

RAPPORT DE STAGE d'IMMERSION EN ENTREPRISE

**SUJET : DÉVELOPPEMENT D'UNE SOLUTION CROSS
PLATFORME POUR LA GESTION DE CLUB ACA ROBOTICS**

Réalisé par :

– **Oumayma AYED**

Encadré par :

– **Mme Kouloud FILALI à ACA Robotics Technologies**

Organisme d'accueil : ACA Robotics Technologies - Technopole El Ghazala



Année universitaire 2025 – 2026

Table des matières

Introduction générale.....	4
Chapitre 1 : Présentation du projet	5
1.1 Introduction.....	5
1.2 Présentation de l'organisme d'accueil	5
1.3 Description du projet.....	6
1.4 Étude de l'existant	6
1.5 Solution proposée	6
1.6 Choix méthodologiques.....	7
1.6.1 Choix de Scrum.....	7
1.6.2 Définition de Scrum.....	7
1.7 Conclusion	8
Chapitre 2: Analyse et spécification des besoins	9
2.1 Introduction.....	9
2.2 Identification des acteurs	9
2.3 Spécification des besoins fonctionnels.....	9
2.4 Spécification des besoins non fonctionnels	11
2.5 Conclusion.....	11
Chapitre 3: Conception	12
3.1 Introduction.....	12
3.2 Diagramme de classes détaillé	12
3.3 Architecture physique	13
3.4 Architecture logique	14
3.5 Flowchart du système	15
3.5 Conclusion	15
Chapitre 4: Réalisation	16
4.1 Introduction.....	16
4.2 Framework et technologies utilisés.....	16
4.3 Environnement de travail	16

4.4	Base de données	17
4.5	Interfaces graphiques réaliser	17
4.5.1	Authentification et inscription.....	17
4.5.2	Tableau de bord administrateur	18
4.5.3	Gestion des enfants.....	19
4.5.4	Gestion des absences	20
4.5.5	Gestion des cours	21
4.5.6	Gestion des paiements.....	22
4.5.7	Gestion des demandes d'inscription	23
4.5.8	Espace parent.....	24
4.6	Conclusion	25
	Conclusion générale et perspectives.....	26

Introduction générale

Dans un contexte marqué par la transformation digitale et l'évolution rapide des technologies de l'information, les solutions numériques jouent un rôle important dans l'amélioration de l'organisation et de la gestion des structures éducatives et technologiques. Les clubs et centres de formation spécialisés dans la robotique et la programmation doivent aujourd'hui s'appuyer sur des outils numériques afin d'optimiser la gestion de leurs activités et faciliter la communication avec leurs utilisateurs.

C'est dans ce cadre que s'inscrit le présent projet réalisé au sein de **ACA Robotics Technologies**, une structure spécialisée dans la formation en robotique et en technologies innovantes. Cette organisation propose des ateliers et des formations destinés aux jeunes afin de développer leurs compétences dans les domaines de la robotique, de la programmation et des nouvelles technologies.

L'objectif de ce projet est de concevoir et de développer une application **cross-platform nommée ACA Robotique Junior**, permettant de digitaliser la gestion des activités du club, notamment la gestion des inscriptions des enfants aux cours, le suivi des présences ainsi que la gestion des paiements. L'application a été développée en utilisant **Flutter** et les services **Firebase**, et elle est actuellement déployée sous forme d'une application web.

Ce rapport est organisé en plusieurs chapitres. Le premier chapitre présente le contexte général du projet ainsi que l'organisme d'accueil. Le deuxième chapitre décrit l'analyse et la spécification des besoins. Le troisième chapitre présente la conception du système. Enfin, le dernier chapitre est consacré à la réalisation de l'application.

Chapitre 1 : Présentation du projet

1.1 Introduction

Ce chapitre a pour objectif de présenter le cadre général du projet, l'organisme d'accueil, ainsi qu'une description détaillée de la solution développée. Il inclut également une étude de l'existant, la solution proposée ainsi que les choix méthodologiques adoptés pour la réalisation du projet.

1.2 Présentation de l'organisme d'accueil

Le projet a été réalisé au sein de **ACA Robotics Technologies**, Fondée en **2017** et située à Technopole El Ghazala (Bloc I3), une structure spécialisée dans la formation et la promotion des technologies innovantes, notamment dans les domaines de la robotique, de la programmation et des systèmes intelligents. L'organisme propose des formations et des ateliers éducatifs destinés aux enfants et aux jeunes afin de développer leurs compétences technologiques et encourager l'apprentissage pratique.

ACA Robotics organise plusieurs activités pédagogiques telles que des clubs de robotique, des ateliers de programmation et des événements technologiques visant à stimuler la créativité et l'innovation chez les jeunes participants. À travers ces initiatives, l'organisme cherche à préparer les nouvelles générations aux métiers du futur liés aux technologies numériques et à l'ingénierie.

Dans le cadre de ses activités, ACA Robotics accueille un grand nombre d'enfants inscrits à différents programmes de formation. La gestion de ces inscriptions, des présences et des paiements nécessite un système organisé permettant de faciliter le suivi administratif et pédagogique des participants.



Figure 1.1 – Logo Aca ROBOTICS

1.3 Description du projet

Le projet réalisé consiste à développer une application nommée **ACA Robotique Junior**, destinée à digitaliser la gestion des activités du club ACA Robotics. Cette application permet de gérer les inscriptions des enfants aux différents cours proposés par le club, de suivre les présences et les absences, ainsi que de gérer les paiements associés aux formations. Elle offre également aux parents un espace dédié leur permettant de consulter les informations relatives à leurs enfants, notamment les cours suivis et l'historique des présences. L'application a été développée en utilisant le framework **Flutter**, permettant de créer une application **cross-platform**. Actuellement, l'application est déployée sous forme d'une **application web**, accessible via un navigateur. Elle utilise également les services **Firebase**, notamment Firebase Authentication pour la gestion des utilisateurs et Cloud Firestore pour le stockage et la gestion des données.

1.4 Étude de l'existant

Avant de développer une nouvelle solution, il est important d'analyser les méthodes actuellement utilisées pour la gestion des activités dans les clubs éducatifs et les centres de formation. Dans plusieurs structures éducatives, la gestion des inscriptions et du suivi des participants est souvent réalisée à l'aide d'outils simples tels que des fichiers Excel, des documents papier ou des formulaires manuels. Cette approche peut fonctionner pour un nombre limité de participants, mais elle devient rapidement difficile à gérer lorsque le nombre d'inscriptions augmente. Ces méthodes présentent plusieurs limites, notamment la difficulté de centraliser les informations, les risques d'erreurs lors de la saisie des données et le manque de visibilité sur les informations liées aux présences et aux paiements. De plus, l'absence d'une plateforme numérique dédiée rend la communication entre les administrateurs et les parents moins efficace.

1.5 Solution proposée

Le projet vise à pallier les limitations des méthodes traditionnelles de gestion des inscriptions, des présences et des paiements utilisées dans certains clubs éducatifs. Afin de répondre à ces besoins, une application numérique nommée **ACA Robotique Junior** a été développée pour digitaliser et centraliser la gestion des activités du club **ACA Robotics**.

Cette solution permet d'améliorer l'organisation des activités du club, de faciliter la gestion administrative et d'offrir aux parents un moyen simple de suivre les informations relatives à leurs enfants.

La solution est conçue en plusieurs modules :

- **Backend** : gestion et centralisation des données à l'aide de la plateforme **Firebase**, incluant l'authentification des utilisateurs, la gestion de la base de données via **Cloud Firestore** et le stockage sécurisé des informations.

- **Frontend : Application cross-platform** : application développée avec **Flutter**, permettant un accès multiplateforme. Comportant 2 espaces :

- ❖ **Espace administrateur** : espace dédié aux administrateurs permettant de gérer les enfants inscrits, les cours proposés, les présences ainsi que le suivi des paiements. Il permet également la validation des inscriptions et la génération de rapports.
- ❖ **Espace parent** : espace destiné aux parents leur permettant d'inscrire leurs enfants aux différents cours, de consulter les présences et les absences, et de suivre les informations relatives aux activités du club.

Cette solution vise à améliorer l'efficacité de la gestion des activités du club, à faciliter la communication entre les administrateurs et les parents, et à offrir une plateforme numérique moderne permettant de centraliser toutes les informations relatives aux formations et aux participants.

1.6 Choix méthodologiques

Pour assurer une gestion efficace du projet et faciliter l'organisation du travail, il a été décidé d'adopter la méthodologie **Scrum**, qui fait partie des méthodes agiles.

1.6.1 Choix de Scrum

Pour ce projet, nous avons opté pour Scrum en tant que méthodologie de travail en raison de sa capacité à offrir une approche flexible et adaptative pour le développement de logiciels. Scrum s'adapte aux changements dans les besoins fonctionnels et non fonctionnels, et permet d'ajuster le projet au fil du temps. De plus, Scrum encourage ainsi la créativité et la communication au sein de l'équipe de travail en favorisant une collaboration active.

1.6.2 Définition de Scrum

Scrum n'est pas une méthode, mais un cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. Le framework Scrum permet de travailler en équipe pour faire de l'amélioration continue sur des livraisons itératives incrémentales de produits afin de satisfaire les besoins des clients. La Figure 1.2 présente le processus de Scrum.

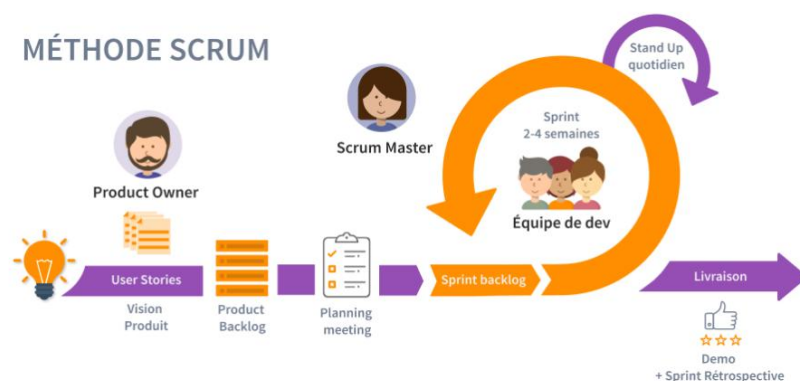


Figure 1.2 – Le processus de Scrum

Scrum est basé sur trois piliers fondamentaux. Il s'agit de :

- La transparence : le fait d'être honnête, de ne rien avoir à cacher, de travailler ensemble au succès du produit/projet en rendant les aspects importants du processus visibles à tous ceux qui sont responsables des résultats.
- L'inspection : le fait de pouvoir s'entraider et inspecter les artefacts Scrum et l'état d'avancement par rapport à un Objectif de Sprint afin de détecter les écarts indésirables.
- L'adaptation : le fait de s'adapter aux changements en général, changements de produit, changements de façon de faire.

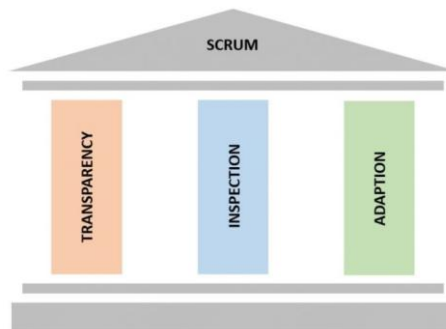


Figure 1.3 – Les trois piliers de Scrum

1.7 Conclusion

Dans ce chapitre, nous avons présenté le contexte général du projet réalisé au sein du club ACA Robotique ainsi que l'organisme d'accueil. Nous avons également décrit le projet développé et analysé les limitations des méthodes de gestion existantes. Enfin, nous avons présenté la solution proposée ainsi que la méthodologie de développement adoptée pour la réalisation du projet.

Le chapitre suivant sera consacré à **l'analyse et à la spécification des besoins du système**, afin d'identifier les fonctionnalités attendues et les exigences nécessaires à la mise en place de l'application.

Chapitre 2: Analyse et spécification des besoins

2.1 Introduction

Dans ce chapitre, nous allons analyser les besoins du système développé au sein du club ACA Robotique. Cette étape est essentielle pour identifier les différents acteurs du système ainsi que les fonctionnalités attendues. Elle permet également de définir les besoins fonctionnels et non fonctionnels afin de garantir une solution adaptée aux exigences du projet.

2.2 Identification des acteurs

Dans le contexte de UML (Unified Modeling Language), un acteur est défini comme une entité externe au système qui interagit avec celui-ci. Ainsi, un acteur peut représenter un utilisateur humain, un autre système informatique ou une organisation qui joue un rôle spécifique dans une ou plusieurs interactions avec le système. Dans le contexte de notre projet, nous identifions les acteurs suivants :

- **Parent** : utilisateur externe qui accède à l'application via mobile ou web afin d'inscrire ses enfants, suivre leurs présences et les paiements.
- **Administrateur** : utilisateur interne responsable de la gestion globale de l'application, notamment la validation des inscriptions, la gestion des cours et le suivi des paiements.

2.3 Spécification des besoins fonctionnels

Gestion de l'authentification

- **BF01** — Le système doit permettre à un utilisateur de s'inscrire
- **BF02** — Le système doit permettre à un utilisateur de se connecter via son email et son mot de passe.
- **BF03** — Le système doit vérifier le rôle de l'utilisateur connecté (parent ou administrateur) afin de lui afficher les fonctionnalités correspondantes.
- **BF04** — Le système doit bloquer l'accès à l'application pour tout utilisateur non authentifié.
- **BF05** — Le système doit permettre à un utilisateur de se déconnecter de son compte.

Gestion des inscriptions

- **BF06** — Le système doit permettre à un parent de soumettre une demande d'inscription pour son enfant en choisissant un club, un local et un horaire.
- **BF07** — Le système doit afficher le statut de la demande d'inscription (EN_ATTENTE, APPROVED, REJECTED).
- **BF08** — Le système doit permettre à l'administrateur de consulter la liste des demandes d'inscription.

- **BF09** — Le système doit permettre à l'administrateur de valider ou de refuser une demande d'inscription.
- **BF10** — Le système doit notifier le parent du résultat de sa demande d'inscription.
- **BF11** — Le système doit bloquer l'accès au compte parent en cas de refus de sa demande d'inscription.

Gestion des enfants

- **BF12** — Le système doit permettre à un parent d'ajouter les informations de son enfant (nom, date de naissance ...).
- **BF13** — Le système doit permettre à un parent de consulter les informations relatives à ses enfants inscrits.
- **BF14** — Le système doit permettre à l'administrateur de consulter la liste des enfants inscrits dans chaque club et de génère un rapport de la liste des enfants en version PDF ou Excel .

Gestion des cours

- **BF15** — Le système doit permettre à l'administrateur de créer, modifier et supprimer des cours.
- **BF16** — Le système doit associer à chaque cours un formateur, un horaire, un local et un niveau.
- **BF17** — Le système doit permettre à un parent de consulter les cours disponibles.
- **BF18** — Le système doit permettre à un parent de choisir un cours lors de la demande d'inscription de son enfant.

Gestion des présences

- **BF19** — Le système doit permettre à l'administrateur d'enregistrer les présences et absences des enfants pour chaque séance.
- **BF20** — Le système doit comptabiliser automatiquement le nombre d'absences par enfant.
- **BF21** — Le système doit alerter l'administrateur lorsqu'un enfant atteint le seuil de 4 absences.
- **BF22** — Le système doit permettre à un parent de consulter les absences de son enfant.

Gestion des paiements

- **BF23** — Le système doit permettre à un parent de consulter l'état de ses paiements.
- **BF24** — Le système doit afficher le montant payé et le montant restant par tranche.
- **BF25** — Le système doit afficher le statut du paiement (PAYE, NON_PAYE, EN_ATTENTE).
- **BF26** — Le système doit permettre à l'administrateur de gérer et suivre les paiements des parents.

Gestion des utilisateurs

- **BF28** — Le système doit permettre à l'administrateur de consulter la liste complète des utilisateurs.
- **BF29** — Le système doit permettre à l'administrateur de bloquer ou débloquent l'accès d'un parent.
- **BF30** — Le système doit afficher le statut de chaque parent (EN_ATTENTE, APPROVED, BLOCKED).

2.4 Spécification des besoins non fonctionnels

Performance

L'application doit offrir un temps de réponse rapide lors des différentes opérations, notamment lors de l'authentification des utilisateurs, de la consultation des données et de la gestion des informations.

Sécurité

Le système doit garantir la sécurité des données des utilisateurs en utilisant un mécanisme d'authentification sécurisé. L'accès aux données doit être limité selon les rôles des utilisateurs (administrateur ou parent).

Ergonomie

L'interface utilisateur doit être simple, intuitive et facile à utiliser afin de permettre aux parents et aux administrateurs d'accéder facilement aux différentes fonctionnalités de l'application.

Accessibilité

L'application doit être accessible depuis un navigateur web, permettant aux utilisateurs d'y accéder à tout moment et depuis différents appareils.

Fiabilité

Le système doit assurer la sauvegarde et la gestion fiable des données afin d'éviter toute perte d'informations relatives aux utilisateurs, aux enfants et aux activités du club.

2.5 Conclusion

Ce chapitre a permis d'identifier et de spécifier les besoins essentiels de l'application **ACA Robotique Junior**, en distinguant les besoins fonctionnels et non fonctionnels. Les besoins fonctionnels définissent les services que l'application doit offrir aux parents et aux administrateurs.

Chapitre 3: Conception

3.1 Introduction

Après avoir analysé et spécifié les besoins fonctionnels et non fonctionnels du système dans le chapitre précédent, il est nécessaire de passer à la phase de conception. Cette étape constitue une phase essentielle dans le processus de développement, car elle permet de définir l'architecture du système ainsi que les différents composants nécessaires à sa réalisation.

3.2 Diagramme de classes détaillé

La Figure 3.1 présente le diagramme de classe

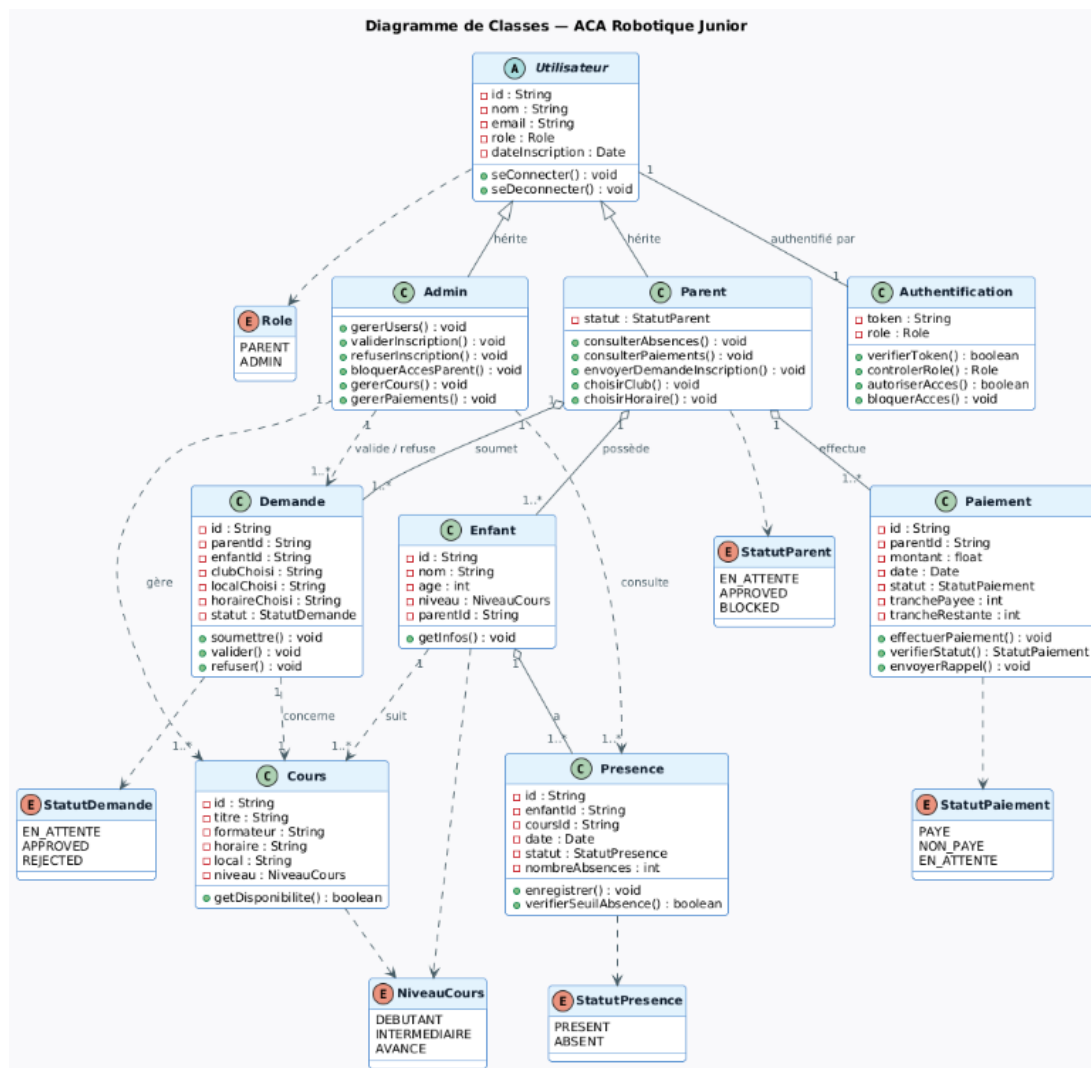


Figure 3.1 – Diagramme de classes

3.3 Architecture physique

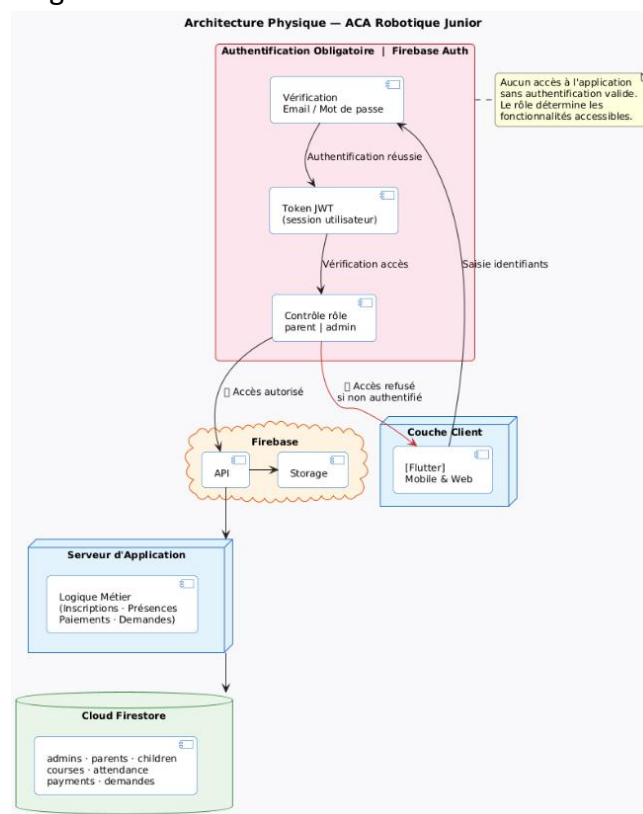
L'architecture physique décrit l'organisation matérielle et logicielle du système ainsi que les interactions entre ses différents composants.

L'application **ACA Robotique Junior** repose sur une architecture **client-serveur** permettant aux utilisateurs d'accéder au système via l'application mobile ou via un navigateur web.

Cette architecture est composée des éléments suivants :

- **Application (Front-end)** : développée avec le framework **Flutter**, elle fournit l'interface utilisateur permettant aux parents et aux administrateurs d'interagir avec le système. Les utilisateurs peuvent consulter les informations, gérer les inscriptions et accéder aux différentes fonctionnalités de l'application.
- **API et services Firebase** : assurent la communication entre l'interface utilisateur et la base de données. Firebase fournit plusieurs services essentiels tels que l'authentification des utilisateurs et la gestion des données.
- **Serveur d'application** : gère la logique métier de l'application, notamment la gestion des inscriptions, des présences et des paiements.
- **Base de données Cloud Firestore** : permet de stocker les informations relatives aux utilisateurs, aux enfants inscrits, aux cours, aux présences et aux paiements.

Cette architecture permet d'assurer une gestion centralisée des données et d'offrir une application accessible en ligne.



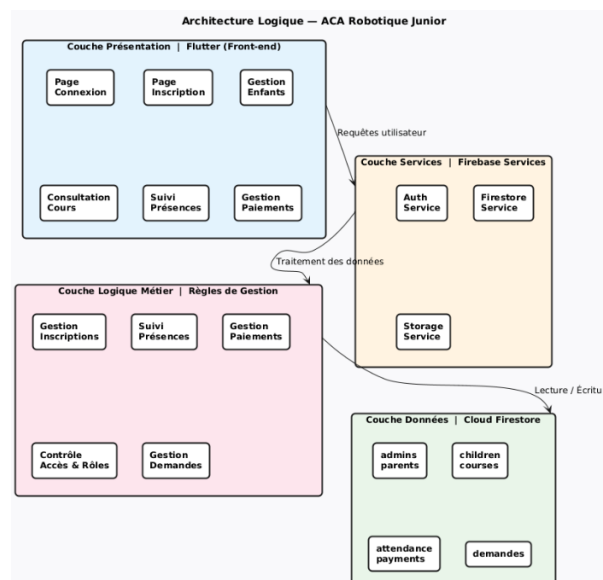
3.4 Architecture logique

L'architecture logique décrit l'organisation interne du système en différentes couches fonctionnelles. Elle permet de séparer les responsabilités du système afin de faciliter sa maintenance et son évolution.

L'architecture logique de l'application **ACA Robotique Junior** repose sur une structure en plusieurs couches :

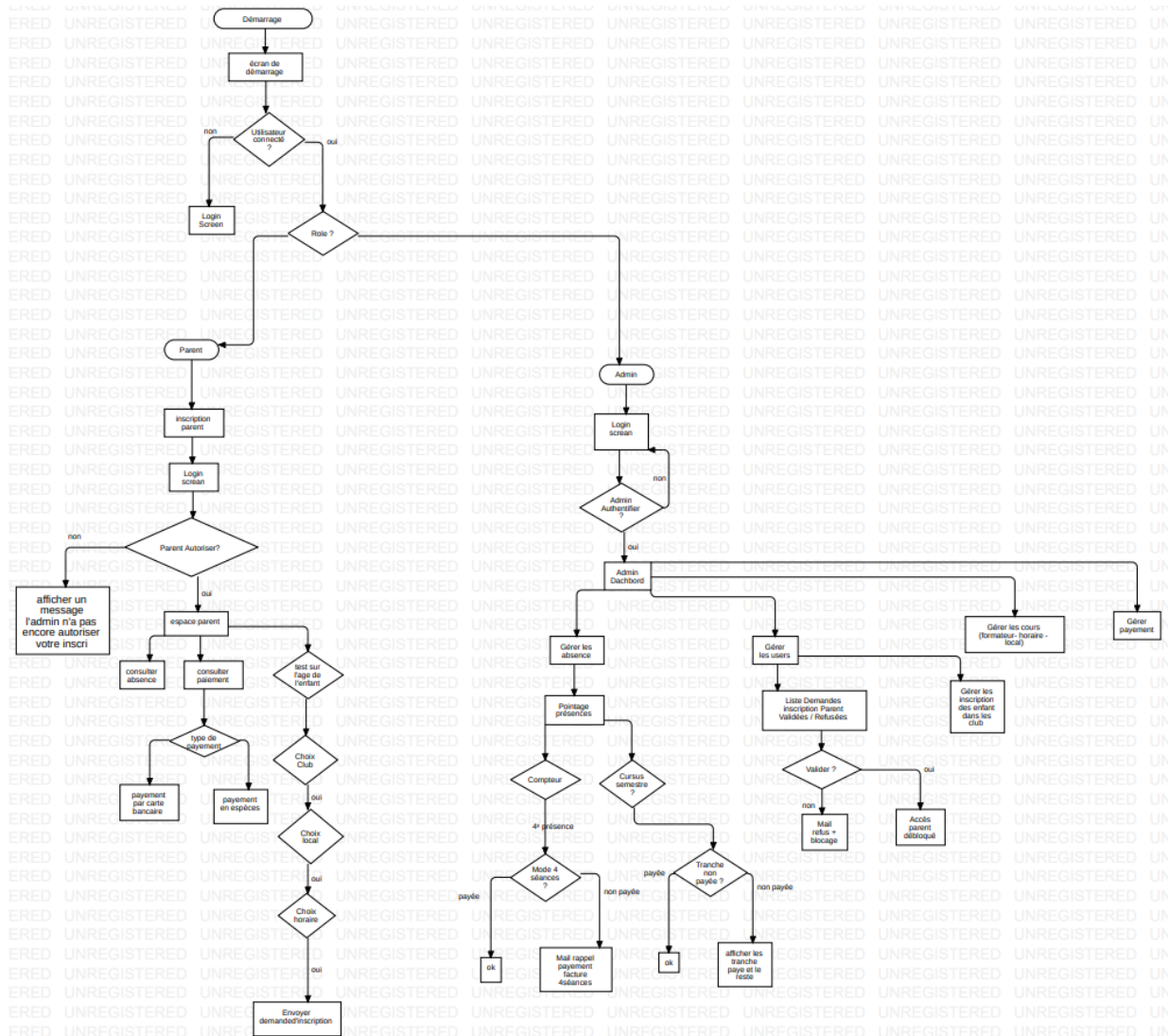
- **Couche présentation (Front-end)** : Cette couche correspond à l'interface utilisateur de l'application développée avec **Flutter**. Elle permet aux utilisateurs d'interagir avec le système à travers différentes interfaces telles que les pages d'inscription, de gestion des enfants, de consultation des cours et de suivi des présences.
- **Couche services** : Cette couche assure la communication entre l'interface utilisateur et les services Firebase. Elle gère les requêtes envoyées par les utilisateurs et permet d'effectuer les opérations nécessaires sur les données.
- **Couche logique métier** : Cette couche contient les règles de gestion du système, notamment la gestion des inscriptions des enfants, le suivi des présences ainsi que la gestion des paiements. Elle permet également de contrôler l'accès aux différentes fonctionnalités selon le rôle de l'utilisateur.
- **Couche données** : Cette couche correspond à la base de données **Cloud Firestore** qui permet de stocker toutes les informations relatives aux utilisateurs, aux enfants, aux cours, aux présences et aux paiements.

Grâce à cette organisation en couches, l'application bénéficie d'une structure claire et modulaire facilitant son évolution et sa maintenance.



3.5 Flowchart du système

Afin de mieux illustrer le fonctionnement global de l'application développée pour ACA Robotics, un diagramme de flux (Flowchart) est présenté ci-dessous. Ce diagramme décrit les différentes étapes du processus ainsi que les interactions entre les différents modules du système.



3.5 Conclusion

Ce chapitre a permis de présenter la conception de l'application à travers le diagramme de classes, les architectures physique et logique ainsi que le flowchart du système. Cette phase constitue une base essentielle pour la réalisation et l'implémentation de la solution développée.

Chapitre 4: Réalisation

4.1 Introduction

L'étape finale, qui englobe toutes les étapes précédentes, est bien évidemment la réalisation du projet. Au cours de ce chapitre, nous allons présenter les principales étapes, en décrivant l'environnement de travail et les technologies utilisées pour la mise en place de notre plateforme ainsi que les différents supports. Et pour finir, nous présenterons quelques captures d'écran de la plateforme.

4.2 Framework et technologies utilisés

Flutter est un framework open-source développé par Google pour créer des applications mobiles multiplateformes (Android et iOS) à partir d'un seul code source. Basé sur le langage Dart, il permet de développer des interfaces utilisateur modernes, réactives et performantes grâce à son moteur de rendu intégré. Dans ce projet, Flutter a été utilisé pour développer l'application mobile **ACA Robotique Junior**, offrant une interface intuitive et un rendu cohérent sur différents appareils. Son approche multiplateforme a permis de réduire le temps de développement et de faciliter la maintenance en regroupant Android et iOS dans une même base de code.



Figure 4.1 – Logo Flutter

4.3 Environnement de travail

Visual Studio Code : Visual Studio Code est un éditeur de code source léger qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages et environnements d'exécution .



Figure 4.2 – Logo VS Code

GitHub : GitHub est une plateforme de développement collaborative basée sur Git. Elle permet de stocker, gérer et partager le code source, de suivre les versions, de collaborer via des branches et des pull requests, et d'intégrer des outils de CI/CD pour automatiser les workflows de développement .



Figure 4.3 – Logo GitHub

4.4 Base de données

Dans ce projet, nous utilisons Firebase, une plateforme de développement d'applications mobiles et web proposée par Google. Son service de base de données, *Cloud Firestore*, est une base de données NoSQL hébergée dans le cloud qui permet de stocker et de synchroniser les données en temps réel entre plusieurs clients. Firebase offre également des fonctionnalités intégrées comme l'authentification, le stockage de fichiers et les notifications push, ce qui simplifie le développement et la maintenance des applications. Grâce à Firebase, notre application mobile Flutter peut accéder rapidement aux données et assurer une expérience utilisateur fluide et réactive.



Figure 4.4 – Logo de Firebase

4.5 Interfaces graphiques réaliser

Cette section présente les principales interfaces de l'application ACA Robotique Junior, organisées selon les deux espaces utilisateurs : l'espace administrateur et l'espace parent.

4.5.1 Authentification et inscription

La Figure 4.5 présente la page de connexion et la Figure 4.6 illustre le processus d'inscription d'un parent, soumis à la validation de l'administrateur avant l'accès à l'application.



La Figure 4.5 : connexion

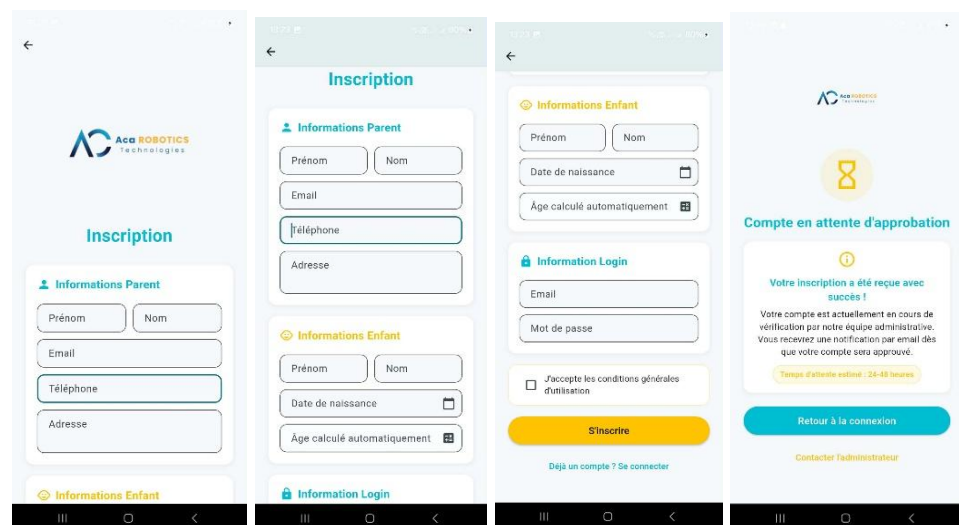
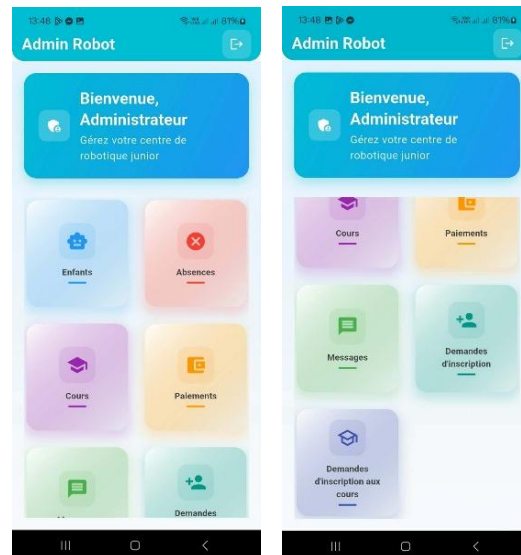


Figure 4.6 : inscription parent

4.5.2 Tableau de bord administrateur

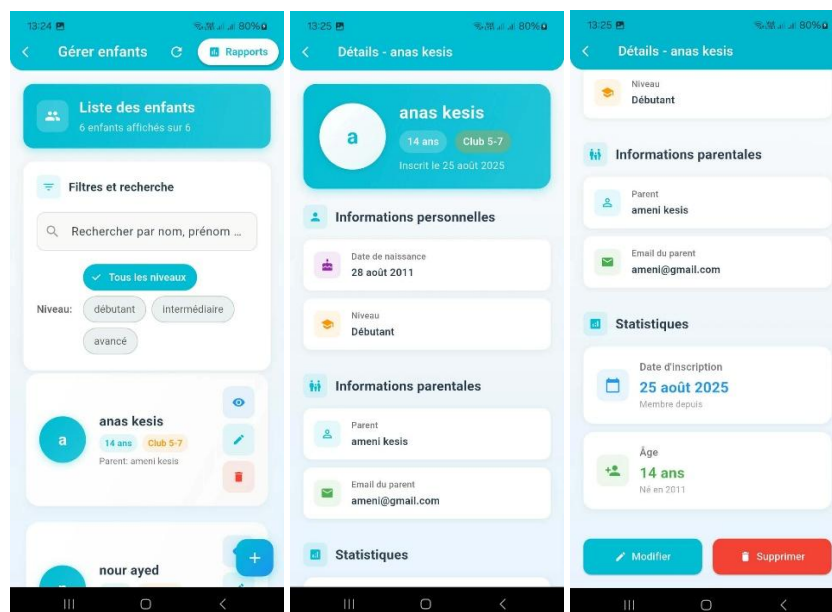
La Figure 4.7 présente le tableau de bord centralisant l'accès aux modules de gestion : enfants, absences, cours, paiements et demandes d'inscription.

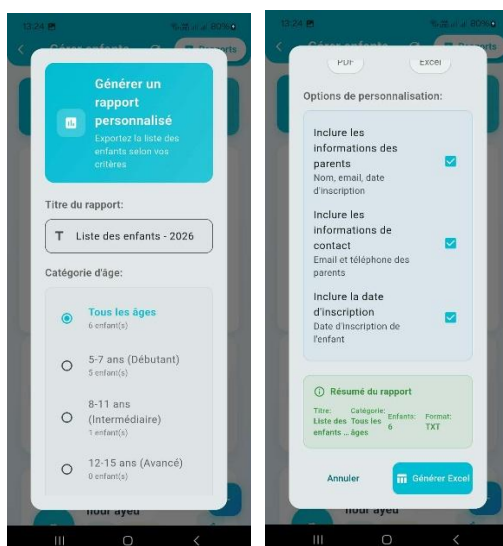


La Figure 4.7 : Dashboard Admin

4.5.3 Gestion des enfants

La Figure 4.8 illustre le module permettant de consulter, modifier, supprimer et rechercher des enfants, avec génération de rapports personnalisés en PDF ou Excel.





ACA ROBOTIQUE JUNIOR
Rapport des enfants - Liste des enfants

INFORMATIONS DU RAPPORT
Catégorie d'âge: Tous les âges Nombre d'enfants: 7
Date de génération: 2026-03-14 14:08:10

Nom	Prénom	Âge	Niveau	Parent
ayed	ouma	15 ans	Débutant	amen amen
kesis	anas	14 ans	Débutant	ameni kesis
ayed	nour	6 ans	Débutant	mouhamed ayed
ziadia	ranime	12 ans	débutant	ahmed ahmed
amine	amine	8 ans	Débutant	ahmed ahmed
ayed	doua	15 ans	Débutant	houcem houcem
ayed	amenallah	10 ans	intermédiaire	tarek ayed

Rapport généré automatiquement par ACA Robotique Junior - 2026

Enregistrement automatique ☐ rapport_enfants_Tous_les_ages_1773493457922 - Mode prot... - Enregistré dans ce l...

Fichier Accueil Insertion Dessin Mise en page Formules Données Révision Affichage Automatiser Aide

MODE PROTÉGÉ Attention aux fichiers provenant d'un emplacement Internet car ils peuvent contenir des virus. Il est recommandé de rester en mode...

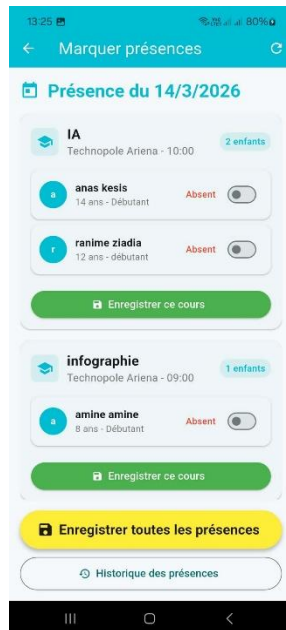
A1 ACA Robotique Junior

	A	B	C	D	E	F	G
1	ACA Robotique Junior						
2	Rapport des enfants - Liste des enfants						
3	Catégorie d'âge: Tous les âges						
4	Nombre d'enfants: 7						
5	Date de génération: 2026-03-14 14:04:17						
6							
7							
8	ayed	ouma	15 ans	Débutant	amen amen	amen@gmail.com	2026-03-14
9	kesis	anas	14 ans	Débutant	ameni kesis	ameni@gmail.com	2025-08-25
10	ayed	nour	6 ans	Débutant	mouhamed ay	mouhameday@gmail.com	2025-08-22
11	ziadia	ranime	12 ans	débutant	ahmed ahmed	ahmed@gmail.com	2025-08-11
12	amine	amine	8 ans	Débutant	ahmed ahmed	ahmed@gmail.com	2025-08-11
13	ayed	doua	15 ans	Débutant	houcem hou	houcem@gmail.com	2025-08-11
14	ayed	amenallah	10 ans	intermédiaire	tarek ayed	tarek@gmail.com	2025-08-07
15							
16							
17							
18							
19							

La Figure 4.8 : Gestion des enfants

4.5.4 Gestion des absences

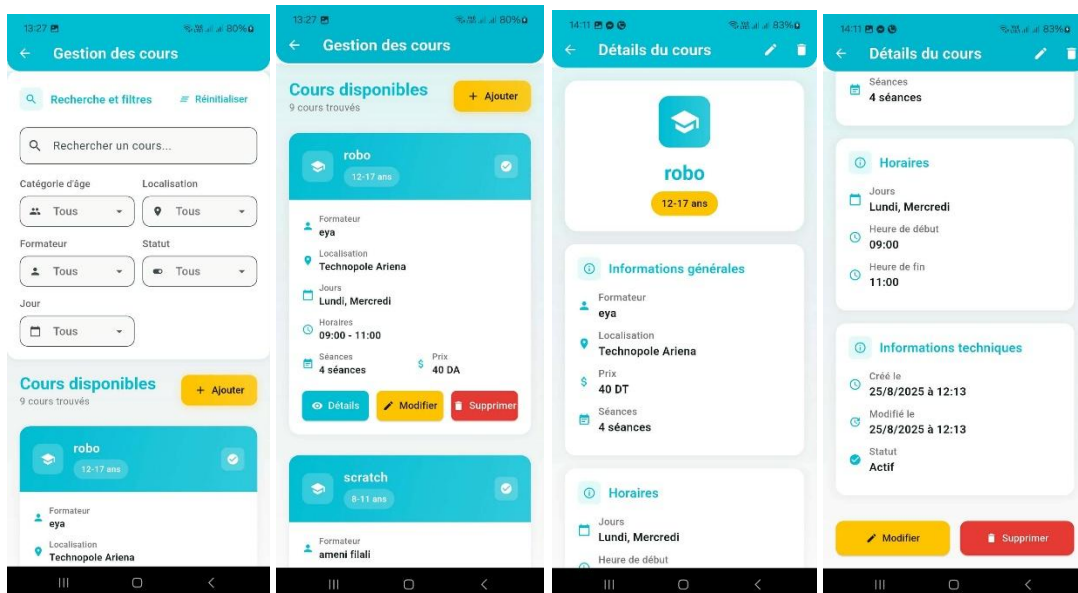
La Figure 4.9 présente le module d'enregistrement des présences par séance avec suivi automatique des absences de chaque enfant.

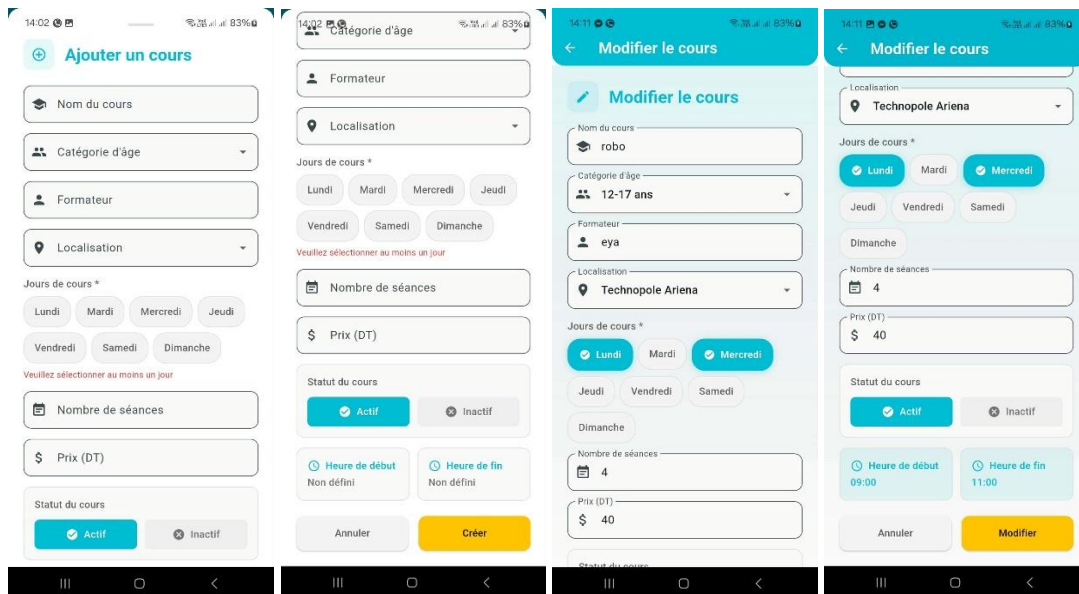


La Figure 4.9 : gestion des absences

4.5.5 Gestion des cours

La Figure 4.10 illustre le module de création et gestion des cours incluant le formateur, la localisation, les horaires et le prix.

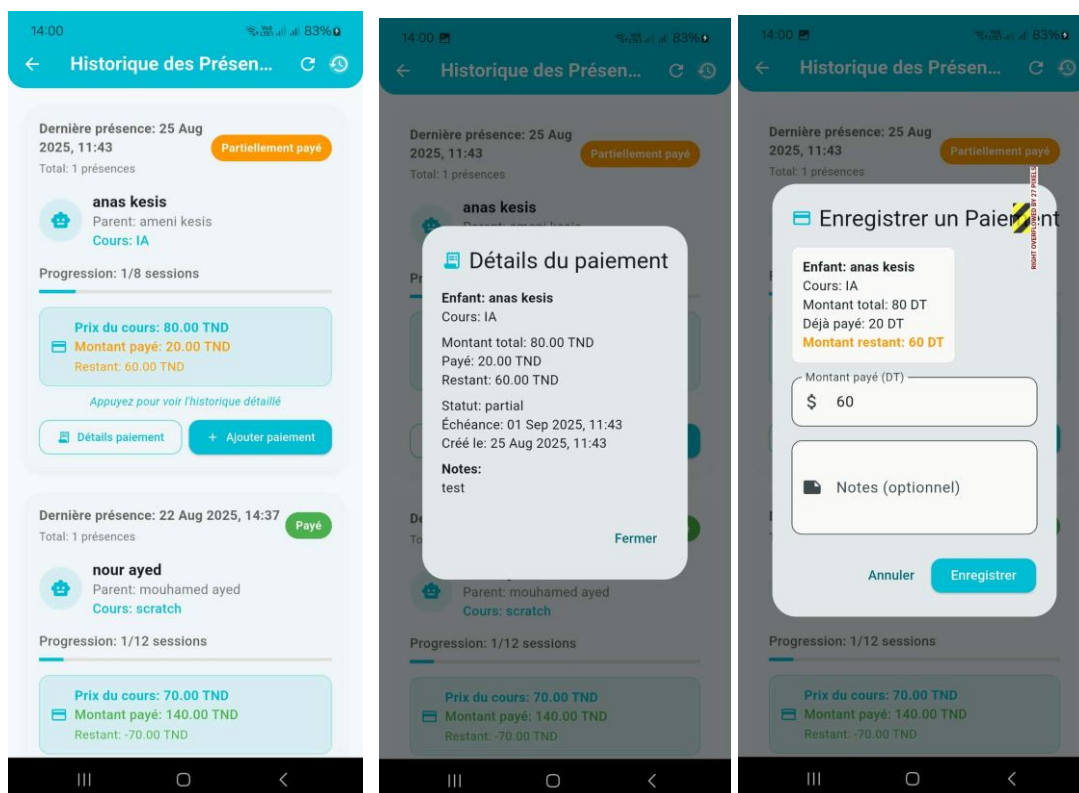




La Figure 4.10 : Gestion Cours

4.5.6 Gestion des paiements

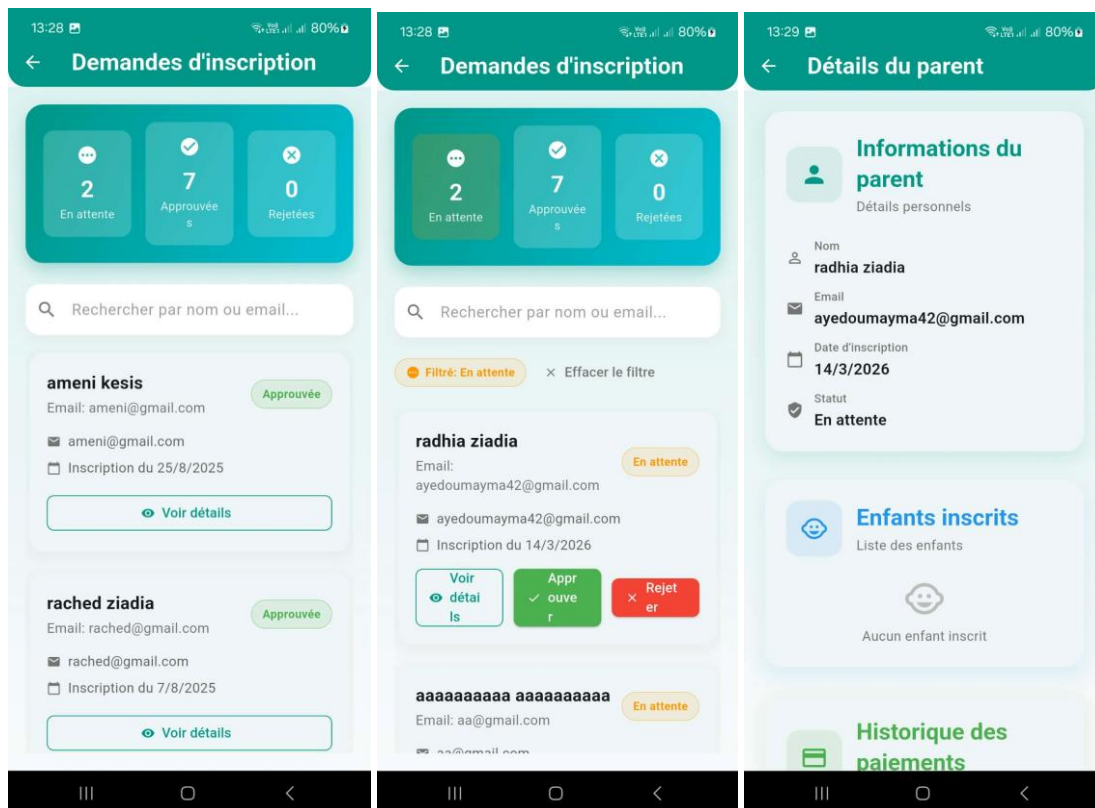
La Figure 4.11 présente le suivi des versements avec affichage du montant payé, du montant restant et du statut de chaque paiement.



La Figure 4.11 : Gestion paiements

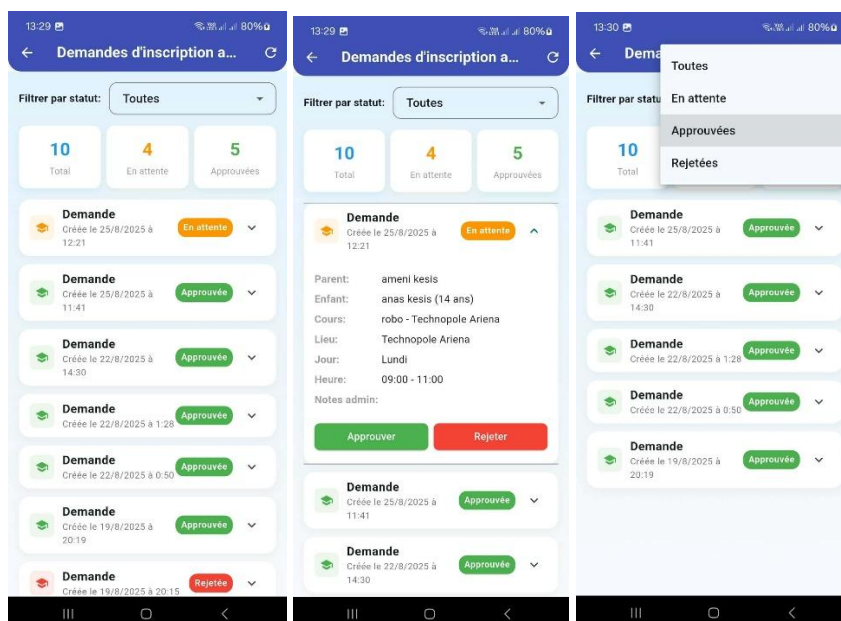
4.5.7 Gestion des demandes d'inscription

La Figure 4.12 illustre le traitement des demandes d'inscription par l'administrateur avec filtrage par statut (en attente, approuvée, rejetée).



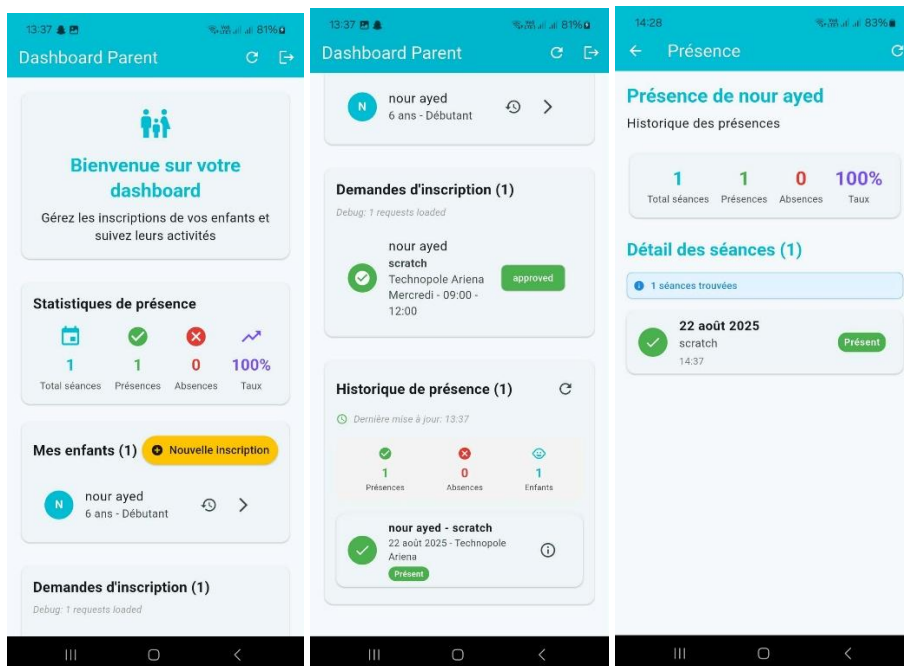
La Figure 4.12 : Gestion des demande d'inscription

Demande d'inscription des enfants aux cours



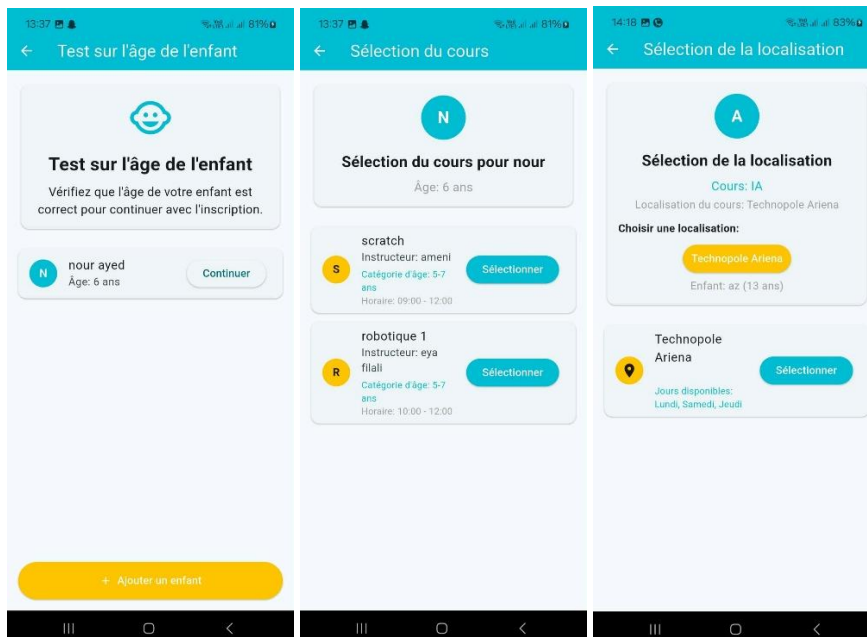
4.5.8 Espace parent

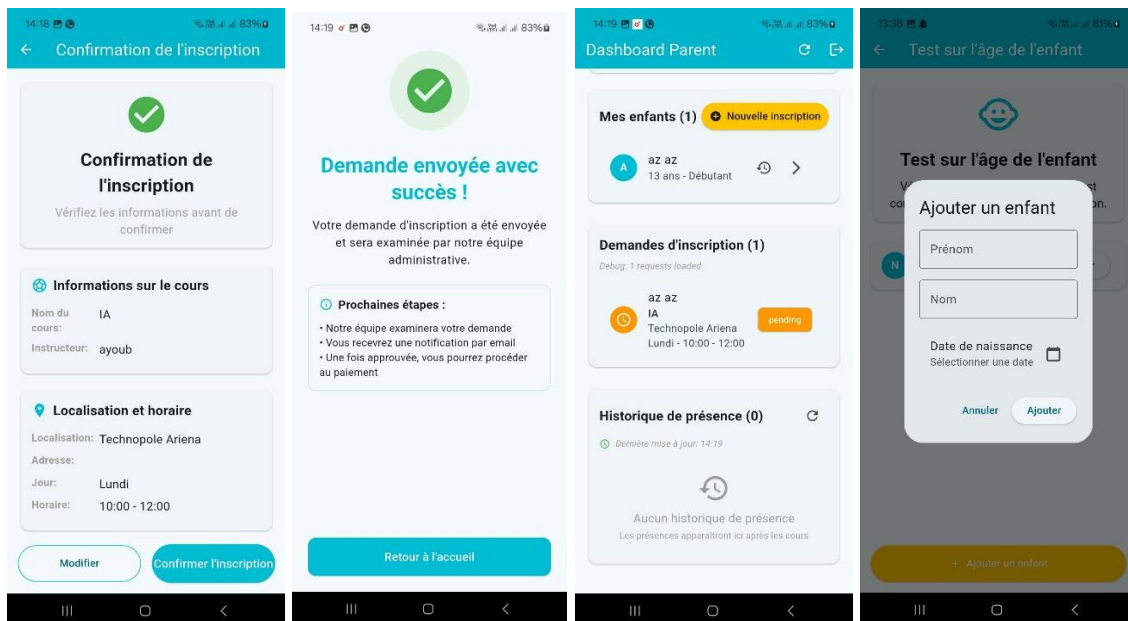
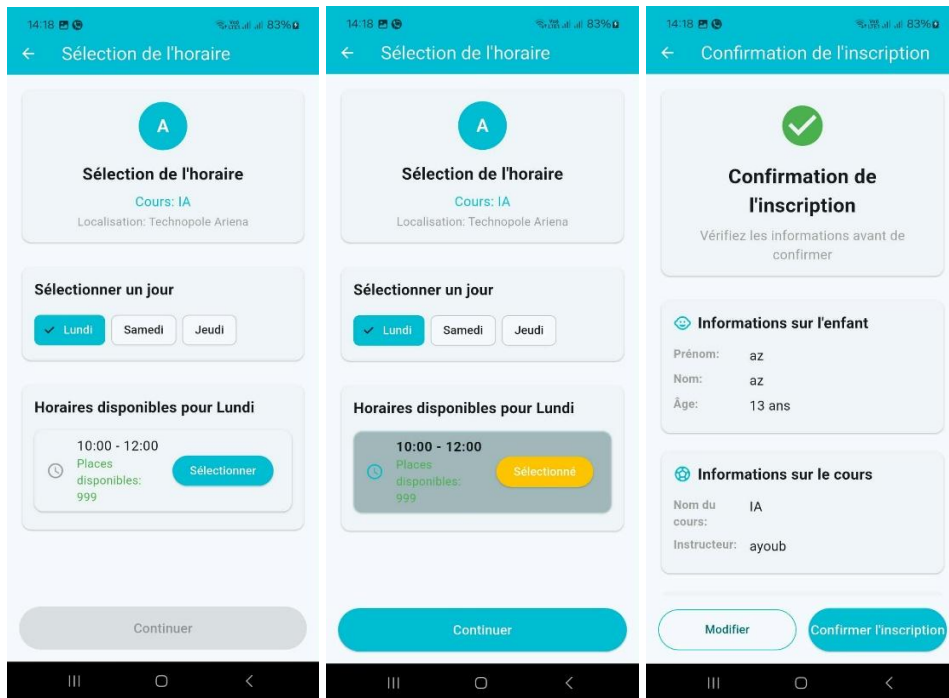
La Figure 4.13 présente le tableau de bord parent affichant les statistiques de présence, les demandes en cours et le processus d'inscription d'un enfant à un cours en cinq étapes.



La Figure 4.13 : Dashboard Parent

Demande d'inscription d'un enfant a un cours suivant des étapes





Ajouter un deuxième enfant par un parent

4.6 Conclusion

La phase de réalisation a permis de matérialiser la vision initiale du projet en une application fonctionnelle, robuste et sécurisée. Les fonctionnalités clés ont été implémentées avec succès, ouvrant la voie à l'exploitation effective de la solution par le club ACA Robotique et ses adhérents.

Conclusion générale et perspectives

Ce projet a permis de concevoir et de développer une application numérique destinée à faciliter la gestion des activités du club **ACA Robotics**. L'objectif principal était de proposer une solution permettant de centraliser la gestion des inscriptions, des enfants, des cours ainsi que le suivi des présences, tout en offrant une interface simple et accessible aux administrateurs et aux parents.

Au cours de ce travail, nous avons analysé les besoins du système, conçu l'architecture de l'application et implémenté une solution basée sur **Flutter** et **Firebase**, permettant de développer une application **cross-platform accessible sur le web**. Cette solution contribue à améliorer l'organisation des activités du club et à simplifier la gestion des informations liées aux participants.

Cependant, certaines fonctionnalités prévues initialement n'ont pas pu être finalisées durant la période du stage. Dans cette perspective, plusieurs améliorations peuvent être envisagées dans les versions futures de l'application. Parmi les perspectives principales figurent **l'intégration d'un module de messagerie** permettant d'améliorer la communication entre les administrateurs et les parents, ainsi que **l'implémentation d'un système de paiement en ligne** pour faciliter la gestion des frais de formation.

À plus long terme, d'autres évolutions pourraient également être envisagées, telles que l'ajout d'un système de notifications, l'amélioration de l'interface utilisateur ou encore le déploiement de l'application sur des plateformes mobiles.