

Rapport De Projet

***Module : processus de développement de logiciel
(PDL)***

***Domaine : Informatique
Master : Systèmes d'Information et
Technologies du Web***

Responsable :	BOUMESSAOUD ABDELKADER
Assistant :	KAMRAOUI ADEL
Membre 1 :	BENZAZA IMEN
Membre 2 :	KAMBOUCHE MANEL

THEME

**APPRENTISSAGE DE LANGUES PAR RENFORCEMENT POUR
LES ENFANTS**

Encadré par : Mr Mohamed Sayah Université Oran 1

INTRODUCTION

L'introduction est structurée autour de l'historique, du contexte, ensuite l'énoncé du problème à résoudre et se termine par la faisabilité.

Historique

Le développement de l'informatique mobile est accéléré depuis le 21^{esiècle}. De nouveaux dispositifs (mobiles, tablette PC, PDA, téléphone mobile, Smartphone, etc.) Offrent une portabilité accrue de l'information et des communications sans fil (Wifi, Bluetooth, UMTS, etc...). Ces technologies ont influencées les changements des comportements et les habitudes des usagers dans beaucoup de secteurs comme l'éducation, la vie quotidienne, et l'apprentissage. Les technologies mobiles peuvent avoir une grande influence sur l'apprentissage. L'apprentissage commence à sortir des salles de cours et à entrer dans des environnements moins classiques, liés au contexte de l'apprenant. L'apprentissage devient ainsi situé, contextuel, personnel, collaboratif, et continu tout au long de la vie de l'apprenant. Utiliser des dispositifs mobiles pour apprendre, n'importe où et n'importe quand c'est l'objectif de l'apprentissage mobile (Mobile Learning M-Learning).

Contexte

Étant donné qu'une langue à son rythme, ses liaisons et son fonctionnement. C'est à force d'apprendre et d'écouter régulièrement qu'on se familiarise avec une langue authentique telle qu'elle est parlée au quotidien. Cet apprentissage de langue s'il est fait par un système mobile facilitera la tâche à l'enfant pour apprendre de manière formelle et non formelle par renforcement.

Énoncé du problème à résoudre

Dans ce travail il est question d'apprendre la langue française par renforcement aux enfants de bas Age [5-12], d'une manière authentiques, telle qu'on l'utilise à l'orale.

Faisabilité

En grande partie, il comprend la conception et la réalisation de l'application. Ainsi une étude préalable du sujet est nécessaire : la présentation du sujet, les besoins fonctionnels et non fonctionnels. Au niveau de la conception nous parlerons de toutes les fonctionnalités de l'application à créer, les acteurs intervenants et les méthodes de conception. Dans la réalisation, nous allons programmer les interfaces et vues permettant de faire fonctionner l'application.

I-PROCESSUS DU DEVELOPPEMENT DU LOGICIEL

ANALYSE ET CLARIFICATION DE LA DEMANDE

1.1 Expression de la demande

Outil d'apprentissage du français pour les enfants

1.2 Reformulation de la demande et verbes d'action

1. L'application génère des phrases à partir de textes compréhensible par un enfant (ex : Fables).
2. Analyse de la syntaxe du texte saisie par l'enfant, à l'aide de l'API Google Translate.
3. Evaluation des phrases introduites par l'enfant avec le retour d'un pourcentage de réussite.

1.3 Résumé et Objectifs (L'essentiel du projet)

Vous voulez que vos enfants grandissent en connaissant la langue française de manière correcte ? Cet outil numérique pratique sous forme d'application mobile peut enseigner aux enfants de tous âges, en se focalisant surtout sur la formulation de phrases correcte. Permettant aux enfants de basculer vers la langue française et aider d'autres à développer leur vocabulaire quelques mots à la fois. En un rien de temps, vos enfants seront sur la bonne voie pour devenir parfaitement fluent en français. De plus, les parents peuvent également les utiliser.

1.4 Fournitures (Description de ce qui est livrable au client)

Application mobile androïde de catégorie enseignement pour enfants, gratuite au téléchargement sur Play Store ou sources parallèle sous forme de fichier 'apk'.

1.5 Définition et Acronymes (Explication des mots spécifiques au projet ou aux techniques utilisées)

APK : Android Package.

UX : User Expérience

API : Application Programming Interface

UML : Unified Modeling Language

SPÉCIFICATION FONCTIONNELLE ET TECHNIQUE (UML)

2 UML

2.1 Modélisation statique

2.1.1 Diagramme de classes

Un diagramme de classe fait partie de la catégorie des diagrammes de structure. C'est un schéma représentant toutes les classes d'un programme, leurs attributs, leurs méthodes, et les relations qu'il comporte.

La figure (1) représente le diagramme de classe de notre application. Nous y trouvons 10 classes

- **Administrateur**
- **Enfant**
- **Exercice**
- **Génération de phrases**
- **Question/réponse**
- **Apprendre**
- **Alphabet**
- **Nombres**
- **Jours**
- **Mois**

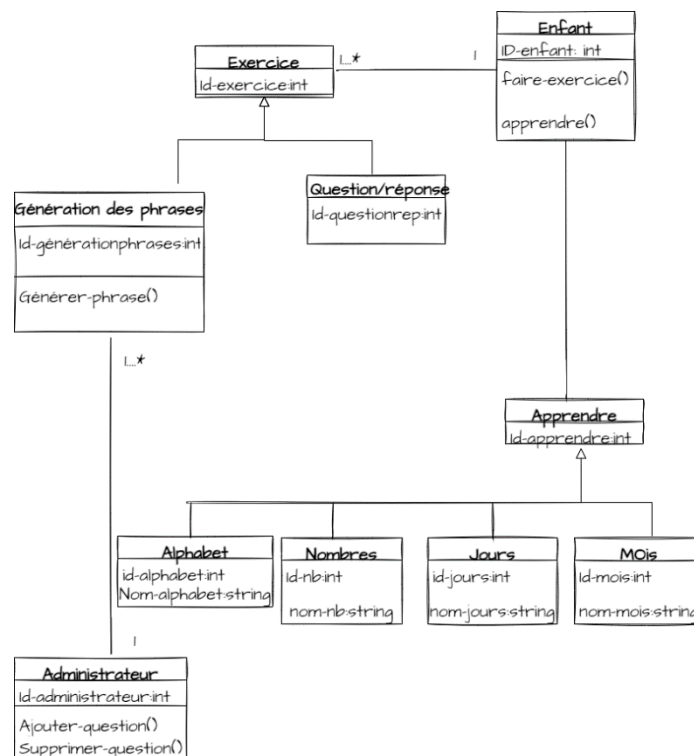


Figure 1: Diagramme de classe de l'application

2.1.2 Diagramme de cas d'utilisation

La figure ci-dessous (2) illustre le diagramme de cas d'utilisation pour les acteurs du système.

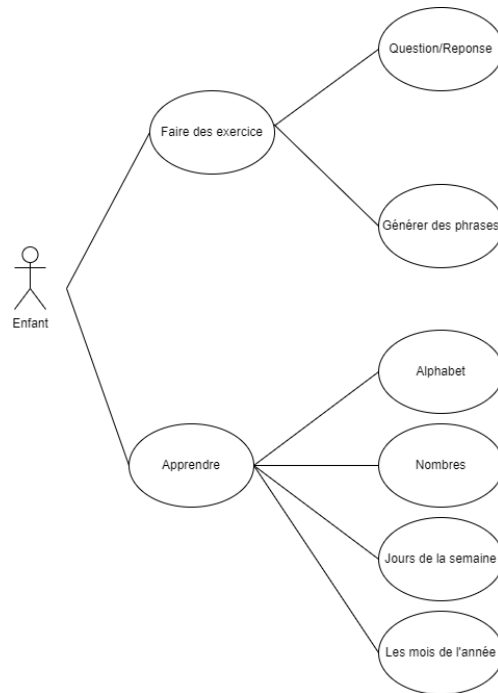


Figure 2 : Cas d'utilisation

2.2 Modélisation dynamique

Cette modélisation nous permettra de donner plus de détails sur quelques cas d'utilisation du système. Nous le ferons à travers une description textuelle illustrée par des diagrammes de séquences.

2.2.1 Diagramme de séquence

Il permet de représenter graphiquement un scénario. Le diagramme de séquence décrit la dynamique d'un système.

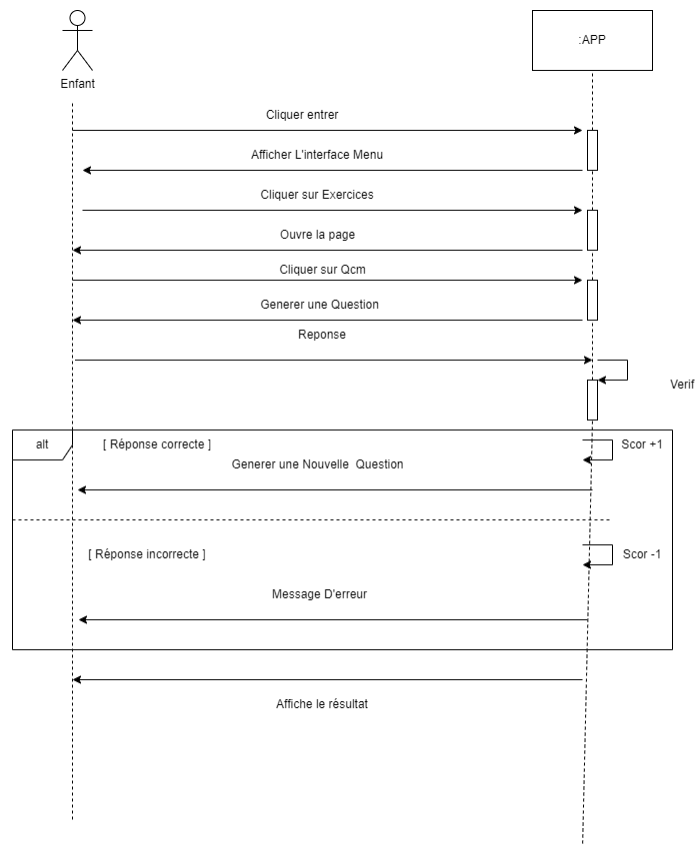


Figure 3: Diagramme de séquence ouverture de l'application et exercice qcm

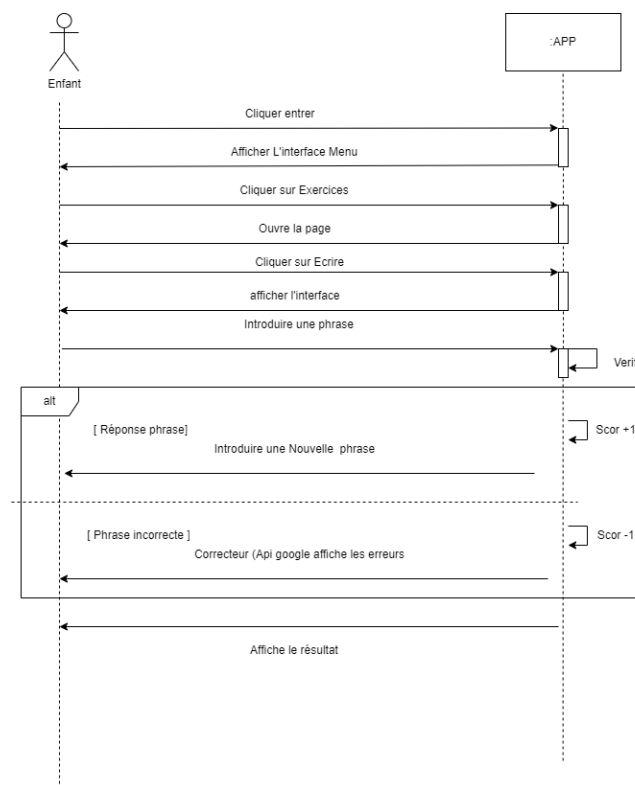


Figure 4: Diagramme de séquence ouverture de l'application et generation_de_phrase

2.2.2 Diagramme d'activité

Le diagramme d'activités permet de déterminer des traitements a priori séquentiels. Ils offrent un pouvoir d'expression très proche des langages de programmation

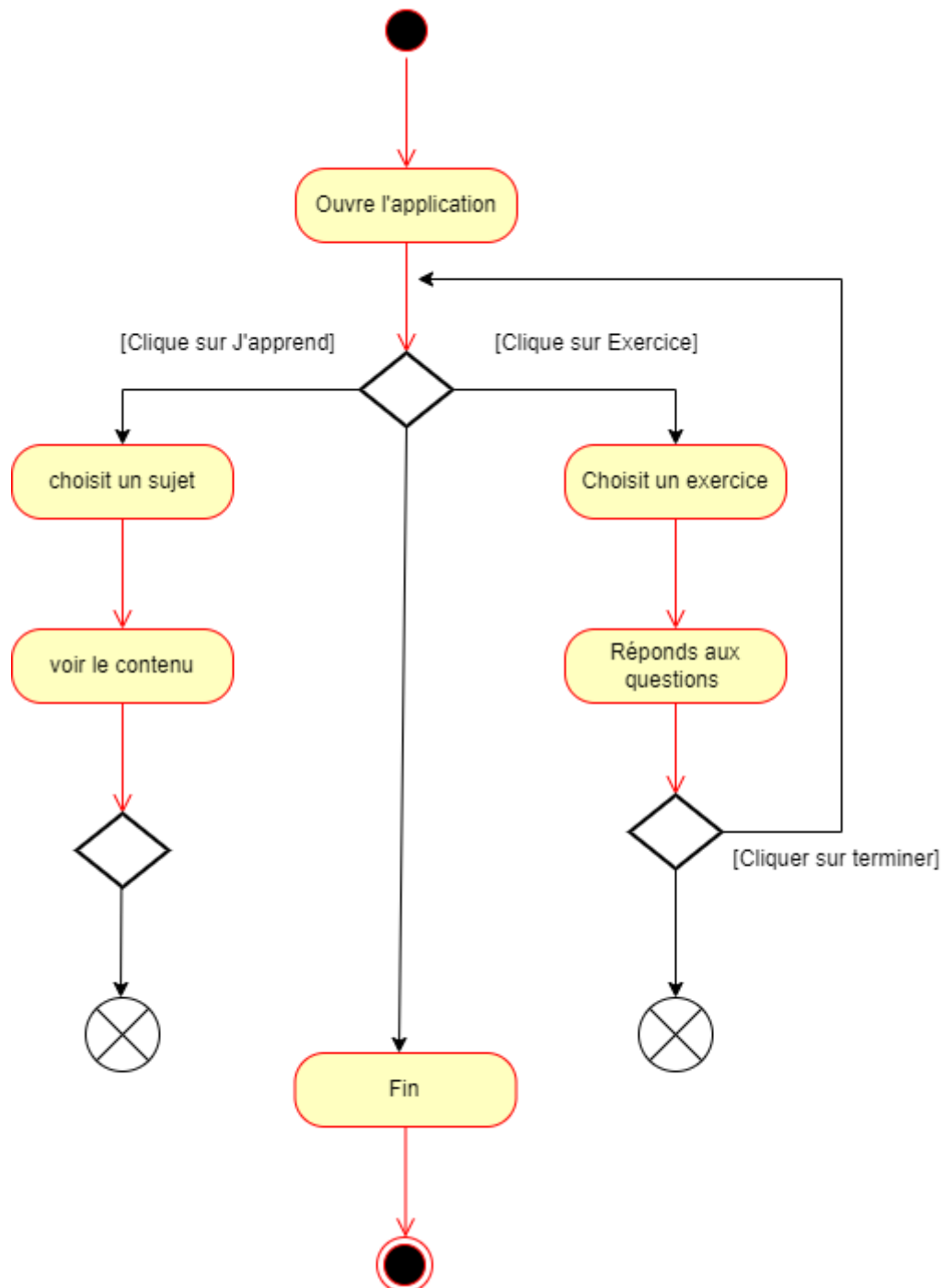


Figure 5 : Diagramme d'activité

CONCEPTION GÉNÉRALE

3.1 Architecture globale

3.1.1 Définition

Dans la plus simple des définitions, l'architecture d'application est une collection de tous les aspects, interfaces et techniques utilisés pour concevoir une application mobile et le comportement global de tous ces éléments structurels. En d'autres termes, l'architecture de l'information pour les applications mobiles fait référence à la feuille de route initiale et à un guide complet selon lequel une application mobile est censée fonctionner. Cela devrait donner une idée précise de la structure des applications mobiles. Si l'architecture de l'application ne définit pas l'ensemble de la vision et de la feuille de route, il est plus probable que l'application finira par échouer.

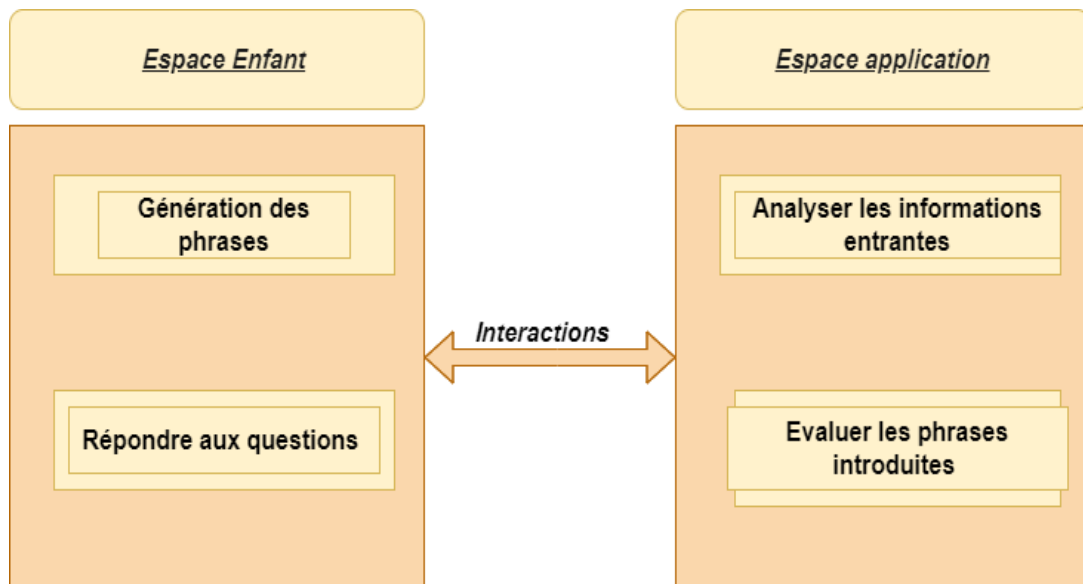


Figure 6 : Architecture Générale

3.1.2 Architecture choisis

Dans notre cas, on a choisi une architecture d'application mobile native. Les applications natives sont des applications mobiles conçues pour un système d'exploitation (OS) particulier. Comme ils fonctionnent conformément aux mécanismes intégraux des appareils mobiles, ils créent une sensation «naturelle». Les utilisateurs sont plus susceptibles de trouver l'apparence et l'utilisation de ces applications pratiques, contrairement aux applications mobile web progressives (PWA) qui peuvent causer plusieurs restrictions dans notre cas.

3.2 Description des composants et Spécification des interfaces

Les composants d'application sont les blocs de construction essentiels d'une application Android. Chaque composant est un point d'entrée par lequel le système ou un utilisateur peut accéder à votre application. Certains composants dépendent des autres. Il existe quatre types différents de composants d'application

Chaque type sert un objectif distinct et à un cycle de vie distinct qui définit la manière dont le composant est créé et détruit. Les sections suivantes décrivent les quatre types de composants d'application.

3.2.1 Activités

Description de l'interface utilisateur, l'interaction de l'utilisateur avec la machine et le traitement de l'écran.

Les activités qui définissent notre application sont :

1/ MainActivity : la première interface qui apparaît lorsqu'on lance l'application



Figure 7 : lancement de l'application

2/ SecondActivity (interface Menu) : cette interface contient un menu des différentes activités d'apprentissage

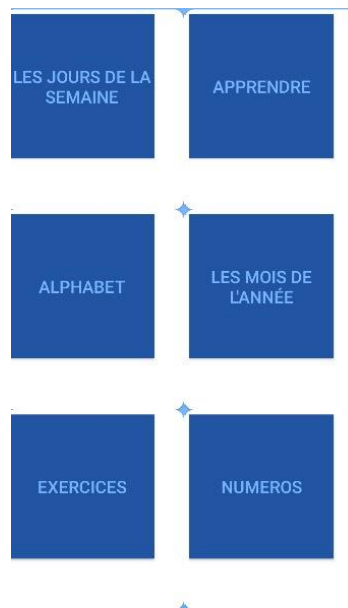


Figure 8 : Interface Menu

3/ Exercices : une interface qui propose deux type d'exercice (Question/Réponse) ou (Générer des phrase)

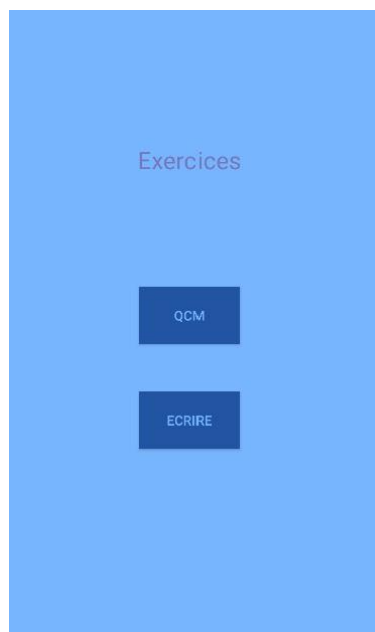


Figure 9 : Interface Exercice

4/ Qcm : cette interface permet à l'enfant de répondre aux différentes questions posé et de savoir son niveau d'apprentissage



Figure 10 : Interface QCM

5/ Generer_phrase: une interface très importante dans l'application ou on analyse et on évalue des phrases générées par l'enfant en corrigeant ses erreurs en utilisant l'api Google

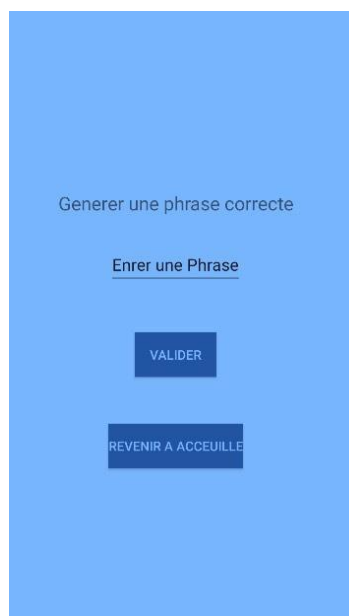


Figure 11 : Interface Génération De Phrases

6/ Alphabet, Mois de l'année, Jours de la semaine, Nombres: Des interfaces d'apprentissage où l'enfant teste son niveau

3.2.2 Services

Des opérations de traitement d'arrière-plan associé à l'application.

3.2.3 Récepteurs de radiodiffusion (Broadcast Receivers)

Système d'exploitation Android et gère la communication entre les applications.

3.2.4 Fournisseurs de contenu (Content Providers)

Traiter le problème des données et la gestion de base de données.

3.3 Spécification des interfaces

3.4 Langages d'implémentation

- **Le langage Java**

Un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

- **Le langage SQL**

Un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

- **Le langage PHP**

HyperText Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

3.5 Choix des outils de développement

Une liste d'outils de développement utilisés est proposée ci-dessous.

- **Android Studio**

Un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux¹⁶. Il permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

- **XAMPP**

Un ensemble de logiciels permettant de mettre en place facilement le serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation.

- **Apache**

Un serveur web chargé de délivrer les pages web aux visiteurs. Il ne gère que les sites web statiques (il ne peut traiter que des pages HTML). Il faut donc le compléter avec d'autres programmes.

- **Plug-in PHP**

Un plug-in pour Apache qui le rend capable de traiter des pages web dynamiques en PHP. En combinant Apache et PHP, l'ordinateur est capable de lire des pages web en PHP.

- **Serveur MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR) libre fonctionnant sous diverses plates-formes telles que UNIX, Linux et Windows, et permettant de manipuler des instructions adressées à la base de données Sous forme de requêtes SQL.

- **Draw.io**

Conçu par Seibert Media, draw.io est un logiciel permettant de créer des diagrammes et des graphiques. Il permet de choisir parmi une fonction de mise en page automatique ou de créer une mise en page personnalisée.

CONCEPTION DÉTAILLÉ – INTÉGRATION

4.1 Implémentation des méthodes/fonctions

```
public void checkButton(View v) {  
    boolean checked = ((RadioButton) v).isChecked();  
  
    switch (v.getId()) {  
        case R.id.radiom1:  
            if (checked)  
                reponse.setText("Mauvaise Reponse");break;  
        case R.id.radiom2:  
            if (checked)  
                reponse.setText("Bonne Reponse");break;  
    }  
}
```

checkButton() : fonction qui compare la réponse sélectionner par l'enfant et la bonne réponse et affiche des messages pour chaque réponse;

```
Button button1 =findViewById(R.id.button1);  
button1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        startActivity(new Intent( packageContext: MainActivity.this,SecondActivity.class));  
    }  
});
```

onClick() : fonction qui permet de naviguer de la première interface vers le menu

```
qcm.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        startActivity(new Intent( packageContext: Exercices.this,Question1.class));  
    }  
});  
ecrire.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        startActivity(new Intent( packageContext: Exercices.this,Generer_Phrase.class));  
    }  
});
```

La fonction qui permet de choisir entre Question/Réponse ou Générer une phrase dans l'interface exercice

4.2 Intégration de l'API

4.2.1 API

Une API Web est une interface de programmation d'applications pour un serveur Web ou un navigateur Web. L'API permet aux développeurs d'interagir avec des services externes. Ce sont les commandes que le développeur du service a déterminé qu'elles seront utilisées pour accéder à certaines fonctionnalités de leur programme.

Dans notre cas on s'intéressera à l'API Google Translate

4.2.2 Activation de l'API

Il faut passer par la plate-forme Google (developers.google.com/apis-explorer) créé spécifiquement pour l'aide aux développeurs d'applications et activer l'API souhaité en se dotant d'une clé à cet effet.

Cloud Translation API

Google Enterprise API

Integrates text translation into your website or application.

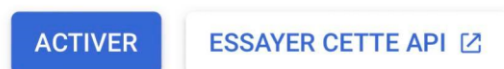


Figure 14 : Activation de l'API Google Translate

Identifiants

Ajouter des identifiants au projet

✓ Déterminer les identifiants dont vous avez besoin
Appel de Google Maps Android API depuis Android

2 Créer une clé d'API

Nom

Limiter l'utilisation de vos applications Android (Facultatif)
Les requêtes API nous sont directement envoyées par les appareils Android. Nous vérifions que chaque requête provient d'une application Android dont le nom de package et l'empreinte de signature SHA-1 correspondent à ce que vous avez indiqué. Pour connaître le nom du package, reportez-vous au fichier AndroidManifest.xml, et pour obtenir l'empreinte, utilisez la commande suivante. [En savoir plus](#)

```
keytool -list -v -keystore mystore.keystore
```

Nom du package	Empreinte du certificat SHA-1
<input type="text" value="com.example"/>	<input type="text" value="12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:AA:BB:CC:DD"/>

[+ Ajouter nom du package et empreinte](#)

[Créer la clé d'API](#)

Figure 12 : Création d'une clé API

4.2.3 L'intégration

L'intégration d'API est la connexion entre deux ou plusieurs applications via leurs API (interfaces de programmation d'applications) qui permettent aux systèmes d'échanger des sources de données.

4.2.3.1 Intégration en XML

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
3     <uses-permission android:name="android.permission.INTERNET"/>
4     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
5     <!--Activities below as usual-->
6
7 </manifest>
```

4.2.3.2 Intégration en JSON

```
1 implementation('com.google.cloud:google-cloud-translate:1.12.0') {
2     exclude group: 'org.apache.httpcomponents'
3     exclude group: 'org.json', module: 'json'
4 }
5 annotationProcessor 'com.google.cloud:google-cloud-translate:1.12.0'
```

4.2.3.3 Intégration en Java

```
1 android {
2     ...
3     packagingOptions {
4         exclude 'project.properties'
5     }
6 }
```


TESTS ET VALIDATION

5.1 Test de conformité

5.1.1 La démarche UX

L'UX est un sigle anglais qui signifie User eXperience, en français on parle d'expérience utilisateur, le design a pour but d'améliorer le parcours de l'internaute pour le rendre plus agréable et instinctif. L'UX est un ensemble de normes et de bonne pratique dont l'objectif est double :

Concevoir un produit qui réponde très exactement aux besoins et aux attentes de l'utilisateur en faisant en sorte que son expérience soit la meilleure possible, il faut travailler sur l'ergonomie du produit et anticiper d'usage. Pour cela il faut d'abord définir les caractéristiques de l'utilisateur. Ce profil utilisateur s'appelle une persona. « *Un bon design commence par une bonne compréhension de la psychologie humaine* » Steve Krug.

Prenons pour exemple un enfant (une persona):

- Il a des journées peu remplies et a beaucoup de temps.
- Il n'aime pas lire beaucoup de texte.
- Il est occasionnellement sur un smartphone.
- Il est toujours curieux.

Les cas d'usage :

- Réduire le nombre de "clics" et limiter la difficulté des actions à réaliser pour interagir avec l'application.
- Envoyer des notifications push pour qu'il soit au courant des nouveaux exercices susceptible d'attiser sa curiosité
- Ajouter des pubs vers d'autres applications d'apprentissage pour les enfants.

5.2 Test structurel

Sachant que les anomalies ne peuvent pas être découvertes si les parties les contenant ne sont pas exécutées, le teste structurel est très important pour la validation d'une application.

Il repose sur des analyses du code source et l'exécution du code binaire.

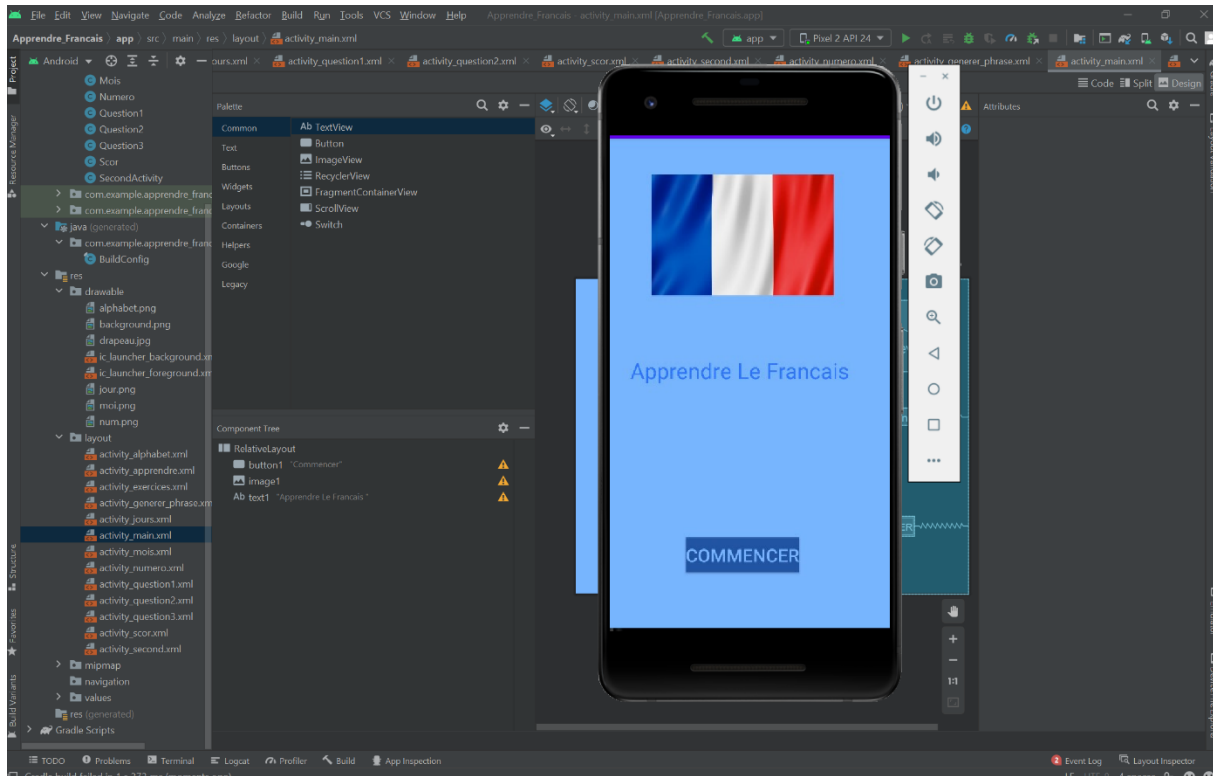


Figure 13 : Test Structurel De L'application

DÉPLOIEMENT

6.1 Livrables – Solutions

Les livrables (déverbales en anglais) et les jalons (mile stones en anglais) sont des termes standard utilisés dans le management de projet. Un livrable est un produit ou un service donné au client. Il s'agit de l'achèvement d'un processus, d'une tâche, d'une partie du projet ou de l'aboutissement de la totalité du projet.

Un projet peut donc avoir plusieurs livrables. Généralement, un livrable est tangible, mesurable et spécifique et il possède une date butoir. Un livrable est le résultat d'un jalon ou d'une échéance inscrit dans le planning du projet. Ce peut être un logiciel, un document de conception, un programme de formation ou tout autre élément, produit ou résultat requis dans le planning du projet.

Dans notre cas les livrables sont :

- Un Rapport De Projet
- Une Application

6.2 Contraintes matérielles

Ordinateur capable d'exécuter Android Studio, c'est-à-dire d'une configuration minimale suivante :

- Microsoft Windows 7/8/10 (32 bits ou 64 bits)
- 3 Go de RAM minimum, 8 Go de RAM recommandés (plus 1 Go pour l'émulateur Android)
- 2 Go d'espace disque disponible minimum, 4 Go recommandés (500 Mo pour IDE plus 1,5 Go pour Android SDK et image système de l'émulateur)
- Résolution d'écran minimale de 1280 x 800.

II-ORGANISATION ET GESTION DU PROJET

ORGANISATION STRUCTURELLE

1.1 Chef de Projet :

BOUMESSAOUD Abdelkader

1. Mise en forme du rapport.
2. Planification du projet.
3. Distribution des tâches.
4. Validation de l'application

1.2 Adjoint du Projet :

KAMRAOUI Adel

1. Demander une clé de l'api de Google translate.
2. Intégrer la clé de l'api sur Android studio.
3. Faire appel aux fonctions pour l'interaction avec l'application.

1.3 Membres du Projet :

BENZAZA Imen

1. Création des diagrammes
2. Codage de l'application
3. Aide à la rédaction

KAMBOUCHE Manel

1. Création de diagrammes
2. Rédaction des préliminaires

ECHEANCIER DU DEVELOPPEMENT

2.1 Planification du projet (GitHub)

Avec GitHub Project Planning, on peut utiliser les référentiels GitHub, les problèmes, les tableaux de bord de projet et d'autres outils pour planifier et suivre notre travail, surtout pour une équipe inter fonctionnelle.

On peut créer des notes dans les colonnes pour servir de rappels de tâches, de références à des problèmes et de demandes d'extraction à partir de n'importe quel référentiel sur GitHub.com, ou pour ajouter des informations relatives au tableau de bord du projet.

On peut créer une carte de référence pour un autre tableau de projet en ajoutant un lien vers une note. Si la note n'est pas suffisante pour nos besoins, on peut la convertir en un problème.

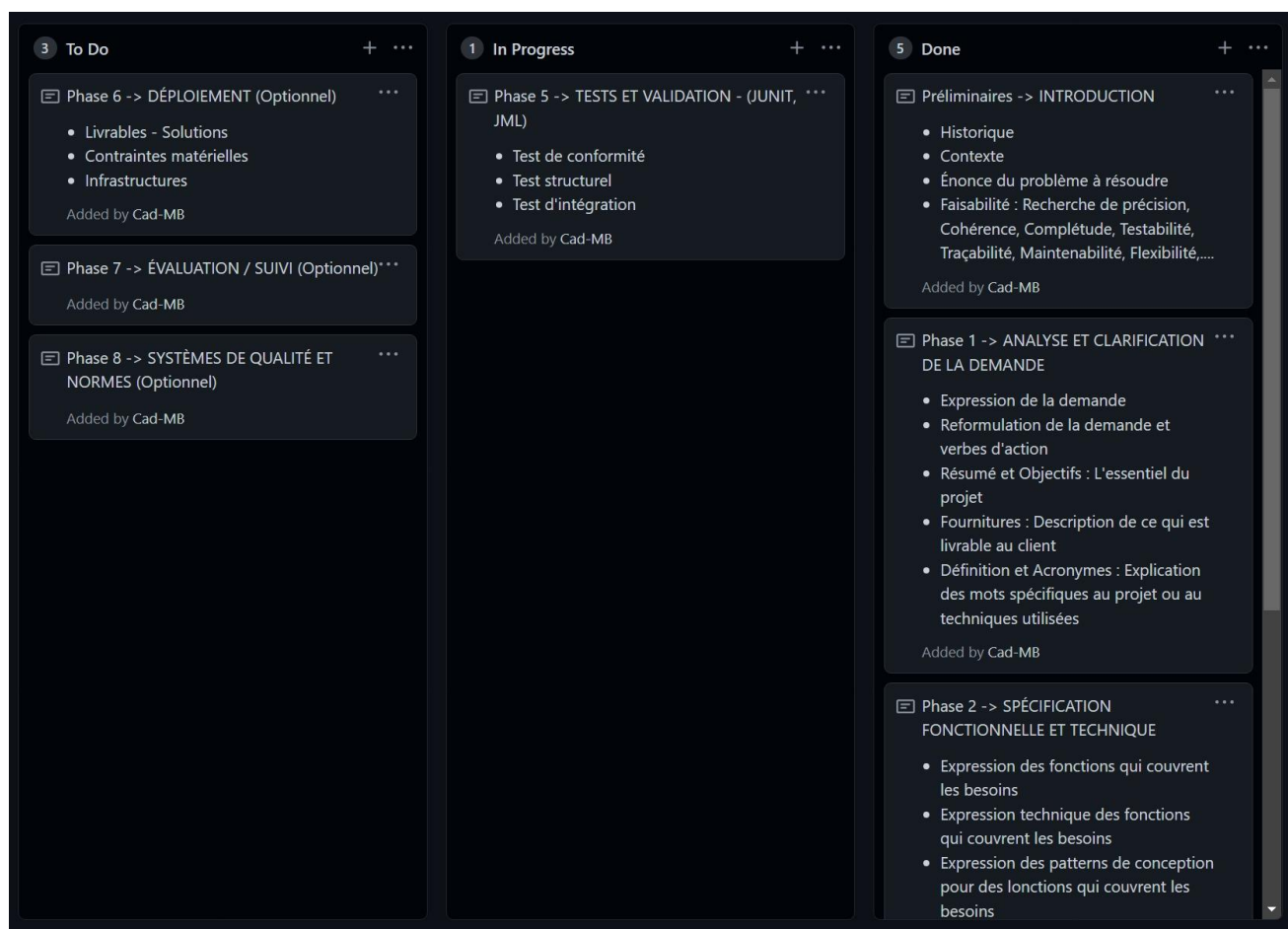


Figure 14 : GitHub Project Planning