.2 Тестування

Нижче наведені скріншоти роботи мобільного додатку на Android версії 7.0.

Перше що побачить Користувач це сторінку з полями “Nickname” та ”Password” (див. рис. 3.1) . Логін потрібно писати англійською мовою. Максимальна довжина паролю 16 символів.

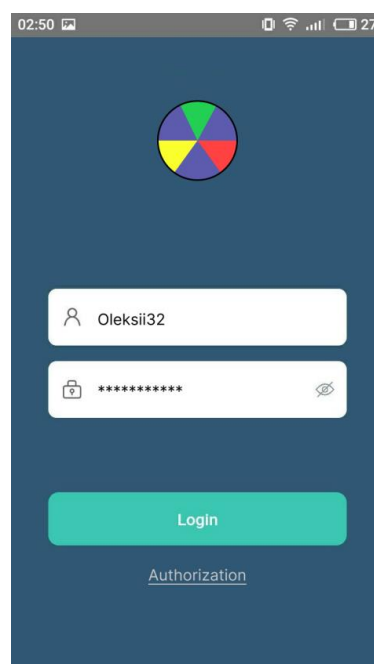
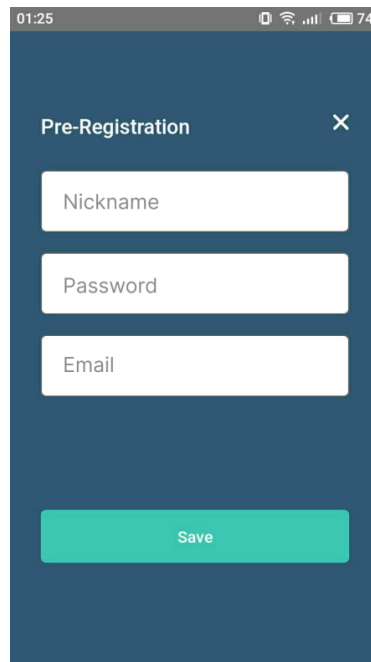


Рисунок 3.1 – Стартова сторінка додатку

У випадку якщо не реєструвалися до цього, натиснемо “Authorization”.  
Нас перекине на сторінку реєстрації (див. рис. 3.2).



01:25

Pre-Registration

Nickname

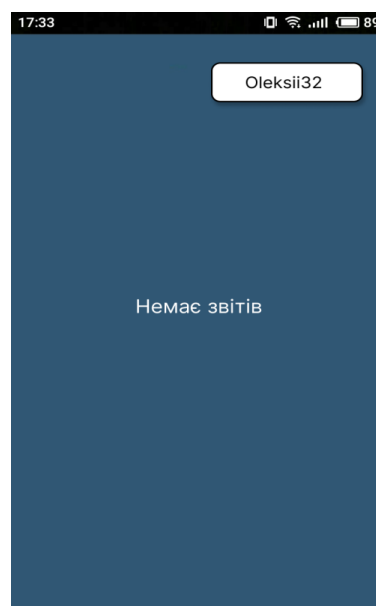
Password

Email

Save

Рисунок 3.2 – Сторінка реєстрації

Якщо ж профіль в додатку вже є, ми просто вписуємо свій пароль та логін та натискаємо кнопку Login. Нас перекине на сторінку звітів профілю (див. рис. 3.3).



17:33

Oleksii32

Немає звітів

Рисунок 3.3 – Сторінка профілю із звітами

Для того щоб переглянути, додати чи видалити звіт, потрібно викликати бокову панель, для цього “прогорнемо” пальцем по екрану зліва направо (див. рис. 3.4).

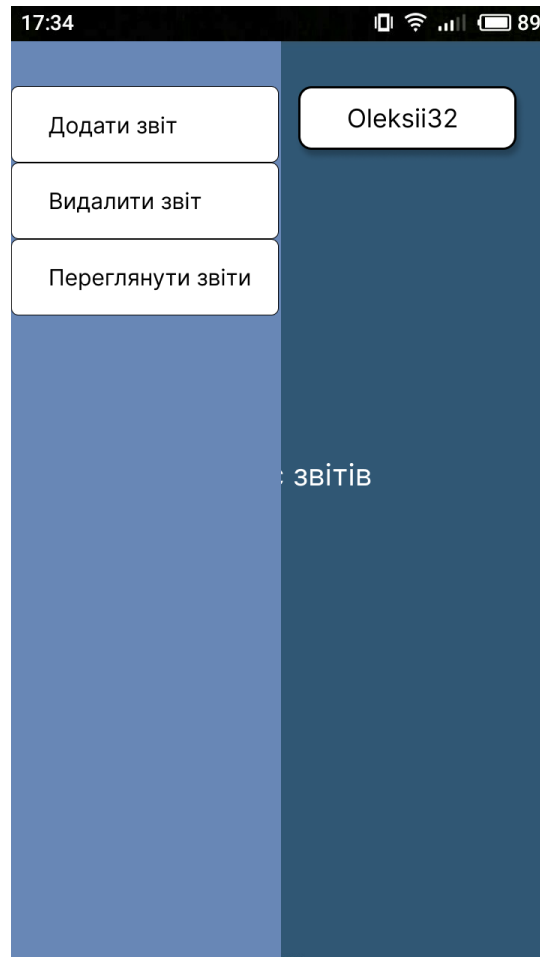


Рисунок 3.4 – Бокова панель

Додамо звіт в профіль, для цього натиснемо кнопку “Додати звіт”. Побачимо вигулькове вікно з двома полями (див. рис. 3.5).

Перше поле означає що потрібно вписати назву excel-файлу з якого додаток буде брати дані. Excel-файл повинен бути у головній директорії хмарного сховища Google Drive, назва файлу обов’язково англійською мовою.

Друге поле означає, що потрібно вписати діапазон комірок excel-таблиці, саме з них і будемо брати дані. Кнопка Add зберігає вибір, та відправляє запит.

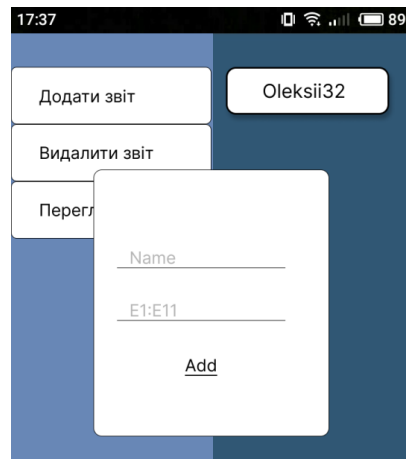


Рисунок 3.5 – Вікно додавки звіту

Для прикладу, створили excel-file з назвою FebruaryReport. Ввели цю назву у вікно Add, та вказали комірку в якій знаходяться потрібні дані.

На сторінці профілю з'являться 3 панелі (див. рис. 3.6).

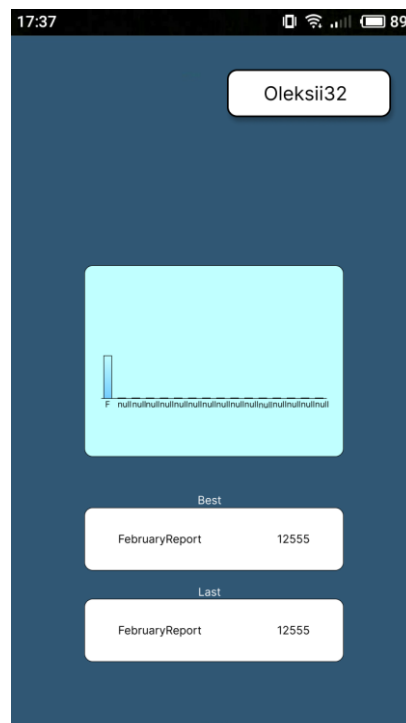


Рисунок 3.6 – Приклад візуалізації даних звіту

Верхня – стовпчаста діаграма, вона візуально порівнює дані експортовані з Google Drive. Так як у нас завантажений лише один звіт, інші стовпці підписані як null, коли завантажений звіт підписаний першою



літерою назви звіту.

Середня показує який звіт має найбільші дані (в контексті бізнесу означає найбільший прибуток чи кількість продажів).

Нижня показує останній завантажений звіт та його дані.

Для порівняння додамо ще один звіт, назовемо його MarchReport, але дані поставимо менші ніж в FebruaryReport. І як бачимо на рисунку 3.7 панель Last змінила інформацію, так як останнім завантаженим звітом став MarchReport.

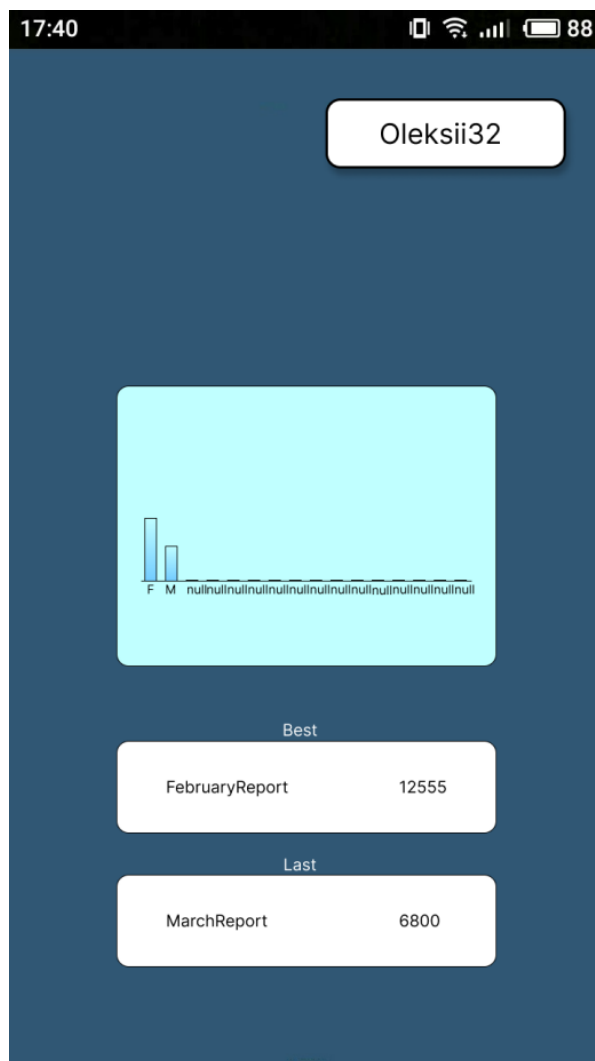


Рисунок 3.7 – Порівняльний аналіз двох звітів

На сторінці “Переглянути звіти” продивимось додані звіти (див. рис. 3.8).

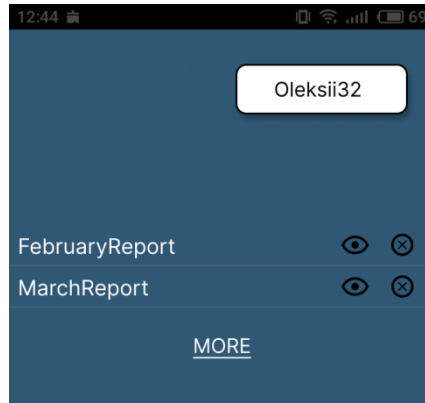


Рисунок 3.8 – Список доданих звітів

Натиснувши “Перегляд звіту” (значок ока). Потрапляємо на сторінку перегляду звіту, з даними, щодо кожного дня. Оскільки в лютому тільки 28 днів, тобто даними заповнені тільки 28 комірок, інші дні будуть позначатися як Null (див. рис. 3.9).

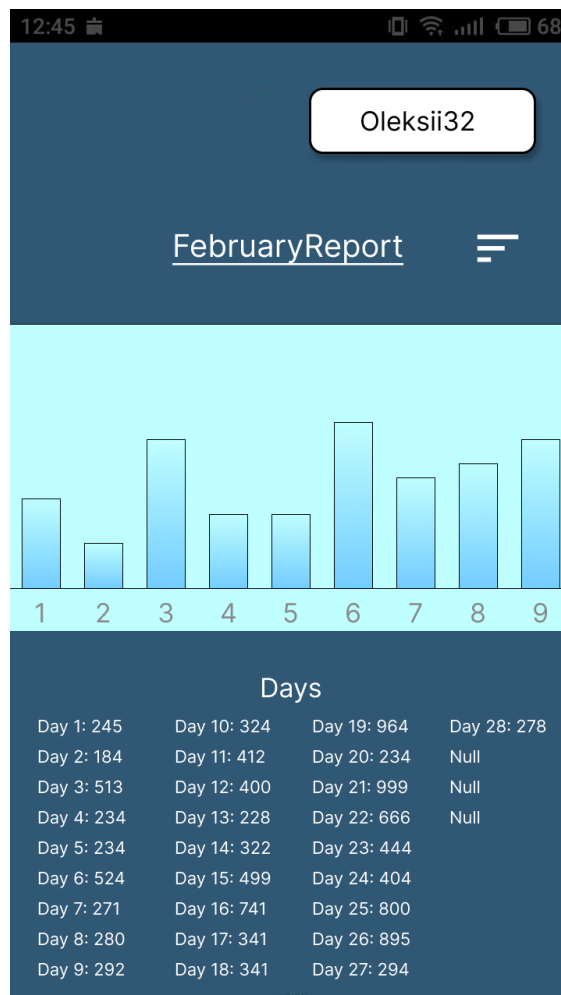


Рисунок 3.9 – Перегляд звіту FebruaryReport

## ВИСНОВКИ

У результаті досліджувальної роботи виявили, що для створення мобільного додатка для аналізу бізнес-даних найкращим підходом є використання об'єктно-орієнтованих методів розробки. Врахували основні етапи життєвого циклу інформаційної системи, такі як аналіз, проєктування та реалізація.

Реалізована система порівняльного аналізу для імпортованих даних.

Для проєктування системи розробили специфікації згідно з моделлю складної системи. Це дозволило представити цільову інформаційну систему у вигляді таких моделей, як логічна, функціональна.

Політика проєктування інтерфейсу передбачає максимальну зручність використання мобільного додатка і мінімізацію складності інформації та об'єктів на сторінках додатка. Намагалися зробити його інтуїтивно зрозумілим для користувачів.

За час розробки, були використані мова програмування Java. Бібліотека для візуалізації MPAndroidChart. Середовище розробки Android SDK.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Albright C., Winston W. Business Analytics: Data Analysis and Decision Making. Boston : Cengage Learning, 2019. 441 p.
2. Bloch J. Effective Java. Boston : Addison-Wesley Professional, 2018. 330 p.
3. Leiva A. Kotlin for Android Developers. California : Leanpub, 2017. 422 p.
4. Meyers S. Effective Modern C++. California : O'Reilly, 2014. 748 p.
5. Flanagan D. JavaScript: The Definitive Guide. California : O'Reilly, 2020. 531 p.
6. Balbaert I. Learning Dart. Birmingham : Packt Publishing, 2014. 402 p.
7. Рейтинг мов програмування 2023. URL: <https://dou.ua/lenta/articles/language-rating-2023/?from=slider> (дата звернення: 10.05.2023).
8. Forta B. SQL in 10 Minutes a Day, Sams Teach Yourself. Indianapolis : Sams Publishing, 2021. 546 p.
9. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. Berkeley : New Riders, 2014. 544 p.
10. Онлайн ресурс Figma. URL: <https://www.figma.com> (дата звернення: 08.05.2023).

## ДОДАТОК А

### Код програмного забезпечення

#### A.1 DataOut.java

```
import com.google.api.client.auth.oauth2.Credential;
import
com.google.api.client.extensions.java6.auth.oauth2.AuthorizationCodeInstalledAp
p;
import com.google.api.client.extensions.jetty.auth.oauth2.LocalServerReceiver;
import
com.google.api.client.googleapis.auth.oauth2.GoogleAuthorizationCodeFlow;
import com.google.api.client.googleapis.auth.oauth2.GoogleClientSecrets;
import com.google.api.client.googleapis.javanet.GoogleNetHttpTransport;
import com.google.api.client.http.HttpTransport;
import com.google.api.client.json.JsonFactory;
import com.google.api.client.json.jackson2.JacksonFactory;
import com.google.api.client.util.store.FileDataStoreFactory;
import com.google.api.services.drive.Drive;
import com.google.api.services.drive.DriveScopes;
import com.google.api.services.drive.model.File;
import com.google.api.services.drive.model.FileList;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
```

```

import java.security.GeneralSecurityException;
import java.util.Collections;
import java.util.List;
public class ReadExcelFromGoogleDrive {
    private static final String APPLICATION_NAME = "Google Drive API Java
Quickstart";
    private static final JsonFactory JSON_FACTORY =
JacksonFactory.getDefaultInstance();
    private static final String CREDENTIALS_FOLDER = "credentials";
    private static final String CLIENT_SECRET_FILE_NAME =
"client_secret.json";
    private static final List<String> SCOPES =
Collections.singletonList(DriveScopes.DRIVE_READONLY);
    private static final String TOKENS_DIRECTORY_PATH = "tokens";
    private static GoogleClientSecrets loadClientSecrets() throws IOException {
        InputStream in =
ReadExcelFromGoogleDrive.class.getResourceAsStream("/")
CLIENT_SECRET_FILE_NAME);
        return GoogleClientSecrets.load(JSON_FACTORY, new
InputStreamReader(in));
    }
    private static Credential authorize(HttpTransport httpTransport) throws
IOException {
        GoogleClientSecrets clientSecrets = loadClientSecrets();
        GoogleAuthorizationCodeFlow flow=new
GoogleAuthorizationCodeFlow.Builder(
            httpTransport, JSON_FACTORY, clientSecrets, SCOPES)
            .setDataStoreFactory(new FileDataStoreFactory
Credential credential = new AuthorizationCodeInstalledApp(
            flow, new LocalServerReceiver()).authorize("user");

```

```

    return credential;
}
private static Drive getDriveService(HttpTransport httpTransport, Credential
credential) throws IOException {
    return new Drive.Builder(
        httpTransport, JSON_FACTORY, credential)
        .setApplicationName(APPLICATION_NAME)
        .build();
}
public static void readExcelFile(String fileId) throws GeneralSecurityException,
IOException {
    HttpTransport httpTransport = GoogleNetHttpTransport.newTrustedTransport();
    Credential credential = authorize(httpTransport);
    Drive driveService = getDriveService(httpTransport, credential);
    File file = driveService.files().get(fileId).execute();
    InputStream inputStream = driveService.files().export(fileId,
"application/vnd.openxmlformats-officedocument.spreadsheetml.sheet")
        .executeMedia()
        .getContent();
    XSSFWorkbook workbook = new XSSFWorkbook(inputStream);
    Sheet sheet = workbook.getSheetAt(0);
    for (Row row : sheet) {
        for (Cell cell : row) {
            System.out.print(cell.getStringCellValue() + "\t");
        }
        System.out.println();
    }
    workbook.close();
}
public static void main(String[] args) throws GeneralSecurityException,

```

```
IOException {  
    String fileId = "658865865";  
    readExcelFile(fileId);  
}
```

## A.2 Swipe.java

```
import android.app.Activity;  
import android.content.Context;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.util.DisplayMetrics;  
import android.view.GestureDetector;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.WindowManager;  
import android.widget.PopupWindow;  
  
public class SwipePopupWindow {  
    private Context context;  
    private PopupWindow popupWindow;  
    private GestureDetector gestureDetector;  
  
    public SwipePopupWindow(Context context) {  
        this.context = context;  
        gestureDetector = new GestureDetector(context, new SwipeGestureListener());  
    }  
    public void showPopupWindow(View anchorView) {  
        if (popupWindow == null) {
```



```

        View                popupView                =                ((Activity)
context).getLayoutInflater().inflate(R.layout.popup_window_layout, null);
        int width = ViewGroup.LayoutParams.WRAP_CONTENT;
        int height = ViewGroup.LayoutParams.WRAP_CONTENT;
        popupWindow = new PopupWindow(popupView, width, height, true);
        popupWindow.setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
        popupWindow.setAnimationStyle(android.R.style.Animation_Dialog);
    }

    popupWindow.showAsDropDown(anchorView);

    WindowManager.LayoutParams    layoutParams    =    ((Activity)
context).getWindow().getAttributes();
    layoutParams.alpha = 0.7f;
    ((Activity) context).getWindow().setAttributes(layoutParams);

    popupWindow.setOnDismissListener(new
PopupWindow.OnDismissListener() {
        @Override
        public void onDismiss() {
            WindowManager.LayoutParams    layoutParams    =    ((Activity)
context).getWindow().getAttributes();
            layoutParams.alpha = 1.0f;
            ((Activity) context).getWindow().setAttributes(layoutParams);
        }
    });
}

public boolean handleTouchEvent(MotionEvent event) {

```

```

        return gestureDetector.onTouchEvent(event);
    }

    private class SwipeGestureListener extends
GestureDetector.SimpleOnGestureListener {
        private static final int SWIPE_THRESHOLD = 100;
        private static final int SWIPE_VELOCITY_THRESHOLD = 100;
        public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX,
float velocityY) {
            float diffX = e2.getX() - e1.getX();
            float diffY = e2.getY() - e1.getY();

            if (Math.abs(diffX) > Math.abs(diffY)
                && Math.abs(diffX) > SWIPE_THRESHOLD
                && Math.abs(velocityX) > SWIPE_VELOCITY_THRESHOLD) {
                if (diffX > 0) {
                    showPopupWindow(((Activity)
context).getWindow().getDecorView().getRootView());
                    return true;
                }
            }

            return false;
        }
    }
}

```

### A.3 DatabaseHelper.java

```

import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private DatabaseHelper databaseHelper;
    private SQLiteDatabase database;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        databaseHelper = new DatabaseHelper(this);
        database = databaseHelper.getWritableDatabase();
        Cursor cursor = getAllBusinessData();
        if (cursor.moveToFirst()) {
            do {
                int id = cursor.getInt(cursor.getColumnIndex("_id"));
                String date = cursor.getString(cursor.getColumnIndex("date"));
                double revenue = cursor.getDouble(cursor.getColumnIndex("revenue"));
                double expenses =
cursor.getDouble(cursor.getColumnIndex("expenses"));
            } while (cursor.moveToNext());
        }
        cursor.close();
        database.close();
    }
}

```

```
}  
private void addBusinessData(String date, double revenue, double expenses) {  
    ContentValues values = new ContentValues();  
    values.put("date", date);  
    values.put("revenue", revenue);  
    values.put("expenses", expenses);  
    database.insert("business_data", null, values);  
}  
private Cursor getAllBusinessData() {  
    return database.query("business_data", null, null, null, null, null, null);  
}  
}
```