ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки

Кафедра ЕОМ

Модель мобільного додатку з застосуванням фреймворку Flutter

Автор:
студент групи СПм-22-4
Бешта В.С.

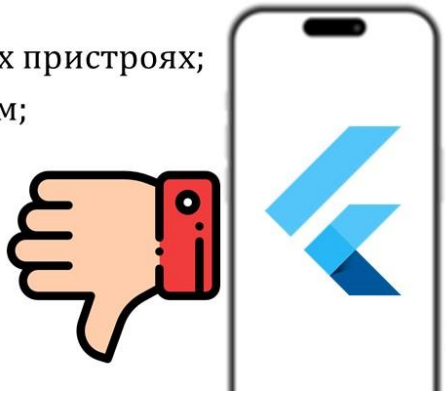
Керівник:
доцент каф. ЕОМ, к.т.н.
Філімончук Т.В.

Мета:

- провести порівняльний аналіз інструментарію для розробки крос платформного мобільного застосунку;
- розробити модель мобільного застосунку з використанням існуючого фреймворку, яка відповідає вимогам:
  - забезпечення безпеки застосунку;
  - реалізації механізмів підключення бази даних;
  - масштабованість;
  - легкість підтримки кодової бази;
- створити та протестувати основні модулі мобільного застосунку, які будуть демонструвати відповідність вимогам розробленої моделі.

Недоліки Flutter

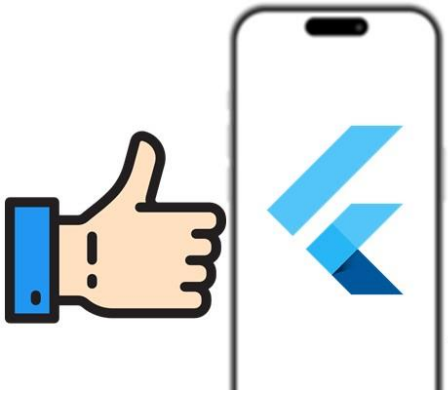
- великий розмір програм;
- необхідність використання унікальних скриптів;
- обмеження доступу до нативних API;
- проблеми з продуктивністю на деяких пристроях;
- обмежена підтримка деяких платформ;
- часті оновлення.



3

Переваги Flutter

- можливість "гарячого" перезавантаження;
- єдина кодова база;
- висока продуктивність;
- багатий набір віджетів;
- підтримка старіших пристроїв;
- організована документація;
- вбудовані компоненти.



4

Модель фреймворку, що існує на даний час

M = {UI, BL, SEC, MT},

- UI (User Interface): інтерфейс взаємодії користувача;
- BL (Business Logic): виконання основних функцій програми;
- SEC (Security): механізм безпеки, який включає аутентифікацію та авторизацію;
- MT (Mobile Testing): модуль для створення та проведення тестів.

5

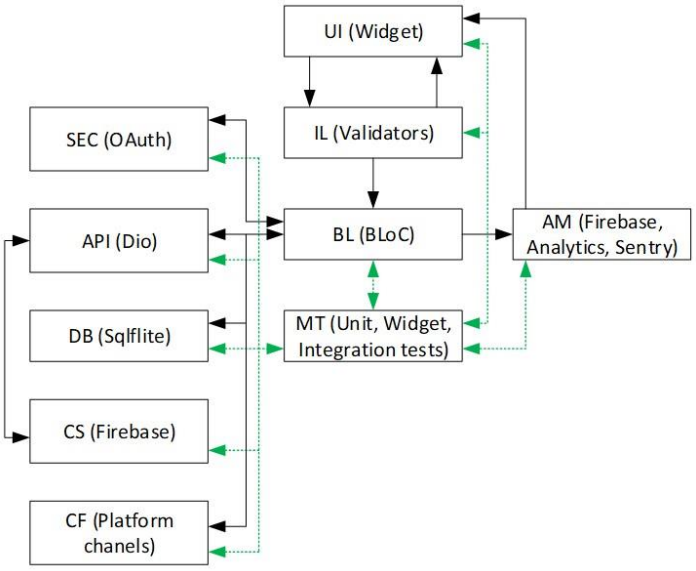
Запропонована модель фреймворку

M = {UI, CF, IL, BL, DB, CS, API, SEC, AM, MT},

- UI (User Interface): інтерфейс взаємодії користувача;
- CF (Client Frontend): клієнтська частина для iOS та Android;
- IL (Interaction Logic): обробка даних та комунікація з функціями програми;
- BL (Business Logic): виконання основних функцій програми;
- DB (Database): зберігання та організація інформації;
- CS (Cloud Services): хмарні сервіси для зберігання файлів та управління ресурсами;
- API (Application Programming Interface): комунікація між клієнтом та сервером;
- SEC (Security): механізм безпеки, який включає аутентифікацію та авторизацію;
- AM (Analytics and Monitoring): аналіз та моніторинг використання додатку;
- MT (Mobile Testing): модуль для створення та проведення тестів.

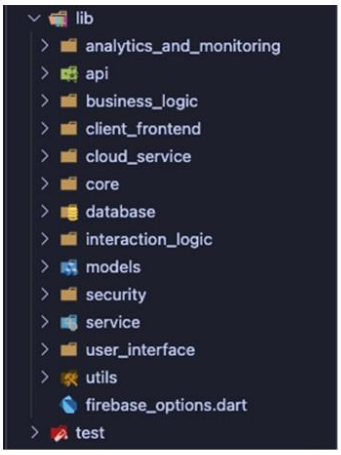
6

Взаємодія модулів



7

Структура тестового додатку



- **user_interface** – директорія для User Interface компонентів;
- **client_frontend** – директорія для CF логіки, реалізація специфічних для платформи (iOS, Android) інтерфейсів та логіки;
- **interaction_logic** – директорія для Interaction Logic;
- **business_logic** – директорія для Business Logic;
- **database** – директорія для Database взаємодії;
- **cloud_services** – директорія для Cloud Services;
- **api** – директорія відповідає за логіку API запитів;
- **analytics_and_monitoring** – директорія для Analytics and Monitoring;
- **core** – містить налаштування навігації та налаштування залежностей.

8

Масштабованість та легкість підтримки кодової бази

```dart
abstract class DatabaseService {
  Future<void> openDatabase(String dbName);
  Future<void> closeDatabase();

  Future<void> insert(String table, Map<String, dynamic> data);
  Future<Map<String, dynamic>> get(String table, String id);
  Future<List<Map<String, dynamic>>> getAll(String table);
  Future<void> update(String table, String id, Map<String,
dynamic> data);
  Future<void> delete(String table, String id);
}
```

Майже кожен модуль має свій інтерфейс

```dart
class SQLiteDatabaseServiceImpl implements DatabaseService {
  SQLiteDatabaseServiceImpl(this._database);
  final Database _database;
  @override
  Future<void> insert(String table, JsonMap data) async {
    await _database.insert(table, data);
  }

  @override
  Future<JsonMap?> query(String table, String id) async {
    final results =
        await _database.query(table, where: 'id = ?', whereArgs: [id]);
    return results.isNotEmpty ? results.first : null;
  }
    /// others methods implementation
}
```

9

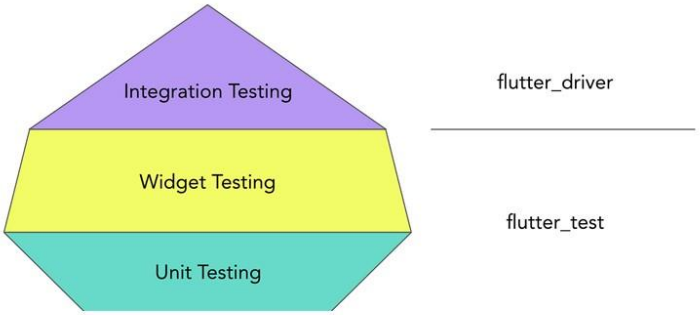
Забезпечення безпеки застосунку

```dart
abstract class SecurityModule {
  Future<bool> twoFactorAuthentication(String username, String
password, String biometricIdentifier);

  Future<String> generateAccessToken();

  Uint8List encryptData(Uint8List data);

  Uint8List decryptData(Uint8List encryptedData);

  Future<void> storeDataSecurely(String key, String value);

  Future<String?> retrieveDataSecurely(String key);
}
```

10

Тестування мобільного додатку

Доступні три типи процесів тестування для повного тестування програми:



11

Висновки

- проведено порівняльний аналіз інструментарію для розробки кросплатформного мобільного застосунку;
- розроблено модель мобільного застосунку з використанням існуючого фреймворку, яка буде відповідає вимогам:
  - забезпечення безпеки застосунку;
  - реалізації механізмів підключення бази даних;
  - масштабованість;
  - легкість підтримки кодової бази;
- створено та протестовано основні модулі мобільного застосунку, які демонструють відповідність вимогам розробленої моделі.

12