



المدرسة العليا
للتكنولوجيا - الصويرة
L'ÉCOLE SUPÉRIEURE DE
TECHNOLOGIE – ESSAOUIRA

Rapport de PFE :

Application mobile de gestion des étudiants

Ecole Supérieure de Technologie d'Essaouira

Génie Informatique

Réalisé par :

KADIRI Mehdi

IDALIOUALI Imad

Encadré par :

M.BENKIRANE Said

Année universitaire : 2022 – 2023

Remerciement

Je souhaite exprimer ma gratitude envers toutes les personnes qui ont contribué à la réalisation de notre projet de fin d'études ainsi qu'à la création de notre application. Je suis sincèrement reconnaissant envers chacun d'entre eux pour leur aide inestimable et leur soutien tout au long de ce voyage enrichissant.

Tout d'abord, je voudrais remercier notre encadrant pédagogique, M.BENKIRANE Said, pour son soutien indéfectible tout au long de ce projet. Ses précieux conseils, orientations et suggestions ont été d'une importance capitale pour nous guider et nous aider à achever ce travail avec succès.

Je tiens également à exprimer ma gratitude envers tout le personnel de l'Ecole Supérieure de Technologie Essaouira, notamment les enseignants et les membres de l'administration, pour leur collaboration et leur soutien continu. Leurs retours constructifs nous ont permis de peaufiner nos idées et de faire évoluer notre application.

Je tiens également à remercier mon binôme, avec qui j'ai travaillé en étroite collaboration tout au long de ce projet. Nous avons travaillé ensemble, partageant nos idées et travaillant dur pour surmonter tous les obstacles. Cette expérience a été très enrichissante pour moi et je suis fier du travail que nous avons accompli ensemble.

Enfin, je suis extrêmement reconnaissant envers ma famille, mes amis et mes proches pour leur soutien inconditionnel. Leur encouragement, leur confiance et leur soutien ont été pour moi une source de motivation et m'ont permis de continuer à avancer dans les moments difficiles.

En conclusion, je suis extrêmement reconnaissant envers toutes les personnes qui ont contribué à la réussite de ce projet. Leur soutien et leur contribution ont été essentiels pour notre succès et je suis fier de faire partie d'une communauté aussi talentueuse et engagée.

Table des matières

I.	Introduction.....	7
II.	Chapitre 1 : Cadre général du projet	8
1.	<i>Introduction :.....</i>	8
2.	<i>Context et la présentation de projet :</i>	8
a)	Présentaion du projet :	8
b)	Le but à atteindre :	8
c)	Travail demandé :	8
3.	<i>Conclusion :</i>	8
III.	Chapitre 2 : Étude conceptuelle	9
1.	<i>Introduction :.....</i>	9
2.	<i>Langage UML :</i>	9
a)	Présentation du langage UML :.....	9
b)	Intérêt de la modélisation :	9
c)	Les avantages d’UML :	9
3.	<i>Conception avec UML :</i>	11
a)	Outil de modélisation :	11
b)	Modélisation avec les diagrammes cas d’utilisation :	11
c)	Le diagramme de Classe :	12
4.	<i>Conclusion :</i>	15
IV.	Chapitre 3 : La réalisation	16
1.	<i>Introduction :.....</i>	16
2.	<i>Environnement matériel :</i>	16
3.	<i>L’environnement logicielle :</i>	16
a)	Système d’exploitation :	16
b)	Choix de langage de programmation :	16
c)	Choix de framework :	17
d)	Choix de L’IDE :	18
e)	Firebase :	19
f)	Contrôle de version :	20
g)	Packages:	20

h)	Outils utilise:	22
4.	<i>Choix de développement</i> :	22
a)	Le modèle :	23
b)	La vue :	23
c)	Le contrôleur :	23
5.	<i>Les interfaces graphiques</i> :	23
i)	Interface Authentification :	24
j)	Interface Accueil :	25
k)	Interface Home :	27
l)	Interface Liste des annonces :	29
m)	Interface de communication entre les étudiants :	31
n)	Interface Profil de l'utilisateur :	32
o)	Interface d'edite de profile :	33
6.	<i>Conclusion</i> :	33
V.	Conclusion :	34
VI.	Références :	35

Liste des figures

<i>Figure 1 : Logo de UML.....</i>	<i>9</i>
<i>Figure 2 : Logo de StarUML.....</i>	<i>11</i>
<i>Figure 3 : diagramme de cas d'utilisation</i>	<i>12</i>
<i>Figure 4 : Diagramme de classe</i>	<i>13</i>
<i>Figure 5 : Logo de JavaScript.....</i>	<i>17</i>
<i>Figure 6 : Logo de React Native.....</i>	<i>18</i>
<i>Figure 7 : Logo de Visual Studio Code.....</i>	<i>19</i>
<i>Figure 8 : Logo de Firebase.....</i>	<i>19</i>
<i>Figure 9 : Logo de Git</i>	<i>20</i>
<i>Figure 10 : Logo de GitHub.....</i>	<i>20</i>
<i>Figure 11 : Schéma du Modèle MVC</i>	<i>23</i>
<i>Figure 12 : Interface Authentification</i>	<i>24</i>
<i>Figure 13 : Interface Accueil Partie Étudiant.....</i>	<i>25</i>
<i>Figure 14 : Interface Accueil Partie Professeur</i>	<i>26</i>
<i>Figure 15 : Interface Home.....</i>	<i>28</i>
<i>Figure 16 : Interface Liste des annonces Partie Étudiant</i>	<i>29</i>
<i>Figure 17 : Interface Liste des annonces Partie Professeur.....</i>	<i>30</i>
<i>Figure 18 : Interface de communication entre les étudiants.....</i>	<i>31</i>
<i>Figure 19 : Interface Profil de l'utilisateur.....</i>	<i>32</i>

Liste des tableaux

<i>Table 1 : Classe User</i>	14
<i>Table 2 : Classe Filière</i>	14
<i>Table 3 : Classe Annonce</i>	14

Liste des Acronymes

Android: Système d'exploitation mobile de Google

IHM: Interface Homme Machine

iOS: Système d'exploitation mobile de Apple

MVC: Model-View-Controller

UML: Unified Modeling Language

I. Introduction

Le présent rapport concerne le projet de fin d'études réalisé au sein de l'Ecole Supérieure de Technologie, portant sur la conception et le développement d'une application mobile de gestion pour l'école. Ce projet a été réalisé en binôme dans le cadre de la formation en cycle supérieur universitaire.

L'objectif de ce projet était de répondre aux besoins de l'école en proposant une solution pratique et efficace pour la gestion des tâches quotidiennes telles que la gestion des emplois du temps, la gestion des emplois des exams, la gestion des notes, la communication entre prof et étudiants. L'application mobile offre une plateforme conviviale, intuitive et accessible à tout moment, permettant ainsi une meilleure organisation et une gestion optimale de l'école.

Ce rapport présente donc les différentes étapes de la réalisation de ce projet, allant de la conception jusqu'à la mise en œuvre de l'application. Il expose les choix technologiques effectués, les difficultés rencontrées ainsi que les solutions apportées pour les surmonter. Enfin, il présente les résultats obtenus, les tests effectués et les perspectives d'avenir pour l'application.

Ce projet a été réalisé avec rigueur, sérieux et engagement. Il nous a permis de mettre en pratique nos connaissances théoriques acquises durant notre formation et de développer de nouvelles compétences en matière de conception et développement d'applications mobiles.

II. Chapitre 1 : Cadre général du projet

1. Introduction :

Ce chapitre présente le projet, ainsi que le démarche à suivre.

2. Contexte et la présentation de projet :

a) Présentaion du projet :

Le projet sur lequel nous travaillons est une application mobile pour la gestion des étudiants, qui vise à améliorer la communication et la gestion des étudiants en permettant aux professeurs de publier des annonces, aux étudiants de communiquer entre eux, et de consulter leur emploi du temps ainsi que les dates des examens et leurs résultats. Nous avons choisi de développer cette application pour répondre aux besoins des étudiants et des enseignants qui souhaitent avoir un outil pratique et facile à utiliser pour gérer leurs activités académiques.

b) Le but à atteindre :

Le but de notre projet est de créer et développer une application mobile pour la gestion des étudiants, qui permettra aux professeurs de faire des annonces, aux étudiants de communiquer entre eux, et de consulter leur emploi du temps ainsi que les dates des examens et leurs résultats.

c) Travail demandé :

Notre tâche permet de :

- ❖ Apprendre la plateforme de développement mobile.
- ❖ Réaliser l'interface graphique pour l'application.
- ❖ Concevoir et développer une base de données et l'intégrer dans l'application
- ❖ Établir la connexion entre l'application et la base de données.
- ❖ Développer une application fonctionnant sur un Smartphone.

3. Conclusion :

Dans ce chapitre, nous avons mis notre projet dans son contexte général. Nous avons présenté le cadre général de notre projet, une brève présentation, le but à atteindre et le travail demandé en finissant avec l'environnement de développement et la planification temporelle.

III. Chapitre 2 : Étude conceptuelle

1. Introduction :

Penser avant d'agir, faire des plans avant de construire, concevoir d'abord, développer ensuite c'est la démarche qui doit être suivie lors du développement d'une application et pour réussir n'importe quel projet.

En effet, La conception d'un système informatique est une étape très importante qui va influencer la qualité et la fiabilité de toute application.

D'abord, nous allons commencer ce chapitre par la partie de conception détaillée où nous présentons l'architecture globale de l'application. Enfin nous citons les différents diagrammes de cas d'utilisation, de séquences et le diagramme de classes.

2. Langage UML :

a) Présentation du langage UML :

UML (en anglais Unified Modeling Language, « langage de modélisation unifié ») est un langage graphique de modélisation des données et des traitements. C'est une formalisation nonpropriétaire de la modélisation objet utilisée en génie logiciel. UML spécifie plusieurs objectifs qui font un outil exact de communication :

- ❖ Comprendre et décrire les besoins.
- ❖ Spécifier un système.
- ❖ Établir l'architecture logicielle.



Figure 1: Logo de UML

b) Intérêt de la modélisation :

L'utilisation de la modélisation conceptuelle dans le développement des systèmes d'information permet de prendre en compte les besoins des applications d'une façon plus adéquats et de présenter d'une manière abstraite certains aspects des systèmes physiques et humains.

c) Les avantages d'UML :

- ❖ UML est un langage formel et standardisé.
 - ◆ Gain de précision.

- ◆ Motivation à l'utilisation d'outils.
- ◆ Gagne de stabilité et de fixité.
- ❖ UML est un support de communication adéquat et compétitif.
 - ◆ Il éclaire et facilite la compréhension de représentation abstraite complexe.
 - ◆ Son caractère plurivalent et sa souplesse en font un langage universel.
 - ◆ UML a pour objectif de spécifier, édifier et documenter les systèmes à base de logiciel.
 - ◆ UML n'est pas une méthode mais une notation qui laisse la liberté de la conception.
 - ◆ UML est un langage qui permet de modéliser tous les types de systèmes informatiques mais, qui nécessite toutefois une méthodologie de conception.

UML normalise les concepts objet, sa notation graphique permet d'exprimer une solution objet, ce qui simplifie la comparaison et l'appréciation des solutions. UML cadre l'analyse objet, il permet non seulement de représenter les concepts objets, mais il sous-entend une démarche d'analyse qui permet de reproduire une solution objet de manière itérative, grâce aux diagrammes, qui supportent l'abstraction.

Un diagramme UML est une représentation graphique, et à chaque vue correspondent des diagrammes qui sont répartis selon leurs aspects statiques ou dynamiques :

- ❖ Statique :
 - ◆ Cas d'utilisation
 - ◆ Classes
 - ◆ Composants
 - ◆ Objets
 - ◆ Déploiement
- ❖ Dynamique (comportementaux) :
 - ◆ Séquences
 - ◆ Activité
 - ◆ État-transition
 - ◆ Collaboration
- ❖ Fonctionnel :

- ◆ Cas d'utilisation
- ◆ Collaboration
- Ces diagrammes ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utilisables sont les diagrammes d'activités, de classes, de cas d'utilisation, d'objets, d'états transitions et de séquence. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'étude à qu'ils permettent de formaliser les contraintes de la réalisation et les solutions.

3. Conception avec UML :

a) Outil de modélisation :

StarUML est un logiciel de modélisation UML (Unified Modeling Language) qui permet de concevoir des diagrammes pour représenter des systèmes logiciels et des processus métier. Il permet également de créer des diagrammes d'activités, de séquences, de classes, de cas d'utilisation, de composants et d'autres types de diagrammes UML.



Figure 2: Logo de StarUML

b) Modélisation avec les diagrammes cas d'utilisation :

Le diagramme de cas d'utilisation permet de déterminer les possibilités d'interférence entre le système et les acteurs, c'est-à-dire déterminer toutes les fonctionnalités que doit fournir le système. Il permet aussi de délimiter ce dernier.

- ❖ Chaque usage effectué par les acteurs est représenté par un cas d'utilisation.
- ❖ Chaque cas d'utilisation symbolise une fonctionnalité qui leur est offerte afin d'engendrer le résultat attendu.
- ❖ Le diagramme de cas d'utilisation décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur.

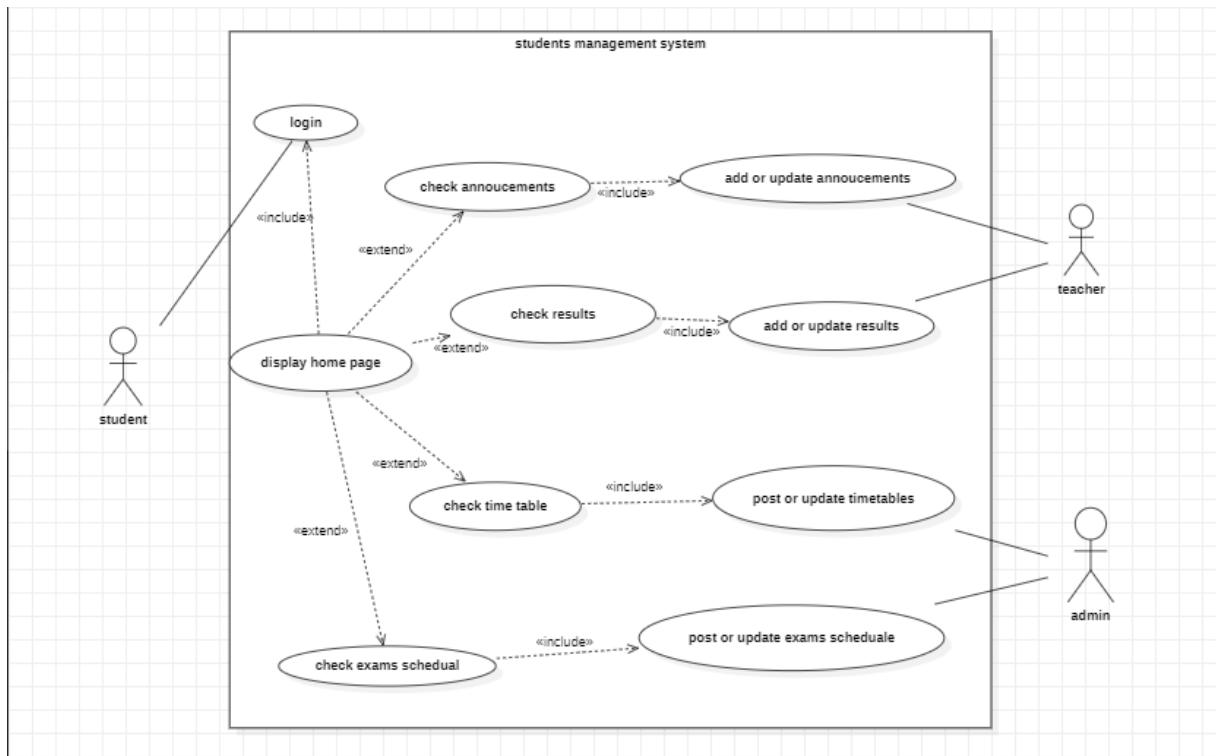


Figure 3: diagramme de cas d'utilisation

La figure 3 représente le diagramme de cas d'utilisation global de notre projet dont les acteurs sont : le professeur et l'étudiant.

c) Le diagramme de Classe :

i. Définition :

Un diagramme de classe est un diagramme UML qui contient des classes, des interfaces, des packages et leurs relations, et qui fournit une vue logique de tout ou partie d'un système informatique.

On construit un diagramme de classes pour simplifier l'interaction des objets d'un système qu'on est en train de modéliser. Ces diagrammes expriment la structure statique d'un système en termes de classes et de relations entre eux. Une classe décrit un ensemble d'objets et une association décrit un ensemble de liens. Un diagramme de classe n'exprime rien de spécifique concernant les liens d'un objet particulier, mais il décrit, le lien potentiel entre un objet et d'autres objets.

La figure 4 décrit le diagramme de classes de notre application :

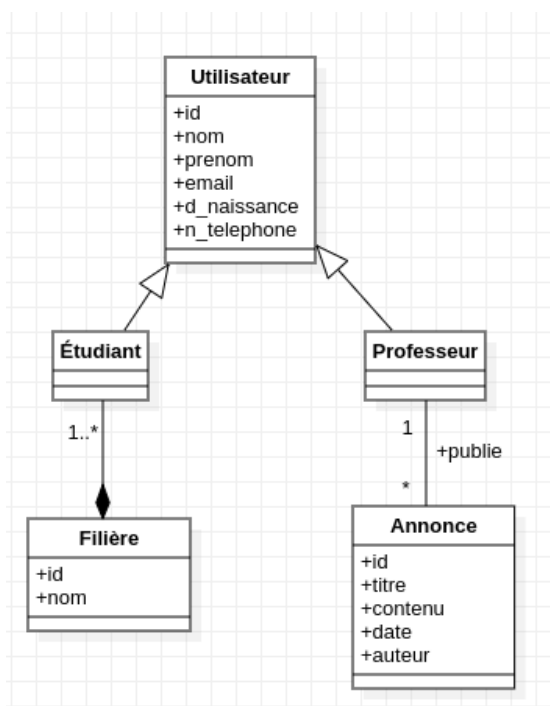


Figure 4: Diagramme de classe

ii. Présentation des classes :

Notre application comporte les classes suivantes :

- ❖ User: Elle contient des informations relatives au utilisateur.
- ❖ Teacher: Elle contient des informations relatives au professeur.
- ❖ Student: Elle contient des informations relatives au l'étudiant.
- ❖ Annonce: Elle contient le numéro, le titre, le contenu et la date de l'annonce.
- ❖ Filière : Elle contient le nom et l'id des filières.

iii. Les classes, les attributs :

Classe: Utilisateur		
Attributs		
Nom	Type	Description
id	String	Identifiant de chaque utilisateur
nom	String	Nom de l'utilisateur

prenom	String	Prenom de l'utilisateur
email	String	L'adresses e-mail de l'utilisateur
d_naissance	Date	Date de naissance de l'utilisateur
n_telephone	int	Numéro de téléphone de l'utilisateur

Table 1: Classe User

**« id » représente la clé primaire de la table Utilisateur

Classe : Filiere		
Attributs		
Nom	Type	Description
id	String	Identifiant de chaque filière
nom	String	Nom de filière

Table 2: Classe Filière

**« id » représente la clé primaire de la table Filière

Classe : Annonce		
Attributs		
Nom	Type	Description
id	String	Identifiant de chaque annonce
titre	String	Titre de l'Annonce
contenu	String	Contenu de l'Annonce
date	Date	Date de l'Annonce
auteur	String	Auteur de l'Annonce

Table 3: Classe Annonce

**« id » représente la clé primaire de la table Annonce

4. Conclusion :

Ce chapitre a été consacré à la modélisation UML du système .Nous avons présenté différents diagrammes : le diagramme de cas d'utilisation, de classe. Dans le chapitre suivant, nous présenterons la réalisation et la mise à l'essai de ce système.

IV. Chapitre 3 : La réalisation

1. Introduction :

De nos jours, chaque technologie évolue pour atteindre le but de perfection comme le développement mobile. En effet, la réussite de ce projet dépend de choix de langage, framework, bon design et architecture.

2. Environnement matériel :

Nous mentionnons les caractéristiques de nos ordinateurs sur lesquelles nous avons développé l'application parce qu'elles peuvent donner une idée sur les conditions du travail.

Donc l'application a été développée sur deux ordinateurs portables dont les caractéristiques principales sont :

❖ 1^{ère} ordinateur

- ◆ Nom : HP EliteBook x360 1030 G2
- ◆ Processeur : Intel® Core™ i5-7300U CPU @ 2.60GHz
- ◆ RAM : 8GB
- ◆ Type de système : Système d'exploitation 64bits

❖ 2^{ème} ordinateur

- ◆ Nom : ThinkPad X250
- ◆ Processeur : Intel® Core™ i5-5300U CPU @ 2.30GHz
- ◆ RAM : 8GB
- ◆ Type de système : Système d'exploitation 64bits

3. L'environnement logicielle :

Nous avons énuméré au cours de cette partie les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

a) Système d'exploitation :

Nous avons utilisé comme système d'exploitation :

- Microsoft Windows 10 Professionnel (64 bits)
- Ubuntu 22.04.2 LTS x86_64

b) Choix de langage de programmation :

La langage de programmation utilisé va beaucoup influencer sur le projet et la manière dont celui-ci sera développé, en fonction des avantages et des inconvénients du langage.

Il est important de bien étudier la language, pour éviter de devoir changer de langage en cours de projet, ce qui constituerai une perte de temps considérable.

Le choix du langage s'est finalement porté sur **JavaScript**.



Figure 5: Logo de JavaScript

JavaScript est un langage de programmation de haut niveau, souvent utilisé pour créer des applications web interactives et dynamiques. Il a été développé à l'origine par Netscape en 1995 et est maintenant largement utilisé pour la création de sites web, d'applications web, de jeux vidéo, d'applications mobiles et de nombreuses autres applications.

JavaScript est un langage orienté objet, ce qui signifie qu'il utilise des objets pour représenter des données et des fonctionnalités. Il est également un langage interprété, ce qui signifie que le code JavaScript est exécuté directement par le navigateur ou l'interpréteur JavaScript sans nécessiter de compilation préalable.

En effet j'ai choisi cette language pour plusieurs avantages :

- Facilité d'utilisation : JavaScript est un langage relativement facile à apprendre et à utiliser. Il est conçu pour être facilement compréhensible par les développeurs débutants et peut être utilisé pour créer des applications web simples ou complexes.
- Large communauté de développeurs : JavaScript est un langage populaire avec une large communauté de développeurs et une abondance de ressources en ligne pour l'apprentissage et la résolution de problèmes.
- Rapidité d'exécution : Les navigateurs modernes ont des moteurs JavaScript très performants qui permettent une exécution rapide des scripts.

c) Choix de framework :

Pour que l'application soit robuste, facile à faire évoluer et réalisable en un temps minimum, un Framework représente un outil idéal.

Il existe une grande quantité de Framework. Chacun présentant des avantages et des inconvénients.

C'est pour cela j'ai choisi **React Native** pour ces grands avantages par rapport à autres Framework :

- Développement rapide et efficace : React Native permet de développer des applications mobiles pour les deux principales plateformes (iOS et Android) en

utilisant un seul code base, ce qui permet de réduire considérablement le temps et les coûts de développement.

- Performances élevées : Les applications mobiles créées avec React Native ont des performances élevées et une grande réactivité grâce à l'utilisation de la technologie de rendu en temps réel.
- Expérience utilisateur fluide : React Native permet de créer des interfaces utilisateur fluides et réactives qui ressemblent à des applications natives, ce qui améliore l'expérience utilisateur.
- Code réutilisable : Les composants React peuvent être réutilisés à travers différents projets, ce qui permet de gagner du temps et de réduire les coûts de développement.

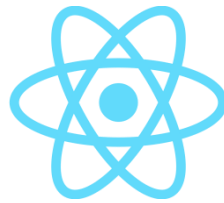


Figure 6: Logo de React Native

React Native est un framework open-source développé par Facebook qui permet de créer des applications mobiles pour les plateformes iOS et Android en utilisant le langage de programmation JavaScript. Il a été créé pour permettre aux développeurs de créer des applications mobiles performantes et réactives avec une expérience utilisateur de qualité tout en utilisant un code JavaScript partagé entre les deux plateformes.

React Native utilise la même syntaxe de programmation que React, une bibliothèque JavaScript populaire pour la création d'interfaces utilisateur web. Il permet aux développeurs de créer des interfaces utilisateur pour les applications mobiles en utilisant des composants préfabriqués, tels que des boutons, des menus déroulants, des barres de navigation, etc. Ces composants sont conçus pour être réutilisables et modulaires, ce qui permet aux développeurs de gagner du temps et de l'efficacité lors du développement de nouvelles fonctionnalités.

d) Choix de L'IDE :

Un IDE complet est un éditeur unique dans lequel vous pouvez écrire, compiler, lancer, déboguer et publier l'application.

Visual Studio Code est un éditeur de code source développé par Microsoft. Il est disponible sur plusieurs plateformes, notamment Windows, macOS et Linux. Il offre une gamme de fonctionnalités pour faciliter le développement de logiciels, telles que la coloration syntaxique, l'autocomplétion, la détection d'erreurs, le débogage, l'intégration avec des systèmes de contrôle de version tels que Git, et bien plus encore.

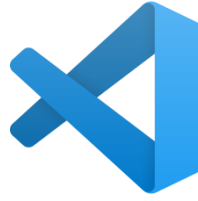


Figure 7: Logo de Visual Studio Code

e) Firebase :

Firebase est une plateforme de développement d'applications mobiles et web développée par Google. Elle offre un large éventail de services et d'outils pour aider les développeurs à créer des applications rapidement et facilement.

Parmi les services offerts par Firebase, on peut citer :

- ❖ L'authentification des utilisateurs, qui permet aux développeurs d'ajouter des fonctionnalités de connexion sécurisées à leurs applications.
- ❖ La base de données en temps réel, qui permet aux développeurs de stocker et de synchroniser des données entre les clients et les serveurs en temps réel.
- ❖ Le stockage Cloud, qui permet aux développeurs de stocker des fichiers tels que des images, des vidéos, des documents, etc.
- ❖ Les notifications push, qui permettent aux développeurs d'envoyer des notifications à leurs utilisateurs en temps réel.
- ❖ L'analyse des utilisateurs, qui fournit des informations sur les performances de l'application, telles que le nombre de téléchargements, le temps d'utilisation, les taux de conversion, etc.
- ❖ Les tests de qualité, qui permettent aux développeurs de tester leurs applications sur différents appareils et configurations.

Firebase est une solution cloud complète pour les développeurs, et est largement utilisé pour la création d'applications mobiles et web à grande échelle.



Figure 8: Logo de Firebase

f) Contrôle de version :

i. Git :

Git est un système de contrôle de version distribué, utilisé pour suivre les modifications apportées à un projet informatique au fil du temps. Il permet à plusieurs personnes de travailler simultanément sur le même projet en conservant toutes les versions précédentes des fichiers, ce qui facilite la collaboration et la gestion des changements. Git offre également des fonctionnalités pour fusionner différentes versions d'un fichier, gérer les conflits de fusion et suivre les modifications apportées par chaque personne au projet. Il est largement utilisé dans l'industrie du logiciel et est considéré comme une compétence essentielle pour les développeurs de logiciels.



Figure 9: Logo de Git

ii. Github :

GitHub est une plateforme en ligne de gestion de développement de logiciels et de contrôle de version basée sur Git, un système de contrôle de version distribué. Il permet aux développeurs de collaborer sur des projets en ligne, de partager du code source et de travailler ensemble sur des projets de logiciels open source ou privés.

GitHub est également utilisé pour le suivi des problèmes, la documentation, le déploiement de projets et l'intégration continue. Il est largement utilisé dans la communauté de développement de logiciels et est considéré comme l'un des principaux outils de développement collaboratif.



Figure 10: Logo de GitHub

g) Packages:

Nous avons utilisé npm (Node Package Manager), un gestionnaire de paquets open-source pour le langage de programmation JavaScript, afin de faciliter le développement de notre projet React Native. Grâce à ce registre de packages, nous avons pu installer, gérer et partager des packages et leurs dépendances en toute simplicité. L'utilisation de

npm nous a permis d'installer et de mettre à jour aisément des packages tiers, ce qui a grandement facilité le processus de développement et évité les erreurs de compatibilité et les conflits de versions.



Figure 9: Logo de npm

Parmi les packages utilisés, on peut citer :

- ❖ **Datetimepicker:** Un composant de calendrier qui permet aux utilisateurs de sélectionner une date et une heure dans une application React Native.
- ❖ **material-bottom-tabs:** Un package de navigation pour React Native qui permet de créer des onglets de navigation en bas de l'application.
- ❖ **react-navigation:** Un package de navigation pour React Native qui permet de naviguer entre les écrans de l'application.
- ❖ **native-stack:** Un package de navigation pour React Native qui permet de créer une pile de navigation pour les écrans.
- ❖ **expo:** Une plateforme de développement d'applications mobiles qui permet de créer des applications multiplateformes en utilisant React Native.
- ❖ **expo-image-picker :** Un package pour Expo qui permet de sélectionner une image utilisant l'appareil photo.
- ❖ **expo-status-bar :** Un package pour Expo qui permet de personnaliser la barre de statut en haut de l'application.
- ❖ **react-native-calendars:** Un package qui permet d'afficher un calendrier dans une application React Native.
- ❖ **react-native-datefield:** Un package qui permet de sélectionner une date dans une application React Native.
- ❖ **react-native-elements:** Une bibliothèque de composants React Native qui permet de créer des interfaces utilisateur élégantes et personnalisées.
- ❖ **react-native-gifted-chat:** Un package qui permet de créer une interface de chat dans une application React Native.
- ❖ **react-native-keyboard-aware-scroll-view:** Un package qui permet de faire défiler le contenu de l'application en prenant en compte le clavier virtuel.
- ❖ **react-native-modal-datetime-picker:** Un package qui permet de sélectionner une date et une heure dans une application React Native.
- ❖ **react-native-paper:** Une bibliothèque de composants React Native qui permet de créer des interfaces utilisateur Material Design.
- ❖ **react-native-reanimated:** Un package qui permet de créer des animations fluides dans une application React Native.
- ❖ **react-native-safe-area-context:** Un package qui permet de gérer les zones de sécurité de l'écran dans une application React Native.

Méthode d'installation des packages :

Pour installer un ou plusieurs packages en même temps, il suffit d'utiliser la commande suivante :

```
'npm i "nom-du-package"'
```

h) Outils utilise:

Nous avons utilisé Expo, qui est une collection d'outils open source permettant de développer des applications mobiles pour iOS, Android et le web. Grâce à cette plateforme, nous avons pu bénéficier d'une bibliothèque de composants préconstruits pour accélérer le développement de notre application. Nous avons également pu écrire notre code en JavaScript et l'utiliser pour créer une application native pour iOS et Android. Expo nous a fourni des outils de test ainsi que des fonctionnalités supplémentaires pour faciliter la création de notre application.



Figure 10 : Logo Expo dev

4. Choix de développement :

Pour le développement, nous avons appliqué le modèle MVC. Ce paradigme divise l'IHM (Interface Homme Machine) en un modèle (modèle de données) une vue (la présentation, l'interface utilisateur) et un contrôleur (la logique de contrôle, et la gestion des événements / synchronisation), chacun a un rôle bien précis.

L'architecture MVC ne résout pas tous les problèmes. Elle fournit souvent une première approche qui peut ensuite être adaptée et elle offre aussi un cadre pour structurer une application. Ce maître d'architecture impose la séparation entre les données, la présentation et les traitements, ce qui nous donne trois parties fondamentales dans l'application : le modèle, la vue et le contrôleur.

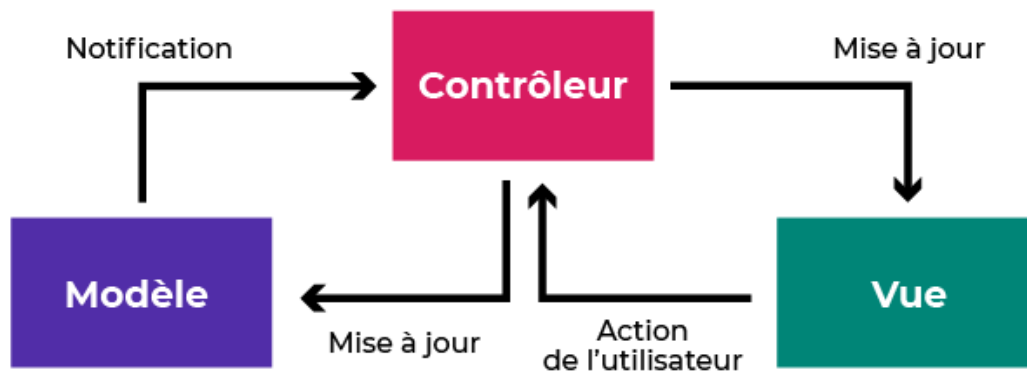


Figure 11: Schéma du Modèle MVC

a) Le modèle :

Le modèle indique le comportement de l'application : traitements des données, interactions avec la base de données. Il décrit l'emplacement des données manipulées par l'application et assure la gestion de ces données et garantit leur intégrité. Dans le cas spécifique d'une base de données, c'est le modèle qui la contient. Le modèle offre des méthodes pour mettre à jour ces données (ajout, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

b) La vue :

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (sélection d'une entrée, boutons, etc.). Ces événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se satisfait d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

c) Le contrôleur :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle, et ce dernier informe la vue que les données ont changé pour qu'elle les mette à jour.

5. Les interfaces graphiques :

L'interface graphique est une partie très importante pour la réalisation d'une application convenable offrant un certain plaisir à l'utilisateur lors de sa navigation. Ainsi, ce critère peut faire la différence entre une application et une autre bien qu'elles aient les mêmes fonctionnalités.

Voici un ensemble de captures d'écrans sur les principaux points d'entrées de l'application :

i) Interface Authentification :

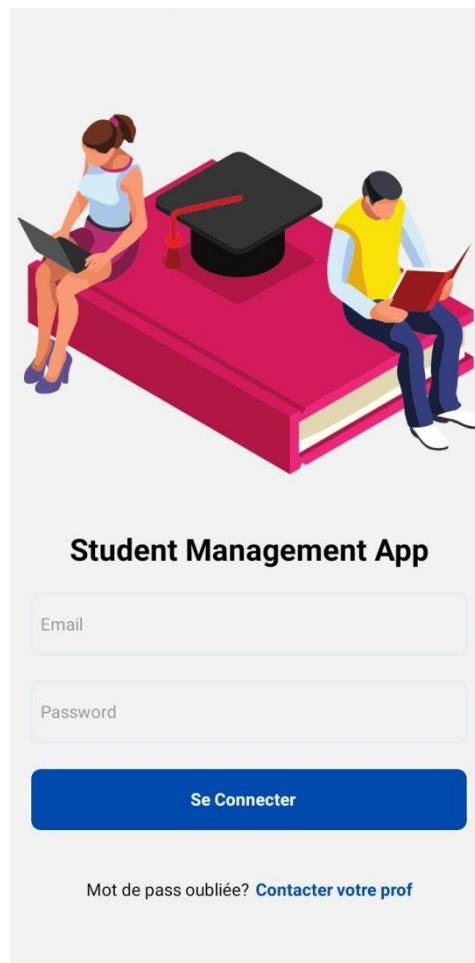


Figure 12: Interface Authentification

La figure 12 représente l'interface d'authentification, cette interface demande aux utilisateurs de saisir leur adresse e-mail et leur mot de passe afin de vérifier leur identité avant de leur permettre d'accéder à l'application. Cette étape d'authentification est importante pour assurer la sécurité et la confidentialité des données de l'application.

j) Interface Accueil :

i. Partie Étudiant :

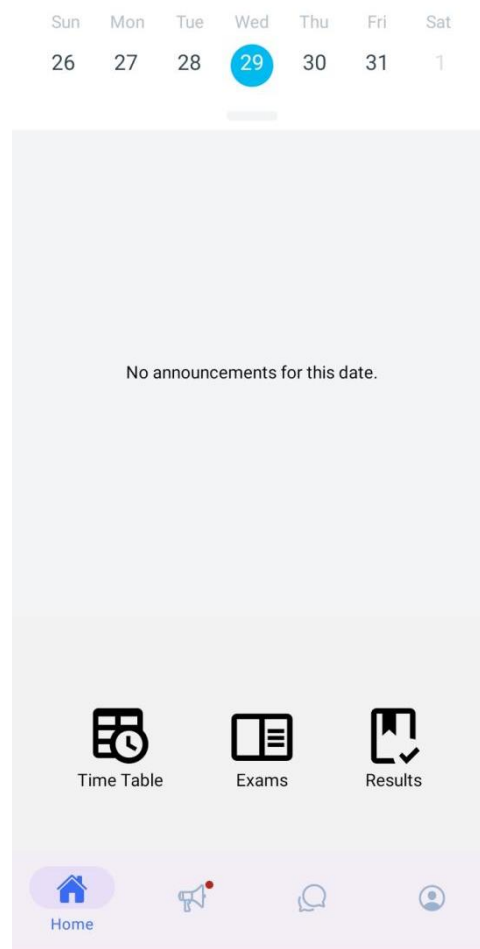


Figure 13: Interface Accueil Partie Étudiant

Si l'utilisateur a saisi l'adresse e-mail et le mot de passe et que le système détecte qu'il est un étudiant, la figure 13 s'affiche, elle contient 4 sous-interfaces différentes, chacune ayant une fonctionnalité spécifique. Ces sous-interfaces sont l'interface Home, l'interface Liste des annonces, l'interface de communication entre les étudiants et l'interface Profil de l'utilisateur. En utilisant ces différentes sous-interfaces, les étudiants peuvent accéder à toutes les informations dont ils ont besoin et communiquer efficacement avec les autres étudiants.

ii. Partie Professeur :

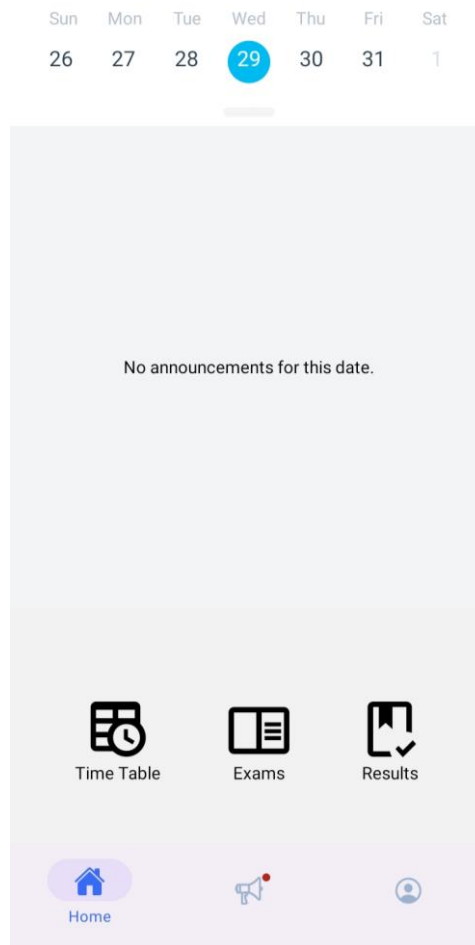
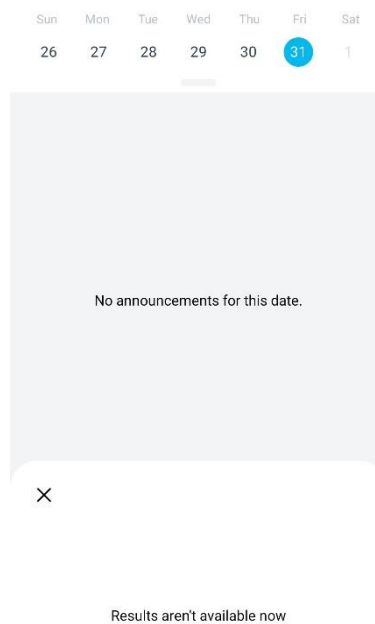
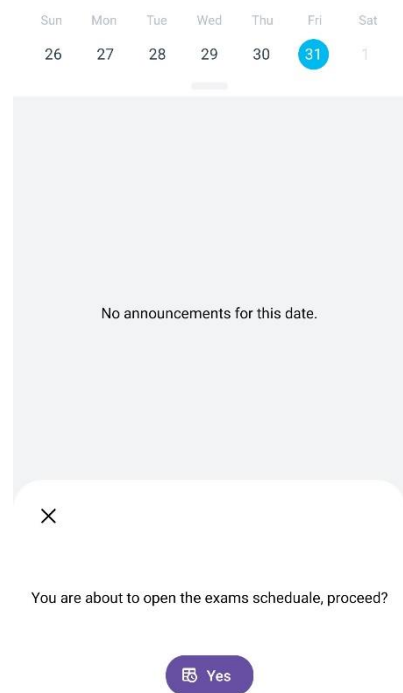
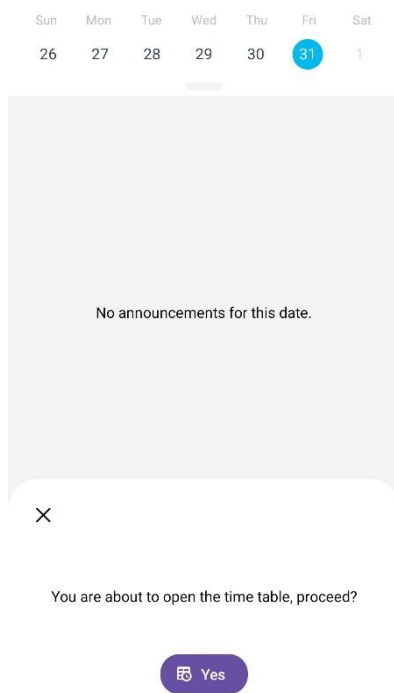


Figure 14: Interface Accueil Partie Professeur

Si l'utilisateur a saisi l'adresse e-mail et le mot de passe et que le système détecte qu'il est un professeur, la figure 14 s'affiche, elle contient 3 sous-interfaces différentes, chacune ayant une fonctionnalité spécifique. Ces sous-interfaces sont l'interface Home, l'interface Liste des annonces et l'interface Profil de l'utilisateur.

k) Interface Home :

i. Partie étudiant



ii. Partie Professeurs

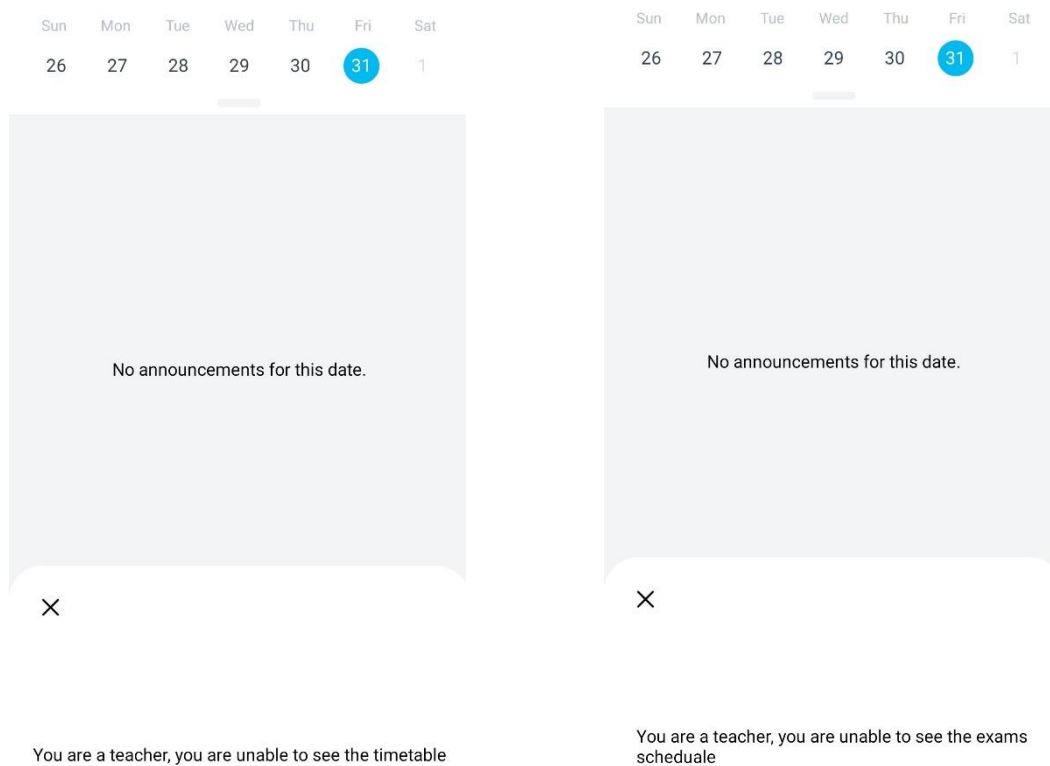


Figure 15: Interface Home

L'interface Home contenir un agenda et des boutons pour accéder à différentes fonctionnalités, notamment l'emploi du temps, les examens et les résultats.

I) Interface Liste des annonces :

Partie Étudiant :

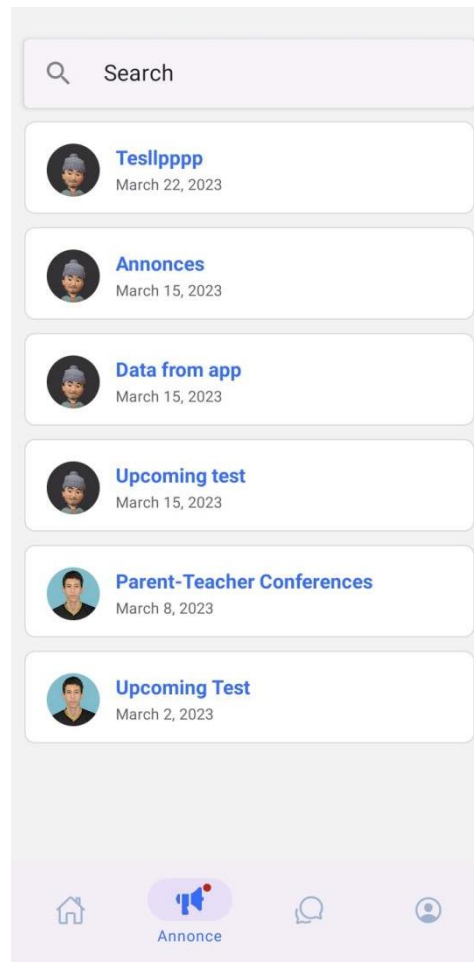


Figure 16: Interface Liste des annonces Partie Étudiant

Cette interface affiche une liste de toutes les annonces disponibles. Les étudiants peuvent utiliser cette interface pour trouver rapidement les annonces qui les intéressent.

Partie Professeur :

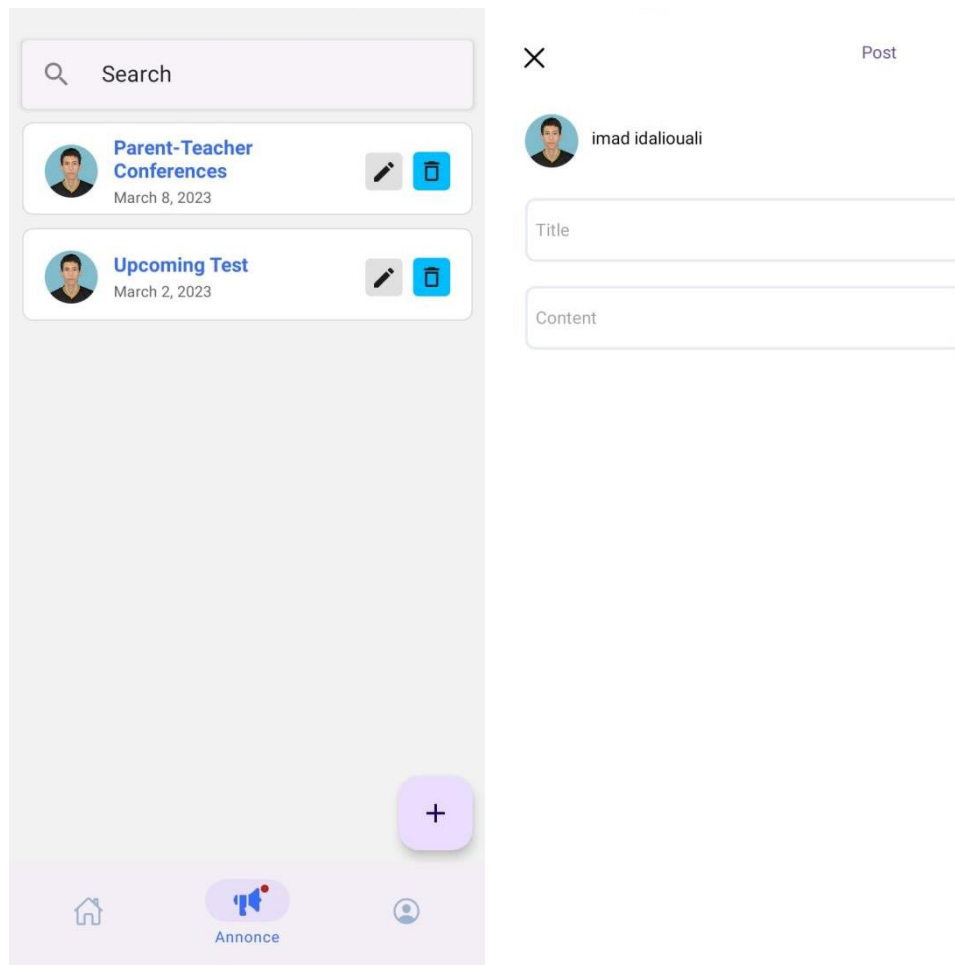


Figure 17: Interface Liste des annonces Partie Professeur

L'interface Liste des annonces peut permettre aux professeurs de publier des annonces pour leurs cours, ainsi que des mises à jour importantes ou des rappels concernant les devoirs, les tests et les autres travaux assignés. Les étudiants peuvent également consulter cette interface pour s'informer des dernières informations concernant leurs cours.

m) Interface de communication entre les étudiants :

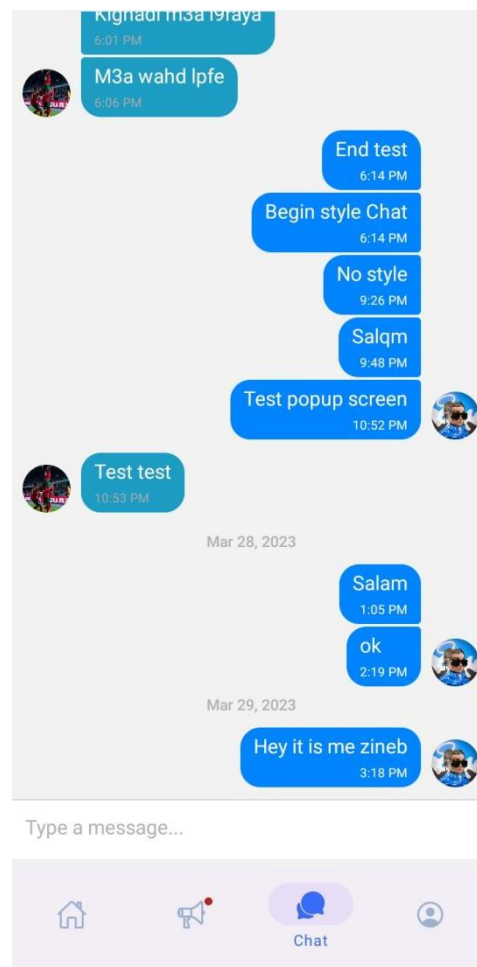


Figure 18: Interface de communication entre les étudiants

Cette interface permet aux étudiants de communiquer entre eux. Elle peut être utilisée pour poser des questions, partager des informations et discuter de divers sujets.

n) Interface Profil de l'utilisateur :

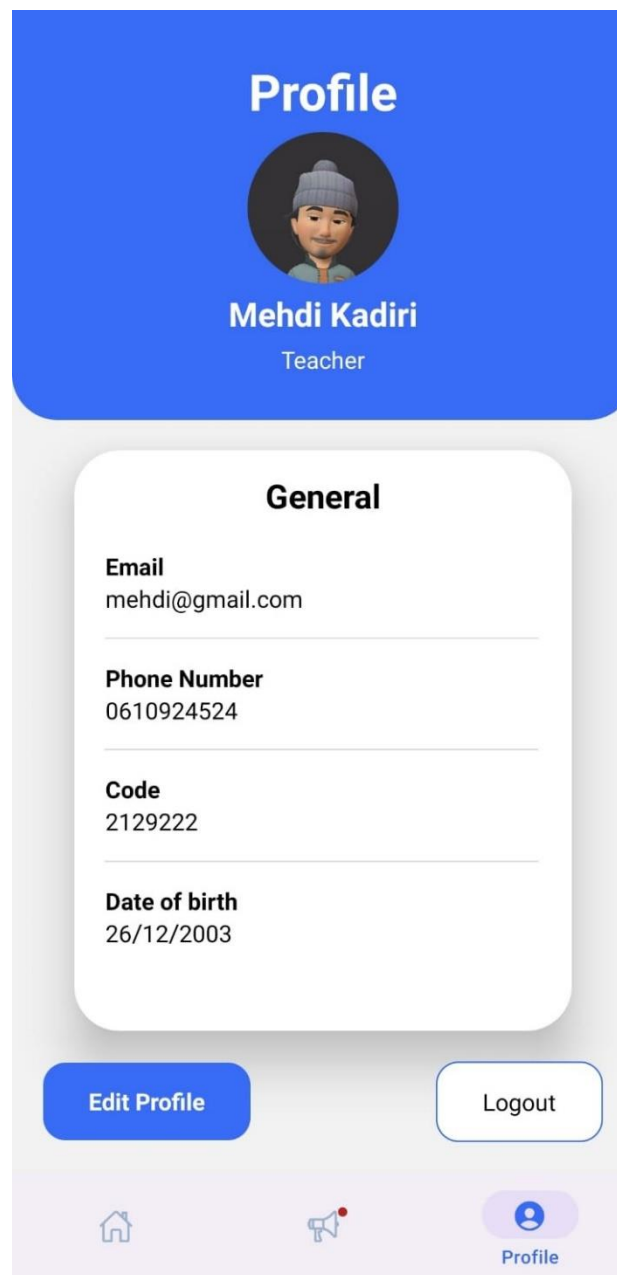


Figure 19: Interface Profil de l'utilisateur

Cette interface affiche les informations du profil de l'utilisateur.

o) Interface d'edite de profile :

Change Image

Mehdi Kadiri

0610924524

26/12/2003

Edit

Profile

Cette interface permet a l'utilisateur de changer ses information personnelle.

6. Conclusion :

La dernière partie était dédiée à la navigation dans notre application. Elle constitue le dernier volet de ce rapport, elle a pour objet de présenter l'environnement logiciel et matériel de réalisation, et de décrire les principales fonctionnalités implémentées suite au développement par des captures écrans présentant les principes interfaces de l'application.

V. Conclusion :

En conclusion, la réalisation de ce projet de fin d'études a été une expérience très enrichissante pour nous. Nous avons pu mettre en pratique les connaissances théoriques acquises durant notre formation et développer de nouvelles compétences en matière de conception et développement d'applications mobiles.

Nous avons ainsi réussi à répondre aux besoins de l'école en proposant une solution pratique et efficace pour la gestion de ses tâches quotidiennes. L'application mobile que nous avons conçue offre une plateforme conviviale, intuitive et accessible à tout moment, permettant ainsi une meilleure organisation et une gestion optimale de l'école.

Bien que nous ayons rencontré des difficultés techniques et organisationnelles au cours de la réalisation de ce projet, nous avons su les surmonter grâce à notre travail acharné, notre détermination et notre collaboration en binôme.

Nous sommes fiers du résultat final obtenu et nous espérons que cette application mobile sera bénéfique pour l'école ainsi que pour toutes les personnes impliquées dans sa gestion.

Enfin, nous tenons à remercier toutes les personnes qui nous ont soutenus et aidés tout au long de ce projet, notamment notre encadrant pédagogique, les enseignants de l'école ainsi que nos familles et amis. Nous leur sommes reconnaissants pour leur confiance et leur soutien sans faille tout au long de cette aventure.

VI. Références :

<https://reactnative.dev/>

https://fr.wikipedia.org/wiki/React_Native

<https://firebase.google.com/>

<https://fr.wikipedia.org/wiki/Firebase>

<https://staruml.io/>

<https://developer.mozilla.org/en-US/docs/Glossary/MVC>