

## **Загальний опис**

**Назва:** Expense Tracker App.

**Мета:** Створення додатку для відстеження особистих фінансів з можливістю перегляду та аналізу витрат.

**Цільова аудиторія:** люди, що мають доходи та витрати.

## **Функціональні вимоги**

### **1. Додавання витрат:**

- Користувач може вручну додавати нові витрати, вказуючи суму, категорію, дату та короткий опис операції.

### **2. Вибір категорії витрат:**

- Користувач може вибирати категорію витрат з представлених варіантів або додати нову категорію.

### **3. Відображення списку витрат:**

- Користувач може переглядати введені витрати у формі списку з можливістю сортування та фільтрації за датою або категорією.

### **4. Статистика витрат:**

- Користувач може переглядати статистику своїх витрат за певний період часу у вигляді графіків та звітів.

### **5. Фільтрація та сортування даних:**

- Користувач має можливість фільтрувати та сортувати дані про витрати за різними параметрами (категоріями, періодом).

### **6. Валютний калькулятор**

- Для конвертації валют на основі зовнішнього API.

## **Нефункціональні вимоги**

### **1. Продуктивність:**

- Додаток повинен бути ефективним та швидким.

## **2. Кросплатформенність:**

- Додаток повинен бути сумісним із різними версіями операційних систем Android та iOS.

## **3. Адаптивність:**

- Інтерфейс додатка повинен бути адаптивним до різних розмірів екранів і роздільної здатності мобільних пристроїв.

## **4. Тестова покриття:**

- Наявність комплексних тестів для перевірки функціональності, продуктивності та безпеки додатка.

## **5. Інтеграція з зовнішніми сервісами:**

- Можливість взаємодії з зовнішніми API для отримання курсів валют та інших необхідних даних.

## **6. Інтерфейс користувача:**

- Інтерфейс повинен бути інтуїтивно зрозумілим та привабливим для користувачів.

# **Технічні вимоги**

## **1. Flutter Framework:**

- Використання Flutter для кросплатформенної розробки додатку.

## **2. Доступ до API для курсів валют:**

- Інтеграція з зовнішнім API для отримання актуальних курсів валют.

## **3. База даних:**

- Використання локальної бази даних для зберігання інформації про витрати користувачів та іншої необхідної інформації.

## **4. Архітектура програмного забезпечення:**

- Використання чіткої та ефективної архітектури для полегшення розширення та обслуговування коду.

## 5. Тестування:

- Розробка та виконання автоматизованих тестів для перевірки функціональності та інтеграції.

## Моделювання



*Діаграма способу використання*

### 1. Користувач:

Головний користувач, що має доступ до усіх функцій додатку.

### 2. Додавання витрат:

Користувач може вручну додавати нові витрати, вказуючи необхідну інформацію:

- Вибір категорії витрат
- Вибір суми витрат
- Вибір валюти витрат

### 3. Відображення списку витрат:

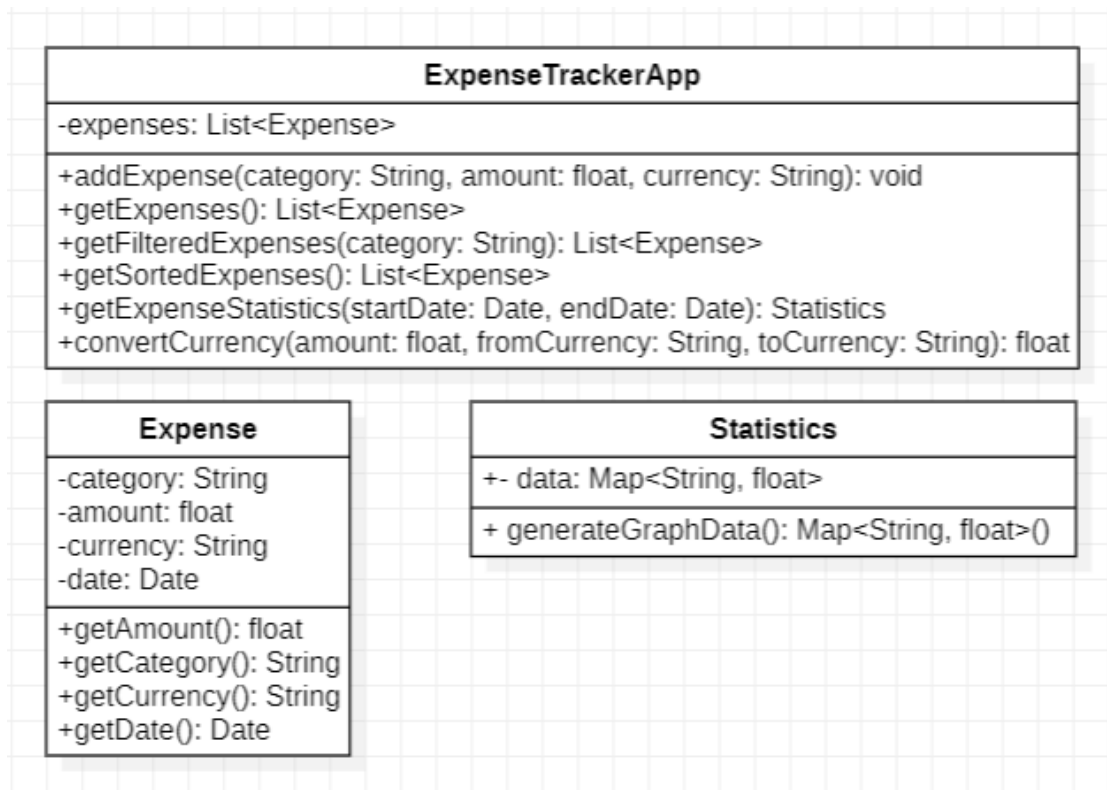
Користувач може переглядати список введених витрат з можливістю сортування та фільтрації. Користувач може переглядати статистику витрат за певний період часу у вигляді графіків.

#### 4. Фільтрація та сортування даних:

Користувач може фільтрувати та сортувати дані про витрати за різними параметрами.

#### 5. Валютний калькулятор:

Користувач може конвертувати валюту на основі зовнішнього API.



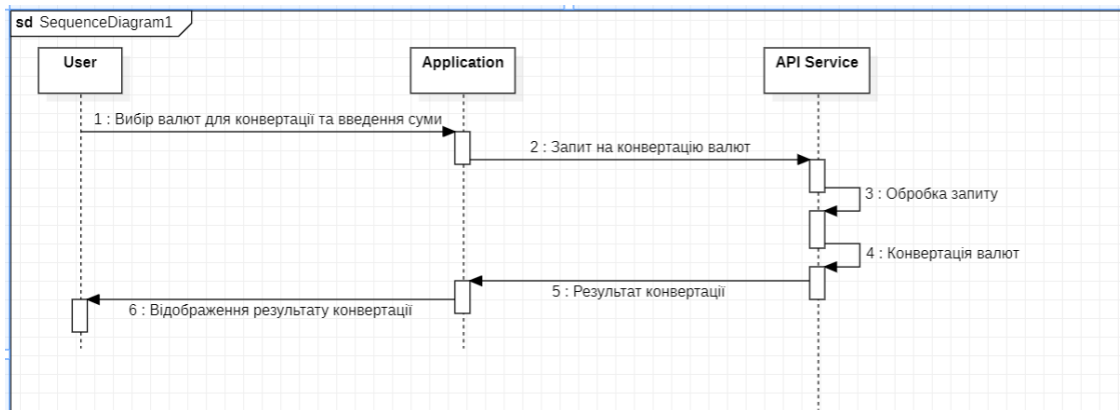
*Діаграма класів*

Класи: ExpenseTrackerApp, Expense та Statistics.

Клас ExpenseTrackerApp містить методи для додавання, отримання, фільтрації, сортування витрат, отримання статистики витрат та конвертації валюти.

Клас `Expense` представляє окрему витрату і має методи для отримання інформації про витрату.

Клас `Statistics` використовується для генерації даних для графіків статистики витрат.



*Діаграма послідовностей*

Діаграма послідовностей для використання API конвертації валют у додатку `Expense Tracker` виглядає наступним чином:

1. Користувач взаємодіє з додатком, вибираючи валюту для конвертації та вводячи суму.
2. Додаток формує запит до API конвертації валют і передає необхідні дані.
3. Сервер API отримує запит на конвертацію валют.
4. Сервер API обробляє запит, виконує конвертацію валют та повертає результат.
5. Додаток отримує результат конвертації від сервера API та відображає його користувачеві.

## Дизайн і інтерфейс

