

КРОСПЛАТФОРМЕННИЙ ЗАСТОСУНОК «SUPER TRANSLATOR»

Створив студент групи ПП-31
Щербина Андрій Віталійович

Огляд додатку

Розглянутий застосунок був написаний на мові програмування Dart, на основі фреймворку Flutter, це першорядний інструмент для перекладу тексту, який дозволяє користувачам швидко та ефективно переводити тексти між різними мовами. Цей додаток відповідає на реальні потреби сучасного світу, де глобальна комунікація є ключовим фактором успіху.

Мова програмування Dart - це сучасна, об'єктно-орієнтована мова програмування, яка була розроблена компанією Google і випущена у 2011 році. Dart призначена для розробки веб-додатків, мобільних застосунків та серверних систем. Основною метою розробки мови Dart було створення ефективного і продуктивного інструменту для розробки широкого спектру програмного забезпечення.

Фреймворк Flutter - це відкритий фреймворк для розробки крос-платформенних мобільних додатків. Він також був створений компанією Google і випущений у 2018 році. Flutter використовує мову програмування Dart для розробки додатків, і його основною перевагою є можливість швидкого створення гарних та високоефективних інтерфейсів користувача для різних платформ, включаючи iOS, Android, веб та десктоп.

Основні функції застосунку

Переклад тексту в реальному часі

Мій застосунок дозволяє користувачам миттєво перекладати тексти під час введення, забезпечуючи швидкий та точний переклад.

Зберігання улюблених перекладів

За допомогою "Super Translator" можна легко зберігати улюблені переклади та звертатися до них у майбутньому, що забезпечує зручність та ефективність в користуванні.

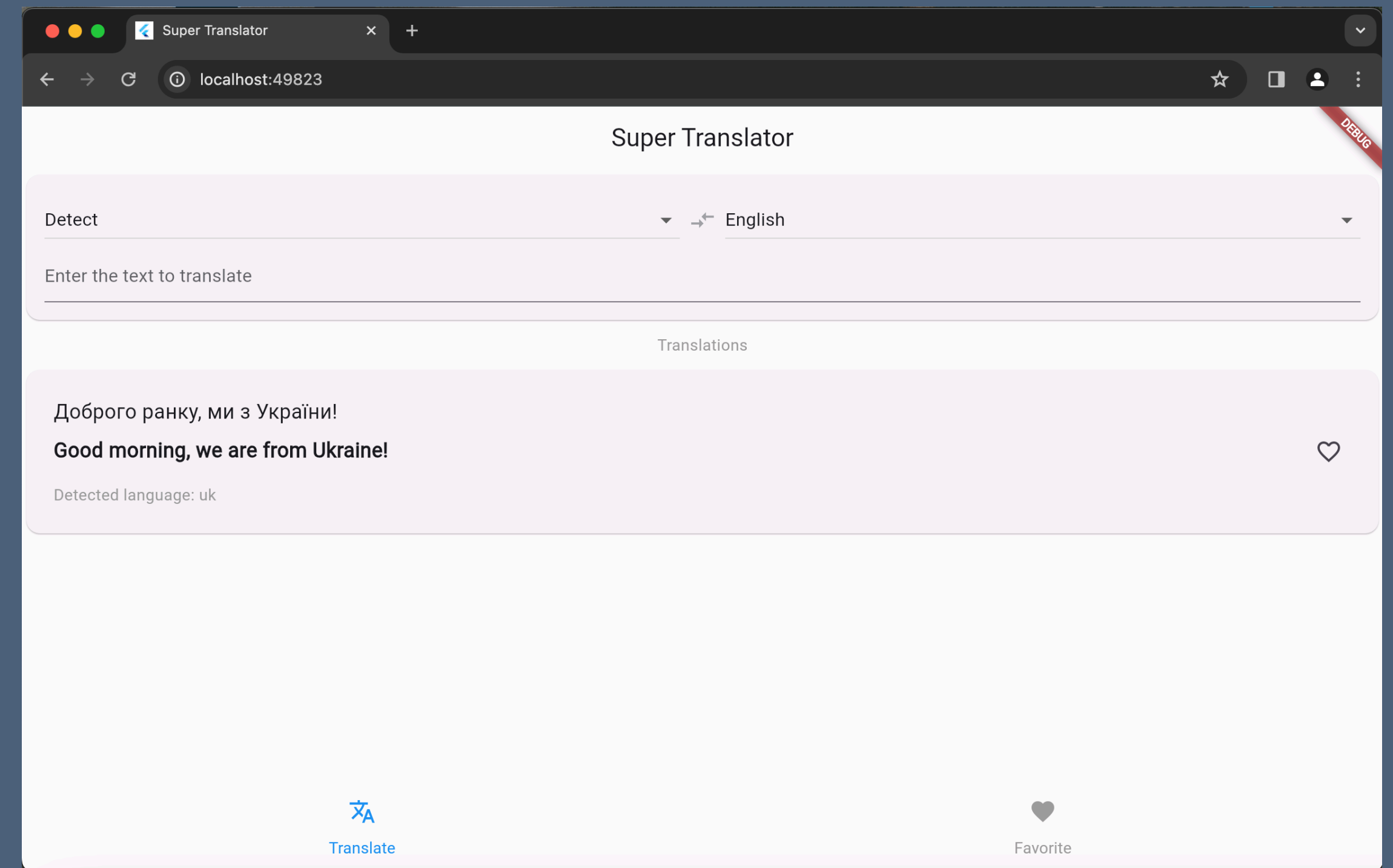
Особисті налаштування та управління мовами

Надає можливість користувачам вибирати мову джерела та мову призначення, щоб забезпечити найкращий переклад залежно від їх потреб та уподобань.

Головний екран

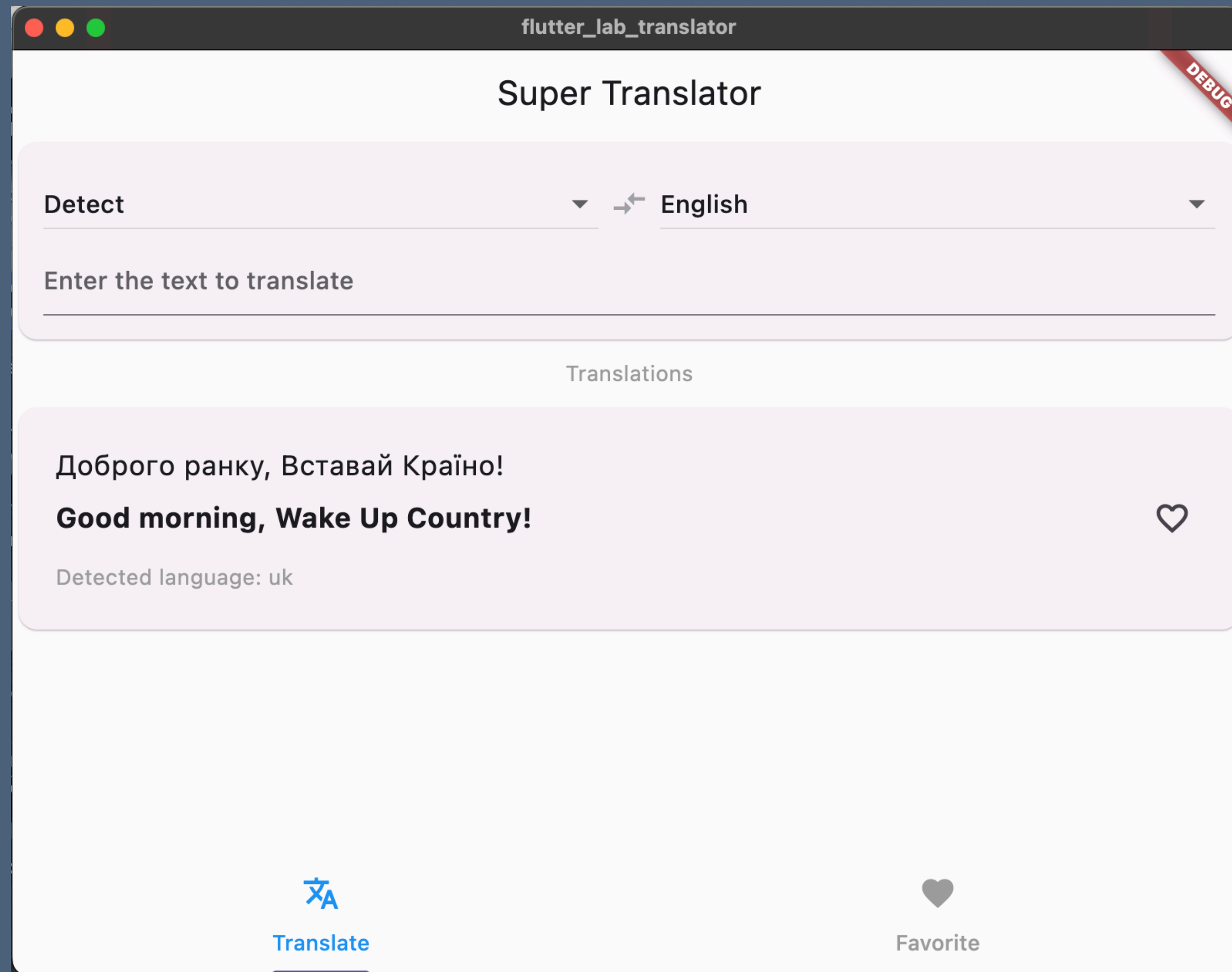


Головний екран Super Translator - це центральна точка доступу до всіх основних функцій перекладача. На цьому екрані користувачі можуть виконувати переклад тексту між різними мовами, а також отримувати доступ до списку підтримуваних мов та їх властивостей.



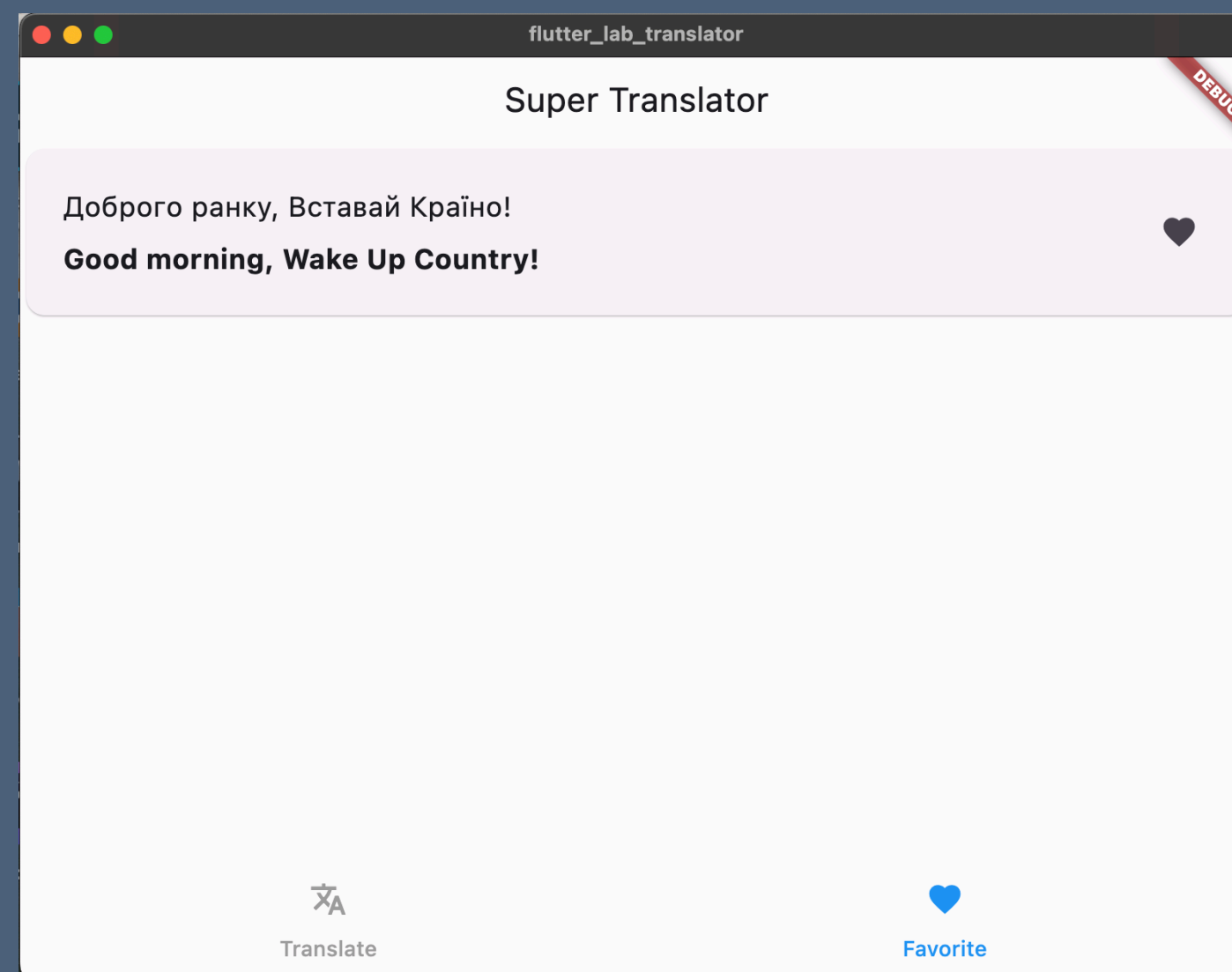
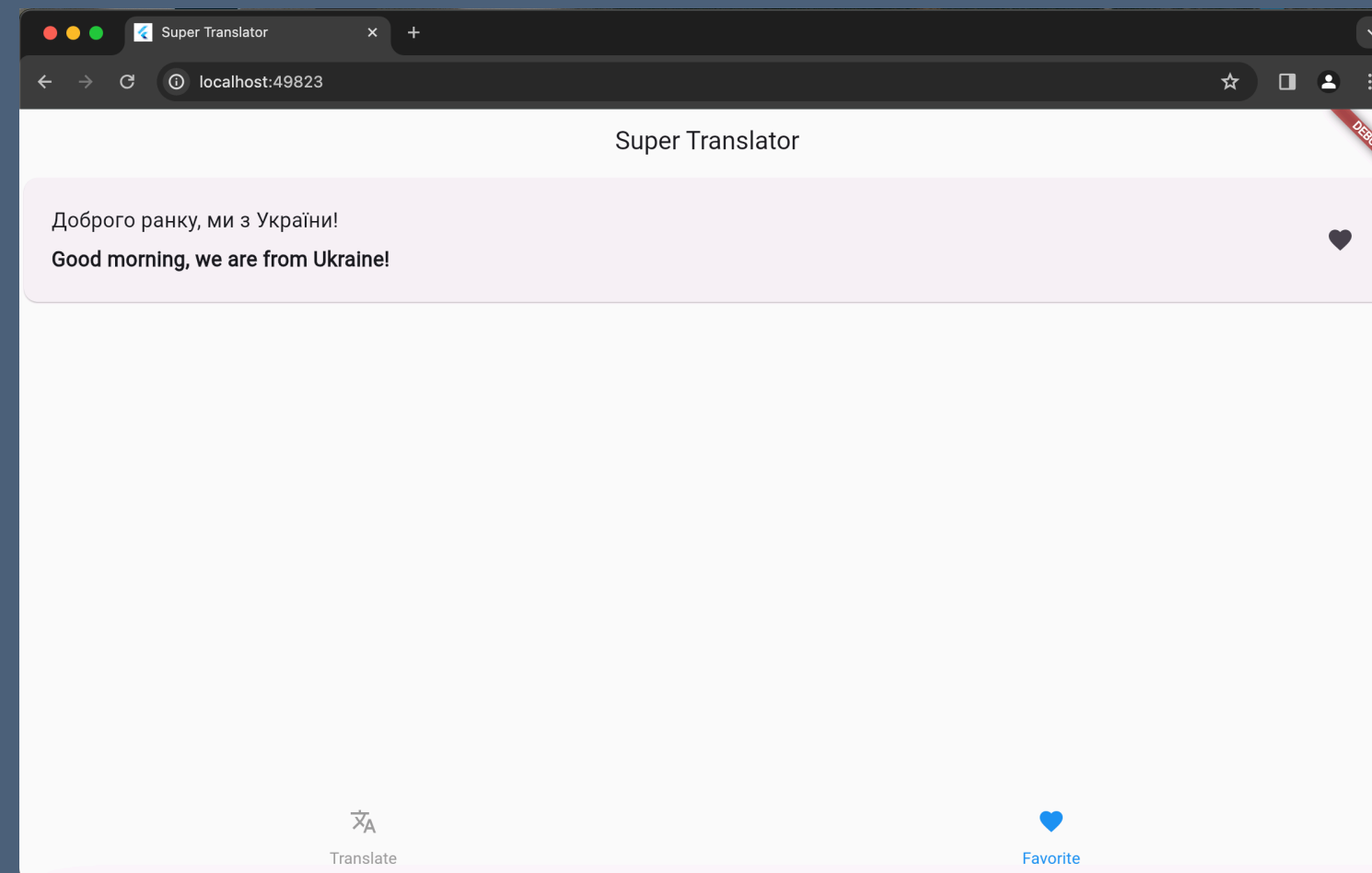
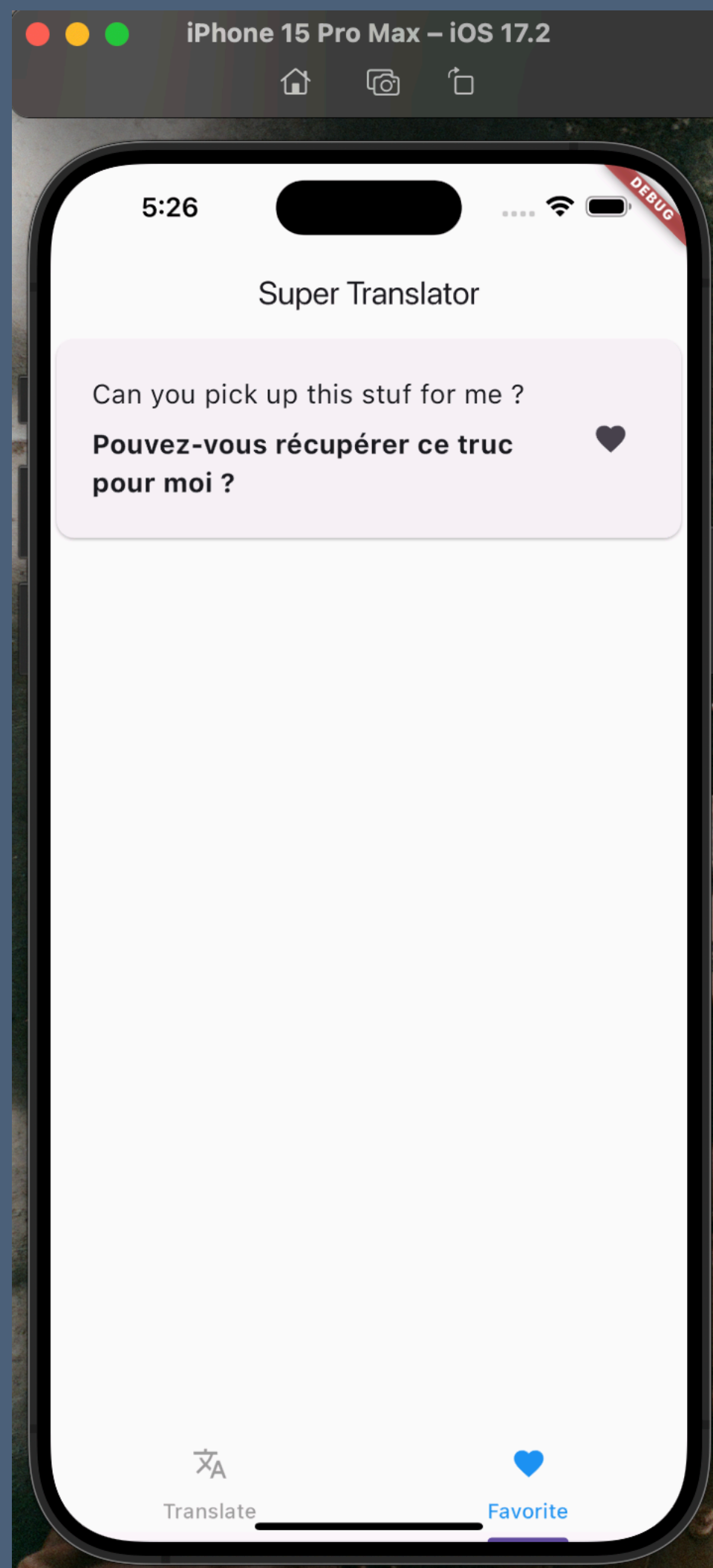
Перш за все, на головному екрані розміщується текстове поле, куди користувачі можуть ввести текст для перекладу. Також тут доступні розкладка клавіатури та інші елементи інтерфейсу для полегшення введення тексту. Під текстовим полем розміщується випадаючий список для вибору мови вихідного тексту та мови, на яку буде здійснено переклад.

Головний екран



У нижній частині головного екрана Super Translator розміщуються дві кнопки для швидкого доступу до інших функціональних складових додатку. Одна з цих кнопок веде на екран «Favorites» або «Улюблені», де користувачі можуть переглядати список своїх улюблених перекладів або збережених фраз.

Екран «Favorites»



Екран «Favorites» у застосунку дозволяє користувачам переглядати та редагувати список їх улюблених перекладів або збережених фраз. Під час розробки цього функціоналу використовувалися такі файли та класи:

- `favorites_screen.dart`: Цей файл містить клас `FavoritesScreen`, який відповідає за відображення екрану зі списком улюблених перекладів.
- `favorites_cubit.dart`: У цьому файлі реалізовано клас `FavoritesCubit`, який відповідає за бізнес-логіку екрану `Favorites`. Він взаємодіє з репозиторієм для зберігання та управління улюбленими перекладами.
- `i_favorite_repository.dart`: Цей файл містить інтерфейс `IFavoriteRepository`, який визначає методи для роботи з улюбленими перекладами, такими як отримання, додавання та видалення.
- `db_favorite_repository.dart`: У цьому файлі знаходиться реалізація класу `DbFavoriteRepository`, який взаємодіє з локальною базою даних для зберігання улюблених перекладів.
- `translation_card.dart`: Використовується для відображення окремого перекладу на екрані `Favorites`.

Екран «Favorites»



На етапі розробки застосунку Super Translator, було використано API Google Translate для здійснення мовних перекладів.

Google Translate API - це інтерфейс програмування застосунків, який надає розробникам доступ до машинного перекладу та інших мовних сервісів Google.

**Generate Google
Translate API Key**

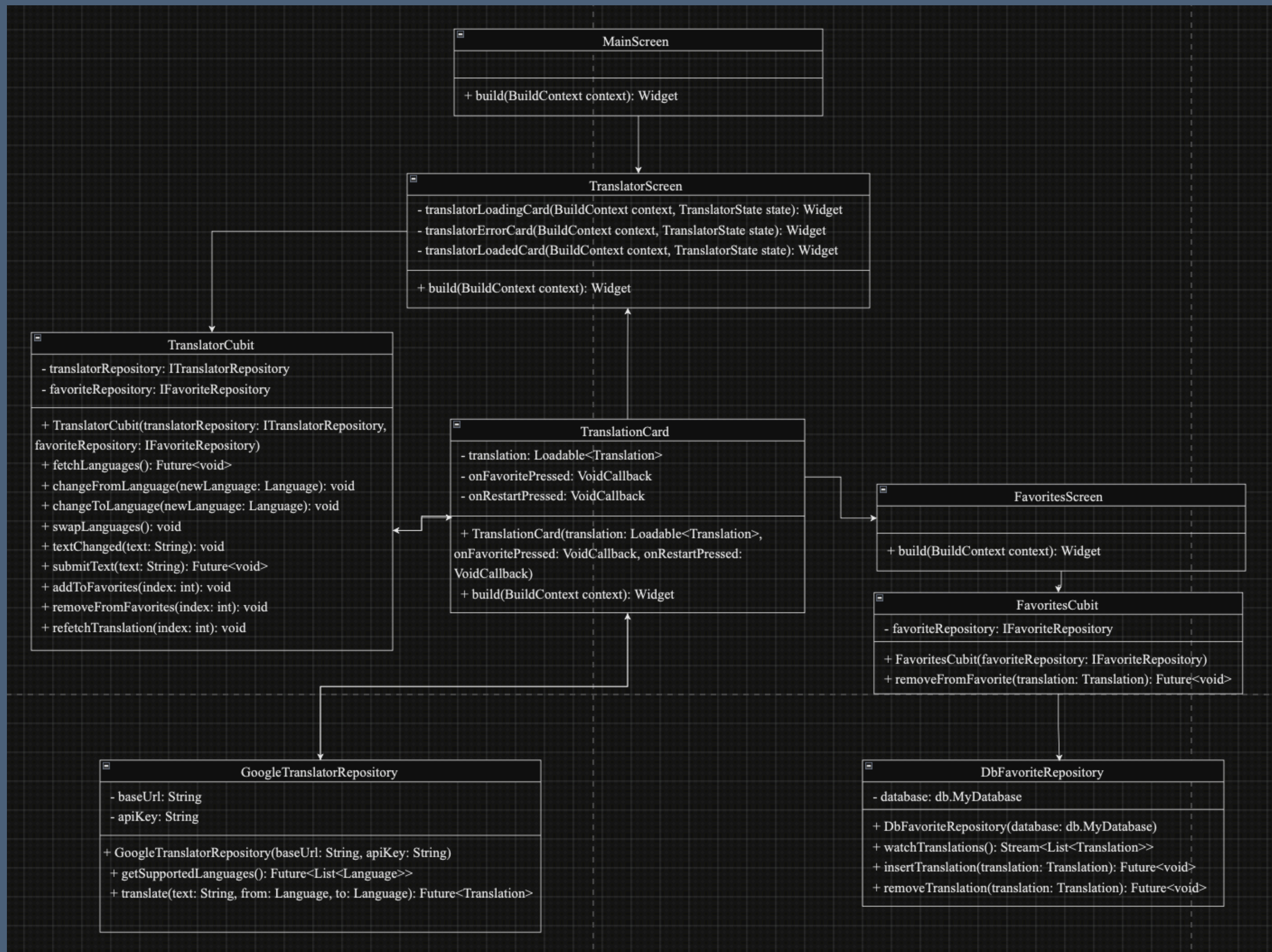


Один з ключових елементів використання Google Translate API - це API Key. API Key - це унікальний ідентифікатор, який використовується для ідентифікації запитів до API та забезпечення безпеки та контролю доступу. Кожен розробник повинен отримати свій власний API Key від Google для користування послугами.

`google_translator_repository.dart`: У цьому файлі реалізовано клас `GoogleTranslatorRepository`, який використовує API Key для виконання запитів до Google Translate API та отримання перекладів.

`main_screen.dart`: Клас `MainScreen`, який представляє головний екран застосунку, може використовувати Google Translate API для перекладу тексту.

Архитектура



Використані технології

1. Використання шаблону Repository:

Для ефективного відокремлення джерела даних від бізнес-логіки додатку ми використовуємо патерн Repository. Це спрощує тестування та розширення функціональності.

2. Використання Dependency Injection:

Для зручності управління залежностями та забезпечення їх доступності в класах Cubit, ми використовуємо Dependency Injection. Цей підхід робить код більш масштабованим та легким для тестування.

3. Використання BLoC та Cubit:

Управління станом додатку здійснюється за допомогою BLoC та Cubit. Це дозволяє ефективно реалізувати реактивну логіку та зручно керувати станом додатку.

4. Модульність та розділення відповідальностей:

Код розділений на окремі модулі, що дозволяє зберігати логічно пов'язаний код окремо. Наприклад, логіка перекладача розділена від презентаційного шару.

5. Використання MultiProvider та MultiBlocProvider:

Для забезпечення залежностей різних частин додатку використовуються MultiProvider та MultiBlocProvider. Це дозволяє ефективно керувати залежностями та станом додатку.

6. Використання API Google Translate:

Для забезпечення безпеки та зручності управління конфіденційними даними використовується константа для зберігання ключа доступу до API Google Translate.

7. Суворі типізованість:

Використання суворо типізованих залежностей та станів допомагає уникнути помилок та забезпечує більшу стабільність додатку.

Висновки

Розроблений додаток для перекладу представляє собою високоефективний та зручний інструмент для користувачів, що потребують миттєвого доступу до перекладу текстів між різними мовами. Вона поєднує в собі передові технології та сучасний дизайн, що дозволяє забезпечити зручність використання та швидкість перекладу.

Розробка базується на використанні та інтеграції кращих практик у сфері програмного забезпечення, таких як кросплатформена основа, мова Dart та фреймворк Flutter дозволяють без проблем поширювати застосунок на будь-якій доступній платформі, патерни проектування, шаблони розробки, та ефективно управління залежностями. Використання суворо типізованих залежностей допомагає уникнути помилок та забезпечує стабільність та надійність додатку.

Крім того, з використанням API Google Translate та інших сучасних технологій, розробка може легко і швидко інтегруватися з іншими додатками та платформами. Такий підхід робить наш продукт більш гнучким та універсальним у використанні, підвищує привабливість для широкого кола користувачів.

Висновки

Виконана робота над проектом надала мені цінний досвід у ряді ключових аспектів програмування та розробки кросплатформеного програмного забезпечення. По-перше, я отримав поглиблені знання та навички у використанні мови програмування Dart та фреймворка Flutter. Ця робота дозволила мені поглибити розуміння архітектури Flutter, розробляти високоякісні інтерфейси користувача та ефективно управляти станом додатку.

По-друге, робота над проектом розширила мої знання про шаблони проектування та практики розробки програмного забезпечення. Використання шаблону Repository, Dependency Injection, та BLoC/Cubit дозволило мені зрозуміти, як ефективно відокремлювати різні частини додатку та забезпечувати його масштабованість та стабільність.

Крім того, цей проект також дав мені можливість практикувати співпрацю з API Google Translate та іншими зовнішніми сервісами, а також використовувати принципи роботи з локальними базами даних та зберіганням даних на пристроях користувачів. Це розширило мої знання та навички у розробці зручних та функціональних додатків для мобільних пристроїв.

Дякую за увагу !