



# Intelligence Artificielle pour le Génie Logiciel

---

## Projet programmation mobile: MeFormer

Auteur :

SAR Alexandre

numéro étudiant: 21900476

Décembre 2024



Lien GitHub du Projet : [https://github.com/texao/projet\\_mobile\\_IA\\_GL](https://github.com/texao/projet_mobile_IA_GL)

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Spécification techniques</b>	<b>3</b>
2.1	Présentation des fonctionnalités principales . . . . .	3
2.2	Différents profils d'utilisateurs . . . . .	4
<b>3</b>	<b>Conception et architecture</b>	<b>5</b>
3.1	Interface utilisateur . . . . .	5
3.1.1	Écran des de bases . . . . .	5
3.1.2	Tableaux de bord . . . . .	5
3.1.3	Écrans de fonctionnalités spécifiques . . . . .	6
<b>4</b>	<b>Tests et validation</b>	<b>7</b>
4.1	Plan de tests . . . . .	7
4.1.1	LoginActivityTest . . . . .	7
4.1.2	ParentDashboardActivityTest . . . . .	7
4.1.3	StudentDashboardActivityTest . . . . .	7
4.1.4	StudentProgressActivityTest . . . . .	7
4.2	Résultats des tests . . . . .	8
4.3	Outils de tests utilisés . . . . .	10
<b>5</b>	<b>Conclusion</b>	<b>11</b>

# 1 Introduction

L'objectif principal de ce projet est de développer une application mobile sous Android destinée à la formation en ligne. L'application vise à offrir des cours et des services éducatifs à différents profils d'utilisateurs : parents et élèves. Elle permet aux utilisateurs de s'inscrire, de consulter des contenus éducatifs et de suivre les progrès des élèves. L'application doit également proposer une utilisation en mode en ligne.

## 2 Spécification techniques

### 2.1 Présentation des fonctionnalités principales

L'application *MeFormer* offre une plateforme intuitive pour l'apprentissage en ligne. Les fonctionnalités principales incluent :

- **Présentation de l'application :** L'application propose une interface conviviale permettant aux utilisateurs de découvrir ses fonctionnalités.
- **Inscription et gestion des utilisateurs :** Deux profils d'utilisateurs sont pris en charge : parent et élève. Chaque profil bénéficie de services adaptés à ses besoins spécifiques, comme l'ajout d'élèves pour les parents ou l'accès direct aux contenus pour les élèves.
- **Accès aux contenus pédagogiques :** Les utilisateurs inscrits peuvent consulter des cours.
- **Mode de connexion :**
  - *Mode online :* Permet d'accéder à l'intégralité des fonctionnalités de l'application.
- **Recommandations personnalisées :** L'application génère des suggestions pour les élèves afin d'améliorer leur progression, basées sur leurs performances.
- **Statistiques et suivi :** Les parents peuvent consulter des statistiques détaillées sur les activités de leurs enfants ou les courbes de progression.

Ces fonctionnalités permettent une expérience utilisateur adaptée à chaque profil, tout en respectant les contraintes techniques et pédagogiques de l'apprentissage en ligne.

## 2.2 Différents profils d'utilisateurs

L'application *MeFormer* propose une gestion des utilisateurs structurée autour de deux profils principaux : les parents et les élèves. Ces profils permettent d'adapter l'expérience utilisateur en fonction des rôles et des besoins spécifiques de chacun.

- **Parents** : Les parents agissent comme superviseurs pédagogiques, ils ont la possibilité d'inscrire leurs enfants, de suivre leur progression, et de configurer des rappels pour encourager l'utilisation régulière de l'application.
- **Élèves** : Les élèves, ce sont les utilisateurs principaux, bénéficient d'un accès direct aux contenus pédagogiques et d'outils interactifs pour apprendre.

Chaque profil est conçu pour interagir avec l'application de manière unique, en fonction des fonctionnalités et responsabilités associées. Cette distinction garantit une expérience personnalisée et efficace pour tous les utilisateurs.

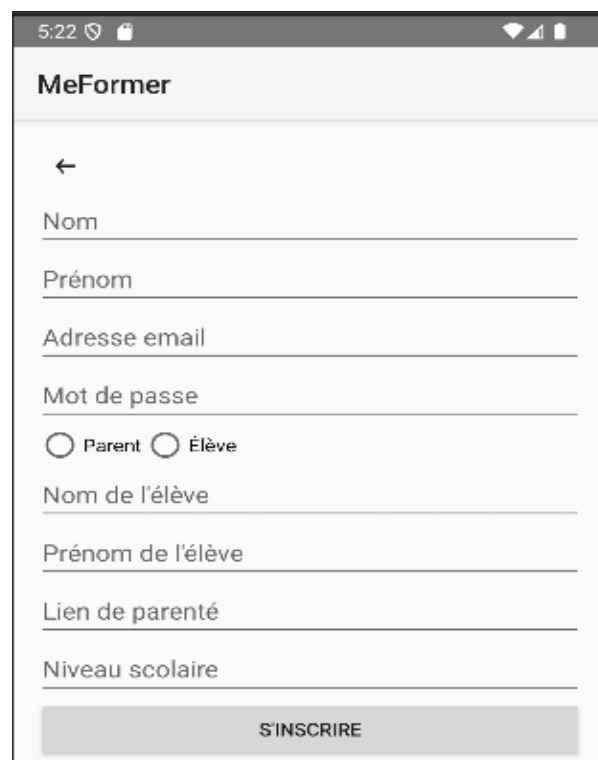
The image shows a mobile application interface for user registration. At the top, the status bar displays the time 5:22 and various icons. Below the status bar, the app title "MeFormer" is centered. A back arrow icon is positioned above the first input field. The form consists of several text input fields: "Nom", "Prénom", "Adresse email", "Mot de passe", "Nom de l'élève", "Prénom de l'élève", "Lien de parenté", and "Niveau scolaire". Below the "Mot de passe" field, there are two radio buttons labeled "Parent" and "Élève". At the bottom of the form, there is a large, light gray button labeled "S'INSCRIRE".

FIGURE 1 – Inscription utilisateur

Sur l'image ci-dessus, nous pouvons voir la page d'inscription d'un utilisateur. En effet, il comporte plusieurs champs (prénom, adresse, parent ou élève, niveau de scolarité).

## 3 Conception et architecture

### 3.1 Interface utilisateur

L'interface utilisateur de l'application *MeFormer* a été conçue pour offrir une expérience intuitive. Chaque composant de l'interface a été développé pour répondre aux besoins spécifiques des différents profils d'utilisateurs (élèves et parents).

#### 3.1.1 Écran des de bases

- **Écran d'accueil (Accueil)** : Cet écran sert de point d'entrée à l'application. Il propose une présentation concise des fonctionnalités principales via du texte. L'utilisateur peut choisir de se connecter ou de s'inscrire.
- **Écran de connexion ou inscription** : Ces écrans permettent aux utilisateurs existants de se connecter en saisissant leurs identifiants (email et mot de passe) ou de s'inscrire.

#### 3.1.2 Tableaux de bord

- **Tableau de bord parent (ParentDashboard)** : Spécifiquement conçu pour les parents, cet écran affiche :
  - Les cours de l'élève.
  - La progression de l'élève.
  - Des recommandations.



FIGURE 2 – Tableau de bord des parents

- **Tableau de bord élève (StudentDashboard) :** Accessible aux élèves, ce tableau de bord regroupe :
  - Les cours disponibles.
  - Les exercices associés.
  - Les recommandations

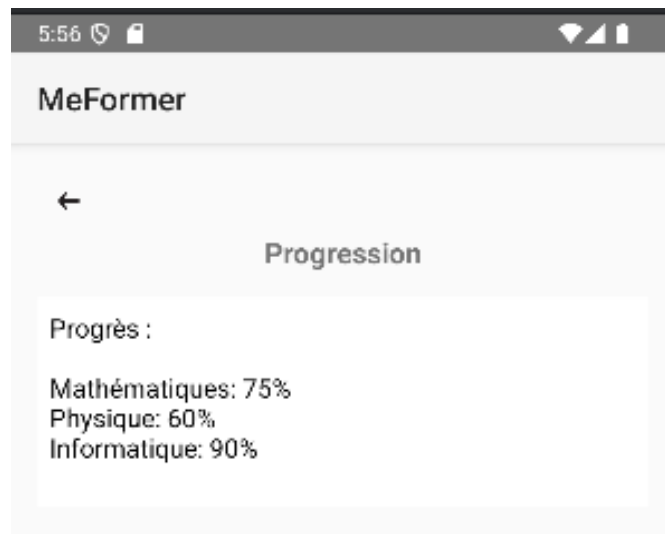


FIGURE 3 – Progrès de l'élève

### 3.1.3 Écrans de fonctionnalités spécifiques

- **Suivi des progrès (ProgressActivity et StudentProgressActivity) :** Ces écrans permettent de visualiser les statistiques liées à la progression des élèves.
  - *ProgressActivity* : Vue détaillée pour les élèves, montrant leur progression individuelle.
  - *StudentProgressActivity* : Vue globale pour les parents.
- **Recommandations (RecommendationActivity) :** Cet écran affiche des suggestions personnalisées basées sur les performances de l'élève. Il propose des conseils pour s'améliorer et des cours/exercices adaptés.

Cette conception centrée sur l'utilisateur assure que chaque écran répond aux besoins spécifiques de l'utilisateur tout en restant intuitif à utiliser.

## 4 Tests et validation

### 4.1 Plan de tests

Cette section détaille le plan de tests mis en œuvre pour valider les fonctionnalités de l'application. Les tests sont organisés par fichier de tests, chacun correspondant à un ensemble de fonctionnalités spécifiques. Pour chaque fichier, les objectifs des tests et les cas testés sont décrits.

#### 4.1.1 LoginActivityTest

- **Objectif** : Vérifier la validité des informations d'identification des utilisateurs (parents et étudiants) et leur gestion.
- **Cas testés** :
  - Validation des identifiants corrects pour les parents et les étudiants.
  - Gestion des identifiants incorrects.
  - Récupération du profil utilisateur (parent, étudiant, inconnu).
  - Cas limites : identifiants vides, nulls, format d'email incorrect.

#### 4.1.2 ParentDashboardActivityTest

- **Objectif** : Tester les fonctionnalités du tableau de bord des parents.
- **Cas testés** :
  - Affichage correct des informations de progression de l'élève.
  - Interaction avec les fonctionnalités du tableau de bord.

#### 4.1.3 StudentDashboardActivityTest

- **Objectif** : Vérifier l'affichage et la fonctionnalité du tableau de bord des étudiants.
- **Cas testés** :
  - Affichage des progrès de l'étudiant.
  - Accès aux fonctionnalités du tableau de bord.

#### 4.1.4 StudentProgressActivityTest

- **Objectif** : Tester la gestion et l'affichage de la progression des étudiants.
- **Cas testés** :
  - Calcul et affichage des progrès.
  - Gestion des erreurs dans les données de progression.

D'autres tests ont également été effectués tels que `UtilsTest` qui gère la validation des emails ou `SetReminderActivityTest` qui permet de valider les recommandations.

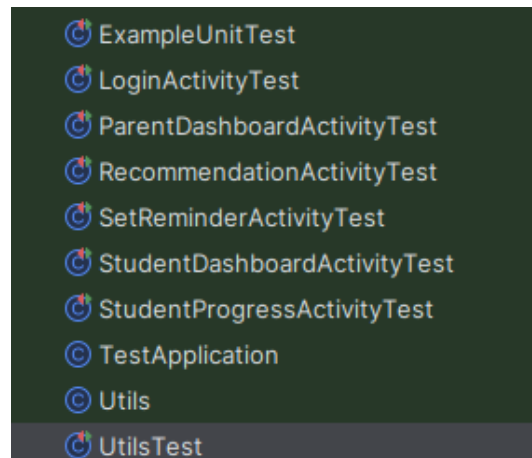


FIGURE 4 – Fichiers de tests

Voici les différents fichiers de fichiers effectués. Chacune teste des fonctionnalités différentes.

## 4.2 Résultats des tests

Les tests ont été exécutés sur l'ensemble des fichiers de test mentionnés dans le plan de tests. Voici les résultats générés par Sonarqube :

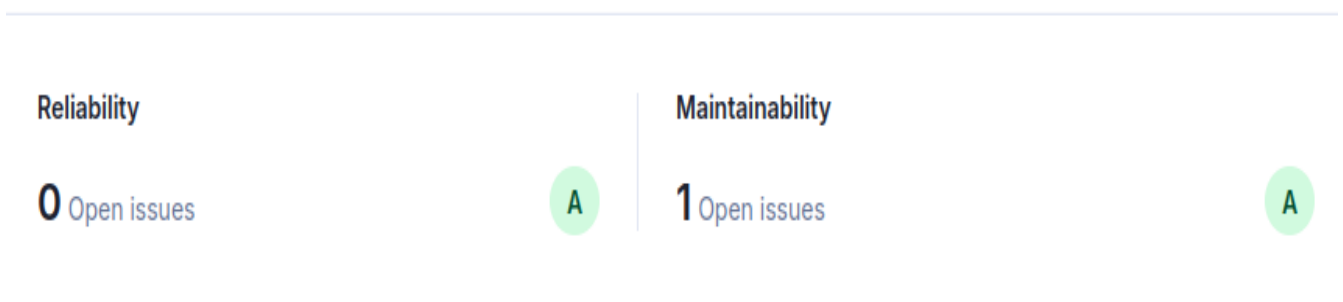


FIGURE 5 – Test Sonarqube

L'image ci-dessus illustre les résultats des analyses effectuées à l'aide de SonarQube pour évaluer la qualité. Deux aspects clés de la qualité du code sont représentés : la fiabilité (Reliability) et la maintenabilité (Maintainability).



Détails des résultats :

Fiabilité (Reliability) : L'indicateur de fiabilité montre qu'il n'y a aucune anomalie critique. Cela indique que le code respecte les bonnes pratiques en termes de gestion des exceptions et d'absence de bugs critiques.

Maintenabilité (Maintainability) : Un problème ouvert a été détecté dans cette catégorie. Ce problème peut être lié à des aspects tels que la complexité, la duplication de code ou l'usage inapproprié de certaines structures de programmation.

Ces résultats témoignent d'un bon niveau global de qualité du code.

#### Couvertures :

Voici les résultats de couverture des tests.

## Package com.example.tp1devmobile

[all](#) > com.example.tp1devmobile

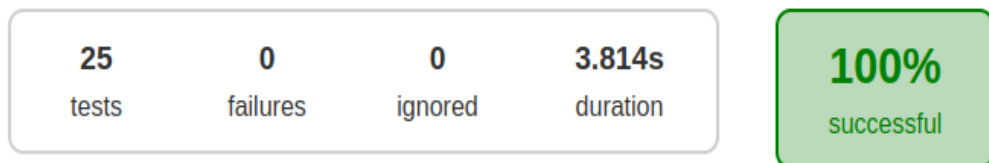


FIGURE 6 – Test Sonarqube

#### Analyse des résultats

L'image présentée fournit un résumé des résultats des tests. Voici les informations :

- **25 tests** : Un total de 25 tests unitaires ont été exécutés.
- **0 failures** : Aucun test n'a échoué. C'est un excellent indicateur de la stabilité de votre code.
- **3.814s duration** : La durée totale de l'exécution de tous les tests est de 3,814 secondes.
- **100% successful** : Le taux de réussite des tests est de 100%, ce qui confirme l'absence d'échecs.

## Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">ExampleUnitTest</a>	1	0	0	0.002s	100%
<a href="#">LoginActivityTest</a>	10	0	0	2.380s	100%
<a href="#">ParentDashboardActivityTest</a>	3	0	0	0.598s	100%
<a href="#">RecommendationActivityTest</a>	1	0	0	0.107s	100%
<a href="#">SetReminderActivityTest</a>	1	0	0	0.099s	100%
<a href="#">StudentDashboardActivityTest</a>	5	0	0	0.432s	100%
<a href="#">StudentProgressActivityTest</a>	3	0	0	0.182s	100%
<a href="#">UtilsTest</a>	1	0	0	0.014s	100%

FIGURE 7 – Tests Couverture

Ce tableau présente de manière détaillée la couverture des tests pour chaque classe de notre application. On observe que tous les tests ont été exécutés avec succès, sans aucun échec ni test ignoré. Cela témoigne de la qualité et de la fiabilité du code développé.

### 4.3 Outils de tests utilisés

Dans le cadre de l'évaluation de la qualité et des fonctionnalités de l'application, j'ai utilisé deux outils de tests principaux : **Espresso** et **Robolectric**. Ces outils permettent de tester différents aspects du comportement de l'application Android, tels que l'interaction avec l'interface utilisateur (UI).

**Espresso** est un framework de test d'interface utilisateur (UI) pour les applications Android. Il permet de simuler des interactions avec l'interface, comme des clics sur des boutons ou des saisies de texte. Espresso garantit que l'UI réagit correctement aux actions de l'utilisateur en s'exécutant sur le même thread que l'interface graphique.

**Robolectric** est un framework de test unitaire pour les applications Android qui permet d'exécuter des tests sur la machine de développement sans nécessiter un appareil ou un émulateur Android. Il simule les composants Android, comme les activités et les vues, directement sur la JVM. Cela permet d'exécuter des tests plus rapidement.

## 5 Conclusion

En conclusion, le projet *MeFormer* m'a permis de développer une application mobile complète répondant aux besoins de formation en ligne pour différents profils d'utilisateurs, à savoir les parents et les élèves.

Pour garantir la qualité du code, j'ai utilisé SonarQube afin d'analyser la fiabilité et la maintenabilité de l'application. Les tests unitaires ont été exécutés avec succès, et SonarQube a fourni un retour sur les aspects à améliorer, notamment concernant la maintenabilité. J'ai également effectué des tests couverture.

Ce projet a été une opportunité pour appliquer les bonnes pratiques de développement mobile sous Android, notamment la qualité du code et l'expérience utilisateur.