



المندوبية السامية للتخطيط
HAUT-COMMISSARIAT AU PLAN



Institut National de
Statistique et d'Economie
Appliquée

Projet Encadré : Application de suivi parental Des performances scolaires

Groupe constitué de : - Omar Farouk Lafdil

- Anas Aaraba
- Mohammed El Alaoui
- Faïçal El Harj

Encadré par : Mr. Erraji Zakarya

Filière : DSE

Année Universitaire : 2021-2022

REMERCIEMENTS	4
RESUME	5
INTRODUCTION.....	6
BUT DU PROJET	7
1 ^{ER} CHAPITRE : RECEUIL DES BESOINS /ETUDE DU MARCHÉ :	8
INTRODUCTION :.....	9
DEFINIR LA CIBLE UTILISATEUR :	9
RASSEMBLEMENT DES BESOINS (QUESTIONNAIRE) :.....	9
CLASSIFICATION DES BESOINS :	11
2 ^{EME} CHAPITRE : VALIDATION DES BESOINS :	13
INTRODUCTION :.....	14
BESOINS FONCTIONNELS VISES :	14
BESOINS NON FONCTIONNELS :	14
3 ^{EME} CHAPITRE : CONCEPTION	16
INTRODUCTION :.....	17
L'APPROCHE UML ADOPTÉE :	17
DICTIONNAIRE DES DONNÉES :	17
DIAGRAMME DE CAS D'UTILISATIONS :	19
DIAGRAMME DE CLASSES :	20
DIAGRAMME D'ACTIVITÉS :	21
MAQUETTE :	23
4 ^{EME} CHAPITRE : TECHNIQUE DE DEVELOPPEMENT :	25
INTRODUCTION :.....	26
ENVIRONNEMENT LOGICIEL :	26
INTERFACES DE NOTRE APPLICATIONS :	28
CONCLUSION :	39
BIBLIOGRAPHIE	40

Table des Figures

Figure 1: premier snap du questionnaire	10
Figure 2: deuxième snap du questionnaire	10
Figure 3: 3ème snap du questionnaire	10
Figure 4: quatrième snap du questionnaire	11
Figure 5: cinquième snap du questionnaire	11
Figure 6: classification des besoins	12
Figure 7: Diagramme de cas d'utilisation.....	19
Figure 8 : Diagramme de classes.....	20
Figure 9: Diagramme d'activité.....	22
Figure 10 : Interface de Login de l'application.....	28
Figure 11 : Menu Principale de l'application (Tuteur)	29
Figure 12 : Profile.....	30
Figure 13 : Menu Enfant	31
Figure 14 : Devoirs	32
Figure 15 : Absences	33
Figure 16 : Liste des professeurs de l'étudiant	34
Figure 17 : Notes des examens	35
Figure 18 : Page de Messagerie (tuteur et professeur)	36
Figure 19 : Devoirs de tous les enfants sous le tutorat de l'utilisateur	37
Figure 20 : Ajout des classe dans la Base de Données	38

Remerciements

Nous tenons à remercier tout d'abord notre professeur encadrant Mr Erraji Zakarya pour son implication, sa disponibilité et son aide.

Nous tenons aussi à remercier nos familles pour leur soutien indispensable.

Nous remercions aussi nos collègues pour leur aide et leurs conseils.

Résumé

Nous avons réalisé ce projet pendant notre deuxième année en Data science and software engineering, 4 ème semestre, à l'institut National de statique et d'économie appliquée dans le cadre du module du projet encadré par notre Professeur Monsieur Erraji Zakarya.

Ce rapport donne un contexte et une explication de notre projet, qui porte sur une application mobile qui facilite le suivi parental des performances scolaires de leurs enfants.

L'Objectif de ce projet est de bien distinguer le besoin de cette idée, et son implémentation d'un coté pratique.

Introduction

Aujourd'hui, le digital est devenu une grande partie de nos vie, il nous est important de bien savoir l'utiliser pour faciliter nos tâches quotidiennes, et aussi de l'utiliser pour implémenter de nouvelles idées qui apparaissaient impossibles sans développement technologique.

Une idée qui pourra simplifier la tâche des parents dans l'éducation de leurs enfants est d'essayer de rapprocher le milieu scolaire de l'enfant au parent, sans que ce dernier se déplace vers l'établissement de son enfant. Ceci pourra être bénéfique puisque beaucoup de parents se trouvent dans des situations où ils ne peuvent pas suivre quotidiennement leurs enfants.

Cette application a comme but d'établir un fil de connexion entre l'administration de l'établissement et les tuteurs des étudiants, ceci en les notifiant des absences, des devoirs et des notes de leurs enfants.

Elle sera aussi un espace de communication entre les professeurs et les tuteurs pour discuter les situations de leurs enfants, et découvrir leurs lacunes et difficultés.

Nous allons étudier le marché ciblé et ses besoins, puis on va passer à réaliser la conception de l'application, puis finalement on passera au côté technique.

But Du Projet

Notre travail va être détaillé et partagé sur des points essentiels, qui vont nous donner la possibilité d'entourer le sujet le maximum possible jusqu'à la génération d'une application adéquate.

Notre travail va être détaillé selon des points précis :

- Collecte du besoin : Dans cette phase, nous allons faire une collecte de besoins à travers l'étude du marché ciblé. On décide ce que cette population attend d'une telle application et leurs souhaits.
- Validation des besoins fonctionnels : Il s'agit ici de finaliser les besoins fonctionnels et non fonctionnels exigés dans l'application.
- Dossier de conception : Cette étape permet d'établir un dossier de conception récapitulant tout ce qui précède (Diagramme de cas d'utilisation, diagramme de classe, etc.)
- Dossier technique : Afin de justifier l'architecture et le choix des outils technologiques de développement, Nous sommes appelée à établir un dossier technique.
- Développement : Dans Cette phase, nous allons nous mettre à développer les interfaces déjà conçu dans les étapes précédentes.

1^{er} Chapitre : Recueil des besoins /Etude du marché :



- ❖ Introduction .
- ❖ Définir la cible utilisateur.
- ❖ Rassemblement des besoins (Questionnaire).
- ❖ Classification des besoins .

Introduction :

Il est critique dans la conception et la mise en œuvre des systèmes d'information, d'élaborer et de suivre une stratégie pour collecter les besoins. La satisfaction des utilisateurs et la réussite des projets en général sont en effet largement dépendants d'une bonne définition du périmètre fonctionnel.

Définir la cible utilisateur :

Notre application est conçue pour être utilisée par les parents/tuteurs, qui savent utiliser un téléphone portable, et veulent suivre les performances scolaires de leurs enfants. Cette cible utilisateur n'a, probablement, pas assez de temps, n'est pas forcément lettrée, et s'attend à une application simple et efficace.

Rassemblement des besoins (Questionnaire) :

On a mis en place un questionnaire pour savoir les attentes du marché ciblé, la population choisie sont des parents d'âge entre 33 et 55 ans habitant dans 2 villes et un village, avec des enfants de niveau scolaire entre la première année en primaire jusqu'aux études au Baccalauréat.

La plupart de la population en question est employée (90% pourcentage estimé non exacte).

Analyse des réponses au questionnaire :

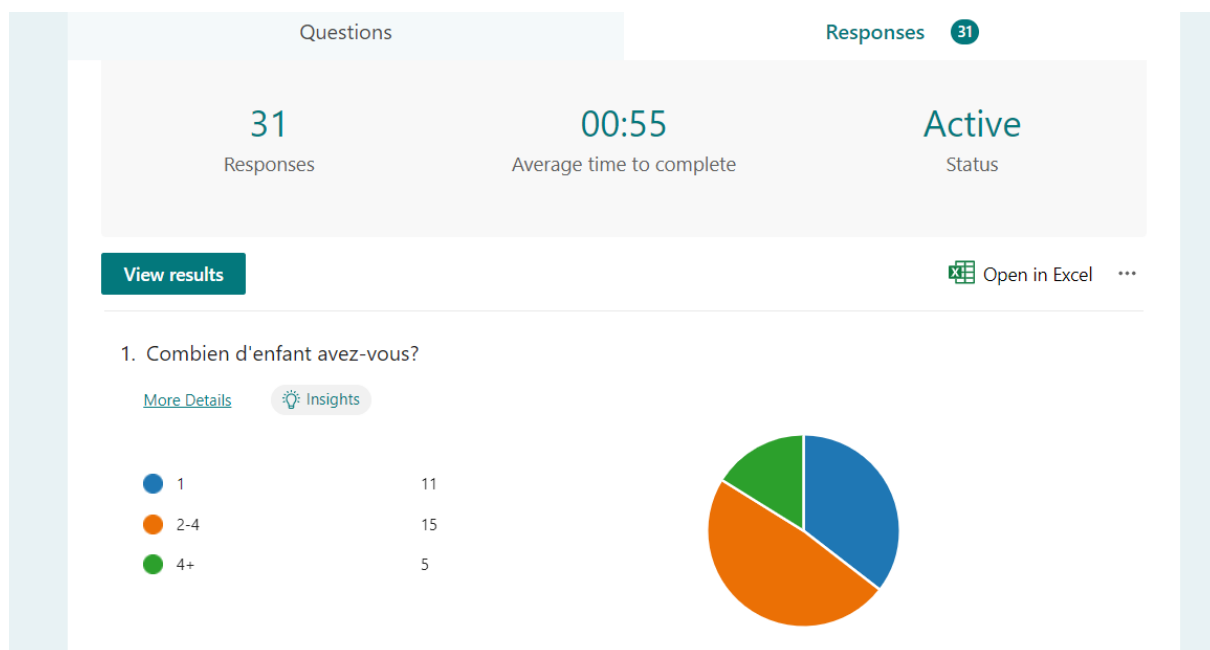


Figure 1: premier snap du questionnaire

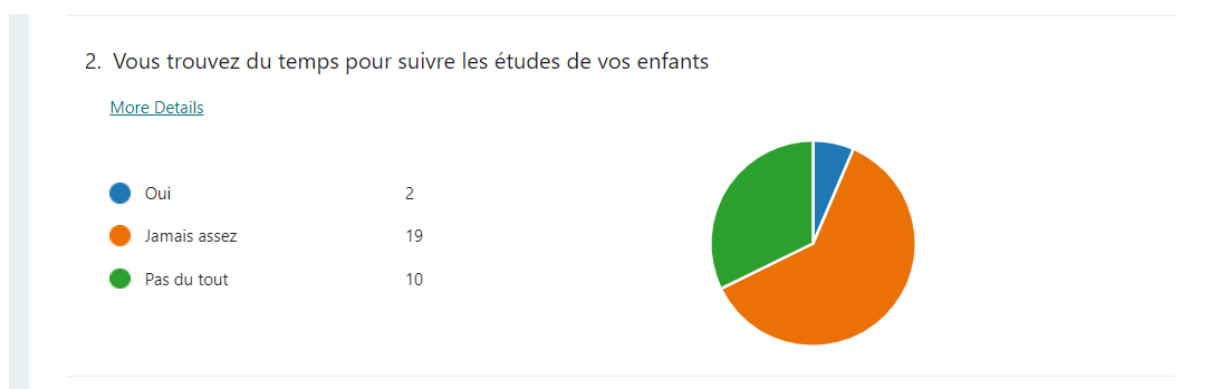


Figure 2: deuxième snap du questionnaire



Figure 3: 3ème snap du questionnaire

4. Si vous avez répondu avec non, Espérez-vous être en contact avec les enseignants et l'administration?

[More Details](#)



Figure 4: quatrième snap du questionnaire

5. Vous pensez que votre enfant est sincère avec vous quand il s'agit de sa performance scolaire

[More Details](#)

Insights

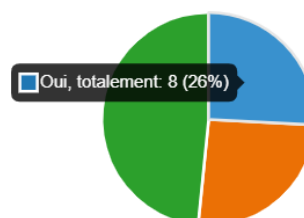
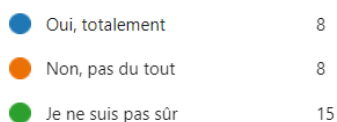


Figure 5: cinquième snap du questionnaire

En analysant les réponses au questionnaires on peut comprendre qu'une bonne partie des parents n'ont pas de temps suffisant pour accompagner leurs enfants dans leur parcours scolaire, ceci n'est certainement pas à cause du manque d'intérêt ou implication, mais à cause des contraintes (distance, travail ...) qui leurs empêchent de faire cette tâche essentielle.

Ces parents souhaitent établir une communication avec les enseignants et l'administration, pourtant n'y arrivent pas. Ils ne sont pas toujours confiant à propos de la sincérité de leurs enfants et estiment qu'ils sont peut être capable de cacher ou déformer des incidents, et pensent qu'il n'est pas hors de question que leurs enfants ne sont pas sincères à propos de leurs performances à l'école.

Classification des besoins :

On peut généraliser notre étude(modeste) et dire que la population ciblé représente des parents qui espèrent avoir un aperçu continu sur les développement du parcours scolaire de leurs enfants, Ceci inclut les notes des examens et contrôles, les devoirs et les travaux

à faire à domicile, l'assiduité en classe et le taux d'absentisme. Ils espèrent aussi avoir un contact avec leurs enseignants et l'administration de l'établissement où ils étudient. La catégorisation des besoins pourra être effectuée en divisant ces besoins en 3 catégories principaux, qui font varier la satisfaction des utilisateurs.

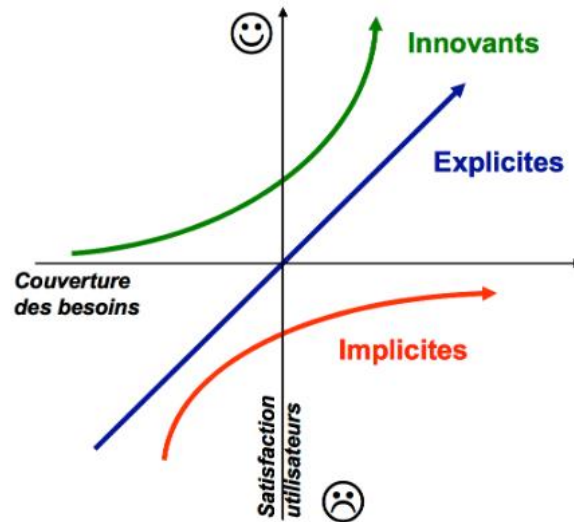


Figure 6: classification des besoins

Les besoins implicites : Des besoins qui sont tellement évidents pour l'utilisateur qu'il ne les exprime pas.

Les besoins explicites : Des besoins ouvertement exprimés par l'utilisateur. La satisfaction ou l'insatisfaction de l'utilisateur croît de façon linéaire avec la performance.

Les besoins latents ou innovants : Des besoins non exprimés par l'utilisateur car il n'en a pas conscience ou n'imagine même pas que ce besoin puisse être couvert.

On peut classer nos besoins donc en :

Besoins implicites : Aperçu sur notes d'examens et absentisme.

Besoins explicites : Aperçu sur les devoirs et les travaux à domicile.

Besoins innovants : Contact avec enseignants.

2^{ème} Chapitre : Validation des besoins :



- ❖ Introduction .
- ❖ Besoins Fonctionnels visés .
- ❖ Besoins non fonctionnels .
- ❖ Classification des besoins .

Introduction :

Cette partie est très importante au début de chaque démarche de développement. Son but est de veiller à développer une application adéquate, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : Quelles sont les fonctions du système ?

Besoins Fonctionnels visés :

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système.

En effet l'application doit répondre aux exigences suivantes :

- ❖ Le système doit pouvoir récupérer des informations de chaque étudiant dans l'école.
- ❖ La suppression des données (tuteur ou étudiant ou classe)
- ❖ Permettre à l'utilisateur de voir la liste de ses enfants et leurs informations.
- ❖ Permettre à l'utilisateur de voir la liste des professeurs de chaque enfant, et la possibilité de leur envoyer un message.
- ❖ Permettre au parent de voir la liste des devoirs et travaux à domicile des enfants, la matière d'étude et le délais d'envoi.
- ❖ Permettre au tuteur de voir les notes des examens de chaque matière .
- ❖ Permettre aux utilisateurs de voir les absences de leurs enfants et si ces absences sont justifiés ou non.

Besoins non fonctionnels :

Les besoins non fonctionnels présentent les exigences internes du système qui sont cachées pour les utilisateurs.

Parmi ces besoins nous citons :

- Les couleurs doivent être harmonieuses et expressives.
- Le contenu doit être structuré, compréhensible, cohérent et lisible.
- L'ergonomie des interfaces homme-machine (IHM) : Il s'agit d'une application Web, elle doit présenter des interfaces utilisateurs conviviales et bien structurées du point de vue contenu informationnel.
- La sécurité : Les droits d'accès au système doivent être bien attribués aux différents partenaires, afin d'assurer la sécurité des données. De plus, le système doit disposer d'un mécanisme d'authentification limitant l'accès aux utilisateurs.
- La compatibilité : L'application doit être compatible avec la majorité des mobiles android avec le maximum de version adoptés.

3^{ème} Chapitre : Conception



- ❖ Introduction .
- ❖ L'approche UML adoptée.
- ❖ Dictionnaire des données.
- ❖ Diagrammes des cas d'utilisations.
- ❖ Diagramme de classe.
- ❖ Diagramme d'activités.
- ❖ Maquette.

Introduction :

Cette partie est consacrée aux étapes fondamentales pour le développement de notre système de gestion de stock du magasin du Ministère. Pour la conception et la réalisation de notre application, nous avons choisis de modéliser en s'appuyant sur le formalisme UML qui offre une flexibilité marquante et s'exprime par l'utilisation des diagrammes.

L'approche UML adoptée :

L'UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

L'UML modélise l'ensemble des données et des traitements en élaborant des différents diagrammes.

En clair, il ne faut pas designer UML en tant que méthode mais plutôt comme une boîte d'outils qui sert à améliorer les méthodes de travail.

On adoptera une vue fonctionnel à partir des diagrammes de cas d'utilisation, de classe et d'activité.

Dictionnaire des données :

Le dictionnaire des données est un tableau qui regroupe toutes les données du système d'informations. Pour chaque donnée il faut préciser :

- ✓ Sa désignation.
- ✓ Sa longueur (en caractère).
- ✓ Attributs.
- ✓ Son type (numérique, alphabétique, alphanumérique).

Le dictionnaire sous forme du tableau suivant :

Entité	Attributs	Type	Taille
Tuteur	Nom	String	0-30
	Prenom	String	0-30
	Téléphone	Number	0-12
Etudiant	Nom	String	0-30
	Prenom	String	0-30
	Email	String	0-30
Classe	Nom	String	0-30
	Nombre d'étudiants	String	0-30
Professeur	Nom	String	0-30
	Prenom	String	0-30
	Email	String	0-60
	Matiere	String	0-30
	Numero de telephone	Number	0-12
Absence	Date absence	DATE	0-30
	matiere	String	0-30
Devoir	Type	String	0-30
	Date	DATE	0-30
Message	Contenu	Text	0-30
	Date d'envoi	DATE	0-30
	Expéditeur	String	0-30
	Destinataire	String	0-30
Note	Note	Number	0-4
	Matière	String	0-30

Diagramme de cas d'utilisations :

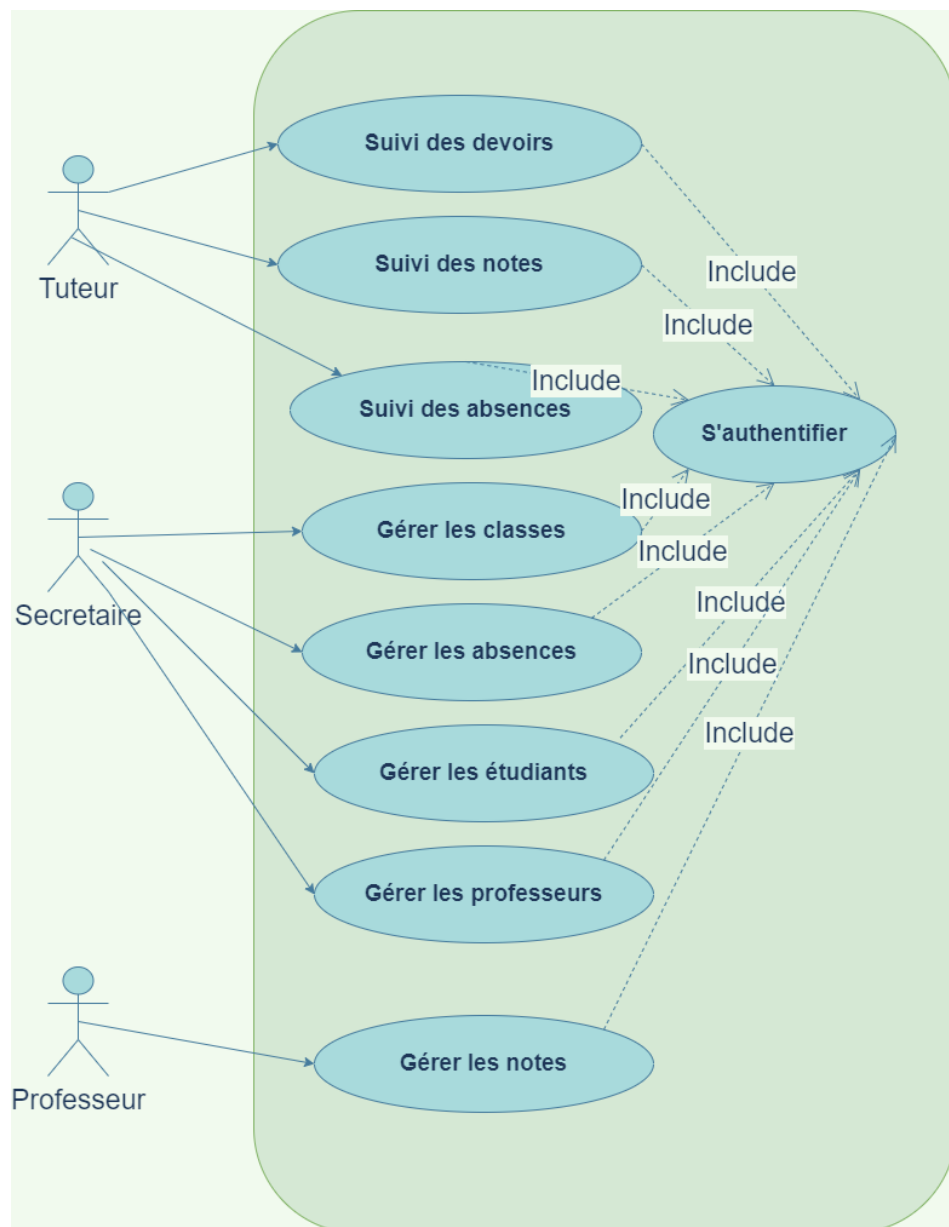


Figure 7: Diagramme de cas d'utilisation

Diagramme de classes :

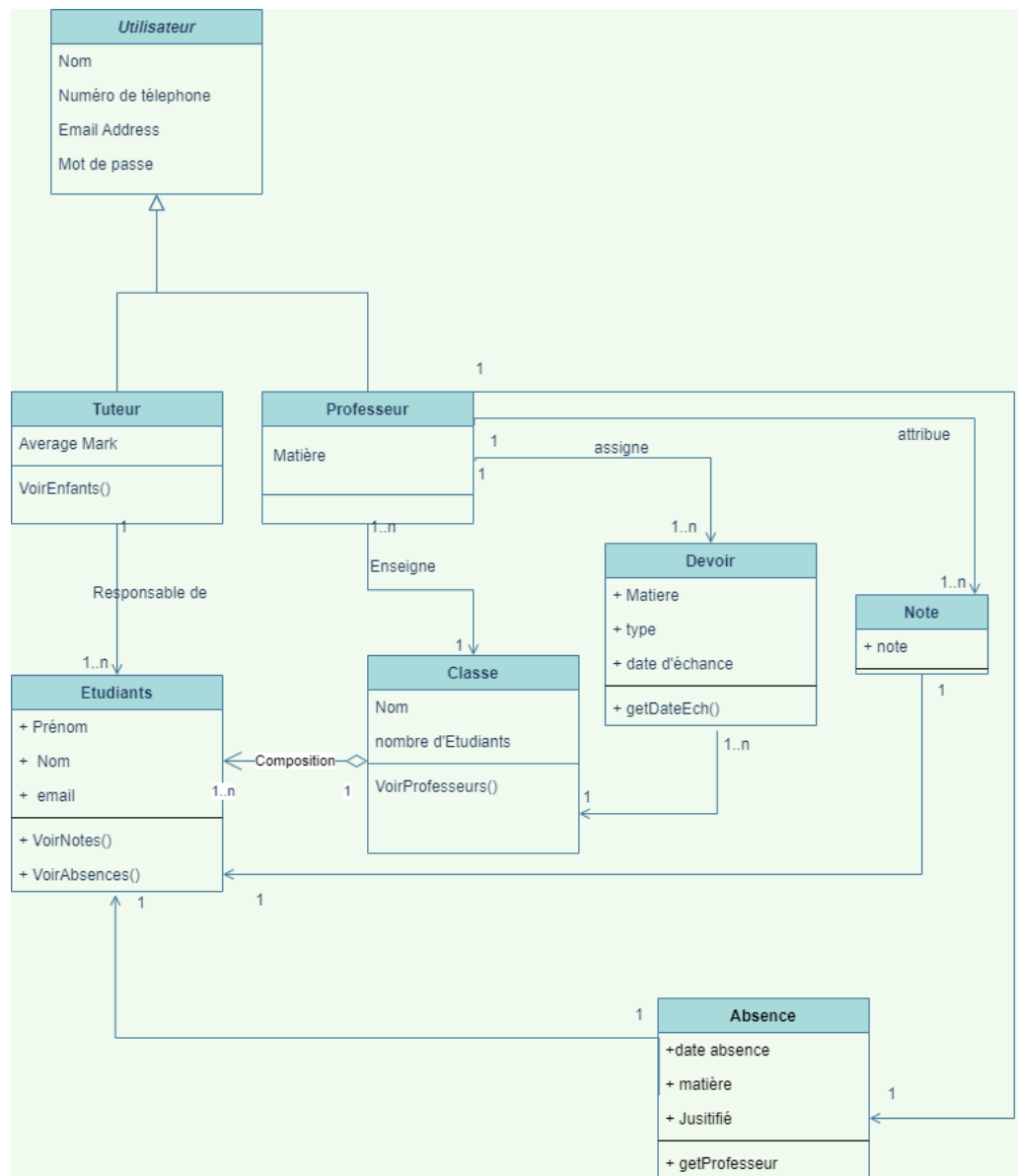


Figure 8 : Diagramme de classes

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté.

- Une classe : Représente la description abstraite d'un ensemble d'objets possédants mêmes caractéristiques. On peut parler également de type.

- Un objet : Est une entité aux frontières bien définies, possédant une identité et encapsulant un état et un comportement. Un objet est une instance (ou occurrence) d'une classe.

- Un attribut : Représente un type d'information contenu dans une classe.

- Une association : Représente une relation sémantique durable entre deux classes.

- Une superclasse : Est une classe plus générale reliée à une ou plusieurs autres classes plus spécialisées (sous-classes) par une relation de généralisation. Les sous-classes « Héritent » des propriétés de leur superclasse et peuvent comporter des propriétés spécifiques supplémentaires .

Diagramme d'activités :

Dans le langage UML, un diagramme d'activité fournit une vue du comportement d'un système en décrivant la séquence d'actions d'un processus. Les diagrammes d'activité sont similaires aux organigrammes de traitement de l'information, car ils montrent les flux entre les actions dans une activité. Les diagrammes d'activité peuvent, cependant, aussi montrer les flux parallèles simultanés et les flux de remplacement.

Dans les diagrammes d'activité, vous utilisez des noeuds d'activité et des bords d'activité pour modéliser le flux de commande et de données entre les actions.

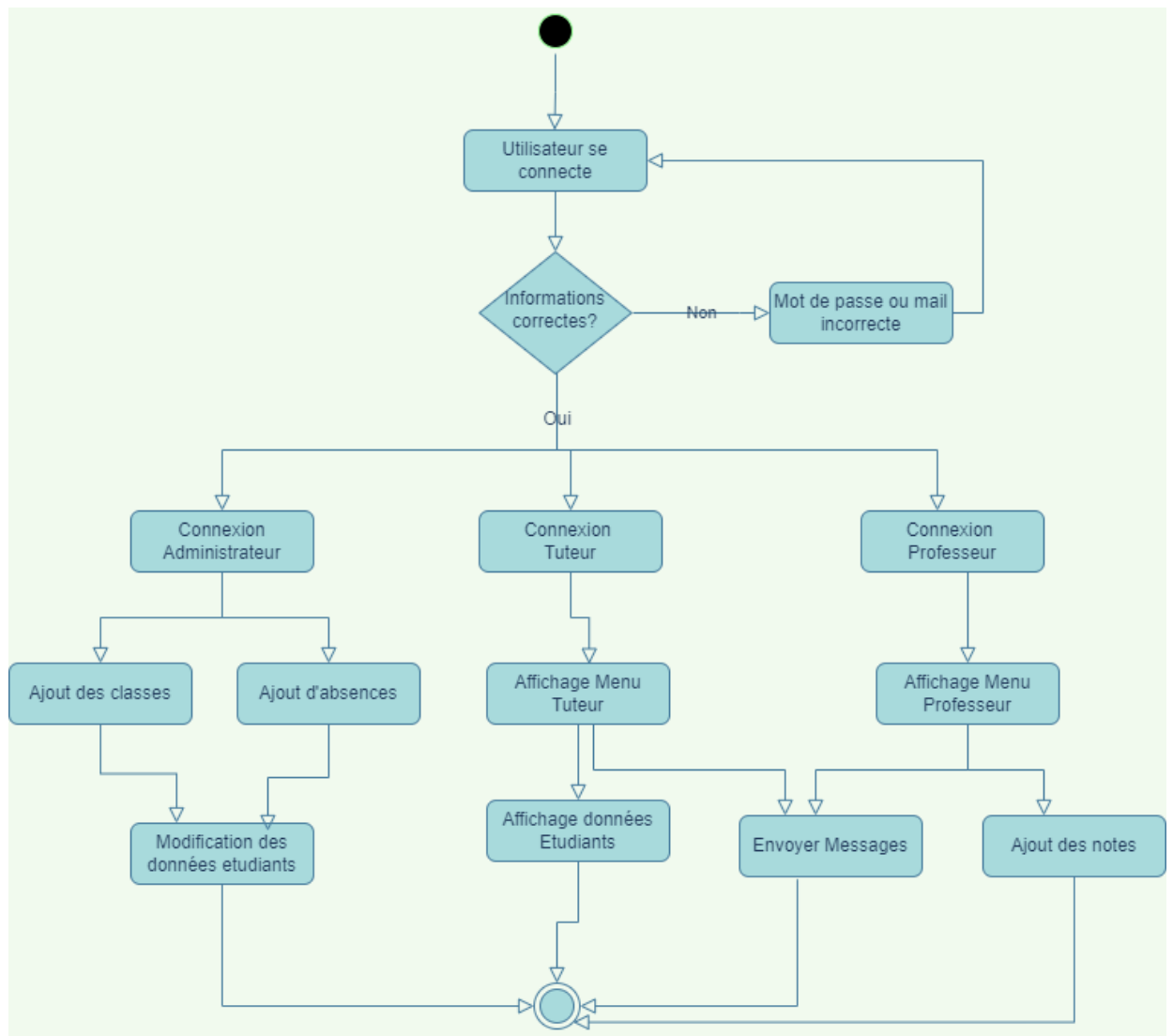
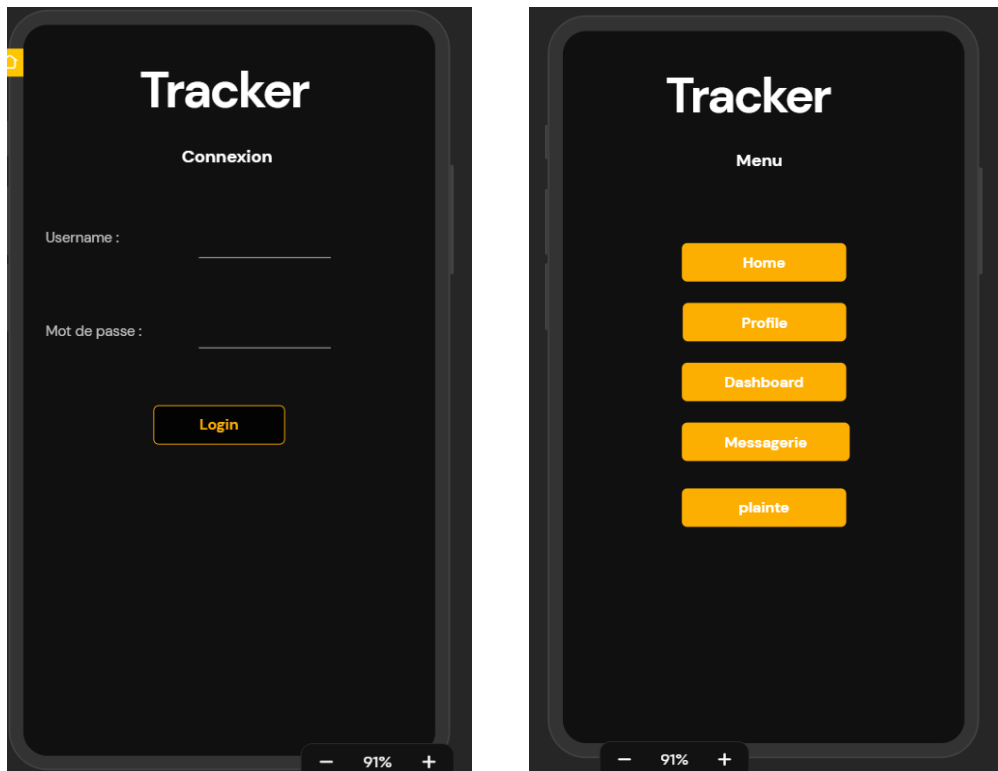
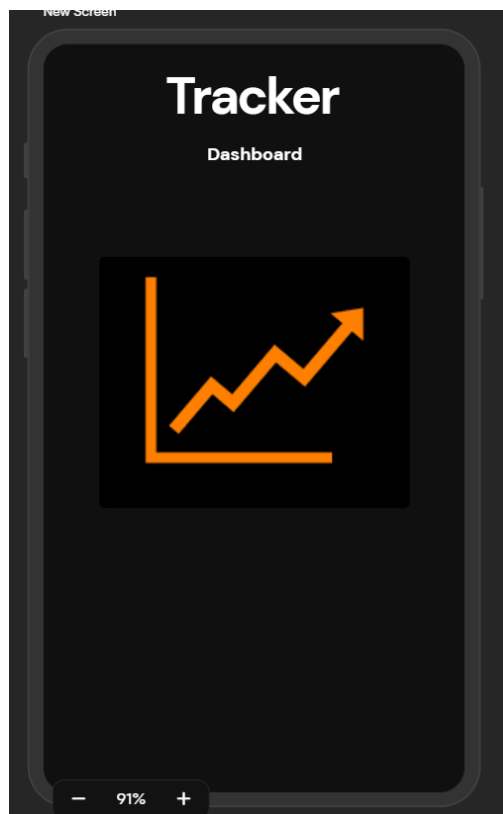
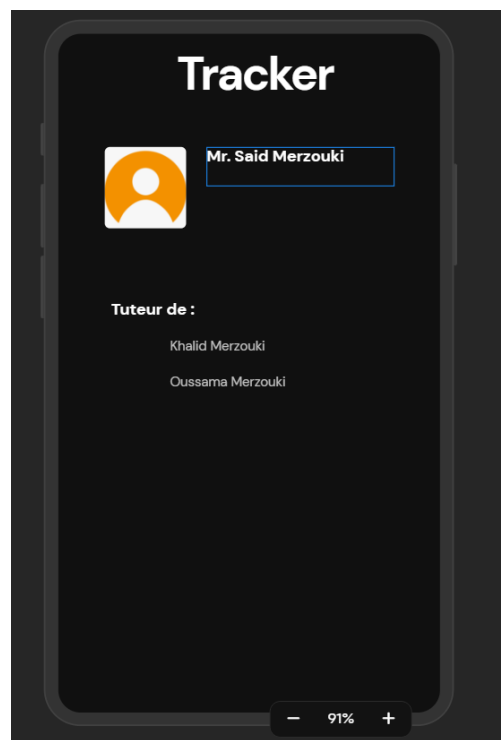
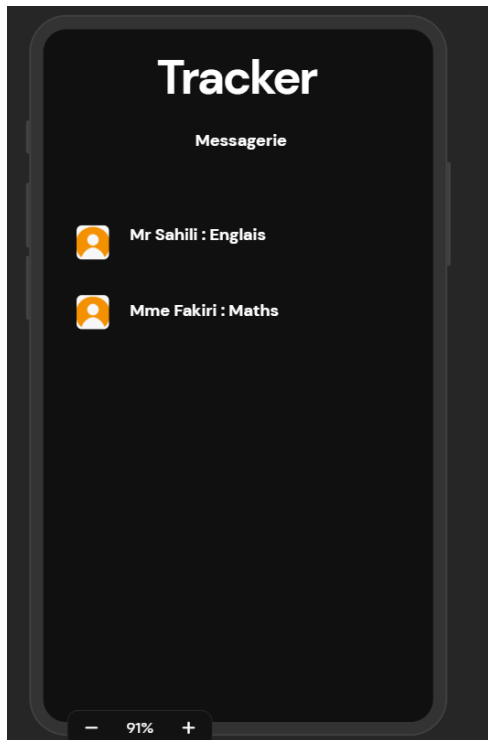


Figure 9: Diagramme d'activité

Maquette :

On présente ici des interfaces conçu pour notre application .





4^{ème} Chapitre : Technique de développement :



- ❖ Introduction .
- ❖ Description de l'environnement de développement intégré.
- ❖ Interfaces de notre application.

Introduction :

Après avoir achevé l'étape de conception de l'application, on va entamer dans ce chapitre la partie réalisation qui constitue le dernier volet de ce rapport et qui a pour objectif d'exposer le travail réalisé. Pour ce faire, on va commencer tout d'abord par préciser l'environnement matériel et logiciel de ce travail.

Environnement Logiciel :

- Spring Boot :

Spring Boot est un framework créé par l'équipe de chez Pivotal, conçu pour simplifier le démarrage et le développement de nouvelles applications Spring. Le framework propose une approche dogmatique de la configuration, qui permet d'éviter aux développeurs de redéfinir la même configuration à plusieurs endroits du code. Dans ce sens, Boot se veut d'être un acteur majeur dans le secteur croissant du développement d'applications rapide.

Pourquoi utiliser Spring Boot ?

Les avantages à utiliser Spring Boot sont multiples :

- ✓ **Légereté :** Spring Boot a la particularité d'être très léger et d'embarquer avec lui le strict minimum pour faire tourner votre service.
- ✓ **Intégration facilitée :** Spring Boot s'intègre particulièrement bien dans une architecture orientée micro services... et c'est l'un des seuls ! En effet, l'adoption des architectures micro services au sein des organisations étant relativement récentes, il n'existait pas dans l'univers de Java de framework capable de créer des services suffisamment légers et performants.

- ✓ Simplicité de prise en main : Spring Boot permet donc de créer une API de services très simplement. Il suffit d'embarquer directement le serveur d'application dans un seul et unique Jar qui est exécutable, par exemple, directement dans un service de conteneur (exemple : Amazon Web Service, dans un App Service sur Microsoft Azure, dans un conteneur Docker ou autre).

- J2EE :

C'est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées.

Elle est composée de deux parties essentielles :

-un **ensemble de spécifications** pour une infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'applications.

-un **ensemble d'API** qui peuvent être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers. Sun propose une implémentation minimale des spécifications de J2EE : le J2EE SDK. Cette implémentation permet de développer des applications respectant les spécifications mais n'est pas prévue pour être utilisée dans un environnement de production. Ces spécifications doivent être respectées par les outils développés par des éditeurs tiers.

L'utilisation de J2EE pour développer et exécuter une application offre plusieurs avantages :

- ✓ une architecture d'applications basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement
- ✓ la possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA ...
- ✓ la possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres J2EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de J2EE.

-Android studio :

Android Studio est 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.

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément18,19. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.

Interfaces de notre applications :



Figure 10 : Interface de Login de l'application

On peut nous connecter à l'application à partir de cette page.

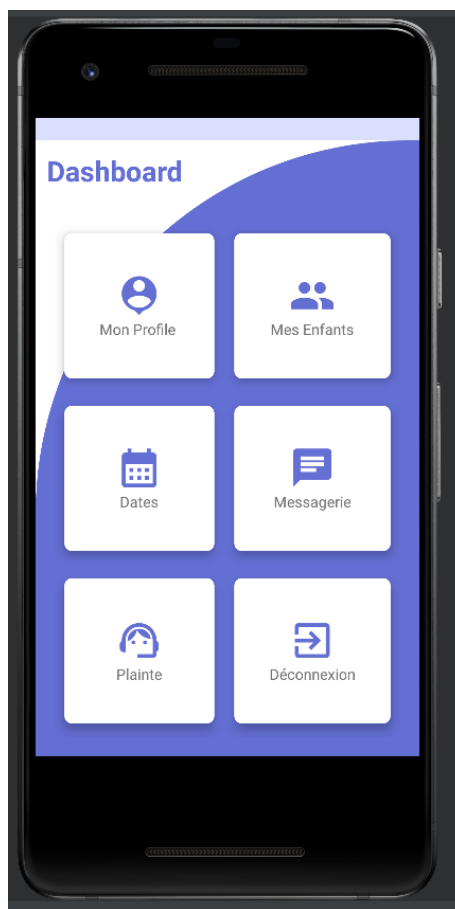


Figure 11 : Menu Principale de l'application (Tuteur)

Cette page représente le menu principal du tuteur où il peut accéder aux différentes fonctionnalités de l'application.

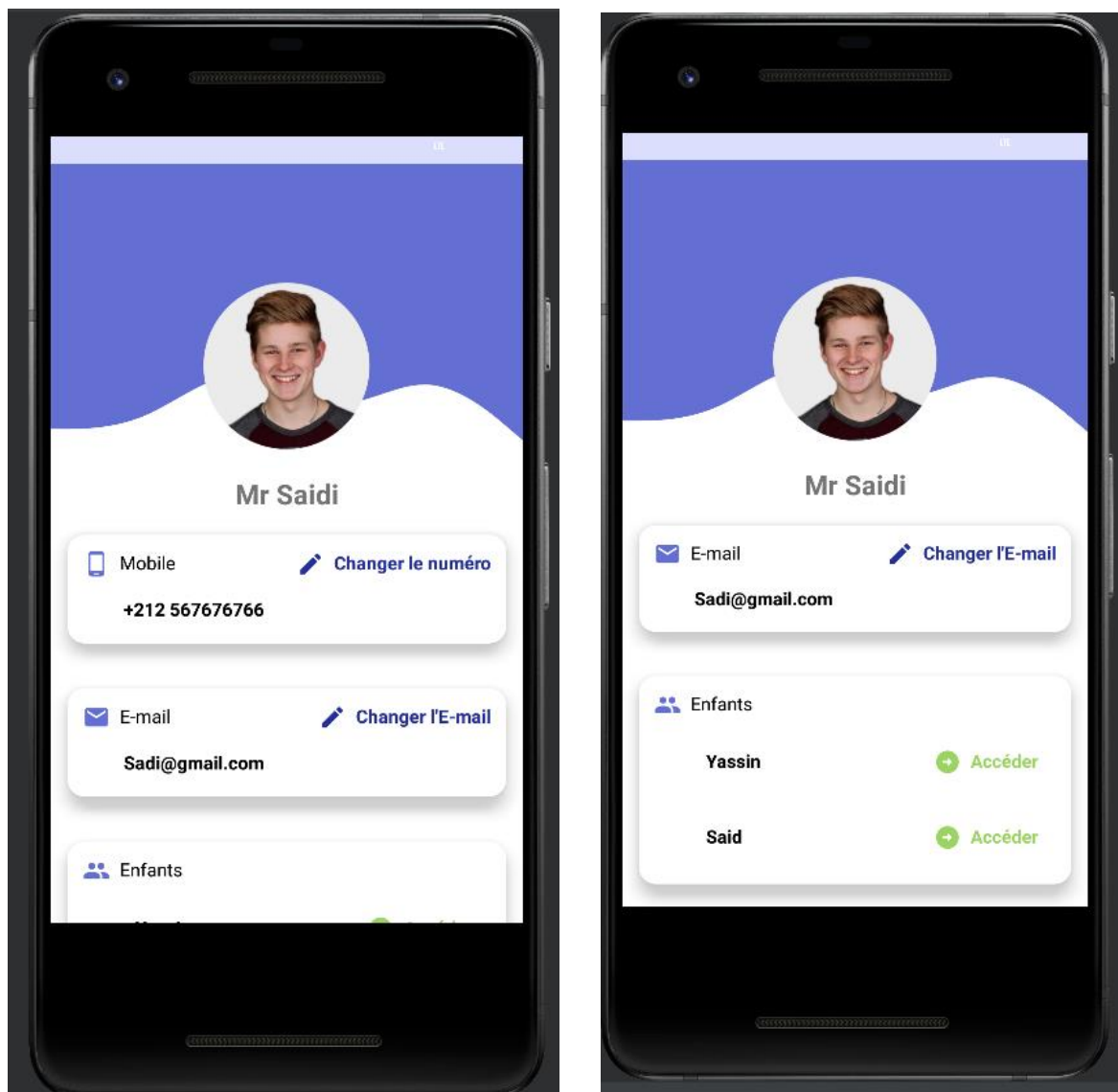


Figure 12 : Profile

Cette page représente le profile du tuteur, elle est scrollable et présente les informations du tuteurs.

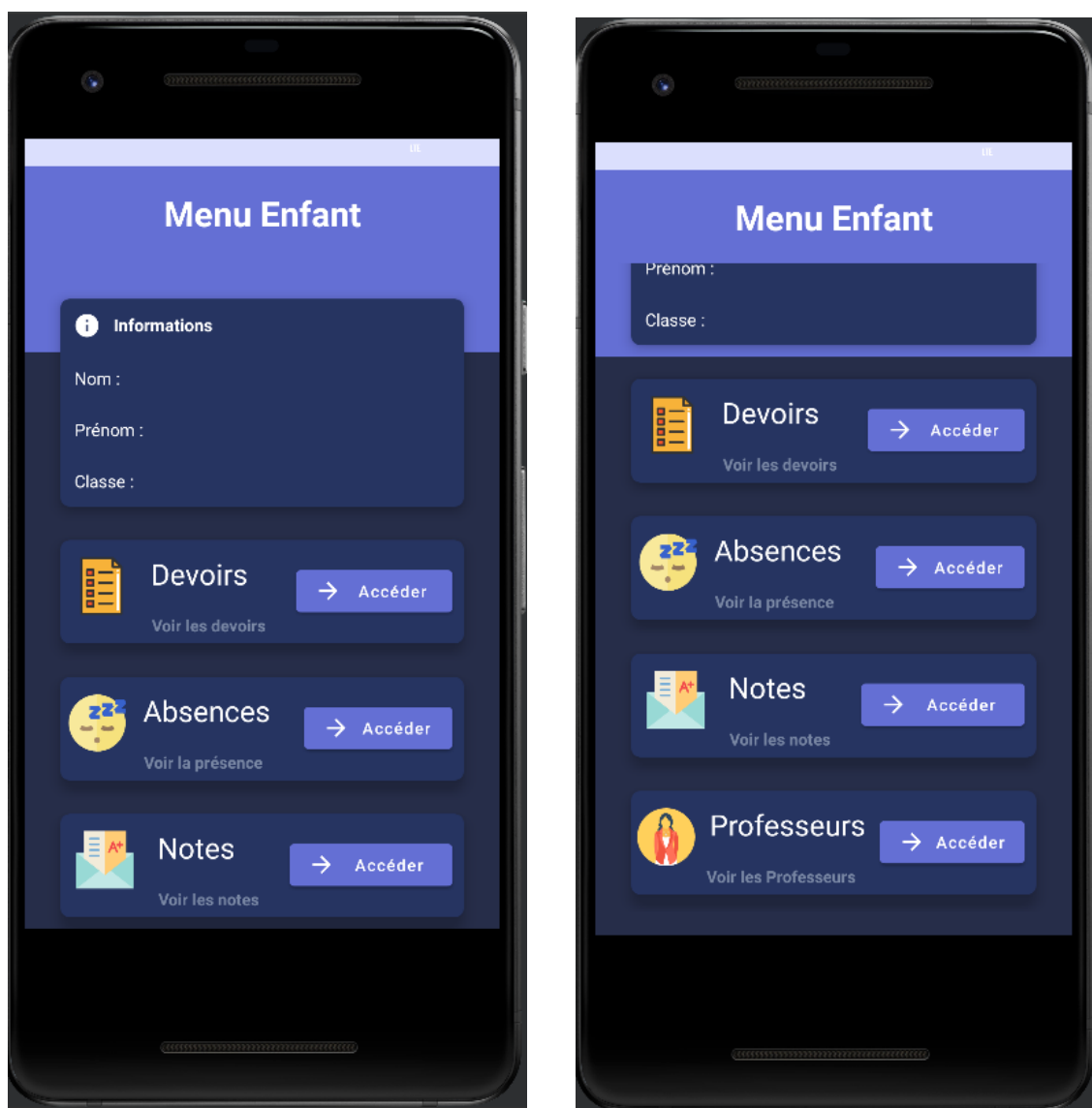


Figure 13 : Menu Enfant

Ce menu représente le menu d'enfant où le tuteur peut accéder au différent volets pour apercevoir les performances de son enfant.

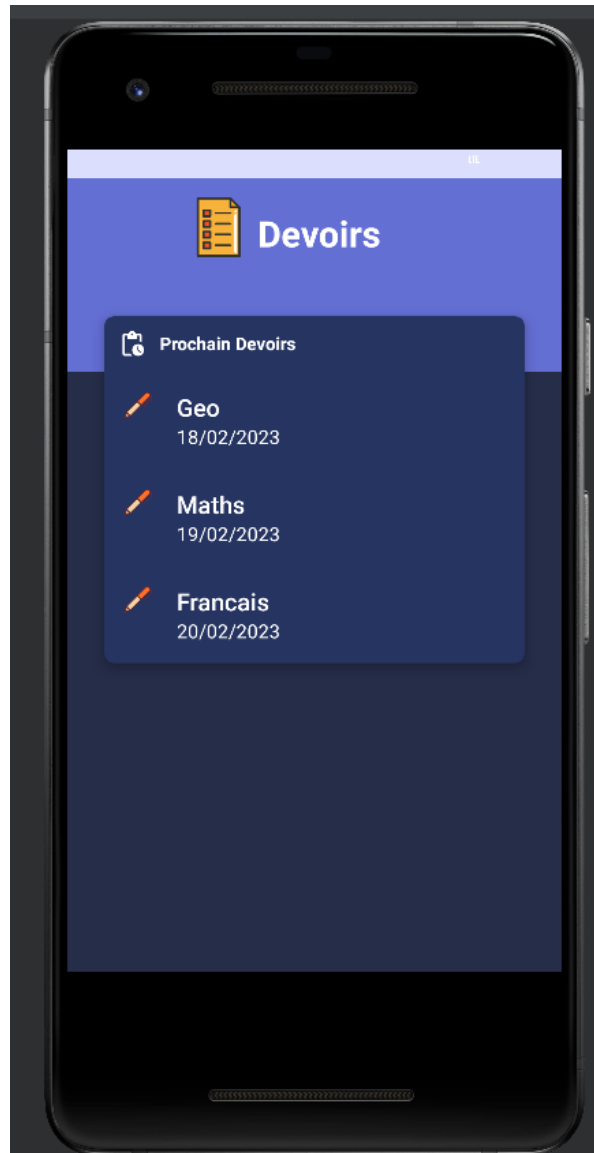


Figure 14 : Devoirs

Cette page représente les devoirs qu'un enfant doit accomplir.

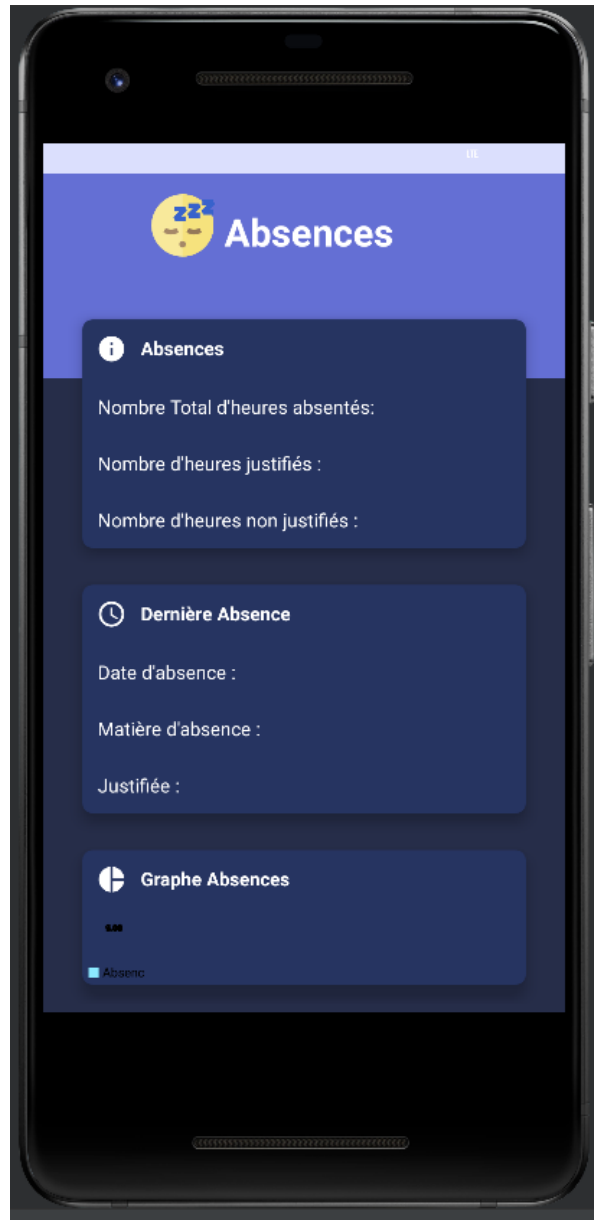


Figure 15 : Absences

Ce volet montre le taux d'absentisme d'un étudiant.



Figure 16 : Liste des professeurs de l'étudiant

Ce menu affiche la liste des professeurs d'un étudiant avec un accès pour leur écrire des messages.

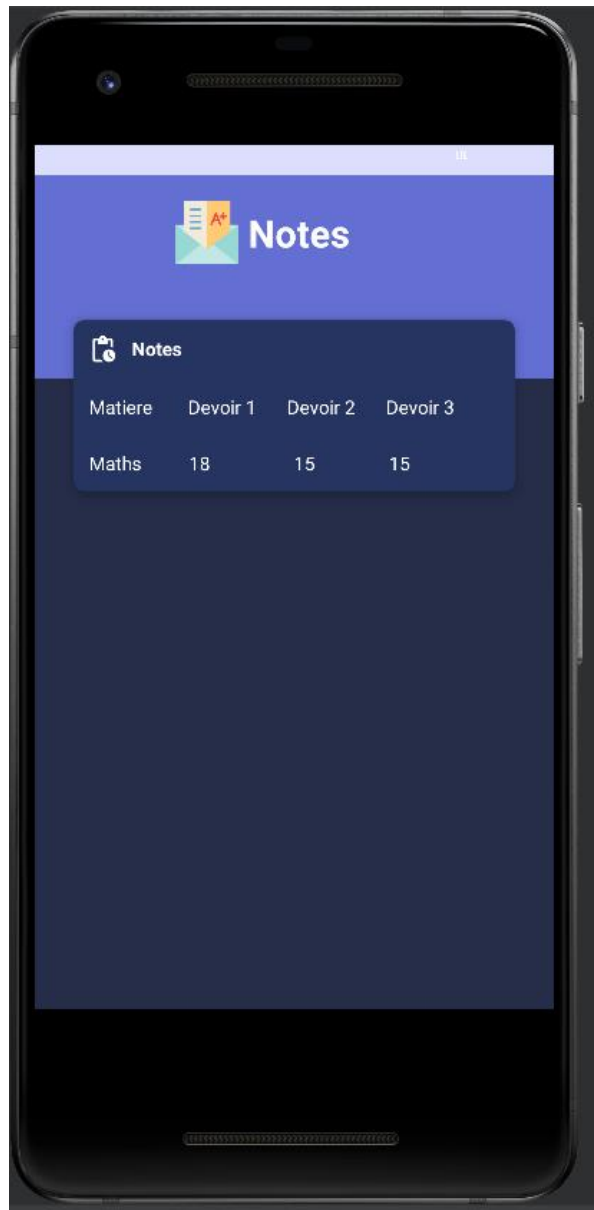


Figure 17 : Notes des examens

Ce volet montre les différentes notes des examens des différentes matières.



Figure 18 : Page de Messagerie (tuteur et professeur)

Ce menu montre la messagerie d'un tuteur ou d'un professeur.



Figure 19 : Devoirs de tous les enfants sous le tutorat de l'utilisateur

Ce menu montre les différents devoirs des différents étudiants sous le tutorat de l'utilisateur.

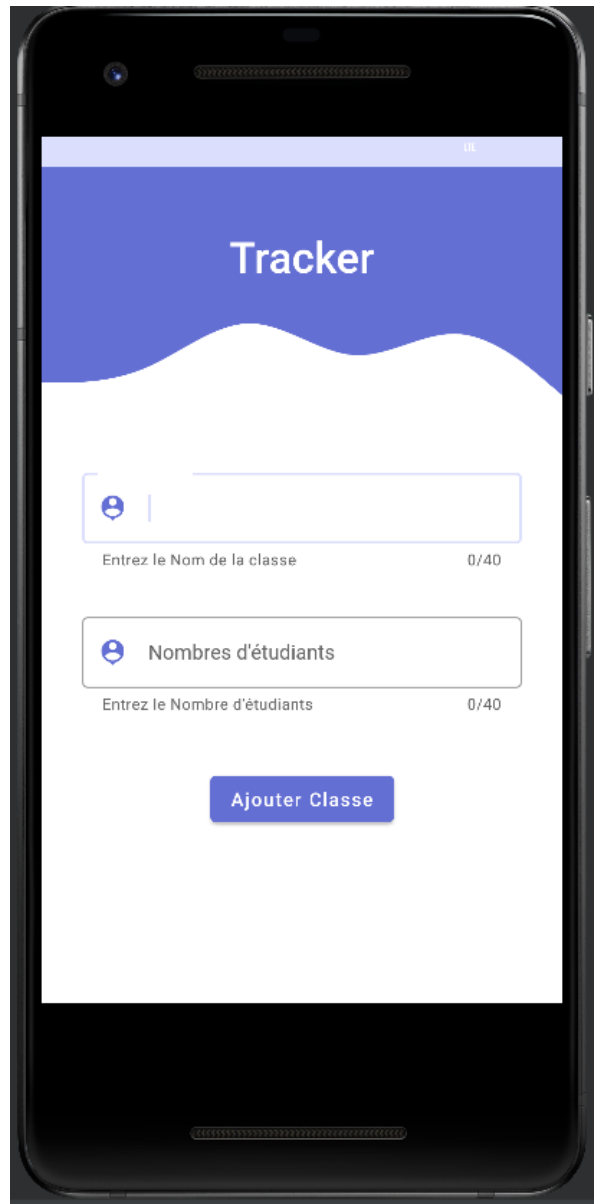


Figure 20 : Ajout des classe dans la Base de Données

Cette interface est conçu pour l'administrateur afin d'ajouter des classes à la base de donnée.

Conclusion :

Ce projet qu'on a réalisé en encadrement de notre professeur Mr Erraji nous a permis d'appliquer les connaissances qui nous ont été enseigné au cours de ces 2 années en DSE, à l'institut national de statistique et d'économie appliquées.

Nous avons réalisé une application Mobile qui permettant de faciliter la tâche de suivi scolaire des parents à leurs enfants.

Au cours de ce projet encadré, nous nous sommes parvenu d'abord à la détermination des besoins du marché. Ainsi nous avons proposé une solution adéquate pour améliorer la situation et aider les écoles et les parents à suivre le parcours scolaire de leurs enfants avec plus de simplicité.

Notre approche dans ce travail consistait à mettre l'accent sur l'affichage des données, et à implémenter des interfaces graphiques offrant la simplicité et la clarté aux utilisateurs pour pouvoir bénéficier des biens de notre application. La réalisation de ce projet a nécessité la maîtrise d'une multitude de langages d'outils de développement comme UML pour la conception, Springboot et JEE pour le développement plus MYSQL.

Bibliographie

https://fr.wikibooks.org/wiki/Programmation_JEE/Pr%C3%A9sentation_du_JEE

<https://www.quora.com/How-can-I-use-Spring-Boot-as-a-backend-in-an-Android-app-Android-Studio>

<https://www.appstud.com/fr/guides/agence-mobile/app069/>

<https://docs.microsoft.com/en-us/azure/azure-sql/?view=azuresql>