

Faculté des technologies de l'information et de la communication

Département informatique appliquée



RAPPORT PROJET FLUTTER

Sentimental Analysis 🤖

Réalisés par : - Nathan Carlinot RANDRIAMIHAJA
- PEMBA Lahizarre

Sous la direction du superviseur : Shiam Beehary

Année scolaire : 2023 – 2024

A- Configuration du projet Flutter : pubspec.yaml

1. Nom de l'application

- Le champ "name" définit le nom de l'application Flutter. C'est un identifiant unique qui est utilisé pour identifier l'application dans l'écosystème Flutter.

2. Description de l'application

- Le champ "description" permet de fournir une brève description de l'application. Cela peut inclure des détails sur ce que fait l'application, ses fonctionnalités principales, son objectif, etc.

3. Publication

- La ligne "publish_to" contrôle le comportement de publication du package sur pub.dev, le référentiel de packages Flutter. Dans ce cas, la valeur "none" est utilisée pour empêcher la publication accidentelle du package.

4. Version et numéro de build

- Le champ "version" définit la version actuelle de votre application. Il suit généralement la convention de versionnement "major.minor.patch". Le numéro de build peut être ajouté à la fin de la version principale pour suivre les itérations de développement.

5. Environnement

- La section "environment" spécifie les contraintes de compatibilité de l'environnement pour exécuter l'application Flutter. Elle peut définir la version minimale requise du SDK Flutter.

6. Dépendances

- La section "dependencies" répertorie les packages externes nécessaires pour que l'application fonctionne correctement. Ces packages sont téléchargés et intégrés à votre application lors de sa construction.

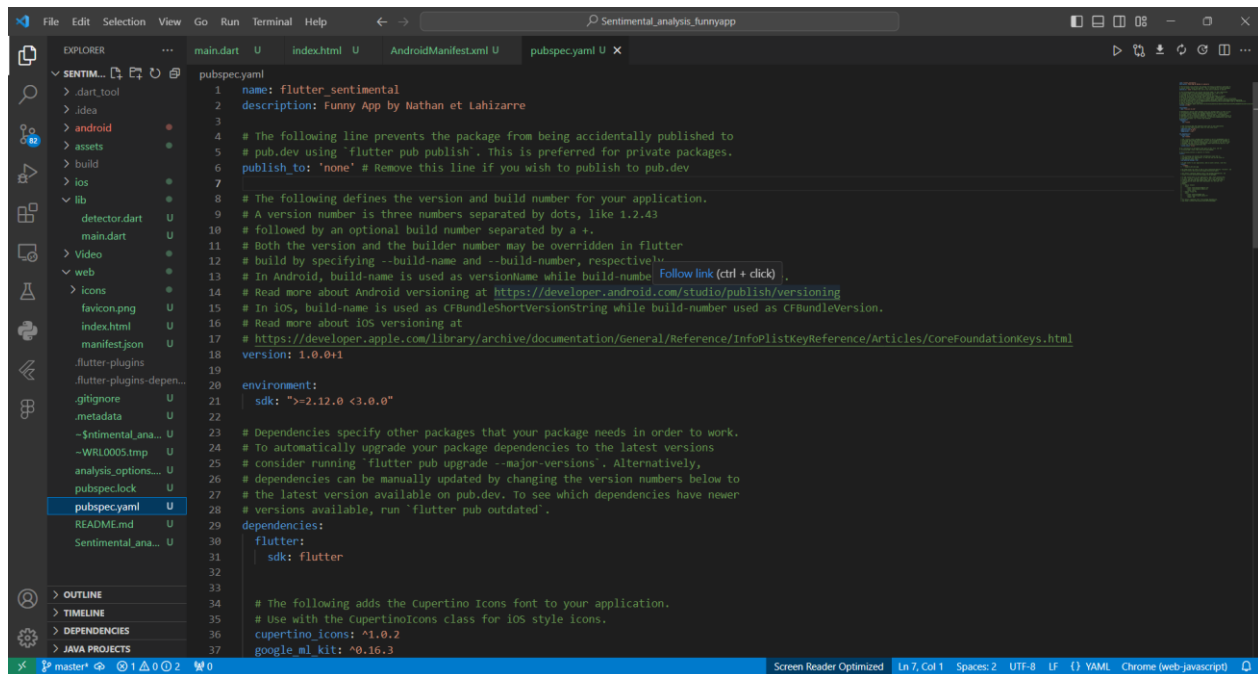
7. Dépendances de développement

- La section "dev_dependencies" contient les packages nécessaires uniquement pour le développement de l'application, tels que les tests unitaires ou les linters.

8. Configuration spécifique à Flutter

- La section "flutter" contient des configurations spécifiques à Flutter telles que l'utilisation de l'icône des Material Icons, la définition des assets utilisés dans l'application, et la configuration des polices personnalisées.

Cette partie du fichier pubspec.yaml est cruciale car elle définit les paramètres de base de l'Application Flutter, ses dépendances externes, et ses configurations spécifiques à Flutter.



B- Point d'entrée de l'application Flutter : main.dart

1- Fonction `main()`

- La fonction principale `main()` est le point d'entrée de l'application Flutter.
- Cette fonction est appelée lorsque l'application est lancée et elle est responsable de créer l'interface utilisateur principale de l'application.

2- Utilisation de `MaterialApp`

- Dans cette application, `MaterialApp` est utilisée comme widget racine.
- `MaterialApp` est un widget fourni par Flutter qui fournit une structure de base pour les applications basées sur le design Material Design de Google.

3- Propriété `home`

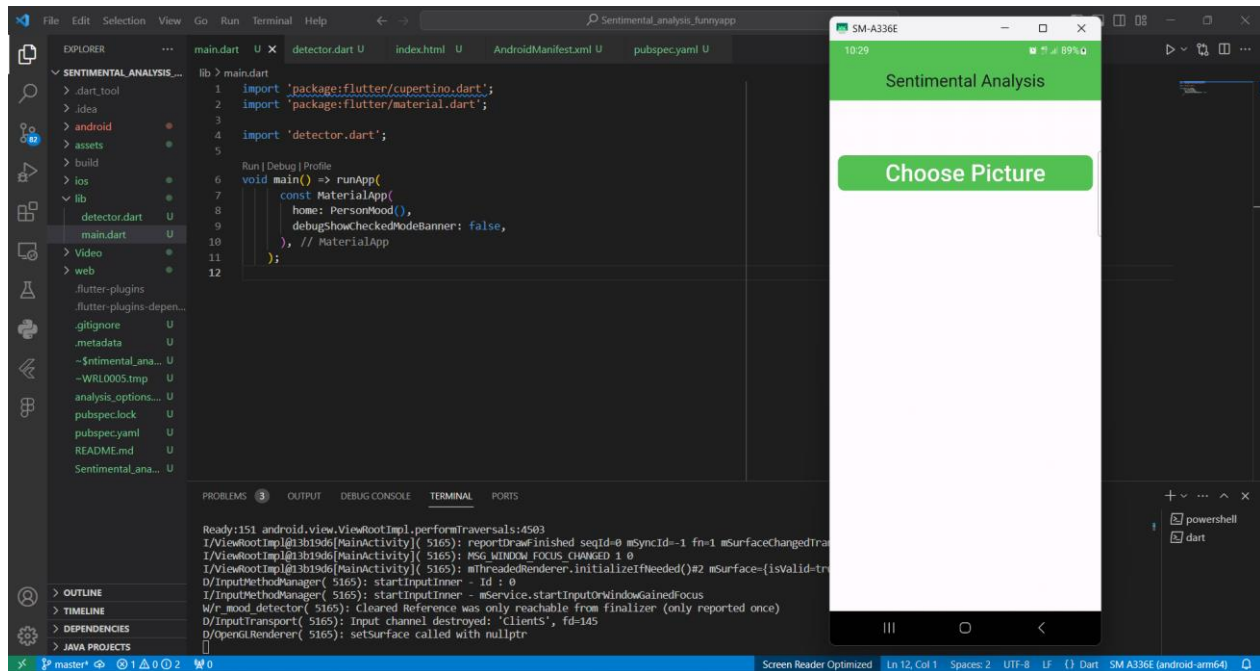
- La propriété `home` de `MaterialApp` est définie sur une instance de la classe `PersonMood`.
- Cela signifie que la page représentée par la classe `PersonMood` est la page d'accueil de l'application, qui sera affichée en premier lorsque l'application sera lancée.

4- Propriété `debugShowCheckedModeBanner`

- La propriété `debugShowCheckedModeBanner` est utilisée pour contrôler l'affichage de la bannière de débogage "Debug" dans le coin supérieur droit de l'interface utilisateur de l'application.

- Dans ce cas, elle est définie sur `false`, ce qui signifie que la bannière de débogage ne sera pas affichée dans l'application en mode de développement.

Le fichier main.dart définit la fonction principale `main()` qui lance l'application Flutter en utilisant `MaterialApp` comme widget racine. Il configure également la page d'accueil de l'application en utilisant la classe `PersonMood` et contrôle l'affichage de la bannière de débogage.



C- Détection de l'humeur à partir d'images : detector.dart

1- Classe `PersonMood`

- La classe `PersonMood` est un StatefulWidget qui représente la page de détection de l'humeur dans l'application.
- Cette classe est responsable de l'interface utilisateur de la page et de la gestion de l'état pour afficher les résultats de la détection d'humeur.

2- Etat de la classe `PersonMood`

- `pathOfImage` : Une variable qui stocke le chemin de l'image sélectionnée par l'utilisateur.
- `moodImagePath` : Une variable qui stocke le chemin de l'image représentant l'humeur détectée.
- `moodDetail` : Une variable qui stocke les détails de l'humeur détectée.
- `isVisible` : Un booléen pour indiquer si l'image et les détails de l'humeur doivent être affichés.

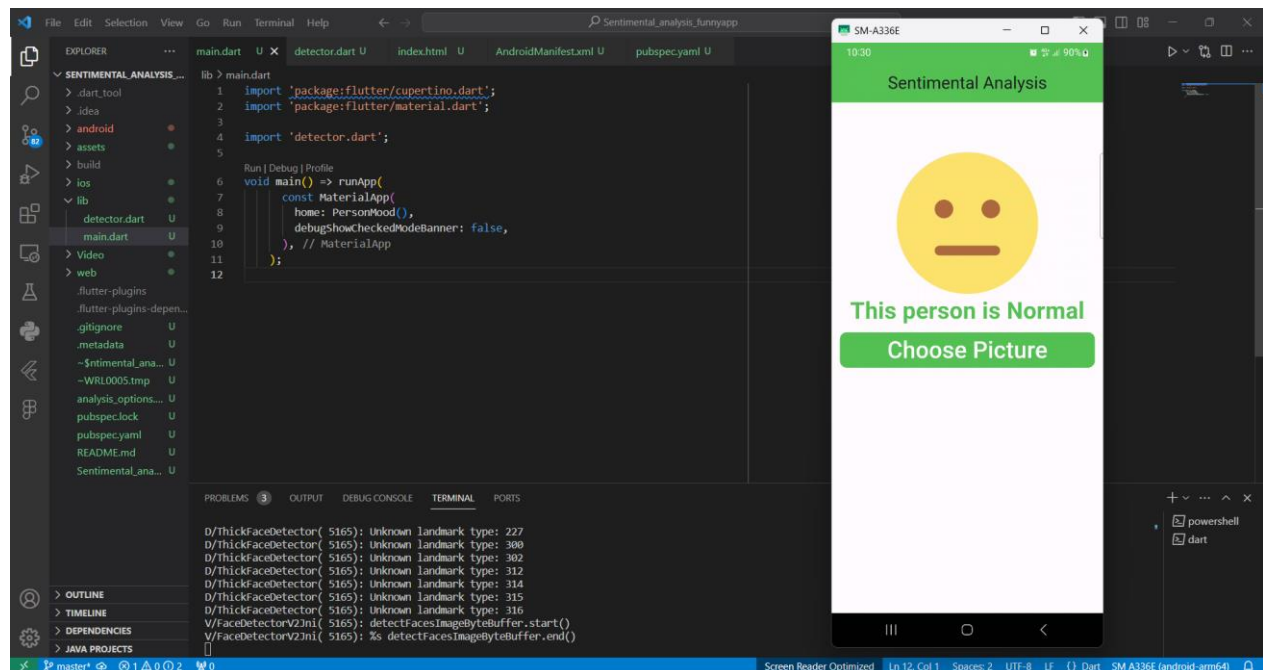
3- Méthode `pickImage()`

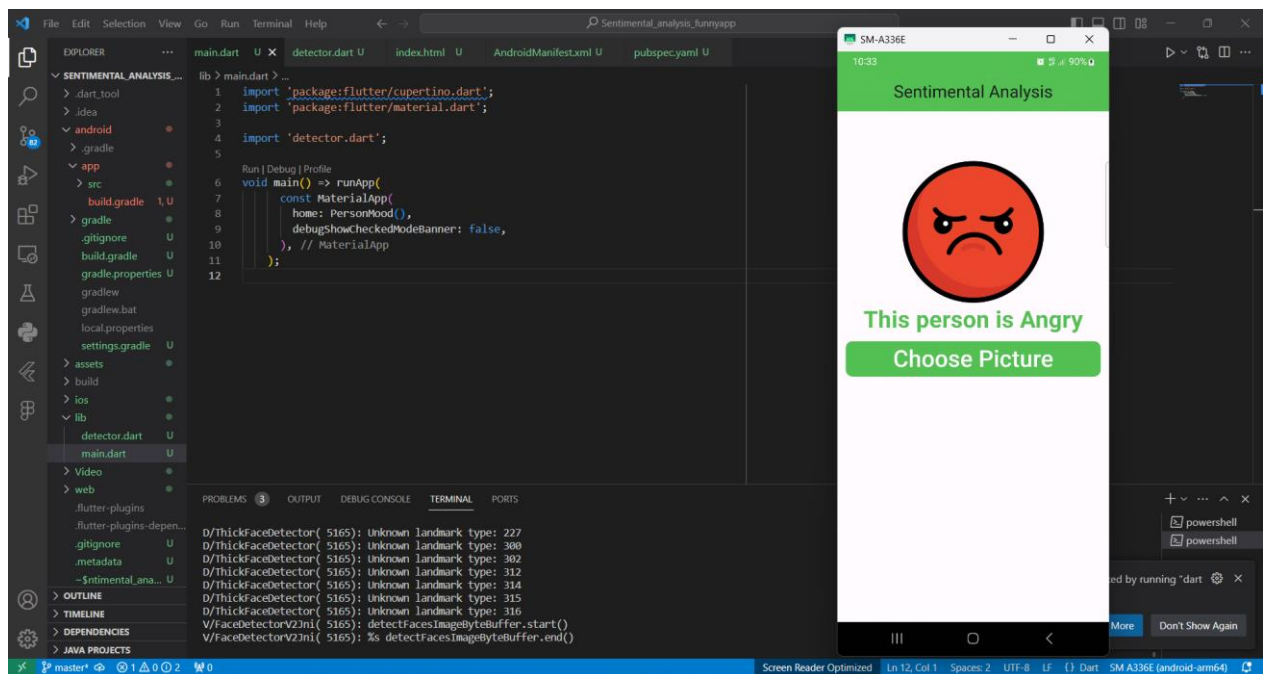
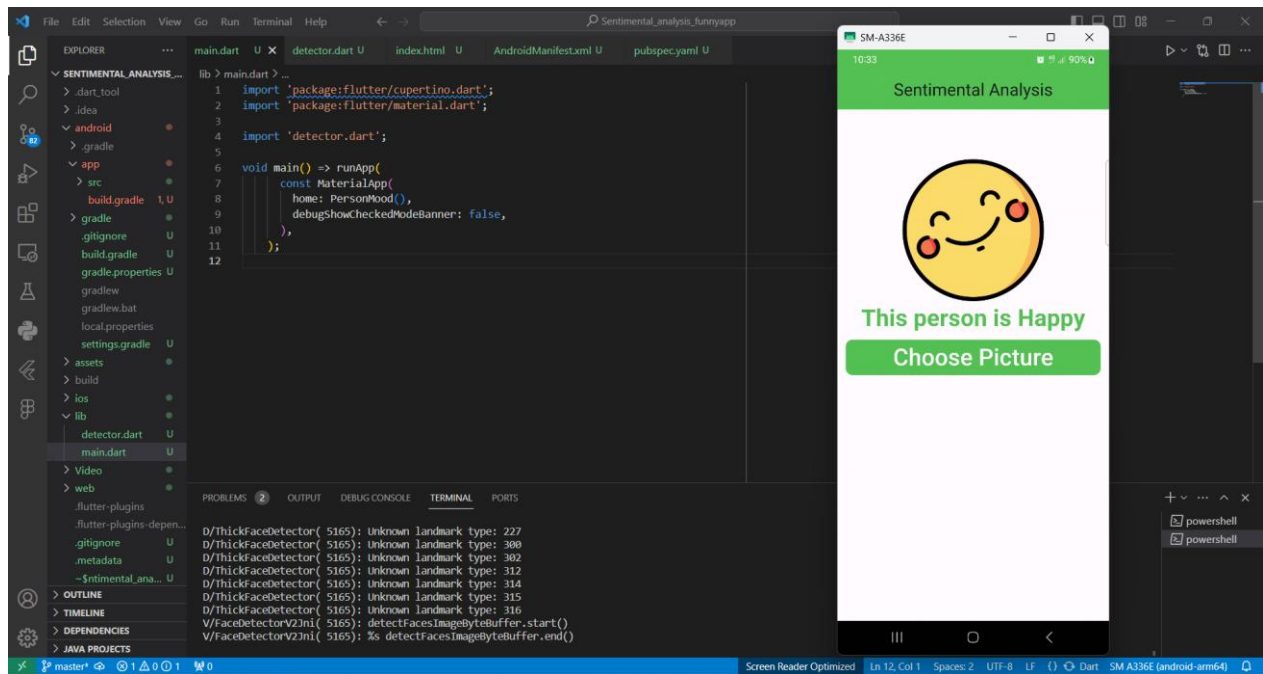
- Cette méthode est appelée lorsque l'utilisateur appuie sur le bouton pour choisir une image dans la galerie.
- Elle utilise le package `image_picker` pour permettre à l'utilisateur de sélectionner une image à partir de la galerie de son appareil.
- Une fois l'image sélectionnée, le chemin de l'image est stocké dans la variable ``pathOfImage``.

Méthode ``extractData(String imagePath)`` :

- Cette méthode est utilisée pour extraire les données de l'image sélectionnée, détecter l'humeur de la personne dans l'image et mettre à jour l'état en conséquence.
- Elle utilise le package Google ML Kit pour détecter les visages dans l'image et analyser leur humeur en fonction de la probabilité de sourire.
- Selon la probabilité de sourire détectée, elle met à jour les variables ``moodDetail`` et ``moodImagePath`` pour représenter l'humeur détectée, puis met à jour l'état pour afficher l'image et les détails de l'humeur.

Le fichier `detector.dart` contient une classe dédiée à la détection de l'humeur à partir d'images. Il utilise des packages externes tels que Google ML Kit et `image_picker` pour la détection faciale et la sélection d'images. La classe ``PersonMood`` gère l'interface utilisateur et l'état de la page de détection d'humeur, tandis que les méthodes ``pickImage()`` et ``extractData(String imagePath)`` permettent respectivement de sélectionner une image et d'extraire les données d'humeur à partir de celle-ci.





D- Configuration du projet Android dans le module Flutter : lib/app/build.gradle

1. Lecture des propriétés locales

- Les lignes initiales permettent de charger les propriétés locales à partir du fichier local.properties du projet, telles que le chemin du SDK Flutter, le code de version et le nom de version de Flutter.

2. Configuration du projet Android :

- Le plugin 'com.android.application' est appliqué pour indiquer qu'il s'agit d'une application Android.
- Le plugin 'kotlin-android' est appliqué pour activer la prise en charge de Kotlin dans le projet Android.
- Le plugin 'flutter.gradle' est appliqué pour configurer le projet Android pour fonctionner avec Flutter.

3. Configuration de la version Android :

- La section 'android' définit les paramètres de compilation et de compatibilité pour le projet Android.
 - 'compileSdkVersion' spécifie la version de l'API Android avec laquelle l'application est compilée.
 - 'defaultConfig' définit les configurations par défaut de l'application, telles que l'ID de l'application, la version minimale et cible de l'API Android, le code de version et le nom de version de l'application.
 - 'buildTypes' définit les types de build, tels que 'release' pour les versions publiées.

4. Configuration de Flutter :

- La section 'flutter' spécifie la source de Flutter pour le module Android.

5. Dépendances :

- La section 'dependencies' spécifie les dépendances du projet Android, telles que les bibliothèques Kotlin standard.

Le fichier build.gradle dans le répertoire lib/app contient la configuration du projet Android pour le module Flutter. Il définit les paramètres de compilation, les configurations de version, les dépendances et les plugins nécessaires pour construire et exécuter l'application Android.

Table de matière

A- Configuration du projet Flutter : pubspec.yaml.....	1
1. Nom de l'application	1
2. Description de l'application.....	1
3. Publication.....	1
4. Version et numéro de build.....	1
5. Environnement.....	1
6. Dépendances.....	1
7. Dépendances de développement	1
8. Configuration spécifique à Flutter.....	1
B- Point d'entrée de l'application Flutter : main.dart.....	2
1- Fonction `main()`	2
2- Utilisation de `MaterialApp`	2
3- Propriété `home`	2
4- Propriété `debugShowCheckedModeBanner`	2
C- Détection de l'humeur à partir d'images : detector.dart.....	3
1- Classe `PersonMood`	3
2- Etat de la classe `PersonMood`	3
D- Configuration du projet Android dans le module Flutter : lib/app/build.gradle	6
1. Lecture des propriétés locales.....	6
2. Configuration du projet Android :	6
3. Configuration de la version Android :	6
4. Configuration de Flutter :	6
5. Dépendances :	6