

Introduction générale

La gestion des jardins d'enfants est une tâche complexe et exigeante qui nécessite une coordination efficace entre les animateurs, les parents, le personnel administratif et les autorités locales. Cependant, de nombreuses difficultés entravent cette coordination et peuvent affecter la qualité de l'enseignement et la sécurité des enfants.

Parmi les principales difficultés rencontrées dans la gestion des jardins d'enfants, on peut citer la difficulté à assurer une communication fluide et efficace entre les différents acteurs impliqués, la gestion des effectifs des enseignants et des élèves, la gestion des horaires et des activités pédagogiques, ainsi que la gestion des ressources financières.

C'est dans ce contexte que l'application de gestion des jardins d'enfants a été développée. Cette application vise à faciliter la gestion quotidienne des jardins d'enfants en offrant des solutions pratiques pour surmonter les défis mentionnés ci-dessus. Elle offre également la possibilité de centraliser toutes les informations relatives à la gestion des jardins d'enfants dans un seul et même endroit, ce qui facilite l'accès à ces informations pour tous les acteurs impliqués.

Au cours de ce rapport, nous allons examiner plus en détail les différentes fonctionnalités de cette application ainsi que les avantages qu'elle offre pour la gestion des jardins d'enfants. Nous allons également évoquer les défis rencontrés lors de sa conception et de son déploiement, ainsi que les opportunités futures pour améliorer encore davantage son efficacité.

Nous rappelons dans un premier temps le contexte général de ce projet et présentons par la même occasion l'organisme d'accueil. Le deuxième chapitre intitulé "Analyse et spécification des besoins" s'intéresse à l'analyse technico-fonctionnelle globale ainsi que la définition des différents besoins et la conception des diagrammes. Ensuite, le chapitre 3 "Réalisation" a pour vocation la présentation de notre environnement de travail et souligne

Introduction générale

par la suite l'architecture logique et physique sur lesquelles repose notre solution. Pour finir, le rapport se termine par une synthèse globale qui identifie les points d'amélioration possibles de notre application ainsi qu'une conclusion générale.

Chapitre I : CONTEXTE GÉNÉRAL

Introduction:

Dans ce chapitre, nous commençons par présenter l'organisme d'accueil et ses différents services. Nous examinons ensuite les solutions actuellement disponibles et en dressons une critique afin de déterminer notre problématique et de proposer une éventuelle solution. Enfin, nous identifions la méthodologie de gestion de projet qui sera adoptée.

1. Cadre général du projet:

Ce stage de fin d'études s'est déroulé au sein de l'entreprise Tinkerview, dans le cadre de l'obtention de mon diplôme de licence en développement web et multimédia. Pendant une période de 4 mois, j'ai eu l'opportunité de travailler sur un projet spécifique, à savoir la création de la version mobile d'une application web existante.

Au sein de Tinkerview, nous avons été chargé de développer la version mobile de l'application web AsterKid, une plateforme de gestion destinée aux établissements de la petite enfance. Mon rôle était de concevoir et de mettre en œuvre une interface utilisateur mobile conviviale, offrant aux utilisateurs la possibilité de communiquer avec les acteurs de l'application, de visualiser l'état des paiements des parents et de consulter le planning, depuis leur téléphone portable .

Pendant ce projet, j'ai travaillé en étroite collaboration avec l'équipe de développement web existante pour assurer la cohérence avec la version web déjà établie. J'ai été impliqué dans toutes les étapes du processus de développement, depuis la conception initiale jusqu'à la réalisation et les tests finaux. J'ai utilisé des technologies mobiles modernes et des pratiques de développement adaptées pour créer une expérience utilisateur fluide et réactive sur les appareils mobiles.

2. Entreprise d'accueil:

Dans cette partie, nous allons présenter et citer les services de l'organisme d'accueil de ce stage de fin d'études.

2. 1. Présentation de l'organisme Tinkerview:

Tinkerview est une jeune entreprise SARL créée en novembre 2022 par deux fondateurs, Mohamed Amine Ziraoui et Amel Barbou. La mission de l'entreprise consiste à créer des projets innovants répondant à des besoins spécifiques dans différents domaines, dans le but de mettre la technologie au service de l'écosystème tunisien et international. Tinkerview a obtenu le label startup en Tunisie et le pré-label en France en janvier 2023, ce qui témoigne de son potentiel et de son engagement à innover. Actuellement, l'entreprise travaille sur deux produits prometteurs : AsterKid et Bilancia, qui visent à répondre à des besoins spécifiques dans différents domaines.

2.2. Présentation de l'application:

AsterKid est l'application sur laquelle je vais travailler pendant le stage, qui est une plateforme de gestion destinée aux établissements de la petite enfance tels que les jardins d'enfants, les garderies scolaires et les crèches.

Durant mon stage je vais travailler sur la version mobile de Asterkid qui, permettra aux utilisateurs de communiquer entre eux grâce à des fonctionnalités de messagerie intégrées. Les parents peuvent facilement contacter le personnel de l'établissement pour poser des questions, demander des mises à jour sur leur enfant ou signaler toute situation particulière. De même, l'établissement peut envoyer des notifications aux parents pour les informer de toute évolution importante, de nouveaux événements ou de changements dans le planning. Cette communication transparente et fluide entre les parents et l'établissement est essentielle

pour garantir le bien-être et la sécurité des enfants, tout en assurant la satisfaction et la confiance des parents vis-à-vis de l'établissement.

3. État de l'existant:

Dans cette section du chapitre, nous présenterons un tableau comparatif des différentes solutions existantes afin d'identifier et d'analyser leurs fonctionnalités distinctes. Notre recherche a révélé un besoin criant au sein des organismes de la petite enfance, où les femmes directrices consacrent beaucoup de temps à des tâches répétitives qui leur font perdre un temps précieux et nuisent à la qualité de leur travail.

Une observation importante que nous avons faite est que les directrices perdent de l'argent chaque année en raison d'un suivi inefficace des paiements, ainsi que de l'utilisation de méthodes traditionnelles peu fiables. Ces problèmes peuvent entraîner des retards de paiement, des erreurs dans les transactions financières et une perte de confiance de la part des parents et des parties prenantes.

3.1. Solution existante dans le marché:

Technologie ----- Solution	SAAS	Application mobile	Communication Instantanée	Gestion financière
AkurSchool	✓	✓	✗	✓
CHOSA	✓	✓	✗	✗
GSA	✓	✓	✗	✓
Meeko	✓	✓	✓	✓
Illumine	✓	✓	✗	✓

3.2.AsterKid application Web:

Les établissements de la petite enfance, tels que les crèches, les jardins d'enfants et les garderies scolaires, utilisent souvent des méthodes de gestion traditionnelles telles que des tableurs et des registres papier pour gérer leurs finances, leurs horaires et les informations relatives aux enfants et aux parents. Cependant, ces méthodes peuvent être fastidieuses et peu pratiques. Par conséquent, il est essentiel d'étudier l'existant afin d'identifier les forces et les faiblesses des méthodes de gestion actuelles, dans le but de proposer une solution adaptée.

Concernant notre application spécifique, celle-ci est conçue sous la forme d'une plateforme web dédiée aux directeurs ou à l'administration des établissements de la petite enfance. Cette version web offre un accès pratique et sécurisé aux fonctionnalités de gestion à partir de tout appareil connecté à Internet, tels que les ordinateurs de bureau, les ordinateurs portables et les smartphones. Elle élimine ainsi la dépendance aux tableurs ou aux registres papier, offrant

Chapitre 1: CONTEXTE GÉNÉRAL

une solution plus moderne et efficace pour gérer les finances, les horaires et les informations relatives aux enfants et aux parents. L'objectif d'AsterKid est de faciliter la gestion interne pour les directeurs ou directrices, en assurant le suivi financier, les tâches quotidiennes et la gestion des employés. En effet, AsterKid permet un suivi efficace des finances grâce à la tenue d'un livre des comptes, le suivi des paiements, des dépenses, des recettes et la facturation. La plateforme simplifie également les tâches quotidiennes du personnel de l'établissement en permettant la gestion des cours, des présences, des activités et des clubs.

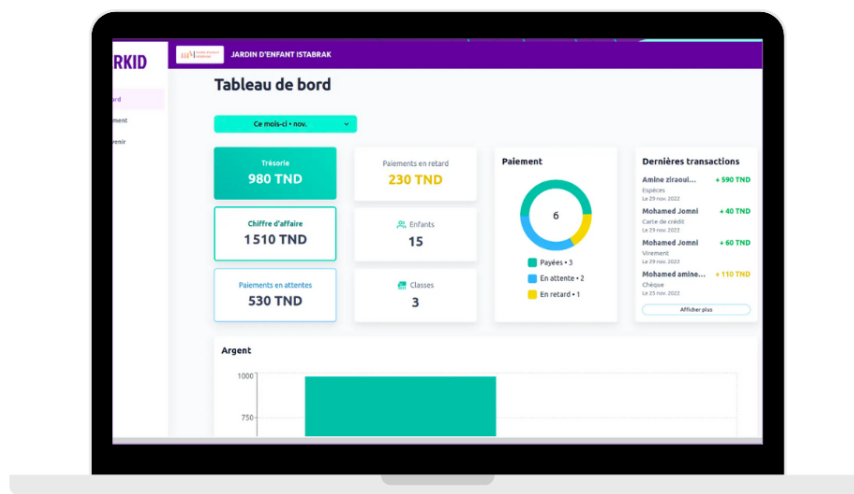


Figure1: tableau de bord AsterKid web

3.3.Critique de l'existant :

En général, pour assurer la confidentialité des informations des utilisateurs, il est indispensable que toute application, y compris notre projet AsterKid, prenne en compte certains prérequis essentiels. Ces prérequis comprennent une gestion efficace des informations de connexion, des données biométriques et de la communication à travers des canaux sécurisés.

Contrairement aux autres solutions présentées dans le Tableau 1, notre solution asterKid a intégré toutes les fonctionnalités essentielles nécessaires pour garantir une communication hybride, rapide et sécurisée entre les acteurs du jardin d'enfants. Notre application mobile

offre une gestion optimale des informations de connexion, permettant aux utilisateurs d'accéder de manière sécurisée à leurs comptes personnels.

De plus, AsterKid assure la confidentialité des données biométriques, en mettant en place des protocoles de sécurité avancés pour protéger ces informations sensibles.

En ce qui concerne la communication, AsterKid propose des canaux sécurisés, empêchant toute interception ou accès non autorisé aux conversations entre les différents acteurs du jardin d'enfants.

Il est important de souligner que notre solution AsterKid est la seule application selon mes recherches à l'échelle nationale qui propose cette approche complète et répond à toutes les exigences de confidentialité établies par les autorités compétentes. Nous avons également obtenu tous les certificats de confidentialité requis, garantissant ainsi la conformité de notre application avec les normes de sécurité établies

3.4. Forces de l'existant :

- Simplification de la gestion financière : l'application permet de tenir un livre de comptes, de suivre les paiements, les dépenses, les recettes et de facturer les parents de manière simple et rapide.
- Optimisation de la planification des tâches : AsterKid permet de gérer les cours, les présences, les activités et les clubs, ainsi que de planifier efficacement les tâches quotidiennes du personnel de l'établissement.
- Facilitation de la gestion des employés : l'application permet de suivre le planning, la présence et les congés des employés, ainsi que de gérer les fiches de paie et les données administratives.
- Offre d'une interface mobile conviviale : pour faciliter la communication entre les différents acteurs des établissements de la petite enfance.

Conclusion:

Au cours de ce premier chapitre, nous avons présenté l'entreprise d'accueil et les solutions qu'elle propose à sa clientèle, ainsi qu'une étude de l'existant dans le marché des applications de gestion de jardin d'enfants en Tunisie. Cette étude nous a permis de déterminer les objectifs à atteindre avec notre solution et de proposer des améliorations pour remédier aux lacunes des solutions existantes. Le rapport mettra également en évidence la méthodologie et le formalisme de modélisation qui ont été adoptés tout au long de la conception du projet. Ces aspects seront abordés plus en détail dans le chapitre suivant, où nous examinerons les approches méthodologiques utilisées pour la planification, la conception et le développement de notre application.

Chapitre II : Modélisation Conceptuelle

Introduction:

Le Modèle conceptuel de données est un élément statique de représentation du système d'information. Son objectif principal est de fournir une représentation claire et cohérente des données manipulées dans le système d'information. Ce Chapitre sera structuré de la manière suivante : nous commencerons par justifier et expliquer notre choix de la méthodologie de conception. Ensuite, nous présenterons l'étude des besoins, et enfin nous identifierons les différents acteurs ainsi que les diagrammes des cas d'utilisation et le diagramme de classe.

1. Méthodologie adoptée:

Le bon choix de la méthode de gestion de projet est nécessaire pour garantir le bon déroulement du projet et aussi pour être vigilant par rapport aux différentes exigences des clients.

1.1. Méthodologie:Scrum :

La méthodologie se réfère à une démarche logique qui suit des méthodes organisationnelles pour obtenir des résultats. Notre choix de méthode de gestion de projet est l'approche agile Scrum. Scrum est une méthode de gestion de projet empirique qui se concentre sur une livraison rapide et régulière. Cette méthode offre divers avantages tels que la transparence de l'état d'avancement du projet pour toute l'équipe, la décomposition des projets complexes en sprints facilement gérables, un meilleur contrôle sur le produit final et la participation du client à la définition et à l'évolution des fonctionnalités.

1.2. Les avantages de la méthode Scrum :

- Visibilité et transparence : Les revues de sprint régulières et les indicateurs de performance permettent une visibilité claire sur l'avancement du projet, facilitant ainsi la détection précoce des problèmes et des retards.
- Découpage en sprints : La découpe du projet en sprints plus courts et gérables facilite la planification, le suivi de l'avancement et la gestion des risques en se concentrant sur des objectifs spécifiques à court terme.
- Implication du client : La participation active du client dans la définition et l'évolution des fonctionnalités garantit que le produit final répondra à ses besoins, assurant ainsi une plus grande satisfaction client.
- Contrôle du produit final : Les équipes de développement ont un plus grand contrôle sur le produit final grâce à une approche itérative et des retours réguliers du client, permettant ainsi des ajustements et des améliorations continus.
- Collaboration et communication : Scrum favorise la collaboration et la communication entre les membres de l'équipe, encourageant une approche ouverte, flexible et autonome qui améliore la cohésion et l'efficacité de l'équipe. [1]

Thème	Approche traditionnelle	Scrum
Flexibilité	La méthodologie traditionnelle suit généralement un plan défini dès le début du projet, avec des étapes linéaires et une portée globale fixe.	Scrum est une méthodologie agile qui met l'accent sur l'adaptabilité et la flexibilité. Les projets Scrum sont découpés en itérations appelées "sprints" et permettent d'ajuster les priorités

		et les fonctionnalités tout au long du projet
Communication et collaboration	La méthodologie traditionnelle met souvent l'accent sur une communication formelle et hiérarchique, principalement par le biais de rapports et de réunions planifiées.	Scrum favorise la communication et la collaboration régulières entre les membres de l'équipe de projet grâce à des réunions fréquentes, comme les réunions quotidiennes de synchronisation.
Gestion du risque	La méthodologie traditionnelle suit une approche plus linéaire, ce qui signifie que les risques sont souvent identifiés plus tard dans le processus, ce qui rend les ajustements plus difficiles à réaliser.	Scrum est conçu pour identifier et atténuer les risques dès le début du projet grâce à une approche itérative. Les problèmes sont détectés rapidement et des ajustements peuvent être apportés en conséquence.
Livraison incrémentale	La méthodologie traditionnelle vise souvent à livrer l'ensemble du projet d'un seul tenant, ce qui peut entraîner des délais plus longs avant que les utilisateurs ne puissent bénéficier des fonctionnalités.	Scrum encourage la livraison régulière d'incréments de fonctionnalités tout au long du projet, ce qui permet de recueillir les commentaires des utilisateurs et de s'adapter rapidement aux changements.

Gestion du changement	La méthodologie traditionnelle, en revanche, est moins adaptable aux changements, car elle suit un plan établi dès le départ.	Scrum reconnaît que les besoins et les exigences des utilisateurs peuvent évoluer au fil du temps. Il offre donc une plus grande souplesse pour gérer les changements de portée.
Contrôle du projet	La méthodologie traditionnelle est souvent plus structurée et hiérarchique, avec une supervision plus étroite de la part des chefs de projet ou des gestionnaires.	Scrum met l'accent sur l'autonomie et la responsabilité de l'équipe de projet, qui est auto-organisée et prend des décisions collectives

1.3. Les moyens de communications

Slack :

Dans le cadre de mon stage, nous avons utilisé Slack comme principal moyen de communication au sein de l'équipe. Cette plate-forme s'est révélée très efficace pour nous permettre de collaborer en temps réel et de partager des informations importantes sur les projets en cours. Grâce à cette solution, nous avons pu communiquer rapidement et facilement, peu importe où nous nous trouvions. En outre, nous avons organisé des réunions hebdomadaires pour discuter des réalisations de la semaine et des objectifs à venir, et Slack nous a permis de faciliter la communication avant, pendant et après chaque réunion en envoyant des rappels et en partageant des fichiers et des informations importantes. En somme, l'utilisation de Slack a grandement contribué à la réussite de notre collaboration en équipe.



Figure 2 : Logo de Slack

1.4. Backlog de produit:

ID	Sprint	Tâche à faire	Durée
1	1	Mise en place de l'architecture, de l'environnement de travail et des écrans de base: <ul style="list-style-type: none"> - Mettre en place l'architecture de base du projet - Créer l'écran d'accueil avec une interface conviviale - Développer les écrans de connexion et d'inscription 	15 jours
2	2	Module Home Page et Messagerie instantanée <ul style="list-style-type: none"> - Créer le composant "Header" pour afficher la barre de navigation - Créer le composant "Footer" pour afficher le pied de page - Mise en place du composant de file d'actualité - Mise en place de la fonctionnalité de publication de messages importants - Conception de l'interface utilisateur pour la file d'actualité et les messages importants - Tests et débogage pour assurer le bon fonctionnement des fonctionnalités - Créer la liste des conversations - Implémenter l'affichage des types de conversation - Gérer la sélection d'une conversation - Afficher les détails des conversations 	28 jours
3	3	Module de paiement pour les parents et Planification:: <ul style="list-style-type: none"> - Créer l'interface de visualisation des paiements - Afficher les listes de paiement pour permettre aux parents de visualiser un récapitulatif des paiements - Envoyer une demande de validation de paiement - Créer l'interface pour afficher le calendrier ou la planification - Implémenter les vues du planning : hebdomadaire , mensuelle .. - Afficher les cours et activités avec les informations essentielles 	18 jours

2. Modélisation :

2.1. Formalisme de modélisation

Pour identifier les besoins, nous avons choisi le langage de modélisation UML qui permet de décrire et visualiser nos besoins et notre architecture. Les diagrammes nécessaires pour la conception du projet sont :

- Diagramme de cas d'utilisation pour une représentation du comportement fonctionnel de notre système logiciel.
- Diagramme de classes pour définir la structure globale de l'application.
- Diagramme de séquences pour présenter les interactions entre le système et les acteurs.

2. Étude des besoins

2.1. Identification des besoins fonctionnels

Module Onboarding :

- Processus d'inscription pour les nouveaux parents
- Processus d'inscription pour les nouveaux enfants

Module Authentification :

- Connexion sécurisée pour les parents et les enseignants

Module Paiement :

- Paiement en ligne des frais de garderie

Module Communication :

- Messagerie instantanée entre les parents, les enseignants et la direction
- Envoi de photos et de vidéos

Module Planning :

- Gestion du planning des activités pour les enfants
- Consultation du planning pour les parents

2.2. Identification des besoins non fonctionnels:

- **Sécurité :**

Assurer la confidentialité et la sécurité des données sensibles des enfants, des parents et du personnel du jardin d'enfants. Cela peut inclure la protection des informations personnelles, l'authentification et l'autorisation sécurisées, ainsi que des mesures de sécurité robustes pour prévenir les intrusions et les accès non autorisés.

- **Performance :**

Pour optimiser les performances de notre application, nous avons mis en place plusieurs stratégies. Tout d'abord, nous avons utilisé des composants fonctionnels plutôt que des composants de classe, car ils sont plus performants.

Nous avons également minimisé les ré-renderings inutiles des composants en utilisant la méthode `useMemo`. Cette méthode nous a permis d'éviter les mises à jour coûteuses du DOM lorsque les props ou l'état ne changent pas, améliorant ainsi les performances globales de l'application.

En combinant ces différentes techniques d'optimisation, nous avons pu garantir des performances optimales pour notre application .

- **Fiabilité :**

Pour l'hébergement de notre application, nous avons opté pour une solution de cloud computing. Les services de cloud computing, tels que Amazon Web Services (AWS), Google Cloud Platform (GCP) ou Microsoft Azure, offrent des infrastructures solides et évolutives pour héberger des applications.

En utilisant le cloud, nous bénéficions de plusieurs avantages en termes de fiabilité. Les fournisseurs de cloud disposent de centres de données hautement sécurisés et redondants, répartis géographiquement, ce qui réduit les risques de temps d'arrêt dus à des pannes matérielles ou à des catastrophes naturelles. De plus, ils proposent généralement des services de sauvegarde et de reprise après sinistre pour assurer la disponibilité continue de l'application en cas de problèmes.

En choisissant un fournisseur de cloud réputé, nous avons accès à des services de surveillance et de gestion des performances avancés. Ces services nous permettent de surveiller en temps réel les performances de l'application, d'identifier les goulots d'étranglement et les problèmes potentiels, et de prendre des mesures préventives pour maintenir la fiabilité de l'application.

- **Convivialité et facilité d'utilisation :**

Pour assurer la convivialité et la facilité d'utilisation de notre application, nous avons opté pour une plateforme d'utilisation qui ressemble aux réseaux sociaux que les utilisateurs utilisent au quotidien. En nous inspirant des interfaces familières et intuitives des réseaux sociaux, nous avons créé une expérience utilisateur conviviale et facile à prendre en main pour les parents, les enseignants et l'administration du jardin d'enfants.

Nous avons conçu une interface utilisateur attractive avec une disposition claire et une utilisation judicieuse des éléments visuels tels que les icônes, les couleurs et les images. Cela permet aux utilisateurs de naviguer facilement dans l'application et de trouver rapidement les fonctionnalités dont ils ont besoin.

Nous avons également accordé une attention particulière à l'ergonomie de l'application. Nous avons organisé les fonctionnalités de manière logique et hiérarchisée, en tenant compte des besoins et des flux de travail spécifiques des différents profils d'utilisateurs. Cela garantit que les utilisateurs peuvent accéder facilement aux informations et aux actions pertinentes, sans confusion ni difficulté.

- **Accessibilité :**

Veiller à ce que l'application soit accessible à tous les utilisateurs, y compris ceux ayant des besoins spéciaux ou des limitations physiques. Cela peut inclure la prise en charge de fonctionnalités d'accessibilité, telles que la compatibilité avec les lecteurs d'écran, la taille de police réglable et les couleurs contrastées pour faciliter la lisibilité.

- **Extensibilité et évolutivité :**

Concevoir l'application de manière à ce qu'elle puisse être facilement étendue et évoluer avec les besoins futurs du jardin d'enfants. Cela implique une architecture modulaire, une conception flexible et une gestion efficace des données afin de permettre l'ajout de nouvelles fonctionnalités et la prise en charge d'une croissance future du nombre d'utilisateurs.

- **Support multiplateforme :**

Assurer une compatibilité et une fonctionnalité cohérentes sur différentes plateformes mobiles, telles qu'Android et iOS, pour permettre aux utilisateurs d'accéder à l'application quel que soit le type de smartphone ou de tablette qu'ils utilisent.

3. Modélisations des besoins :

Dans le processus de conception de notre application, nous avons choisi d'utiliser le langage de modélisation graphique UML (Unified Modeling Language) en raison de sa capacité à fournir une représentation claire et cohérente des données manipulées dans le système d'information, ainsi que pour sa compatibilité avec la conception orientée objet. UML permet de spécifier, visualiser, modifier et construire les documents nécessaires au développement d'un logiciel orienté objet, en fournissant un standard de modélisation pour représenter l'architecture logicielle. Les outils de modélisation UML permettent également de générer automatiquement une partie du code et de représenter diverses projections grâce aux vues, constituées d'un ensemble de diagrammes de vue statique et de vue dynamique. La conception de notre interface a été élaborée en suivant une démarche structurée et rigoureuse, en utilisant les diagrammes de cas d'utilisation, de classe et de séquence pour identifier les acteurs et les interactions avec le système, ainsi que pour représenter la structure et le fonctionnement du système.

La conception de notre interface a été élaborée en suivant la démarche suivante :

3.1. Identification des acteurs :

Un acteur représente l'abstraction d'un rôle joué par des entités externes qui interagissent directement avec le système, comme une personne ou un robot.

Pour notre application, il y aura quatre acteurs :

- Les employés de l'établissement : ce sont les utilisateurs secondaires de l'application, tels que les enseignants, les éducateurs, les animateurs, les cuisiniers, les aides-soignants, qui utilisent l'application pour gérer leur planning, les activités, les présences des enfants, les notes, etc.
- Les parents d'élèves : ils sont les utilisateurs finaux de l'application, et peuvent utiliser l'application pour accéder aux informations sur leurs enfants, leur facturation, leurs présences, les activités, les notes, les communications avec l'établissement, etc.

Ces différents acteurs ont des besoins différents en termes de fonctionnalités de l'application, et il est important de tenir compte de leurs besoins spécifiques lors du développement de l'application.

3.2. Diagramme de cas d'utilisation global application mobile:

Les fonctionnalités de base doivent être fournies par le système est décrit dans le diagramme de cas d'utilisation global. Le diagramme de cas d'utilisation général de notre application est illustré dans le diagramme ci-dessous.

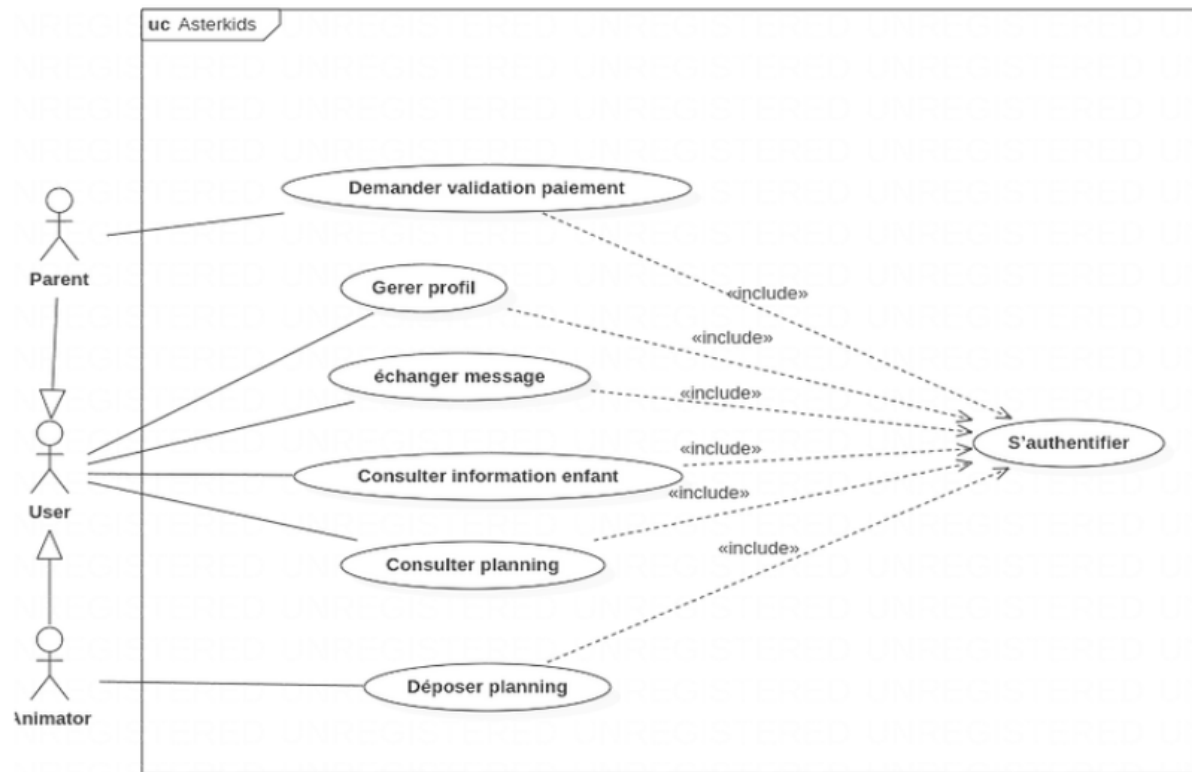


Figure 3 : Cas d'utilisation global

3.3. Diagramme de classes global:

Le diagramme de classes aide à clarifier les nombreuses relations entre les classes de l'application. Nous avons obtenu le diagramme de classes suivant après avoir étudié notre projet :

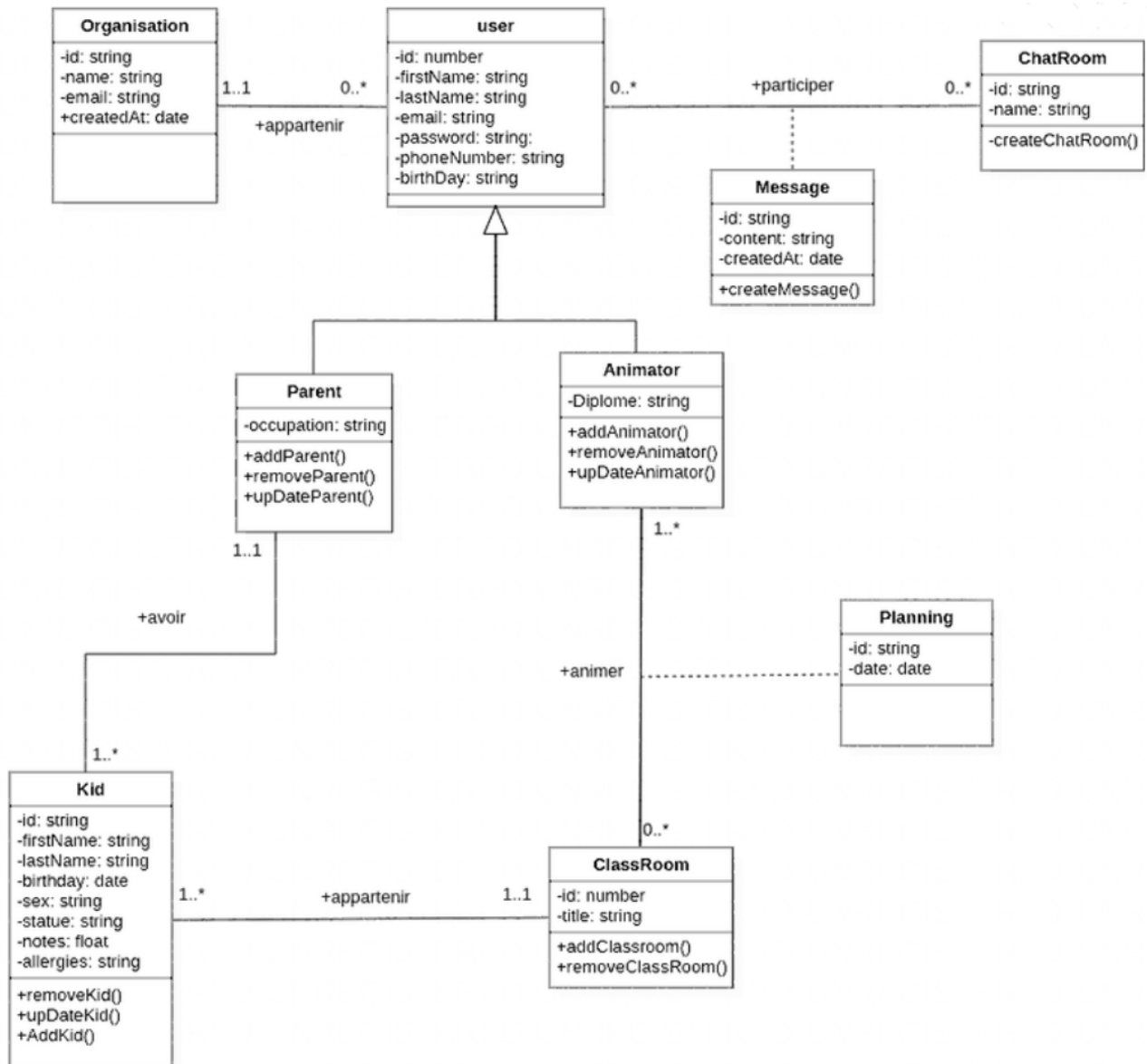


Figure 4: Diagramme de Classes

3.4 Diagramme de séquences:

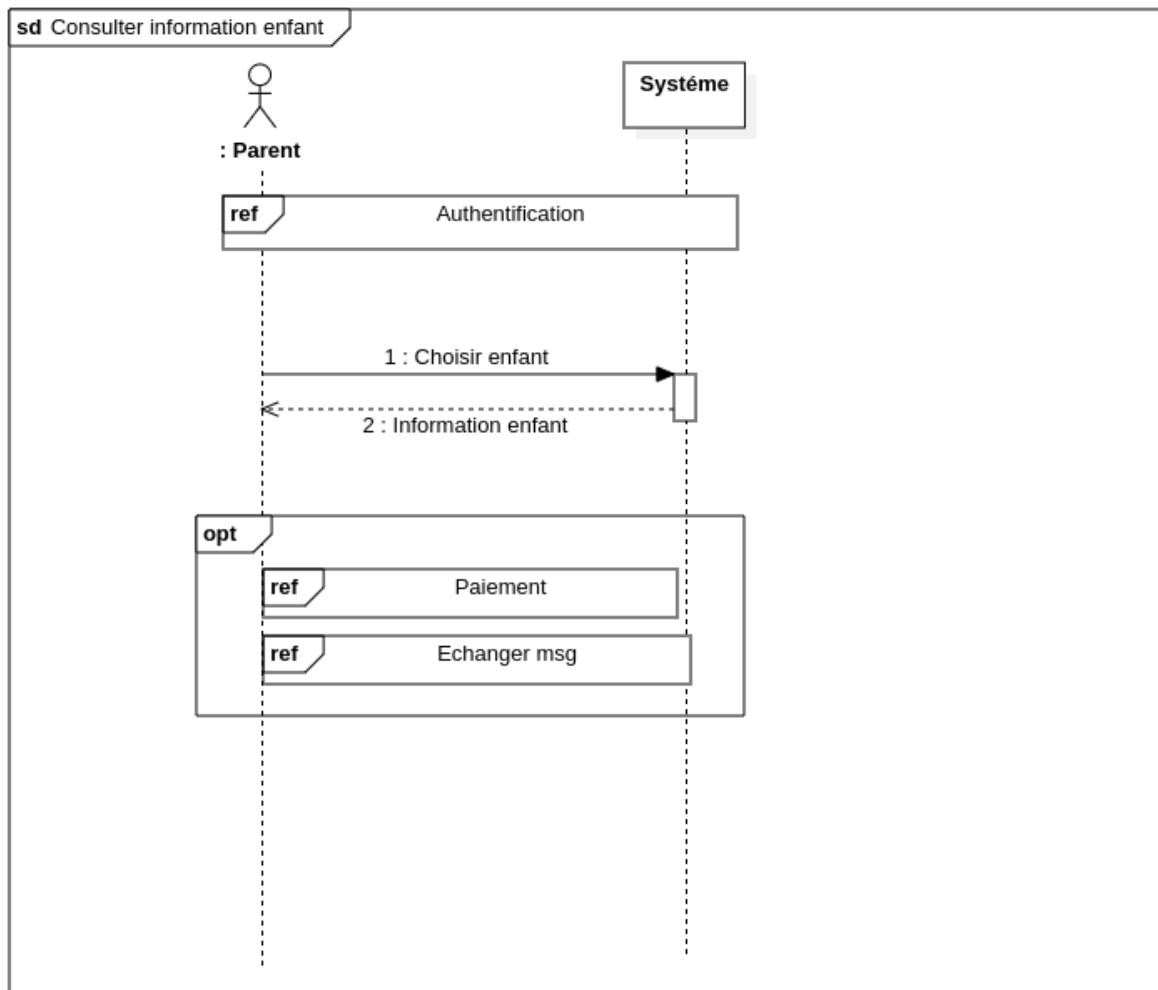


Figure 5: Diagramme de Séquences

4. Architecture physique du projet:

4.1. MVVM pour le Front :

Afin de simplifier et d'accélérer le processus de développement, de mise à jour et de réutilisation du code, le modèle de conception Model-View-ViewModel (MVVM) est utilisé pour structurer le code et segmenter les applications en modules distincts. Ce modèle est composé de trois éléments principaux :

1. Le modèle (Model) : Il contient les données qui seront présentées dans l'application. Cela peut inclure des données provenant de sources externes telles que des API, des bases de données ou d'autres services.
2. La vue (View) : C'est la partie visible de l'application, c'est-à-dire les pages ou les interfaces utilisateur que les utilisateurs interagissent avec. La vue est responsable de l'affichage des données et de la réception des actions de l'utilisateur.
3. Le modèle de vue (ViewModel) : Il joue le rôle d'intermédiaire entre le modèle et la vue. Le ViewModel est responsable de la communication avec le modèle pour récupérer les données nécessaires à l'affichage et de les préparer pour la vue. Il gère également les interactions utilisateur et met à jour le modèle en conséquence.

En utilisant le modèle MVVM, on peut séparer clairement les responsabilités et les préoccupations du code. Cela facilite la maintenance, la mise à jour et la réutilisation du code, car les modifications apportées à un module spécifique n'ont pas d'impact direct sur les autres parties de l'application. De plus, la séparation du modèle et de la vue permet de rendre l'interface utilisateur plus réactive et indépendante du modèle sous-jacent.

Il convient de noter que bien que le modèle MVVM soit un concept général, il peut être mis en œuvre de différentes manières selon le framework ou la technologie utilisée. Dans le cas de React Native, le framework utilise des composants génériques qui correspondent aux vues, et le développement basé sur les principes de MVVM facilite la création de composants réutilisables et la gestion des données dans l'application.

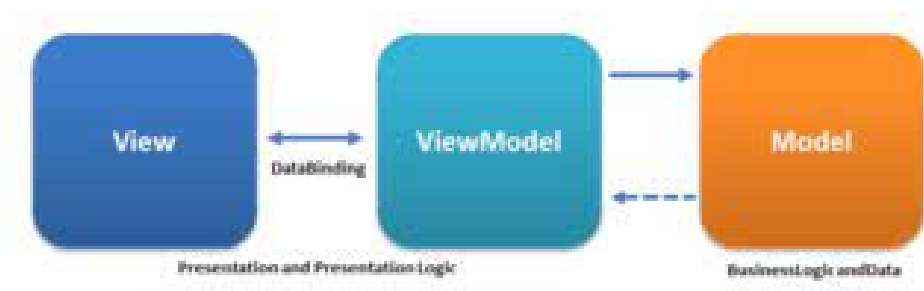


Figure 6: Modèle MVVM

4.2. MVC pour le Back:

Le modèle MVC est un modèle d'organisation du code qui sépare les différentes responsabilités d'une application en trois composants principaux :

1. Le modèle (Model) : Il représente les données et la logique métier de l'application.
Le modèle est responsable de l'accès aux données, de leur manipulation et de leur validation. Il représente la couche qui interagit directement avec la base de données ou d'autres sources de données.
2. La vue (View) : Elle représente l'interface utilisateur de l'application. La vue est responsable de l'affichage des données et de l'interaction avec l'utilisateur. Dans le contexte du back-end, la vue peut être une API REST ou un service qui expose les données aux clients.
3. Le contrôleur (Controller) : Il fait le lien entre le modèle et la vue. Le contrôleur reçoit les requêtes de l'utilisateur à travers la vue, traite ces requêtes en utilisant les fonctionnalités du modèle, puis renvoie les résultats à la vue pour affichage ou transmission au client.

Le modèle MVC facilite la séparation des préoccupations et la modularité du code. Il permet une meilleure organisation et maintenabilité du code en divisant les responsabilités en différents composants. Cela facilite également la réutilisation du code, car les modèles et les contrôleurs peuvent être utilisés dans différents contextes de l'application.

Il est important de noter que le modèle MVC est principalement utilisé pour organiser le code du côté serveur (back-end) dans le cas de NestJS. Pour la partie front-end de l'application, d'autres modèles tels que MVVM (Modèle-Vue-Modèle de vue) ou MVC peuvent être utilisés, en fonction du framework ou de la technologie utilisé

Conclusion :

Dans ce chapitre nous avons posé les bases de la modélisation conceptuelle de notre application mobile pour jardin d'enfants. Il nous a permis de comprendre les besoins fonctionnels et non fonctionnels du système, d'identifier les acteurs impliqués et de

Chapitre 2: : Modélisation Conceptuelle

visualiser l'architecture du système. Ces éléments seront utilisés dans les chapitres suivants pour guider le développement de l'application.

Sprint 1: Mise en place de l'architecture et module d'authentification

Introduction :

Dans ce chapitre, nous présenterons le travail réalisé lors du sprint 1. Nous commencerons par mettre en place le Product Backlog spécifique à ce sprint, puis nous passerons à la spécification fonctionnelle et à la conception. Enfin, nous présenterons les différentes tâches accomplies en illustrant celles-ci à l'aide d'interfaces.

1. Environnement de travail:

1.1. Environnement logiciels:

Visual Studio:

Visual Studio est un environnement de développement intégré (EDI) puissant et polyvalent. Il a été utilisé dans la réalisation de l'application pour 1 et gérer efficacement le code source. Il offre des fonctionnalités avancées telles que l'autocomplétion, le débogage, la gestion des versions et la compilation, ce qui facilite le processus de développement et permet de garantir la qualité du code.[3]



Figure 7: Logo de VS code

React Native Expo:

React Native Expo est un framework qui permet de développer des applications mobiles multiplateformes. Il a été utilisé dans la réalisation de l'application pour concevoir une interface utilisateur réactive et performante. Expo fournit un ensemble d'outils et de bibliothèques qui facilitent le développement d'applications mobiles en utilisant JavaScript et React Native. Cela permet de partager une grande partie du code entre les plateformes iOS et Android, ce qui accélère le processus de développement.[4]



Figure 8: Logo de react native and expo

Node.js:

Node.js est un environnement d'exécution côté serveur basé sur JavaScript. Il a été utilisé pour développer les fonctionnalités côté serveur de l'application. Node.js offre une grande scalabilité et permet de développer des applications Web rapides et évolutives. Grâce à Node.js, il a été possible de mettre en place des opérations côté serveur, telles que le traitement des requêtes, l'accès aux bases de données et la gestion des API.[5]



Figure 9: Logo de node js

Sprint 1 : Mise en place de l'architecture et module authentification

Magnus UI:

Magnus UI est une bibliothèque de composants d'interface utilisateur réutilisables. Elle a été utilisée dans la réalisation de l'application pour simplifier le développement et l'ajustement des éléments visuels. Magnus LUI offre une collection de composants prédéfinis et personnalisables, ce qui permet de gagner du temps et de maintenir une cohérence visuelle dans toute l'application.[6]



Figure 10:Logo de Magnus

Nest:

Nest est un framework de développement d'applications back-end basé sur Node.js. Il a été utilisé pour développer l'API côté serveur de l'application. Nest offre une structure modulaire et une architecture maintenable, ce qui facilite l'organisation et la gestion du code. Il propose également des fonctionnalités telles que la validation des requêtes, la gestion des routes et la gestion des erreurs, ce qui simplifie le processus de développement de l'API.[7]

React Navigation:

React Navigation est une bibliothèque populaire pour la navigation dans les applications mobiles développées avec React Native. Elle fournit des composants et des API permettant de gérer la navigation entre les écrans, la gestion des onglets, les modèles de navigation et bien plus encore.[8]

GitLab:

GitLab est une plateforme DevOps basée sur le web qui fournit un ensemble complet d'outils pour gérer et suivre les dépôts de code source, l'intégration continue (CI) / livraison

continue (CD) et la collaboration de projet. Elle offre une plateforme centralisée pour le contrôle des versions, le suivi des problèmes, les revues de code et la collaboration d'équipe, ce qui facilite le travail en équipe et la livraison efficace de logiciels.

Dans l'ensemble, GitLab fournit une plateforme complète pour gérer l'ensemble du cycle de vie du développement logiciel, de la gestion des versions au déploiement. Il offre une solution unique et intégrée. [9]

React Navigation :

React Navigation est une bibliothèque de gestion de la navigation dans les applications React Native. Elle facilite la création de flux de navigation et la transition entre différentes vues ou écrans dans une application mobile. Avec React Navigation, il est possible de mettre en place des fonctionnalités de navigation telles que la navigation par pile, la navigation par onglets, la navigation par côte, et bien d'autres. Elle offre également des fonctionnalités avancées telles que la gestion des paramètres de navigation, la navigation contextuelle et la gestion des transitions d'animation.[10]

1.2.Environment matériel :

L'environnement matériel utilisé dans la réalisation de l'application comprenait un PC HP équipé d'un processeur Intel Core i3 de 7e génération. Pour améliorer les performances, une mise à niveau matérielle a été effectuée, notamment l'ajout d'une barrette de RAM supplémentaire, portant la capacité totale à 8 Go de mémoire centrale. De plus, un disque SSD a été installé pour accélérer le temps de chargement des applications et des fichiers. En ce qui concerne le système d'exploitation, Ubuntu a été ajouté en parallèle à Windows, offrant ainsi une flexibilité et une compatibilité avec différents environnements de développement. Grâce à ces améliorations matérielles et à la configuration du système, il a été possible de bénéficier d'une plateforme solide et performante pour le développement de l'application.

2. Spécification fonctionnelle :

2.1. Cas d'utilisation Authentification:

Cas d'utilisation	S'authentifier
Acteur	Parent, Animateur
Pré-Condition	L'utilisateur doit être connecté
Post-Condition	L'utilisateur accède à son espace
Enchaînement	<ul style="list-style-type: none">- L'utilisateur ouvre l'interface d'authentification- L'utilisateur saisit son code d'accès et mot de passe.- En cliquant sur « Se connecter » l'utilisateur sera redirigé vers son espace.- Le système vérifie les données saisies.- le système sauvegarde les nouvelles données.
Cas d'exception	<ul style="list-style-type: none">- Champ obligatoire vide.- Saisie erronée.- Compte existant dans le système

2.2. Diagramme de séquences d'authentification :

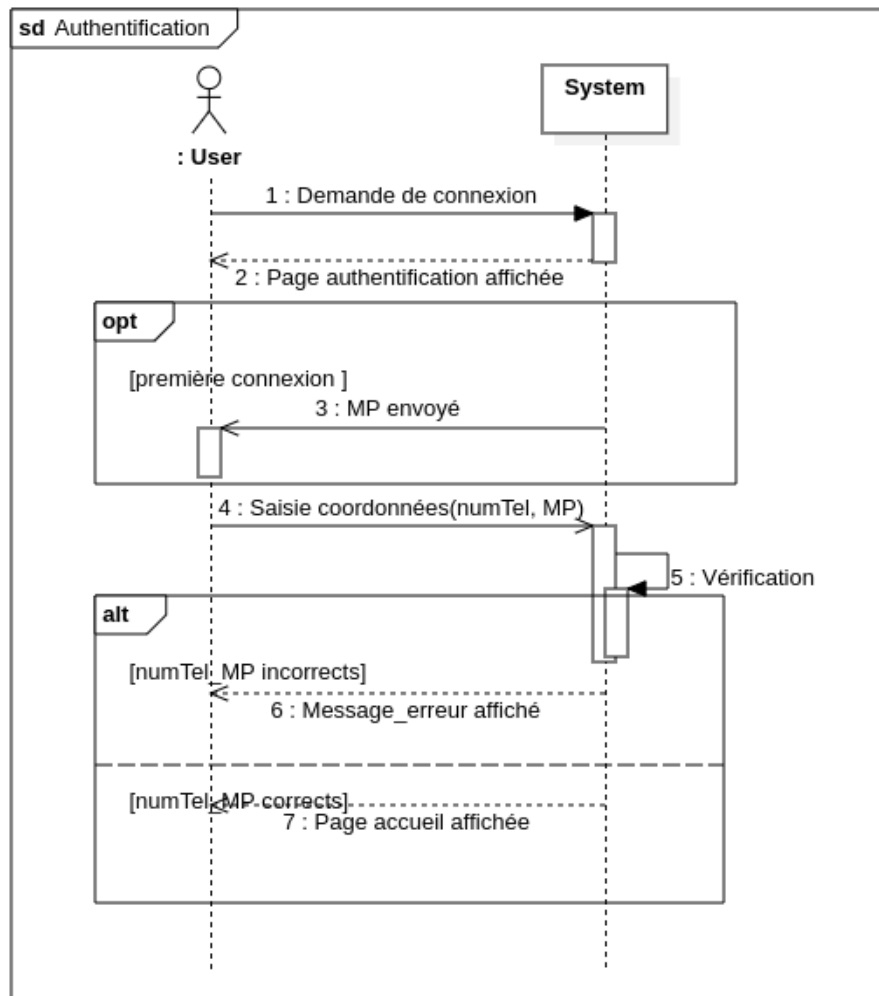


Figure 11: Diagramme de Séquence Authentification

3. Interfaces réalisées:

3.1. Page Onboarding:

La page Onboarding a été conçue pour accueillir les utilisateurs dès leur première interaction avec l'application. Elle offre une expérience conviviale et informative en présentant des écrans attrayants et intuitifs. Cette interface guide les utilisateurs à travers les étapes nécessaires pour créer un compte ou se connecter à un compte existant. Des éléments visuels attrayants tels que des images, des icônes et des textes explicatifs sont utilisés pour faciliter la compréhension et l'engagement des utilisateurs dès le début.

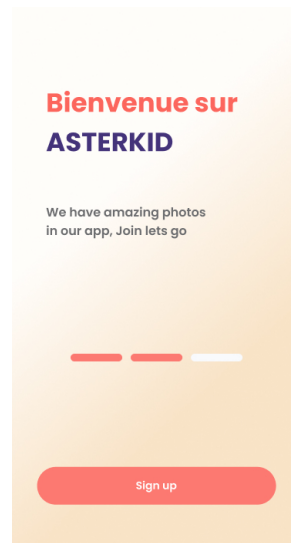


Figure 12: Onboarding Page

3.2. Module Login:

Le module de connexion permet aux utilisateurs enregistrés d'accéder à leur compte personnel. Elle présente un formulaire de connexion convivial où les utilisateurs peuvent saisir leurs informations d'identification, telles que leur code d'accès et leur mot de passe. Des fonctionnalités supplémentaires telles que la récupération du mot de passe ou la possibilité de se connecter via des comptes de réseaux sociaux peuvent également être intégrées pour améliorer l'expérience utilisateur et faciliter l'accès à l'application.

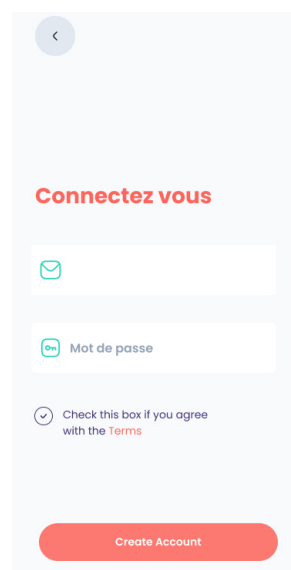


Figure 13: Login Page

Conclusion :

En conclusion, ce premier sprint nous a permis de mettre en place un solide fondement technique et d'avancer sur des fonctionnalités essentielles. Nous sommes maintenant prêts à aborder les prochains sprints avec confiance, en nous concentrant sur l'ajout de nouvelles fonctionnalités et la réalisation de notre vision pour l'application.

Sprint 2 : Module Homepage et Messagerie Instantanée

Introduction :

Dans le deuxième sprint de notre projet, nous nous concentrons sur la mise en place des fonctionnalités de la page d'accueil qui sont : file d'actualité et de messages importants et qui se concentre sur le développement du module de messagerie instantanée de notre application. L'objectif principal de ce sprint est de permettre aux utilisateurs de rester informés des dernières nouvelles et des annonces importantes liées à l'établissement de jardin d'enfants et de mettre en place la liste des conversations et d'implémenter les fonctionnalités de base pour interagir avec les utilisateurs.

1. Backlog deuxième sprint:

ID	Tâche à faire	Durée
2.1	Créer le composant "Header" pour afficher la barre de navigation	2 jours
2.2	Créer le composant "Footer" pour afficher le pied de page	2 jours
2.3	Mise en place du composant de file d'actualité	3jours
2.4	Mise en place de la fonctionnalité de publication de messages importants	2 jours
2.5	Conception de l'interface utilisateur pour la file d'actualité et les messages importants	3 jours

2.6	Créer la liste des conversations	2 jours
2.7	Implémenter l'affichage des types de conversation	2 jour
2.8	Gérer la sélection d'une conversation	2 jour
2.9	Afficher les détails des conversations	10 jours

2. Spécification fonctionnelle :

2.1. Cas d'utilisation Messagerie:

Cas d'utilisation	Echanger message
Acteur	Parent
Pré-Condition	L'utilisateur doit être connecté
Post-Condition	L'utilisateur accède à la page messagerie
Enchaînement	<ul style="list-style-type: none">- Le parent est invité à un groupe de Chat- Le parent démarre une conversation avec un animateur

2.2. Diagramme de séquences Messagerie :

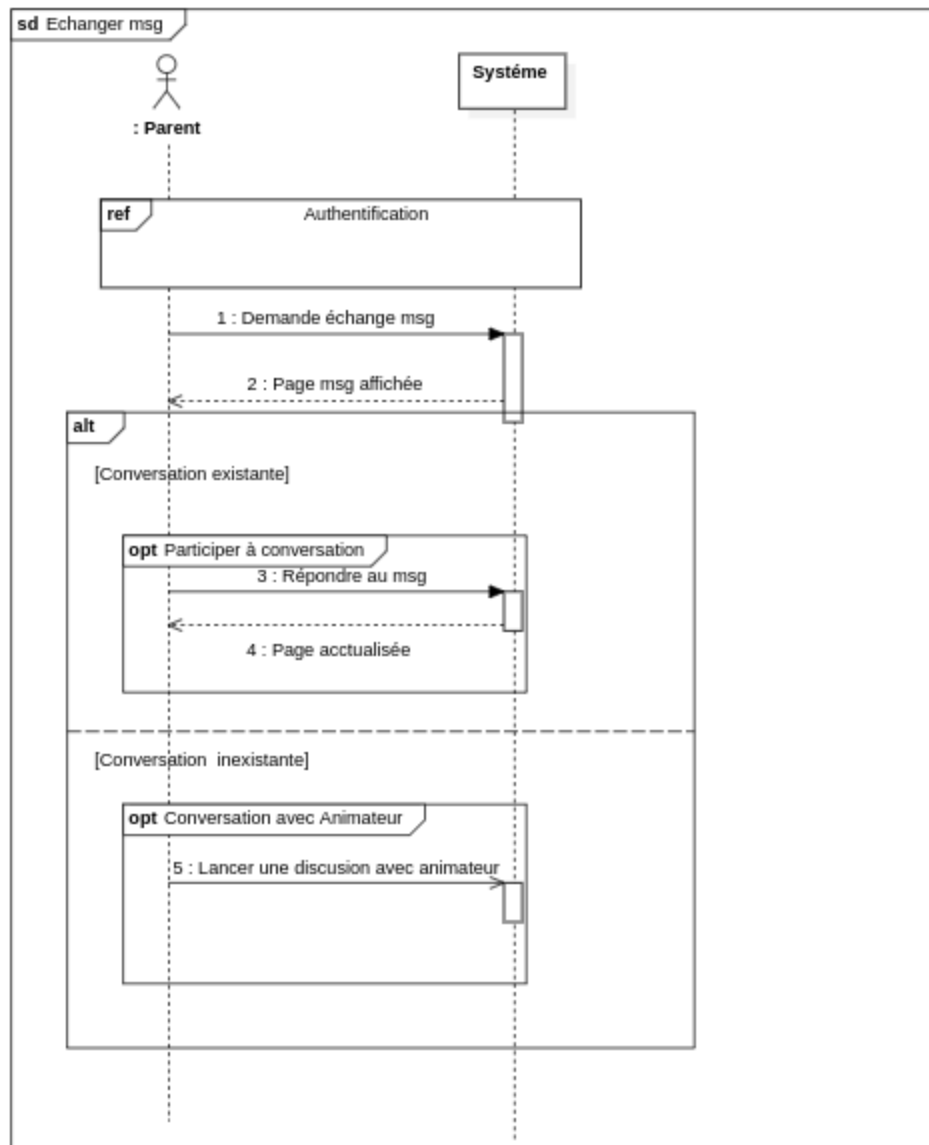


Figure 14: Diagramme de séquences Échanger message

3. Interface réalisée:

3.1. Module d'accueil:

Le module d'accueil de l'application regorge de fonctionnalités clés pour offrir aux parents une expérience complète. Tout d'abord, elle comprend une file d'actualités dynamique qui permet aux parents de rester informés des dernières nouvelles, des mises à jour et des

événements importants liés au jardin d'enfant. Cette fonctionnalité garantit aux utilisateurs d'être constamment à jour et de ne rien manquer.

Ensuite, la page d'accueil affiche également les messages importants qui nécessitent l'attention des utilisateurs. Ces messages peuvent inclure des annonces, des notifications ou des alertes concernant des événements futurs, des changements de politique ou toute autre information pertinente. Cette fonctionnalité permet de communiquer efficacement entre l'établissement et les parents et de leur transmettre des informations essentielles de manière claire et visible.

Enfin, la page d'accueil permet aux utilisateurs d'ajouter leurs propres commentaires aux publications ou aux contenus disponibles. Cette fonctionnalité encourage l'interaction et la participation active des utilisateurs en leur donnant la possibilité d'exprimer leurs idées, de poser des questions ou de partager leurs expériences. Les utilisateurs peuvent ainsi contribuer au contenu de l'application et s'engager dans des discussions enrichissantes.

Dans l'ensemble, la page d'accueil de l'application offre une expérience complète en rassemblant une file d'actualités dynamique, des messages importants, l'affichage des commentaires existants et la possibilité d'ajouter de nouveaux commentaires. Elle crée ainsi un espace interactif où les parents peuvent rester informés, interagir avec d'autres utilisateurs et participer activement à la communauté de l'application.

Sprint 2: Module homepage et messagerie instantanée

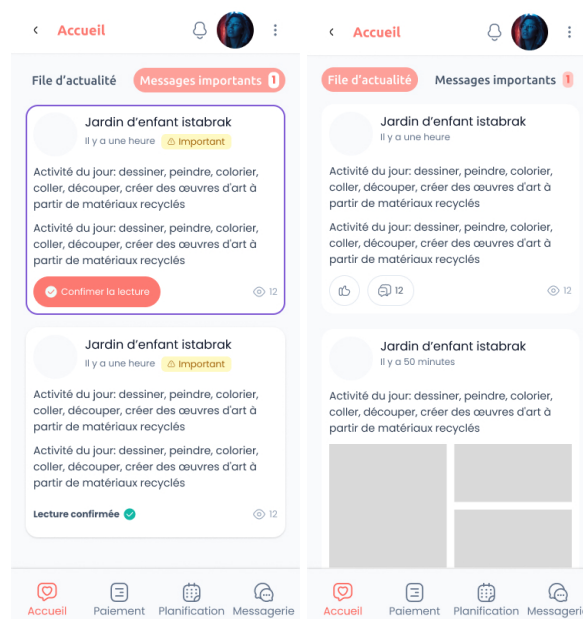


Figure 14: :Home Page

3.1.1.Mise en place du composant de file d'actualité:

Un composant responsable de l'affichage des informations d'un post dans le fil d'actualité. Il affiche des éléments tels que l'image de profil de l'auteur du post, son nom, la date de publication, le contenu du post, le nombre de commentaires et le nombre de vues. Les informations sont présentées de manière claire et attrayante à l'aide des composants fournis par React Native Magnus. Le composant utilise des styles prédéfinis pour assurer une apparence cohérente avec le reste de l'application. Grâce à ce composant, les utilisateurs peuvent visualiser et interagir avec les posts du fil d'actualité de manière conviviale et intuitive.

3.1.2.Mise en place du composant message important :

Un composant est utilisé pour afficher des messages importants dans l'application. Il affiche des informations telles que l'auteur du message, la date de publication, le contenu du message et le nombre de vues. Le composant utilise des icônes spécifiques pour indiquer si

le message a été confirmé comme étant lu ou s'il est marqué comme important. Il offre également la possibilité de confirmer la lecture du message. Grâce à ce composant, les utilisateurs peuvent facilement visualiser et interagir avec les messages importants.

3.1.3. Le composant Header :

Le composant Header que nous avons élaboré est un composant générique utilisé pour afficher la barre de navigation en haut de chaque écran de l'application. Ce composant permet d'ajouter des fonctionnalités telles que le bouton de retour, l'icône de notification, l'avatar utilisateur et l'icône de menu. Le composant prend plusieurs paramètres pour personnaliser son affichage..

Le composant utilise des icônes importées à partir du module "react-native-magnus" pour représenter les différentes fonctionnalités. Les actions associées à chaque fonctionnalité peuvent être définies en utilisant des fonctions de gestion d'événements.

Le composant Header est enveloppé dans un composant "Header" fourni par la bibliothèque "react-native-magnus"..

Ce composant permet d'obtenir une barre de navigation cohérente et personnalisable dans toute l'application, ce qui améliore l'expérience utilisateur en offrant une navigation intuitive et des fonctionnalités supplémentaires.

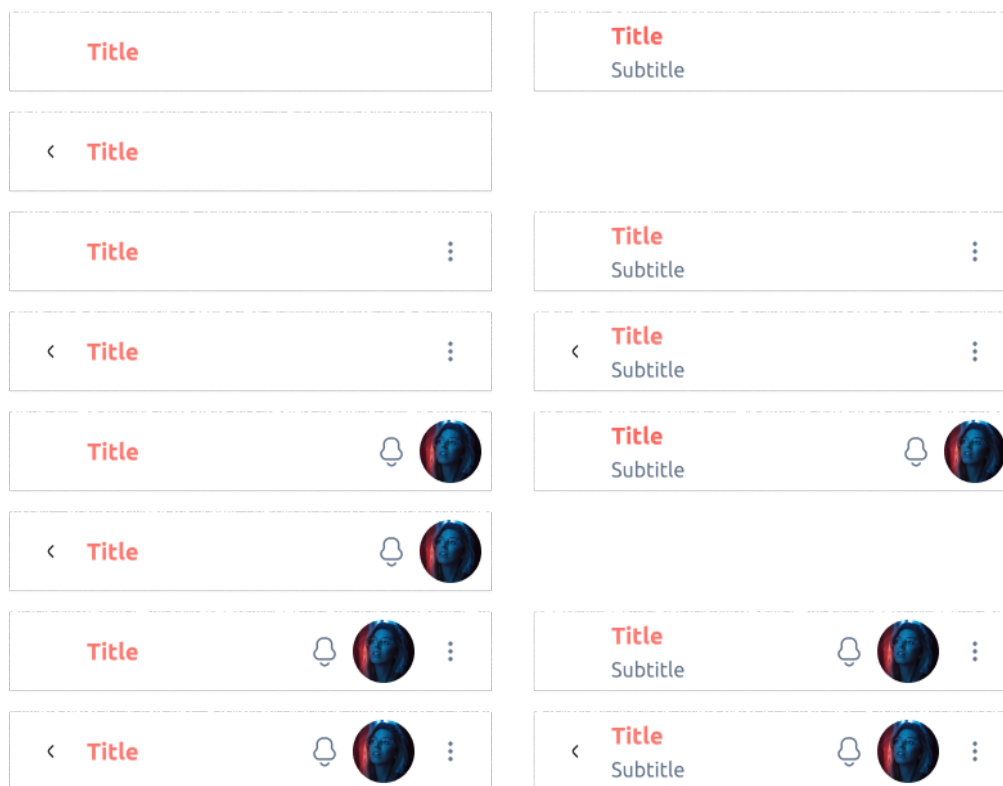


Figure 16: :Header component

3.1.4. Le Composant Footer :

Le composant Footer que nous avons utilisé est basé sur la bibliothèque "react-native-navigation". Nous avons utilisé le composant " Bottom TabNavigator" pour créer une barre de navigation en bas de l'application avec des onglets.

Le composant Footer est enveloppé dans le composant qui configure le comportement et l'apparence des onglets. Nous avons spécifié différentes options pour chaque écran d'onglet, y compris l'icône et l'étiquette à afficher.

Chaque écran d'onglet est défini avec le composant et associe un nom d'écran à un composant de page correspondant. Nous avons également défini les icônes à afficher pour chaque onglet, en utilisant des icônes importées à partir de la bibliothèque.

Le composant Footer offre une navigation facile entre les différentes fonctionnalités de l'application. Chaque onglet représente une fonctionnalité spécifique, telle que l'accueil, les paiements, la planification et la messagerie. Les utilisateurs peuvent interagir avec les onglets en appuyant dessus pour accéder à la page correspondante.

Ce composant ajoute une navigation intuitive et visuellement attrayante à l'application, ce qui permet aux utilisateurs de passer facilement d'une fonctionnalité à l'autre en utilisant la barre de navigation en bas de l'écran.



Figure 17: :navbar component

3.2. Module Messagerie:

Le module de messagerie de l'application de jardin d'enfants offre un espace de communication essentiel pour les utilisateurs. Ce dernier peut prendre différentes formes, que ce soit des groupes de classe dédiés à une classe spécifique, des discussions ouvertes à tous les utilisateurs de l'application ou des conversations privées en tête-à-tête avec un seul utilisateur.

Les groupes de classe permettent aux enseignants, aux parents et aux autres membres de la communauté de classe de communiquer efficacement et de partager des informations spécifiques à leur groupe. Ils offrent un moyen pratique de discuter des événements à venir, des devoirs, des projets ou de coordonner des activités spéciales liées à la classe. Les discussions de groupe favorisent la collaboration et la coopération entre les membres de la classe, facilitant ainsi une communication transparente et une participation active de tous les intervenants.

En plus des groupes de classe, la page de messagerie offre également la possibilité de participer à des discussions ouvertes à tous les utilisateurs de l'application. Ces discussions peuvent porter sur des sujets généraux liés à l'éducation, à l'évolution des enfants, aux conseils pratiques, aux idées créatives, etc. Elles permettent aux parents, aux enseignants et aux autres membres de la communauté de partager leurs connaissances, leurs expériences et de bénéficier de l'expertise collective de la communauté.

Sprint 2: Module homepage et messagerie instantanée

En outre, la page de messagerie offre la fonctionnalité de conversations privées en tête-à-tête avec un seul utilisateur. Cela permet aux parents, aux enseignants ou à d'autres membres de la communauté de discuter de manière confidentielle, d'échanger des informations personnelles, de poser des questions spécifiques ou de résoudre des problèmes particuliers. Ces conversations privées favorisent une communication plus ciblée et individuelle, permettant ainsi de répondre aux besoins spécifiques des utilisateurs.

En résumé, la page de messagerie de l'application de jardin d'enfants offre un espace de communication polyvalent, comprenant des groupes de classe, des discussions ouvertes et des conversations privées. Elle facilite l'échange d'informations, la collaboration, le partage d'expériences et le soutien mutuel au sein de la communauté de l'application, contribuant ainsi à renforcer les liens entre les parents, les enseignants et les membres de la communauté éducative.

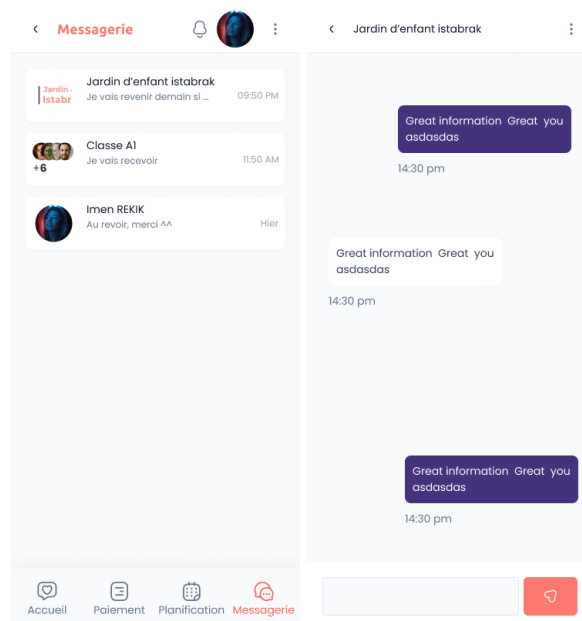


Figure 18: :Chat Page

3.2.1.Créer la liste des conversations :

Dans cette tâche, nous allons concevoir et développer la fonctionnalité permettant d'afficher la liste des conversations disponibles dans l'application. Cela comprend l'affichage des noms d'utilisateur ou des noms de groupes, ainsi que les derniers messages échangés. Nous

utiliserons des composants réutilisables et une approche modulaire pour garantir une conception optimale.

3.2.2 Implémenter l'affichage des types de conversation :

Cette tâche consistera à ajouter la possibilité de filtrer les conversations par type, telles que les conversations individuelles ou les conversations de groupe. Nous créerons des interfaces utilisateur intuitives et des boutons de sélection pour permettre aux utilisateurs de naviguer facilement entre les différentes catégories de conversations.

3.2.3 Gérer la sélection d'une conversation :

Nous allons mettre en place la fonctionnalité permettant aux utilisateurs de sélectionner une conversation spécifique dans la liste. Cela inclura la gestion des événements de clic sur les éléments de la liste et la mise en évidence visuelle de la conversation sélectionnée. Nous veillerons à ce que cette fonctionnalité soit réactive et conviviale.

3.2.4. Afficher les détails de chaque conversation :

Dans cette tâche, nous développerons les fonctionnalités permettant d'afficher les détails d'une conversation spécifique. Cela comprendra l'affichage de l'historique des messages, des participants de la conversation, ainsi que des options supplémentaires telles que la possibilité de supprimer ou archiver une conversation. Nous nous assurerons que toutes les informations nécessaires sont présentées de manière claire et attrayante.

Conclusion :

Au cours du deuxième sprint de notre projet, nous avons réussi à mettre en place les fonctionnalités essentielles de la page d'accueil, notamment la file d'actualités et les

Sprint 2: Module homepage et messagerie instantanée

messages importants et le développement du module de messagerie instantanée de notre application. Nous avons travaillé sur la création de la liste des conversations, l'affichage des types de conversation, la gestion de la sélection d'une conversation et l'affichage des détails de chaque conversation. Ces fonctionnalités sont essentielles pour offrir une expérience utilisateur fluide et interactive dans notre application. Dans le prochain chapitre, nous aborderons le quatrième sprint, qui se concentrera sur d'autres aspects du module de messagerie instantanée, tels que l'envoi de messages et les notifications.

Sprint 3 : Module de Paiement et Planification

Introduction :

Ce chapitre se concentre sur le troisième sprint de notre projet, qui se consacre au développement du module de paiement et planning de l'application de jardin d'enfants. Ces modules offrent aux parents la possibilité de consulter le statut de leurs paiements et d'interagir avec le système pour signaler un paiement effectué et permet aussi aux utilisateurs de visualiser et de gérer le planning des cours et des activités pour les enfants. Nous mettrons en place une interface claire et organisée, offrant une vue hebdomadaire ou mensuelle du planning, avec toutes les informations nécessaires. Les parents peuvent visualiser un récapitulatif de leurs paiements passés et en cours, ainsi que signaler un paiement effectué via une fonctionnalité dédiée.

1. Backlog du troisième sprint :

ID	Tâche à faire	Durée
5.1	Créer l'interface de visualisation des paiements	2 jours
5.2	Afficher le récapitulatif des paiements passés	3 jours
5.3	Afficher les détails de chaque paiement	2 jours
5.4	Ajouter la fonctionnalité de signaler un paiement effectué	1 jours
4.1	Créer l'interface pour afficher le calendrier ou la planification	2 jours
4.2	Implémenter la vue hebdomadaire du planning	3 jours

4.3	Implémenter la vue mensuelle du planning	2 jours
4.4	Afficher les détails des cours et activités	2 jours
4.5	Ajouter la fonctionnalité de gestion du planning	3 jours

2. Spécification fonctionnelle :

2.1. Cas d'utilisation Paiement:

Cas d'utilisation	Consulter paiement
Acteur	Parent
Pré-Condition	Accéder à la page paiement
Post-Condition	Affichage de la liste de paiement
Enchainement	<ul style="list-style-type: none"> - Affichage de toute la liste de paiement - Sélectionner un paiement - Affichage des détails du paiement
Cas d'exception	<ul style="list-style-type: none"> - Paiement Payé ou annuler : pas de bouton marquer comme payé - Paiement en Retard ou Partiellement Payé : Affichage de bouton Marqué comme payé

2.2. Diagramme de séquences paiement:

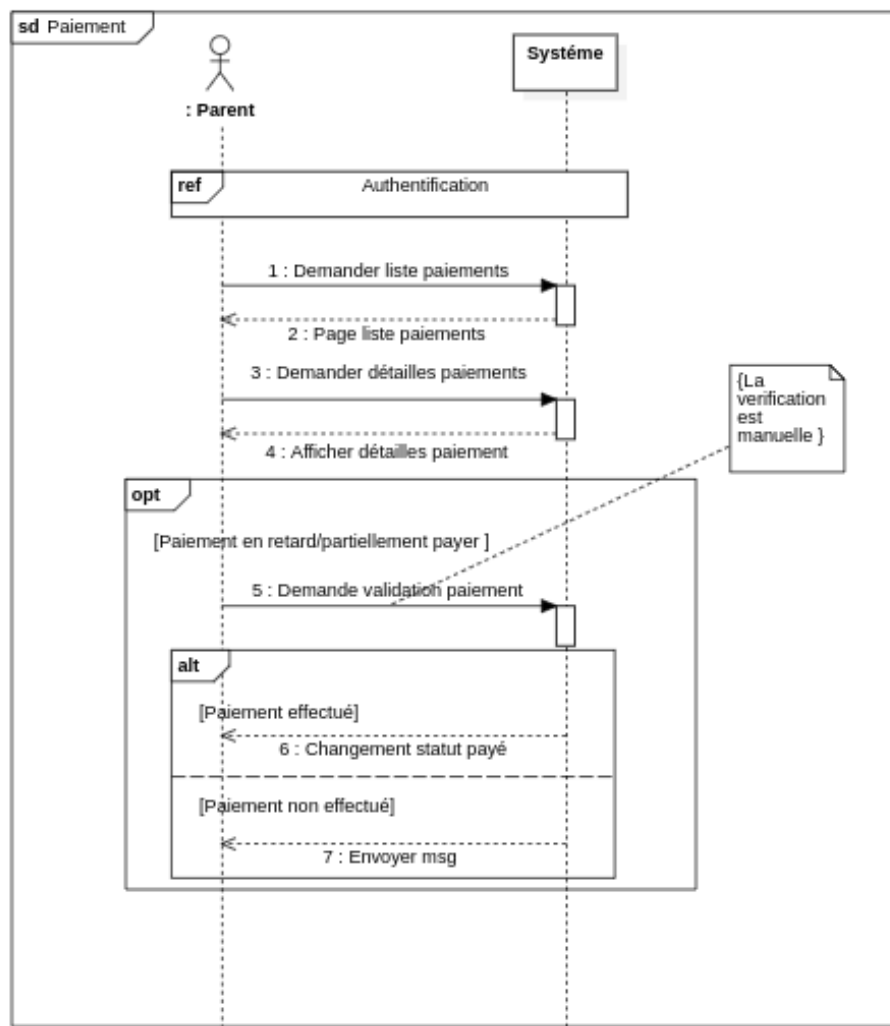


Figure 19: Diagramme de Séquence paiement

3. Interface réaliser :

3.1 Module Paiement :

Dans le module de paiement de l'application de jardin d'enfants, les parents ont la possibilité de consulter le statut de leurs paiements et d'interagir avec le système pour signaler un paiement effectué.

Sur cette page de paiement, les parents peuvent visualiser un récapitulatif de leurs paiements passés et en cours. Ils ont accès aux détails de chaque paiement, tels que la date, le montant

dû et le montant déjà payé. Cela leur permet d'avoir une vue d'ensemble claire de leurs obligations financières envers l'établissement.

En outre, les parents peuvent signaler un paiement effectué en utilisant une fonctionnalité dédiée. Ils peuvent soumettre une demande pour marquer un paiement comme payé en fournissant les informations pertinentes, telles que la référence de paiement ou le numéro de transaction. Cette demande est ensuite traitée par l'administration de l'établissement, qui vérifie les informations fournies et met à jour le statut du paiement en conséquence.

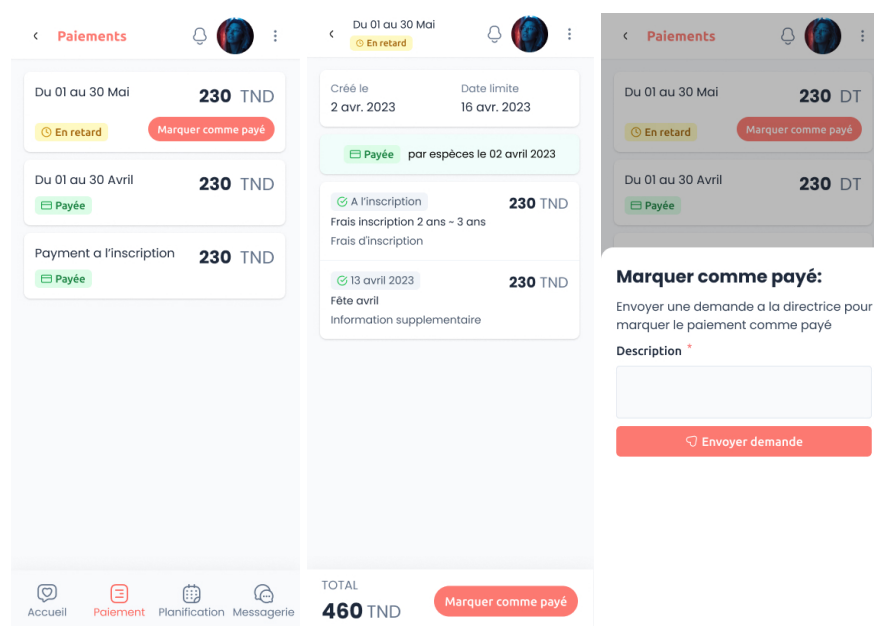


Figure 20: :Payment Page

3.1.1 Créer l'interface de visualisation des paiements :

Dans cette tâche, nous concevons et développons l'interface utilisateur pour la visualisation des paiements. Nous mettrons en place une mise en page claire et conviviale, où les parents pourront accéder aux informations relatives à leurs paiements.

3.1.2 Afficher le récapitulatif des paiements passés :

Nous mettrons en place la fonctionnalité permettant aux parents de visualiser un récapitulatif de leurs paiements passés. Cette vue affichera les paiements effectués, avec des détails tels que la date du paiement, le montant dû et le montant déjà payé.

3.1.3 Afficher les détails de chaque paiement :

Dans cette tâche, nous développerons la fonctionnalité permettant aux parents de consulter les détails de chaque paiement. Ils pourront accéder à des informations spécifiques, telles que la date, le montant dû, le montant déjà payé et éventuellement d'autres détails pertinents.

3.1.4 Ajouter la fonctionnalité de signaler un paiement effectué :

Nous mettrons en place une fonctionnalité permettant aux parents de signaler un paiement effectué. Ils pourront soumettre une demande pour marquer un paiement comme payé en fournissant les informations nécessaires, telles que la référence de paiement ou le numéro de transaction.

3.1.5 Mettre en place le traitement des demandes de paiement signalé :

Dans cette tâche, nous développerons la logique de traitement des demandes de paiement signalées. L'administration de l'établissement reçoit les demandes, vérifie les informations fournies par les parents et met à jour le statut du paiement en conséquence.

3.2 Module Planification :

Le module de planification de l'application de jardin d'enfants est un espace dédié à la visualisation et à la gestion du planning des cours. Elle offre une vue claire et organisée de toutes les activités et les cours prévus pour les enfants.

Sur cette page, les utilisateurs peuvent consulter le planning hebdomadaire ou mensuel, selon leurs préférences, afin de connaître les différents cours et activités qui auront lieu.

Sprint 3 : : Module de Paiement et Planification

Chaque cours est accompagné d'informations essentielles telles que l'heure de début et de fin, le nom de l'enseignant ou de l'animateur, ainsi que la description de l'activité.

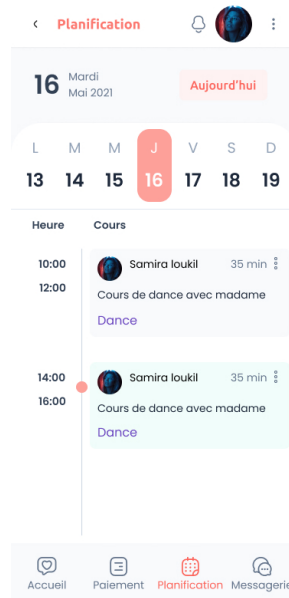


Figure 21: :Planification Page

3.2.1 Créer l'interface de visualisation du planning :

Dans cette tâche, nous concevons et développons l'interface utilisateur pour la visualisation du planning. Nous mettrons en place une mise en page claire et intuitive, avec des composants réutilisables pour afficher les différents cours et activités prévus.

3.2.2 Implémenter la vue hebdomadaire du planning :

Cette tâche consistera à développer la fonctionnalité permettant d'afficher le planning sous forme hebdomadaire. Nous afficherons les cours et les activités organisés par jour de la semaine, en mettant en évidence les heures de début et de fin, ainsi que les informations principales de chaque événement.

3.2.3 Implémenter la vue mensuelle du planning :

Nous ajouterons la possibilité d'afficher le planning sous forme mensuelle, offrant ainsi une vue d'ensemble des cours et des activités prévus tout au long du mois. Nous concevons une interface conviviale, permettant aux utilisateurs de naviguer facilement entre les différentes semaines du mois.

3.2.4 Afficher les détails des cours et activités :

Dans cette tâche, nous développerons les fonctionnalités permettant d'afficher les détails des cours et des activités sélectionnés. Cela inclura des informations telles que l'heure de début et de fin, le nom de l'enseignant ou de l'animateur, ainsi qu'une description de l'activité. Nous nous assurerons que toutes ces informations sont présentées de manière claire et compréhensible.

3.2.5 Ajouter la fonctionnalité de gestion du planning :

Nous mettrons en place la fonctionnalité permettant aux utilisateurs de gérer le planning en ajoutant, modifiant ou supprimant des cours et des activités. Nous concevons des formulaires intuitifs et des boutons d'action pour permettre une gestion facile et efficace du planning.

Conclusion :

Ce troisième sprint s'est concentré sur le développement du module de paiement de notre application. Nous avons travaillé sur la création de l'interface de visualisation des paiements, l'affichage du récapitulatif des paiements passés, l'affichage des détails de chaque paiement, l'ajout de la fonctionnalité de signaler un paiement effectué

Conclusion générale :

Dans le cadre de ce projet de développement d'une application de jardin d'enfants, nous avons réalisé un rapport détaillé présentant les différentes étapes de conception et de développement de l'application. Nous avons abordé les principaux modules de l'application, tels que la page d'accueil, le module de messagerie, le module de paiement et de planification, en fournissant des spécifications fonctionnelles, des diagrammes et des captures d'écran illustrant l'interface utilisateur.

Au cours du premier sprint, nous avons développé la page d'accueil de l'application, en mettant en place des fonctionnalités telles que la file d'actualités, les messages importants et la barre de navigation en bas de l'écran. Ces fonctionnalités offrent une expérience utilisateur intuitive et attrayante, permettant aux utilisateurs de naviguer facilement entre les différentes fonctionnalités de l'application.

Le deuxième sprint était dédié au module de messagerie, qui constitue un espace essentiel de communication au sein de l'application. Nous avons développé des fonctionnalités telles que les groupes de classe, les discussions ouvertes et les conversations privées, offrant aux utilisateurs différents moyens de communiquer et de collaborer. L'interface de messagerie offre une navigation fluide entre les différentes conversations et favorise l'échange d'informations et le soutien mutuel au sein de la communauté éducative.

Le troisième sprint s'est concentré sur le module de paiement et de planification. Nous avons créé une interface conviviale pour la visualisation des paiements, permettant aux parents de consulter leur historique de paiements, de voir les détails de chaque paiement et de signaler les paiements effectués. De plus, nous avons mis en place une interface de planification hebdomadaire et mensuelle, offrant aux utilisateurs une vue claire des cours et des activités prévus.

Tout au long du projet, nous avons adopté une approche modulaire et réutilisable, en utilisant des composants de bibliothèques populaires telles que React Native et React Navigation. Cela nous a permis de développer efficacement les différentes fonctionnalités

Conclusion générale

de l'application tout en garantissant une interface utilisateur cohérente et une maintenance simplifiée.

En conclusion, ce rapport met en évidence les différentes étapes de conception et de développement d'une application de jardin d'enfants. Les modules développés, tels que la page d'accueil, le module de messagerie, le module de paiement et de planification, offrent une expérience utilisateur complète et répondent aux besoins des parents, des enseignants et des membres de la communauté éducative. Ce projet nous a permis de mettre en pratique nos compétences en développement d'applications mobiles et de contribuer à la création d'un outil efficace pour la communication et la gestion au sein d'un jardin d'enfants.

En somme, ce projet représente une étape importante dans le développement d'une application de jardin d'enfants, mais il ouvre également de nouvelles perspectives pour l'avenir. En continuant à améliorer l'application, à explorer les technologies émergentes, à collaborer avec d'autres acteurs de l'éducation et à adapter l'application à d'autres contextes éducatifs, nous pouvons contribuer de manière significative à l'amélioration de la communication et de la gestion au sein des établissements éducatifs.

Bibliographie :

[1] : Livre : "Scrum: The Art of Doing Twice the Work in Half the Time"

[2] : <https://about.gitlab.com/>

[3]: <https://code.visualstudio.com/>

[4]: <https://reactnative.dev/>

[5]: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>

[6]: <https://magnus-ui.com/>

[7]: <https://kinsta.com/knowledgebase/nestjs/>

[8]: <https://reactnavigation.org/>

[9]: <https://www.techtarget.com/whatis/definition/GitLab>

[10]:<https://reactnavigation.org/docs/getting-started/>