



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université des Sciences et de la Technologie Houari Boumediène

Faculté d'électronique et d'informatique

Département Informatique

## Mémoire de Licence

Spécialité:

Informatique Académique

Thème

**Agenda Electronique partagé destiné aux parents**

Présenté Par :

AIRA Cylia

BERROUANE Wafa

Proposé Par :

Mme GUEBLI Wassila

Devant le jury :

Dr Smaili                      Président

M<sup>me</sup> Mekahlia              Membre

**PROJET 019/21**

# **SOMMAIRE**

<i>Introduction Générale</i> .....	6
Chapitre 1 : « Etude préalable et Etat de l’art » .....	7
1.1 Introduction .....	7
1.2 Les agendas électroniques .....	7
1.2.1 Définition .....	7
1.2.2 Type agenda .....	7
1.2.3 Les avantages d’une application agenda électronique(4).....	7
1.2.4 Etude des agendas existants .....	8
1.2.5 Avantages et inconvénients des agendas existants.....	10
1.3 Conclusion .....	10
Chapitre 2 : « Conception et modélisation ».....	11
2.1 Introduction .....	11
2.2 Problématique .....	11
Objectifs .....	11
2.3 Architecture de système.....	12
2.4 Définition des besoins .....	12
2.4.1 Spécification des besoins fonctionnels.....	12
2.4.2 Analyse des besoins non fonctionnels.....	13
2.5 Identification des acteurs .....	13
2.6 Diagramme de cas d’utilisations.....	14
2.7 Diagramme de séquence .....	17
2.8 Diagramme de classe .....	19
2.9 Modèle relationnel .....	20
2.9.1 Définition .....	20
2.9.2 Schéma relationnel .....	<b>Erreur ! Signet non défini.</b>
2.10 Conclusion .....	21
Chapitre 3 : « Implémentation et réalisation» .....	22
3 .....	22
3.1 Introduction .....	22
3.2 Les outils de développement .....	22
3.2.1 DéfinitionAndroid studio .....	22

3.2.2	Définition XML.....	22
3.2.3	Langage JAVA.....	22
3.2.4	FIREBASE.....	22
3.2.5	NoSQL .....	23
3.3	La présentation de quelques interfaces de l'application.....	23
3.3.1	Page de connexion et déconnexion .....	23
3.3.2	Interface calendrier et menu.....	23
3.3.3	Interface des évènements .....	24
3.3.4	Interface gestion de maison.....	25
3.3.5	Interfaces suivi enfant .....	26
3.4	Conclusion.....	28
	<i>Conclusion Générale</i> .....	29

## **Liste figures :**

Figure 1.1	Interface Google Agenda.....	8
Figure 1.2	Interface JORTE .....	9
Figure 1.3	Interface PICNIIC.....	9
Figure 2. 1	Architecture de système.....	12
Figure 2. 2	Diagramme des cas d'utilisation du system.....	14
Figure 2. 3	Diagramme des cas d'utilisation de gestion des évènements .....	15
Figure 2. 4	Diagramme des cas d'utilisation de gestions des tâches.....	15
Figure 2. 5	Diagramme des cas d'utilisation de suivi des enfants .....	16
Figure 2. 6	Diagramme des cas d'utilisation de gestions des listes des cours .....	17
Figure 2. 7	Diagramme de séquence de cas d'utilisation d'authentification.....	17
Figure 2. 8	Diagramme de séquence d'ajout d'enfant.....	18
Figure 2. 9	Diagramme de séquence d'ajout d'évènement .....	18
Figure 2. 10	Diagramme de séquence de suivi d'un enfant et sa suppression .....	19
Figure 2. 11	Diagramme de séquence supprimer tâche.....	19
Figure 2. 12	Diagramme de classe .....	20

Figure 3. 1 Logo firebase .....	23
Figure 3. 2 Interface login .....	23
Figure 3. 3 Interface déconnexion.....	23
Figure 3. 4 Interface calendrier .....	24
Figure 3. 5 Interface menu .....	24
Figure 3. 7 Interface tâche.....	25
Figure 3. 8 Interface liste d'achat.....	26
Figure 3. 9 Interface suivie enfant.....	27
Figure 3. 10 Interface programme d'un enfant .....	27
Figure 3. 11 Interface d'état de santé d'un enfant.....	28

### **Liste tables :**

Tableau 1. Comparaison entre les agendas électroniques .....	10
--	----

# *Remerciement*

*La réalisation de ce mémoire a été possible grâce au concours de plusieurs Personnes à qui je voudrais témoigner toute ma reconnaissance.*

*Je tiens à remercier mon encadreur « Mme GUEBLI Wassila » pour sa patience et ses judicieux conseils qui ont contribué à alimenter ma réflexion.*

*Je remercie également mes chers parents et toute ma famille pour leur soutien constant et leur encouragement.*

*Je tiens à témoigner toute ma reconnaissance pour mes amis spécialement ma binôme BEROUANE Wafa, Eldjida, Raissa qui m'ont accordé leur confiance, support morale et intellectuel et l'aide pour réaliser ce mémoire.*

**AIRA Cylia**

*En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné le courage, la force et la patience d'achever ce travail. Je tiens à remercier toutes les personnes qui ont contribué au succès de ce projet, et qui nous ont aidés lors de la rédaction de ce rapport.*

*J'adresse mes remerciements à notre encadreur « Mme GUEBLI Wassila » pour son aide, ses conseils, son encouragement et sa disponibilité.*

*Je remercie également ma famille pour leur présence et leur encouragement durant mes années d'études.*

*Je ne saurais oublier de remercier tous mes amis, un grand merci à ma binôme AIRA Cylia et Mahdi pour sa confiance et son soutien inestimable durant la réalisation de ce travail.*

**BERROUANE Wafa**

## ***Introduction Générale***

Le manque de temps est une réalité courante chez de nombreux parents aujourd'hui, en raison de la difficulté à concilier vie personnelle et professionnelle.

Du matin au soir, les parents ont l'impression de courir pour accomplir des tâches liées à la vie de famille : travail à domicile, activités prévues, rendez-vous divers (médecins, clubs, etc.) alors qu'ils ne peuvent pas se souvenir de tout. La charge physique, mental et émotionnel pèse lourdement sur les épaules des parents, en particulier ceux qui ont de jeunes enfants.

Pour cette raison, beaucoup de parents utilisent différents agendas électroniques partagés qui offre une meilleure organisation et plus de liberté d'esprit. Cette application leur donne la possibilité d'améliorer leur quotidien et de leur apporter plus de confort.

Différents types d'agendas électroniques sont disponibles. Certains sont simples, tandis que d'autres sont des applications puissantes et polyvalentes. Comme souvent, le choix est donc vaste et chaque application répondra au besoin de chacun. Mais ils ne répondent pas à tous les besoins des parents.

C'est pourquoi nous sommes tenus de mettre en place et de proposer un assistant personnel électronique pour les parents qui intègre un maximum de fonctions pour leur donner la possibilité de planifier, d'organiser et de se rappeler des différents tâches à faire.

Notre application doit permettre aux utilisateurs de mieux organiser leurs temps entre leur vie personnelle et professionnelle.

Pour réaliser notre projet, nous avons répartis le mémoire en trois chapitres :

❖ **Chapitre 1**« Etude préalable et état de l'art »

Ce chapitre présent l'étude de faisabilité de l'application ainsi les types d'agendas existants.

❖ **Chapitre 2**« Analyse des besoins et conception »

Ce chapitre est consacré à la conception et la modélisation des données de l'application en suivant la modélisation UML.

❖ **Chapitre 3**« Implémentation et réalisation »

Ce chapitre présentera les outils de développement et la technologie utilisée pour implémenter le projet et les différentes interfaces de l'application.

# **Chapitre 1 : « Etude préalable et Etat de l'art »**

## **1.1 Introduction**

La première étape avant de proposer toutes solution de n'importe quel problème c'est de faire une étude préalable pour rechercher toutes les informations existantes concernant la problématique et à on faire une synthèse. Cette étude nous permet de nous introduire au domaine et de connaître ce qui existe comme solutions.

Dans ce premier chapitre nous abordons d'abord la définition des agendas électroniques et leurs avantages. Ensuite, nous ferons une étude sur les types d'agendas existants. Enfin, nous terminons par une étude comparative entre ces agendas.

## **1.2 Les agendas électroniques**

### **1.2.1 Définition**

Un agenda est un outil pour la gestion des données personnelles permettant d'associer des actions à des moments, et d'organiser ainsi son temps. Il est utilisé afin de donner à son utilisateur la possibilité de planifier, de noter son emploi du temps et ses rendez-vous.(1)

### **1.2.2 Type agenda**

Un agenda peut se présenter sous forme matérielle ou numérique (logiciel de calendrier installé sur un Smartphone).

L'agenda papier est toujours aussi populaire, c'est parce qu'il présente de nombreux avantages : accessible par tout le monde, facile à utiliser, plus naturel et plus visuel, pas de risque de perte ou de vol de données informatiques et visualiser d'un seul coup d'œil nos événements passés et à venir. L'utilisation d'un agenda papier n'est pas une solution parfaite sans inconvénient par exemple : ça prend de la place, c'est vieux jeu, risque de perte physique. Ainsi, on peut lui reprocher le fait d'avoir des fonctionnalités manquantes comme les rappels, le partage avec d'autres personnes ou encore l'intégration avec d'autres applications.(2)

C'est pourquoi les programmeurs ont inventé les agendas numériques installés sur pc afin d'améliorer et compléter les avantages des agendas papiers et éliminer leurs inconvénients, mais ils ne sont pas assez pratique dont les gens ne peuvent pas consulter leur agenda n'importe où ou à n'importe moment.

Pour cette raison divers développements technologiques tel que les technologies Mobiles ont développé l'application agenda électronique (numérique ) pour une planification efficace et rapide avec plus de fonctionnalités tel que le partage d'agendas entre plusieurs personnes, partage des tâches et la synchronisation avec multiples appareil. (3)

### **1.2.3 Les avantages d'une application agenda électronique(4)**

- **Accessible** : la plupart des agendas électroniques sont gratuits, et on peut accéder à notre agenda à partir d'un téléphone ou d'un ordinateur portable en tout temps
- **Planification** : elles nous donnent la possibilité de faire des planifications sur le long terme.
- Le **partage** entre plusieurs personnes.
- **Programmer des rappels** : après avoir noté un événement /tâche/programme, on peut ajouter un rappel en l'envoyant une notification. (5)
- La possibilité de **gérer plusieurs agendas** : à partir d'un même calendrier numérique, nous sommes en mesure de gérer plusieurs agendas différents. (6)

#### 1.2.4 Etude des agendas existants

Les agendas n'ont pas été programmés de la même manière. Le choix est donc vaste et chaque application répondra aux besoins de chacun. Parmi les applications agenda qui existent : Google Agenda, JORTE, PICNIIC.

Dans ce qui suit, nous présentons les solutions les plus courantes de gestion de temps, on va extraire par la suite les points forts et les points faibles de chacune de ces solutions.

##### ❖ GOOGLE AGENDA

C'est une application Google qui permet de partager des événements des tâches et son utilisation nécessite d'ouvrir un compte Google, elle est disponible sur IOS et ANDROID

On peut créer plusieurs agendas avec la confidentialité voulu (public, privée, partagé avec d'autres utilisateur), ajouter des événements, des rappels et des tâches.(7)

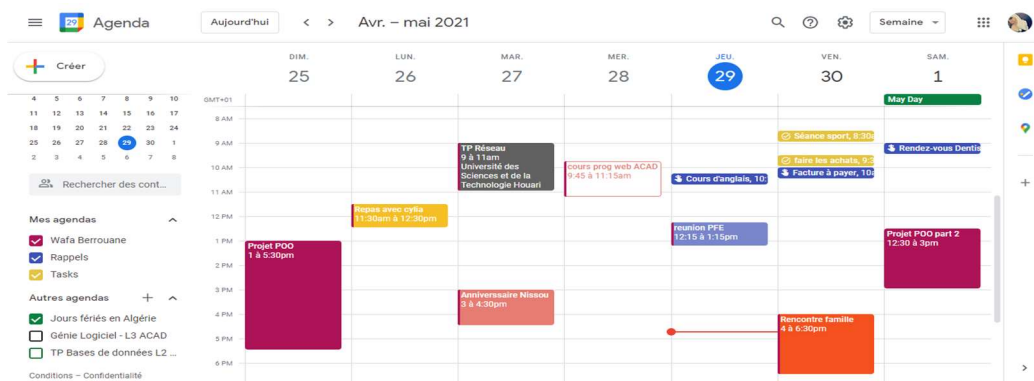


Figure 1.1 Interface Google Agenda

##### ❖ JORTE

L'agenda JORTE ressemble à Google agenda sauf qu'il n'est pas nécessaire d'avoir un compte pour pouvoir l'utiliser, elle est plus évoluée au niveau design (thèmes et couleurs) elle dispose aussi d'états de tâche c'est-à-dire la possibilité de sélectionner



l'états « active », « terminé », « reporté », elle permet aussi la synchronisation avec la plupart des agendas en ligne.(8)

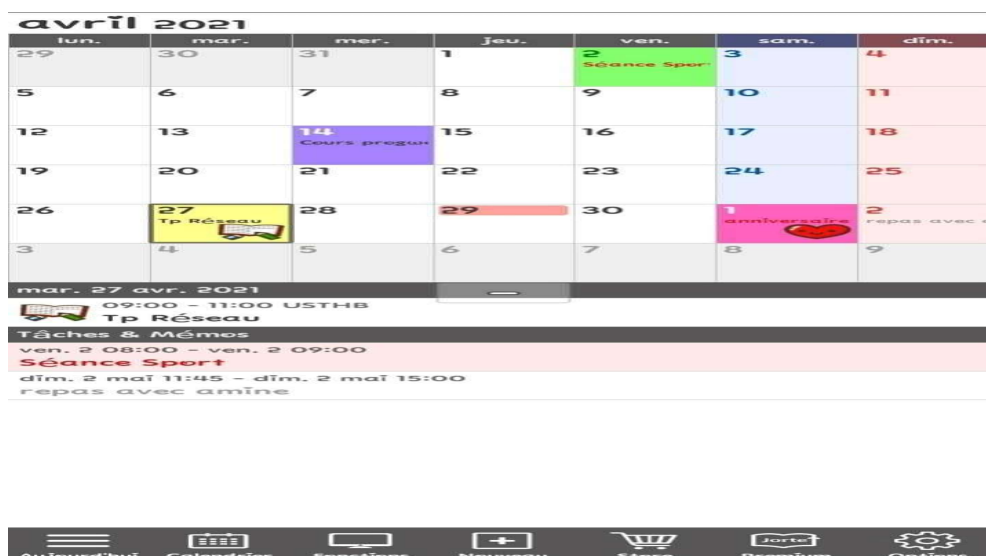


Figure 1.2 Interface JORTE

## ❖ PICNIIC

PICNIIC est un peu plus qu'une application d'agenda. C'est une application d'organisation familiale qui aide les familles à être plus organisés et bien connecté pour que tous les membres soit au courant de l'organisation en famille avec les fonctionnalités :

- carte en temps réel (localisation en temps réel des membres)
- calendrier familial, listes de courses, recettes et conseils, menus, casiers de données
- journal d'activité «Family News» : voir toutes les activités des membres de votre famille sur l'application PICNIIC (permet de suivre les tâches accomplies ...etc.)(9)

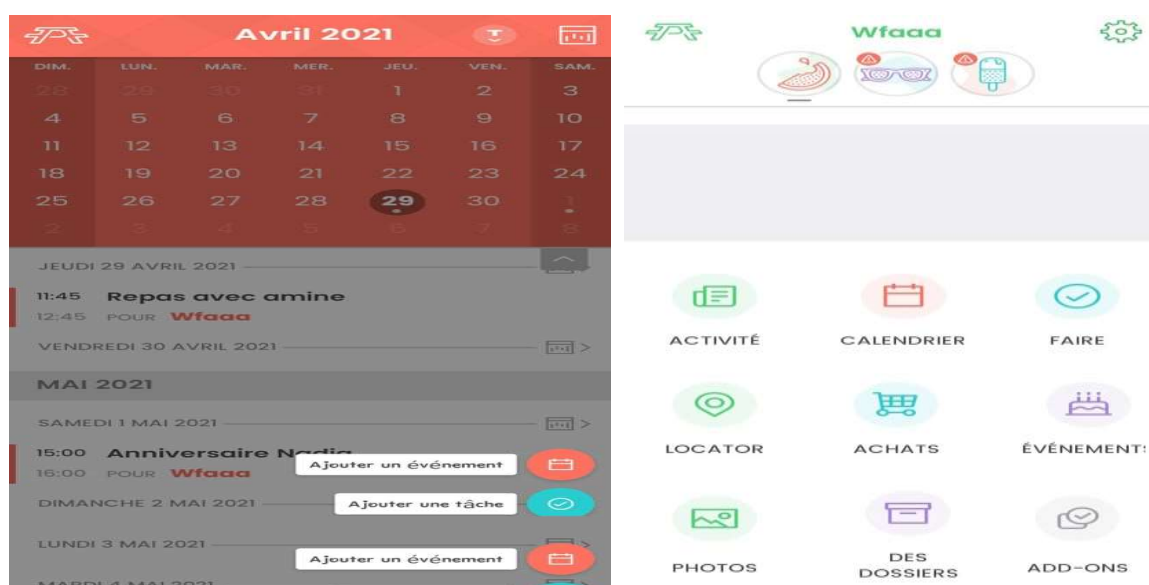


Figure 1.3 Interface PICNIIC

### 1.2.5 Avantages et inconvénients des agendas existants

<u>Outils</u>	<u>Avantages</u>	<u>Inconvénients</u>
GOOGLE AGENDAS	<ul style="list-style-type: none"><li>-gratuit et Confidentialités des données (Privat, partagés)</li><li>-création des calendriers séparés et codés par couleurs (réunion /évènements personnels)</li><li>-intégré à l'ensemble de l'environnement de Google. -recevoir des rappels, notifications.</li></ul>	<ul style="list-style-type: none"><li>-ne peut pas suivre l'état des tâches décrites</li><li>-la planification des tâches sont mélangés entre tâche professionnel et personnel</li></ul>
JORTE	<ul style="list-style-type: none"><li>-permet d'ajouter des emplacements</li><li>-dispose d'états de tâches</li><li>-permet d'ajouter automatiquement les évènements qui nous intéressent</li></ul>	<ul style="list-style-type: none"><li>-dispose des fonctionnalités payantes</li><li>-la planification ne sépare pas les tâches professionnelles des tâches personnelles</li></ul>
PICNIIC	<ul style="list-style-type: none"><li>-En plus de pouvoir suivre la position des membres ,on peut définir les lieux importants et de recevoir des notifications dès l'arrivée d'un membre.</li></ul>	<ul style="list-style-type: none"><li>-on n'a pas l'accès gratuit à toutes les fonctionnalités</li><li>-l'absence de la gestion de confidentialités : tous les membres peuvent voir toutes les données</li></ul>

**Tableau 1. Comparaison entre les agendas électroniques**

Après cette brève comparaison on constate que ces outils disponibles n'offrent pas la totalité des fonctionnalités nécessaires pour résoudre la problématique de gestion de planification et d'organisation chez les parents.

## 1.3 Conclusion

Dans ce chapitre, nous avons présenté l'agenda, ses types puis nous avons cité quelques agendas électroniques qui existent. Dans le prochain chapitre comportera la conception et l'analyse des besoins dont nous allons présenter les besoins fonctionnels et non fonctionnels afin de détecter des difficultés potentielles.

## Chapitre 2 : « Conception et modélisation »

### 2.1 Introduction

L'analyse et la spécification des besoins est une phase importante dans le développement d'une application. Pour réaliser n'importe quel projet il faut faire d'abord une étude complète sur l'analyse et la conception des besoins de ce projet, afin de décrire et expliquer tous les besoins des utilisateurs.

Dans ce chapitre nous allons spécifier l'ensemble des besoins fonctionnels et non fonctionnels liées à notre application afin d'élaborer un cahier de charge qui décrit notre application de façon précise et complète à l'aide des diagrammes des cas d'utilisation et de séquence. Ensuite nous passons à la phase de conception de notre application représentée par le diagramme des classes.

### 2.2 Problématique

La plupart des gens souffrent de la mauvaise organisation de leurs temps entre la charge de la vie professionnelle et personnelle surtout pour les femme qui doivent gérer leur vie de maman et les tâches en travail et en famille :Liste d'achat, suivie d'état de santé des enfants malades (heure de prise de leurs médicaments ,leurs rendez-vous chez le médecin ) ou même leurs activités culturelles (par exemple les séances de sport, les cours de soutiens, ou les programmes avec leurs amis...etc.), gérer les Rendez-vous de travail (rappel des dates clés, des réunions de suivi de projet, des réunion avec les collaborateurs...etc.) et les occasions familiales (un mariage, des anniversaires, rencontre entre membre famille...etc.). En plus des travaux ménagers peuvent être listés comme des tâches urgentes à accomplir avant une date limite (paiement des factures,...etc.).

#### Objectifs

Le but de la réalisation de ces agendas électroniques est d'assurer une meilleure planification des tâches chez les parents. Pour cela nous séparons les tâches professionnelles des tâches personnelles. Parmi les objectifs de réalisation de cette application nous situons :

- Planifier le travail quotidien pour gagner du temps et atteindre vos objectifs personnels et professionnels. Il permet également une collaboration et une communication beaucoup plus fluides entre les membres de la famille.
- Lister les tâches à réaliser en avance, et les partager entres les membres de familles.
- Générer des notifications pour informer l'utilisateur lors d'un événement, ou le délai limite d'une tâche.
- Faciliter le suivi collectif des enfants par les parents et la nourrice (ou la personne qui garde l'enfant) : leur état de santé (température, poids, médicaments, les cours à suivre pour chaque enfant en plus des activités sportives ou culturelles...etc.).
- Nous définissons ci-dessous. les termes utilisés dans notre application
  - ❖ **Un évènement** : est un fait personnel ou professionnel, caractérisé par une date de déroulement, un lieu et une description. Il peut être partagé avec

d'autres utilisateurs et il sera affiché sur le calendrier de ces derniers.  
Exemple : « Anniversaire ».

- ❖ **Une tâche** : est un travail à faire avant un délai précis, caractérisé par une date limite un titre, le choix de partage avec des utilisateurs, et des rappels  
Exemple : « paiement d'une facture ».
- ❖ **Liste d'achat** : est une liste des produits à acheter.
- ❖ **Programme** : inclus les activités sportives et culturelles et rdv (médecin, nourrice...etc.) pris pour l'enfant
- ❖ **Santé** : pour suivre état de santé de l'enfant quand il tombe malade.
- ❖ **Liste contact** : ensembles des emails des utilisateurs avec qui on partage les données.

## 2.3 Architecture de système

Dans cette section nous présentons l'architecture globale de notre système, qui est composé principalement de :

- Un Serveur.
- Une base de données.
- Client.

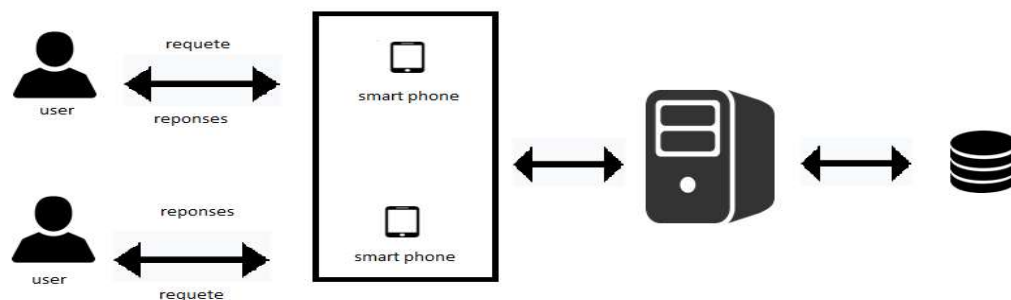


Figure 2. 1 Architecture de système

## 2.4 Définition des besoins

La spécification des besoins représente la première phase du cycle de développement d'une application. Elle doit décrire sans ambiguïté l'application à développer.

### 2.4.1 Spécification des besoins fonctionnels

Dans cette section nous représentons les besoins fonctionnels auxquels doit répondre notre application.

Notre application comprendra les fonctionnalités suivantes :

- L'authentification par un compte Google .

- Partage avec des personnes utilisateurs. Par exemple partager la liste d'achat, ou une tâche commune à faire avec les membres de la famille, un événement entre plusieurs personnes, partager des données d'un enfant avec sa nourrice...etc.
- Gestion des événements personnels et professionnels. Ajouter un événement avec un rappel.
- Afficher la liste des événements.
- Gestion de tâches personnelles avec notification avant le délai limite.
- Affichage de la liste de tâches personnelles.
- Gestion enfants : créer un profil enfant et ajouter des rappels de prise de médicaments et des activités ou rendez-vous pris pour l'enfant.
- Consulter les profils enfants, modifier les informations.
- Gestion de la liste des contacts (ajout, suppression).

### 2.4.2 Analyse des besoins non fonctionnels

Les besoins non fonctionnels spécifient les propriétés du système telles que les contraintes sous lesquelles le logiciel doit opérer, et les relier aux besoins fonctionnels. Par exemple, la performance, la maintenance, l'extensibilité et la flexibilité, contrainte de temps, espace mémoire, représentation des données, langage.

Certains besoins non fonctionnels sont généraux et ne peuvent pas être rattachés à un cas d'utilisation particulier.

#### ➤ Contraintes ergonomiques :

Liées à l'adaptation entre les fonctionnalités de l'application, leurs interfaces et leur utilisation. Pour notre application, nous devons obéir aux contraintes suivantes :

- Permettre un accès rapide de l'information.
- Interface simple et compréhensible.
- L'organisation des rubriques, des volets, ...etc.

#### ➤ Contraintes techniques :

- Il faut que toute interface de l'application soit homogène, les différentes pages (activités) doivent suivre le même modèle de représentation.
- Le code doit être extensible et maintenable pour faciliter toute opération d'amélioration ou d'optimisation.

#### ➤ Contraintes de déploiement :

- notre application doit être téléchargeable.

## 2.5 Identification des acteurs

Vu la description du projet nous distinguons un seul acteur externe qui est l'utilisateur. Pour que l'utilisateur dispose de toutes les fonctionnalités de cet agenda il doit se connecter à un compte Google.

Un utilisateur peut consulter son agenda, ajouter un événement, le partagé avec un autre utilisateur pour être affiché sur le calendrier de ce dernier. En plus il peut ajouter un enfant, consulter ses données.

## Présentation UML

Le Langage de Modélisation Unifié (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet, destiné principalement à faciliter la conception des documents nécessaires au développement d'un logiciel orienté objet. (10)

### 2.6 Diagramme de cas d'utilisations

Les diagrammes de cas d'utilisation (DCU) sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. En effet, un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur et un système. (11)

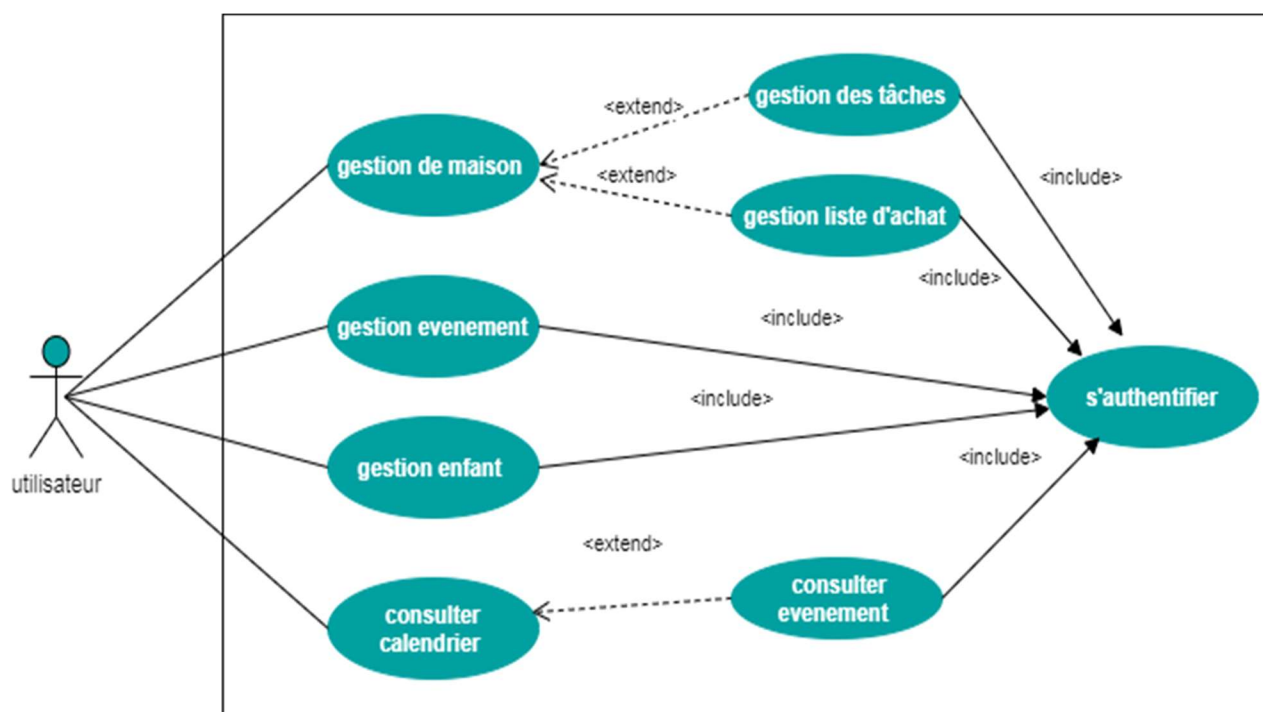


Figure 2. 2 Diagramme des cas d'utilisation du system

Le cas d'utilisation	D'authentification
Acteurs	User
Scénario d'exécution	Utilisateur clique sur « se connecter » le compte ouvert sur l'appareil sera récupéré automatique et de même on a la possibilité de choisir un autre compte Google

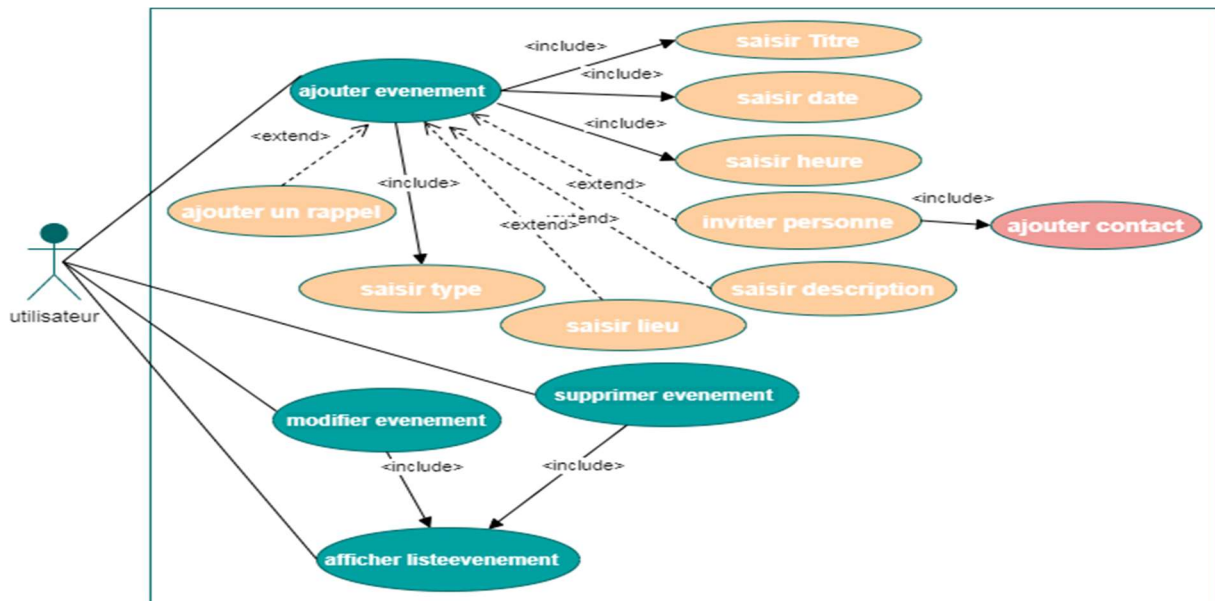


Figure 2. 3 Diagramme des cas d'utilisation de gestion des évènements

cas d'utilisation	gestion évènement
Acteurs	User
Scénario nominal	<ol style="list-style-type: none"> <li>1- utilisateur accède au menu dont on a 3 fonctionnalités (gestion maison, évènement, Suivre enfant)</li> <li>2- l'utilisateur sélectionne la barre d'évènement</li> <li>3- une interface s'affiche pour remplir les données de l'évènement</li> <li>4- une fois cliquant sur le bouton valider l'évènement sera inséré dans la base de données et s'affichera sur calendrier et dans la liste des évènements</li> <li>5- dans la liste des évènements on a la possibilité de la modification et la suppression de chaque évènement</li> </ol>
Scénario alternative	Si la date et l'heure de l'évènement sont occupées un message s'affiche informant l'utilisateur que cette date est déjà occupée et on donne le choix s'il veut l'enregistrer ou l'annuler

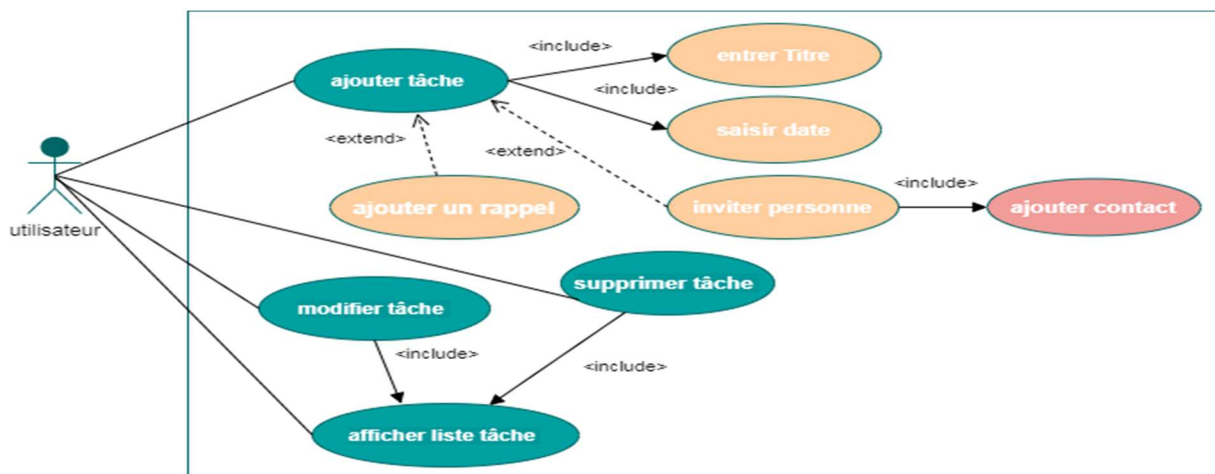


Figure 2. 4 Diagramme des cas d'utilisation de gestions des tâches

cas d'utilisation	Gestion des tâches
-------------------	--------------------

Acteurs	user
Scénario nominal	<p>1- Une interface s'affiche dont on a 3 fonctionnalités (gestion maison, événement, Suivi enfant)</p> <p>2- L'utilisateur sélectionne gestion maison puis la barre des tâches</p> <p>3- Pour ajouter une tâche l'utilisateur saisi les informations</p> <p>4- Une fois cliquant sur le bouton valider la tâche sera insérée dans la base de données et s'affichera dans la liste des tâches</p> <p>5- Pour consulter les tâches on clique sur l'icône liste</p> <p>6- Une liste des tâches s'affiche avec la possibilité de modification ou de suppression de chaque tâche.</p>

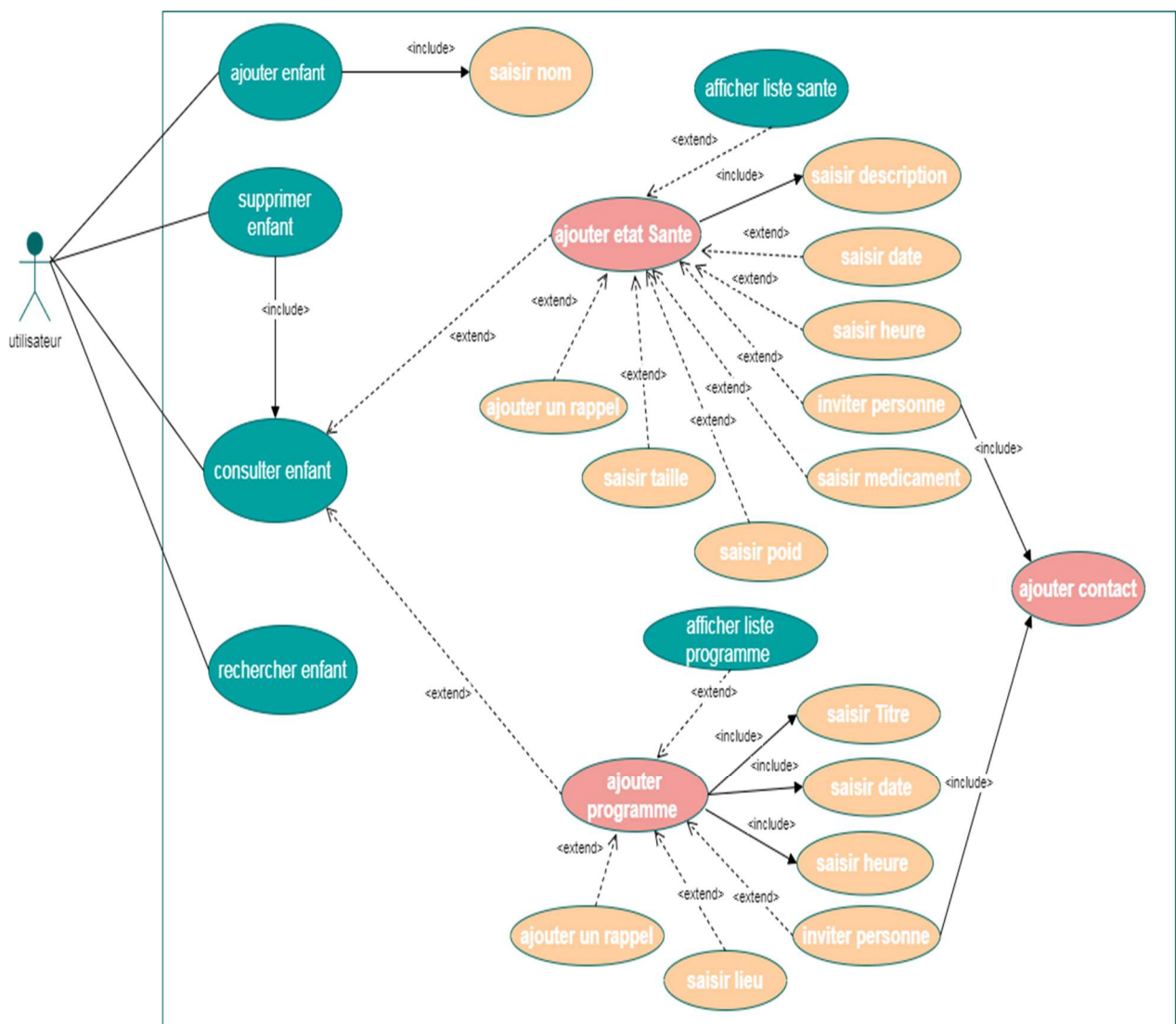


Figure 2. 5 Diagramme des cas d'utilisation de suivi des enfants



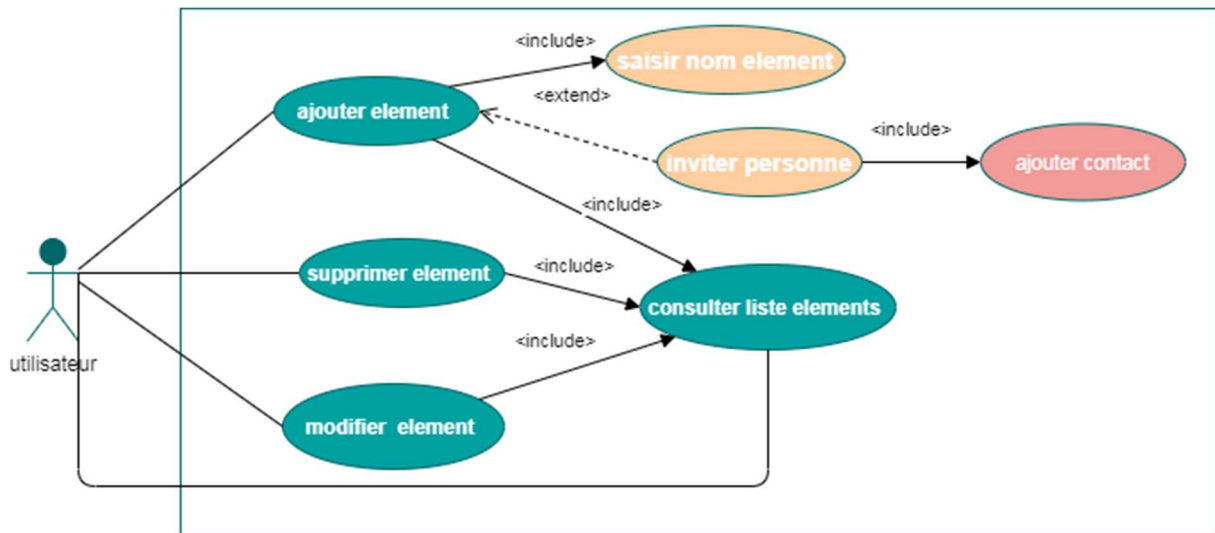


Figure 2. 6 Diagramme des cas d'utilisation de gestions des listes des cours

## 2.7 Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML, il permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets. La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent à l'aide de messages. (12)

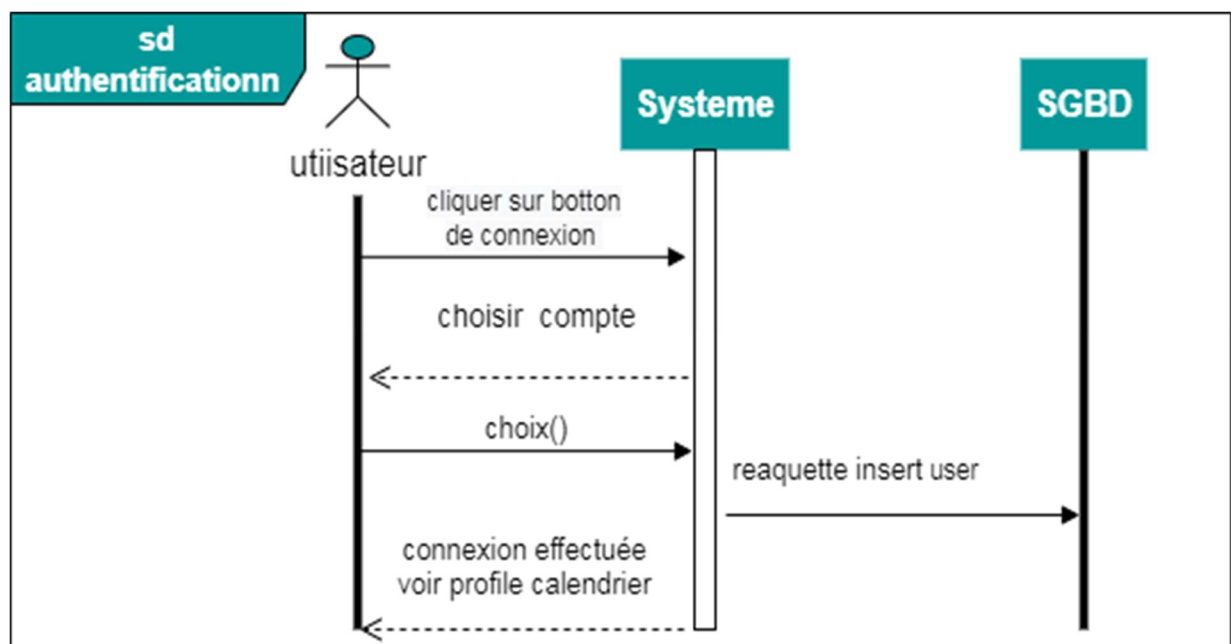


Figure 2. 7 Diagramme de séquence de cas d'utilisation d'authentification

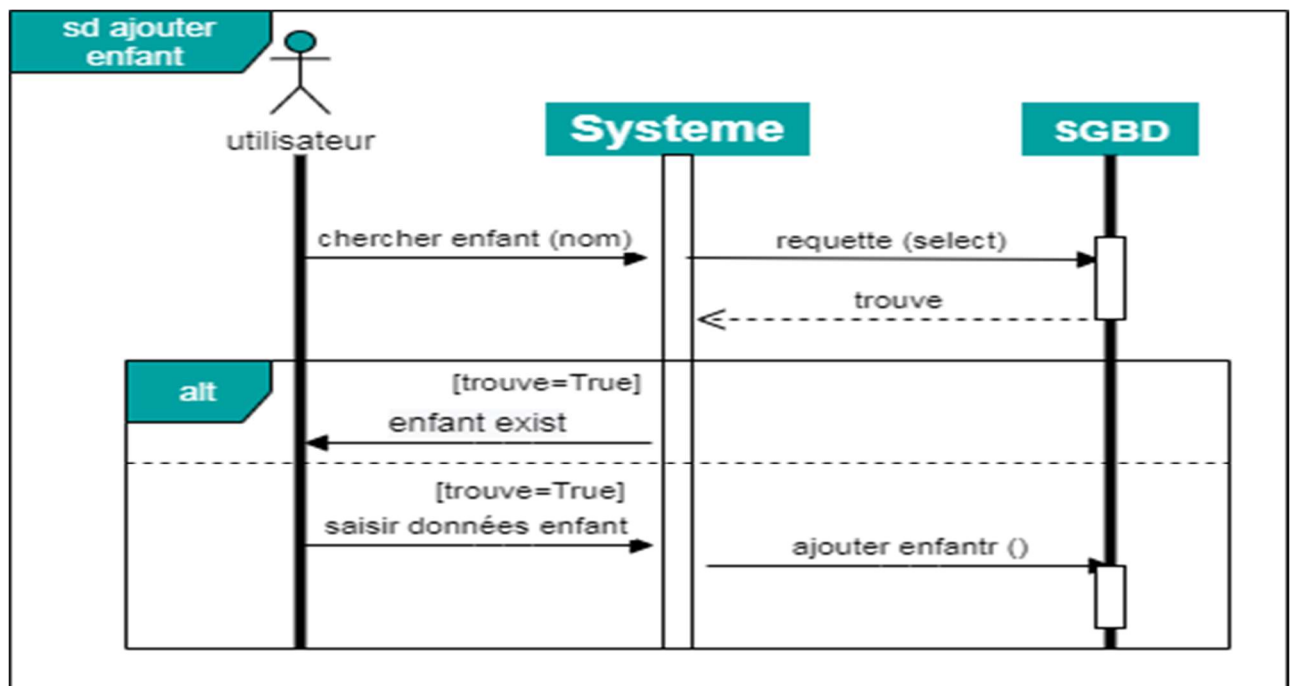


Figure 2. 8 Diagramme de séquence d'ajout d'enfant

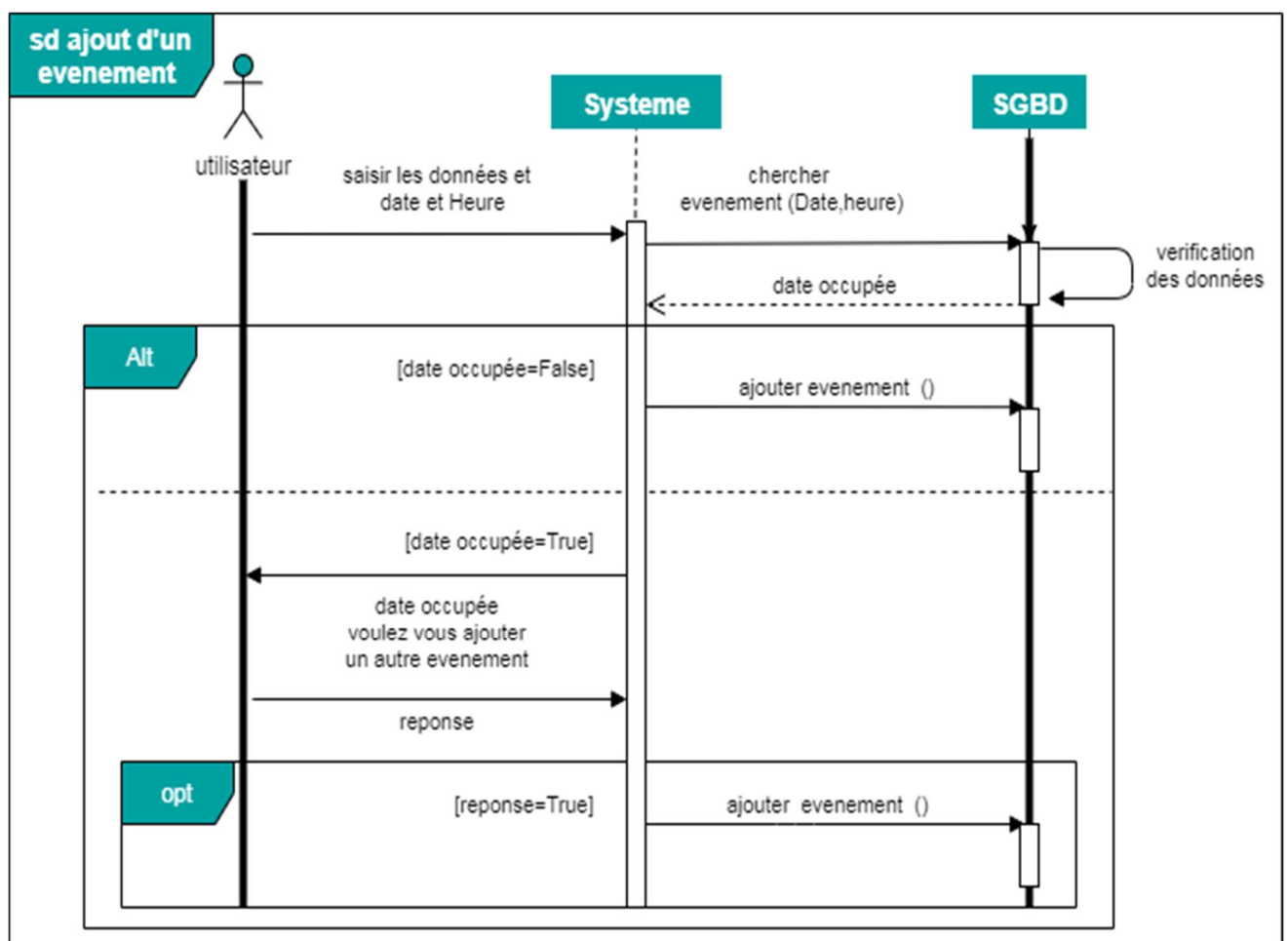


Figure 2. 9 Diagramme de séquence d'ajout d'évènement

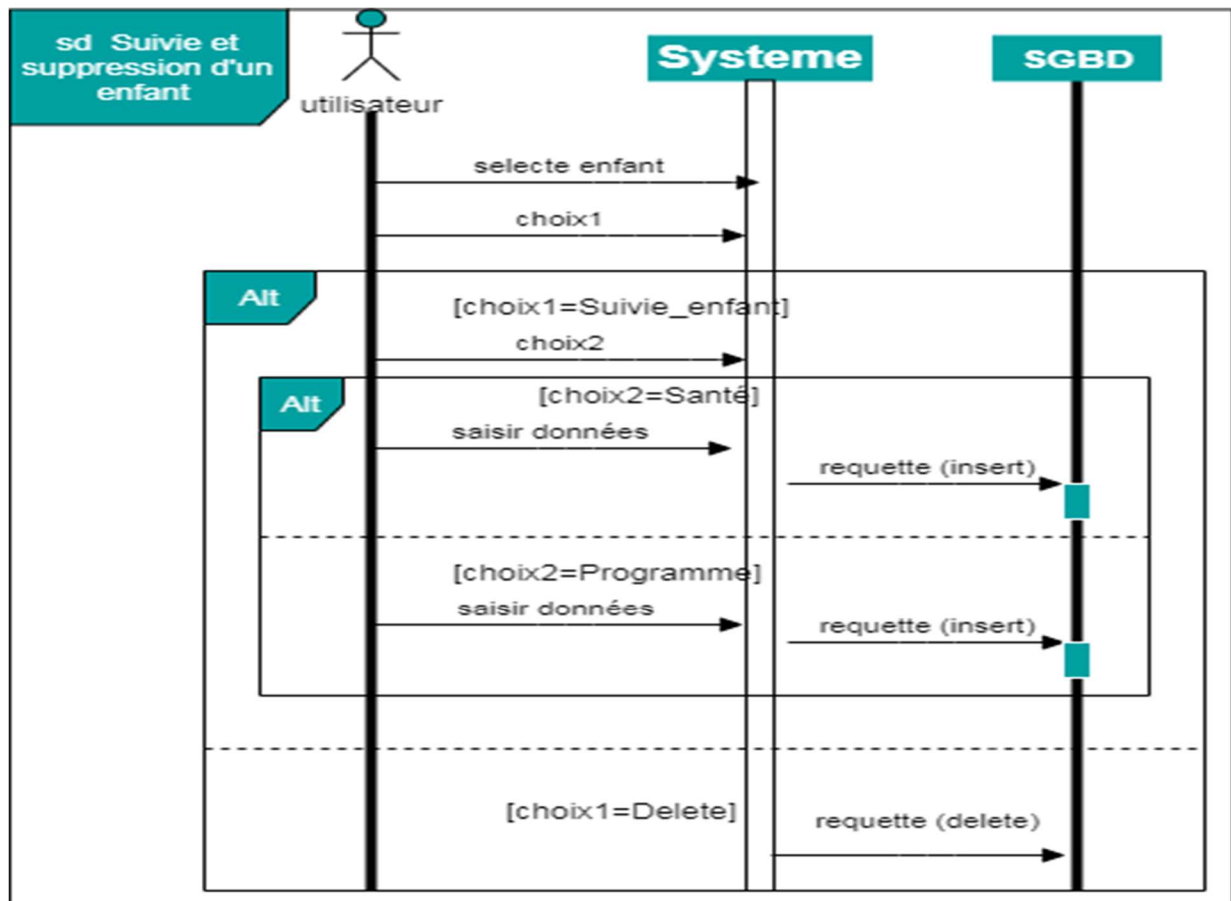


Figure 2. 10 Diagramme de séquence de suivi d'un enfant et sa suppression

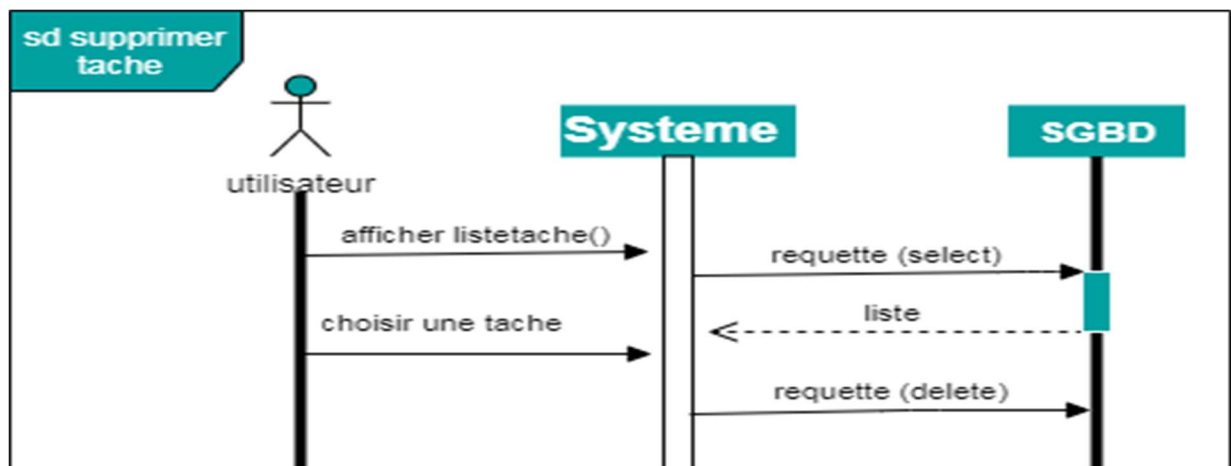


Figure 2. 11 Diagramme de séquence supprimer tâche

## 2.8 Diagramme de classe

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un

même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système.

Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances.(13)

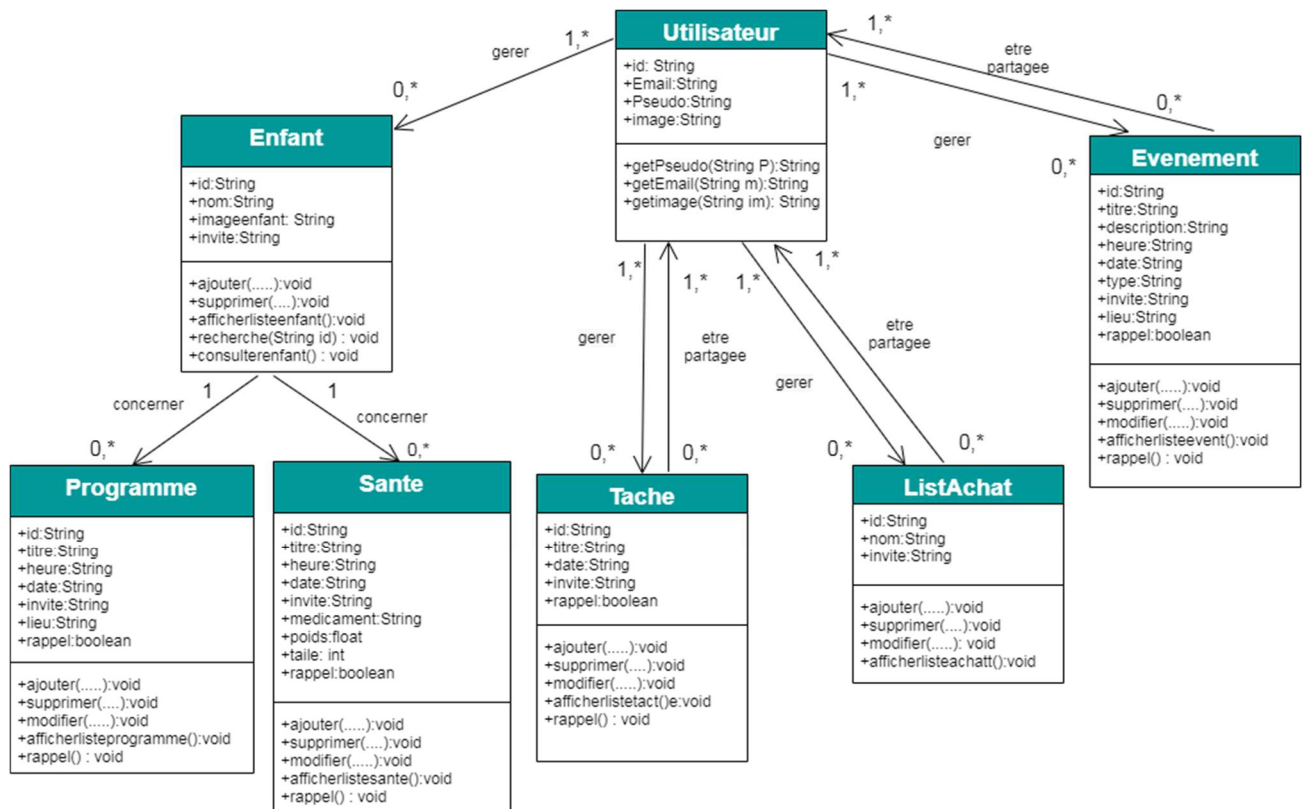


Figure 2. 12 Diagramme de classe

## 2.9 Modèle relationnel

### 2.9.1 Définition

Le modèle relationnel est une manière de modéliser les informations contenues dans une base de données qui repose sur des principes mathématiques inventés par E.F. Codd. Ce qu'on appelle relation est un ensemble d'attributs qui définissent un fait. Chaque instance est appelée un tuple. Les relations sont d'ordinaire représentées sous forme tabulaire, et l'on confond table et relation; on confond de même, ligne dans la table et tuple. Par définition, chaque tuple d'une relation est unique, et est identifié par un ou une combinaison de plusieurs attributs qui forme la clef. L'ordre des tuples est non significatif.(14)

Règles de passage du modèle conceptuel au modèle relationnel

- Règle 1 :
  - toute entité est transformée en table

- Ses propriétés deviennent des champs de la table
- L'identifiant devient la clé primaire de la table
- **Règle 2 :**
  - Toute association binaire de type  $\{*,1\} \{*, N\}$  (\* peut être 0 ou 1) va disparaître et on duplique l'identifiant de l'entité du côté (\*, N) dans celle du côté (\*,1)
- **Règle 3 :**
  - Toute association binaire de type  $[(*, N), (*, N)]$  devient une table ayant comme clé le couple composé par les deux clés des deux tables issues des entités
- **Règle 4 :**
  - Dans une association à trois participants ou plus, l'association devient une table ayant comme clé le composé des clés de toutes les tables issues des entités participantes à cette association.

Notre schéma relationnel est le suivant :

- **Utilisateur** (iduser, Email, Pseudo)
- **Evenement** (idevent, titre, description, lieu, date, heure, type, invite, rappel, iduser)
- **Tache** (idtache, titre, date, invite, rappel, iduser)
- **ListeAchat** (idliste, nomelement, invite, iduser)
- **Enfant** (idenfant, nomenfant, iduser, invite)
- **Programme** (idprogramme, idenfant, titre, date, heure, lieu, invite, rappel, iduser)
- **Sante** (idsante, idenfant, titre, date, heure, medicament, poids, taille, invite, rappel, iduser)

## 2.10 Conclusion

La phase conceptuelle est une étape fondamentale pour la réalisation de n'importe quel projet. Elle permet de représenter les différentes étapes de conception et considérée comme préparation pour la phase de la réalisation et l'implémentation de la base de données. Car pour programmer n'importe quel projet informatique, il ne convient pas de commencer avec l'écriture de code, il nécessite l'organisation des idées et la recherche des moyens et les outils possibles pour développer le projet, puis démarrer la réalisation du travail.

Par la suite nous présenterons les détails de réalisation et implémentation du projet.

## **Chapitre 3 : « Implémentation et réalisation »**

### **3.1 Introduction**

Après avoir élaboré la conception de notre application, dans ce chapitre nous allons présenter notre application ainsi les outils de développement et les langages de programmation utilisés pour la réalisation de notre projet. Nous finirons ce chapitre par présenter les interfaces obtenus pour illustrer le fonctionnement de notre application.

### **3.2 Les outils de développement**

#### **3.2.1 Définition Android studio**

Android studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux. Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android. (15)

#### **3.2.2 Définition XML**

Le XML, pour Extensible Markup Language, désigne un langage informatique (ou métalangage pour être plus précis) utilisé, entre autres, dans la conception des sites Web et pour faciliter les échanges d'informations sur Internet. Ce langage de description a pour mission de formaliser des données textuelles. Il s'agit, en quelque sorte, d'une version améliorée du langage HTML avec la création illimitée de nouvelles balises. Le XML se classe dans la catégorie des langages de description. Il est donc naturellement utilisé pour décrire des données en s'appuyant sur des balises et des règles personnalisables. (16)

#### **3.2.3 Langage JAVA**

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation. Il donne la possibilité de développer des programmes pour téléphones portables. Enfin, il peut-être utilisé sur internet pour des petites applications intégrées à la page web ou encore comme langage serveur. (17)

#### **3.2.4 FIREBASE**

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. (18)



Figure 3. 1 Logo firebase

### 3.2.5 NoSQL

Le NoSQL, pour "not only SQL", développé à l'origine pour gérer du big data, l'utilisation de NoSql a explosée depuis quelques années et désigne une nouvelle manière de stockage pour cela il existe la famille clé /valeur pour stocker nos données, de tel sort que la clé identifie la data de manière unique et la valeur peut contenir n'importe quel type de donnée. (19)

## 3.3 La présentation de quelques interfaces de l'application

### 3.3.1 Page de connexion et déconnexion

Après avoir cliqué sur « Se connecter » notre compte Google sera récupérée automatiquement ou on peut choisir un autre compte, ensuite une interface se génère dont on a le choix soit de déconnecter en cliquant sur « LOGOUT » ou entrer dans l'application en cliquant sur le bouton arrondi.



Figure 3. 2 Interface login



Figure 3. 3 Interface déconnexion

### 3.3.2 Interface calendrier et menu

La première interface dans notre application contient un calendrier dont chaque date comporte tous les événements programmés dans ce jour, ainsi un menu qui nous permet d'aller à n'importe fonctionnalité dans l'application.

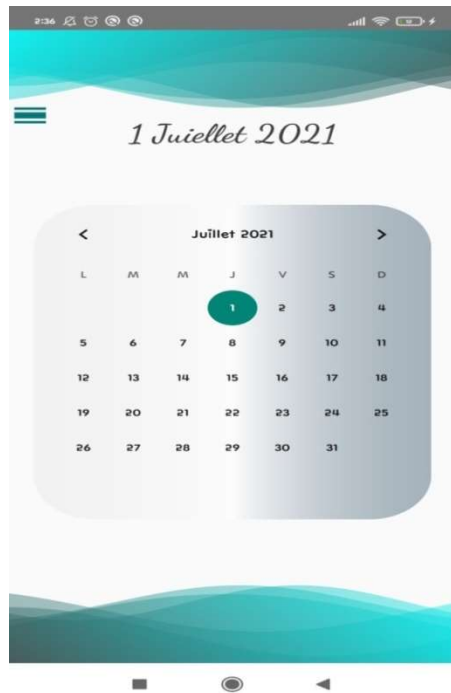


Figure 3. 4 Interface calendrier

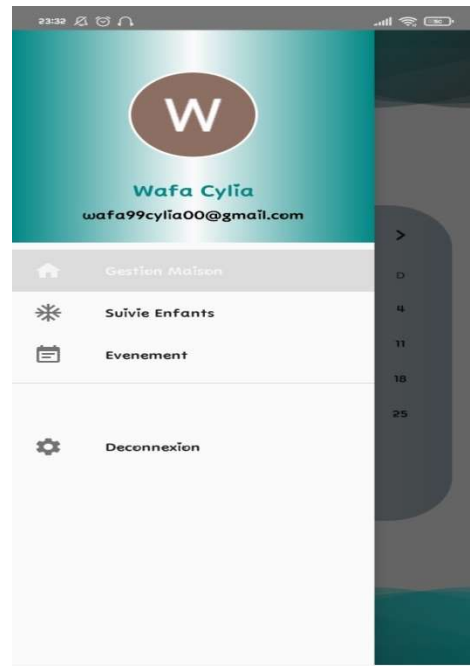


Figure 3. 5 Interface menu

### 3.3.3 Interface des évènements

Après avoir remplir et sauvegarder un évènement, on peut consulter la liste des évènements existants ainsi on a une possibilité soit de supprimer ou modifier chaque évènement.

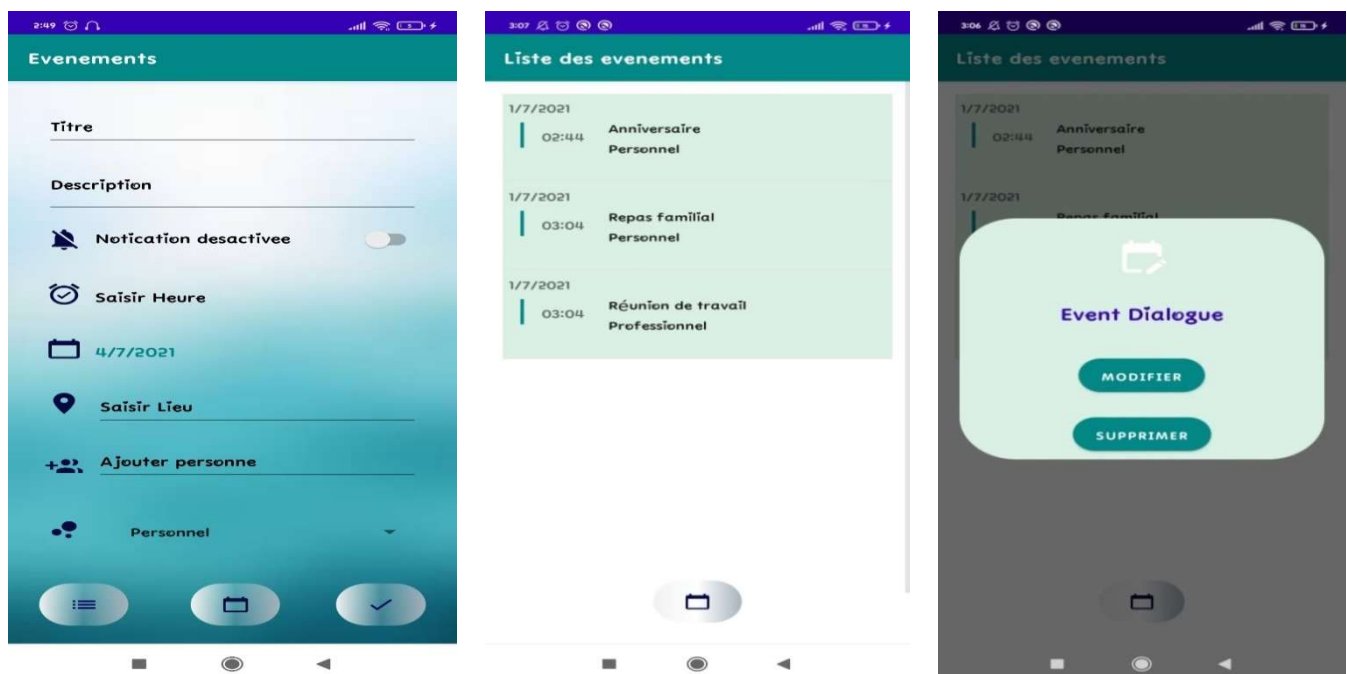


Figure 3. 6 Interface évènement



### 3.3.4 Interface gestion de maison

Ce volet comporte deux fonctionnalités :

#### ❖ Liste Tâches

Il faut remplir les champs de cette interface afin de créer une tâche, après l'avoir sauvegardée, on peut consulter la liste des tâches existantes ainsi on a une possibilité soit de supprimer ou modifier chaque tâche.

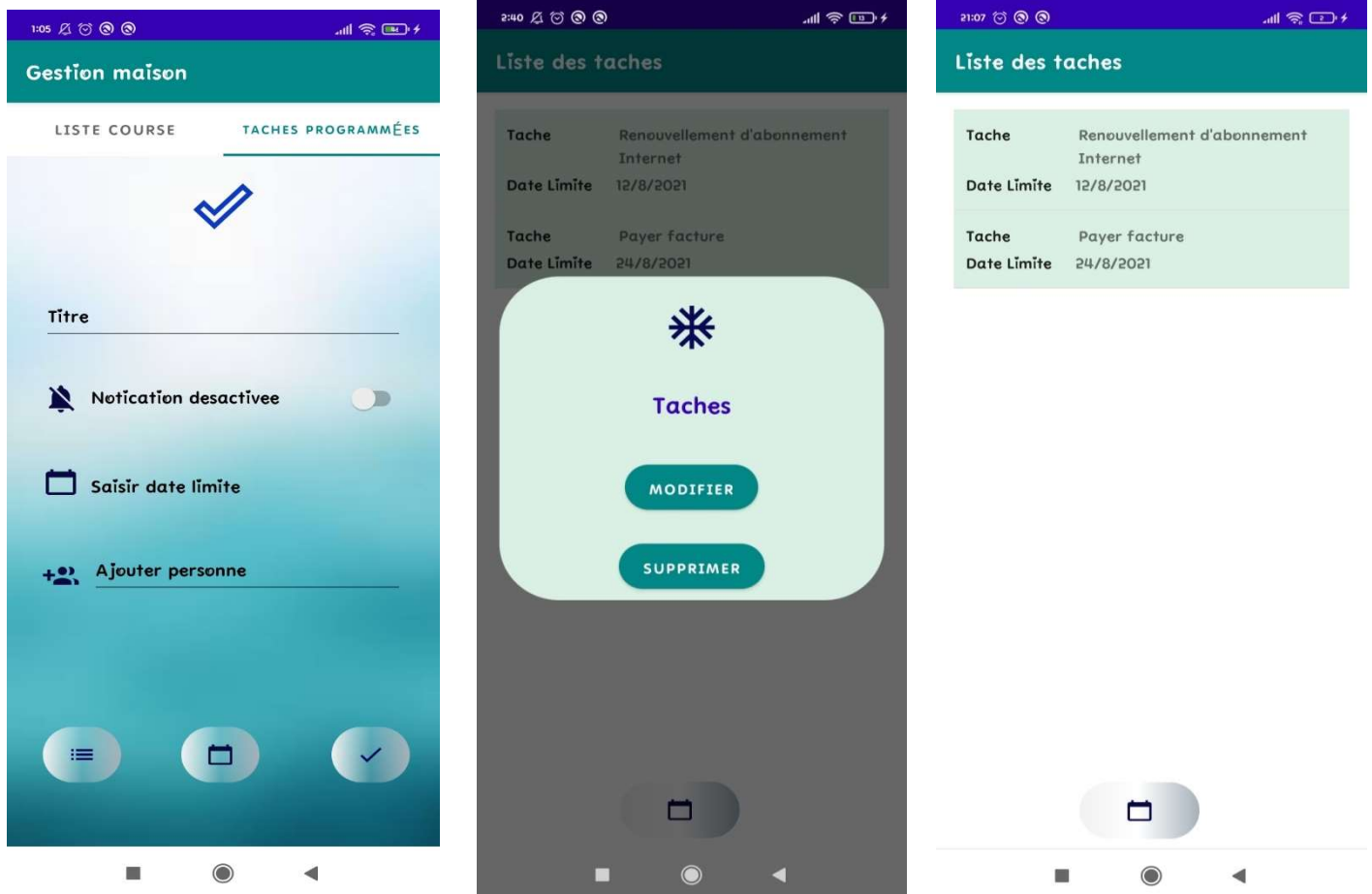


Figure 3. 7 Interface tâche

#### ❖ Interface de liste achat

Chaque utilisateur saisit un élément qui sera enregistré dans la liste avec une possibilité de partage de ce dernier avec une autre personne, comme on a la possibilité soit de modifier ou supprimer l'élément entré.

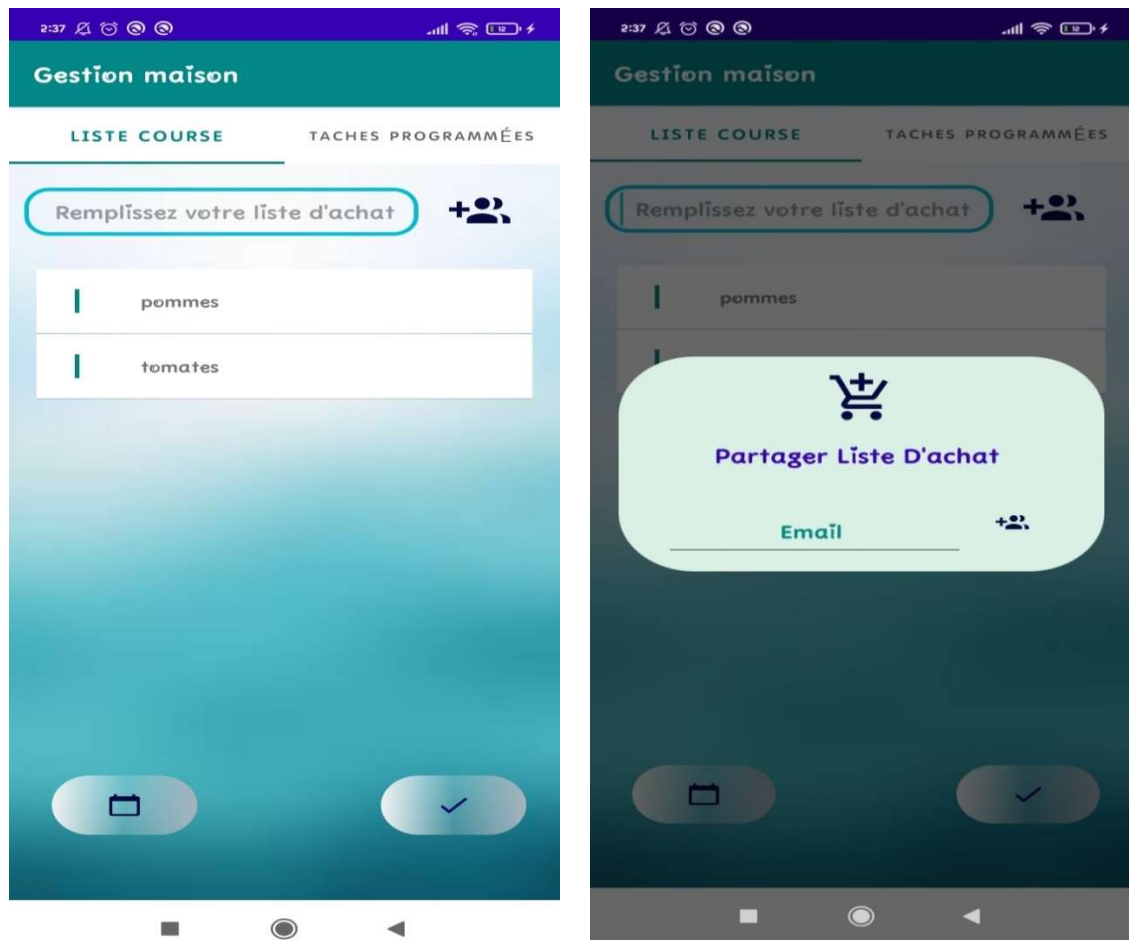


Figure 3. 8 Interface liste d'achat

### 3.3.5 Interfaces suivi enfant

Cette interface contient des enfants identifier par leur image et nom, on fait une recherche sur un enfant en saisissant son nom dans la barre de recherche. Chaque enfant a son profile dont on peut ajouter leur programme ou l'état de leur santé et ça se fait en cliquant sur « Suivi » comme on peut le supprimer dans la liste en cliquant sur « supprimer ». L'icône de partage nous permet de partager chaque enfant avec une autre personne pour faciliter le suivi d'enfant en entrant l'adresse email de cette personne.

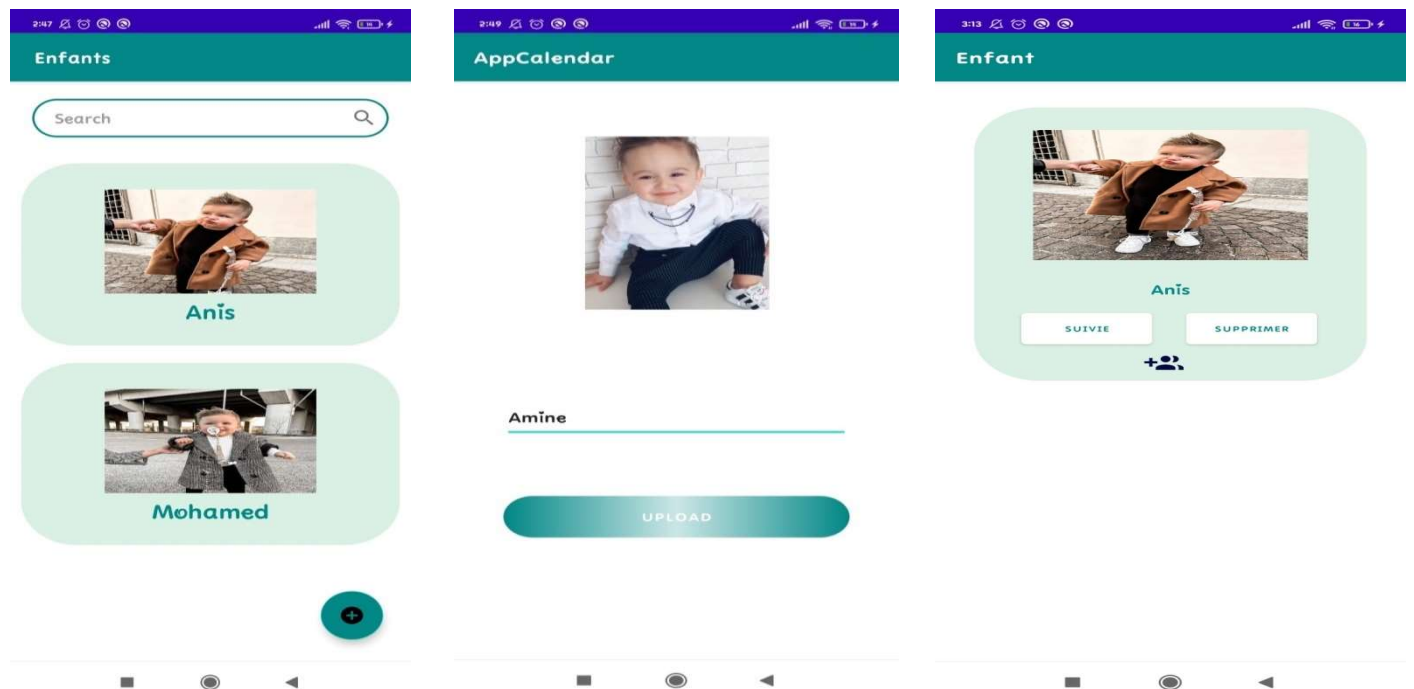


Figure 3. 9 Interface suivie enfant

### ❖ Profile des programmes d'un enfant

Il faut remplir les champs de cette interface afin de créer un programme, après l'avoir sauvegardé, on peut consulter la liste des programmes existants pour chaque enfant ainsi on a une possibilité soit de supprimer ou modifier chaque programme.

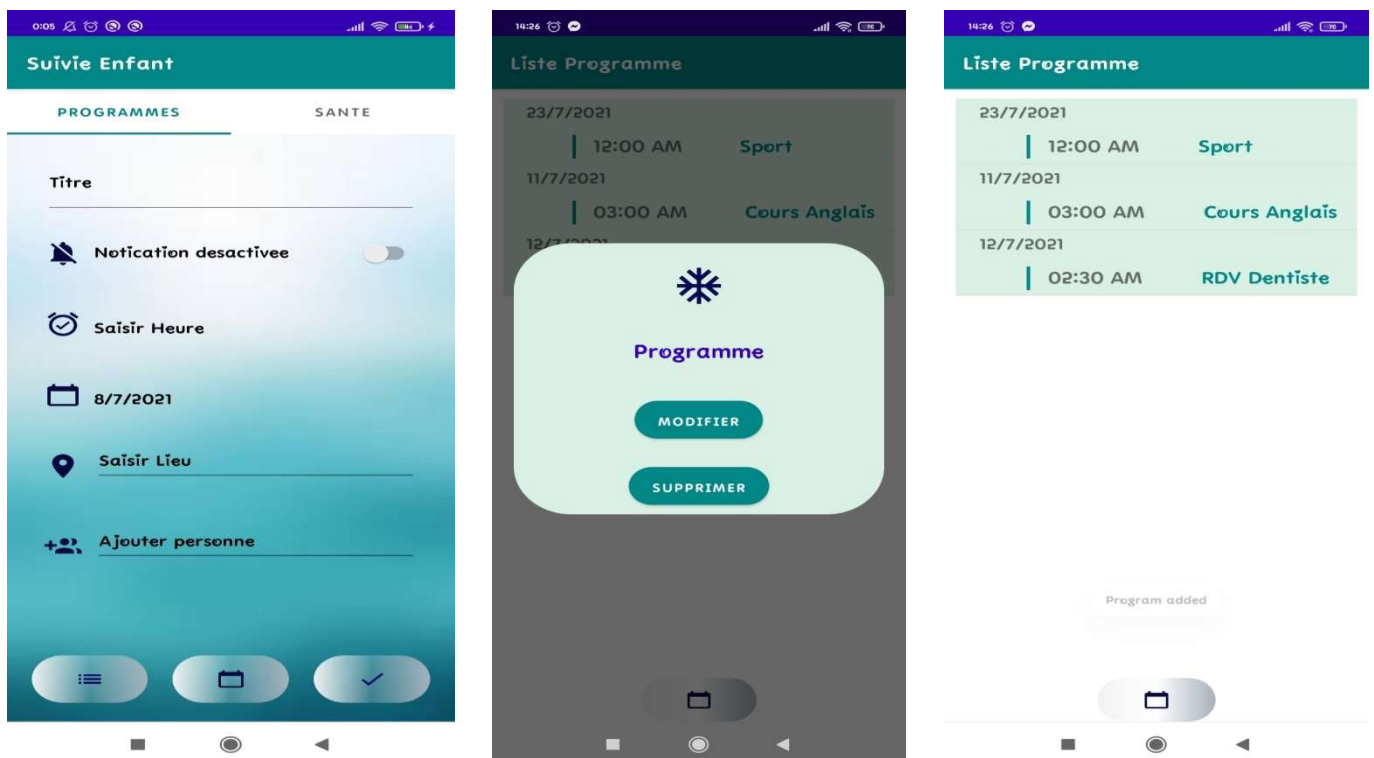


Figure 3. 10 Interface programme d'un enfant

### ❖ Profile d'état santé d'un enfant

Il faut remplir les champs de cette interface afin de créer un évènement d'état de santé, après l'avoir sauvegardé, on peut consulter la liste d'état de santé existants ainsi on a une possibilité soit de supprimer ou modifier chaque évènement d'état de santé.

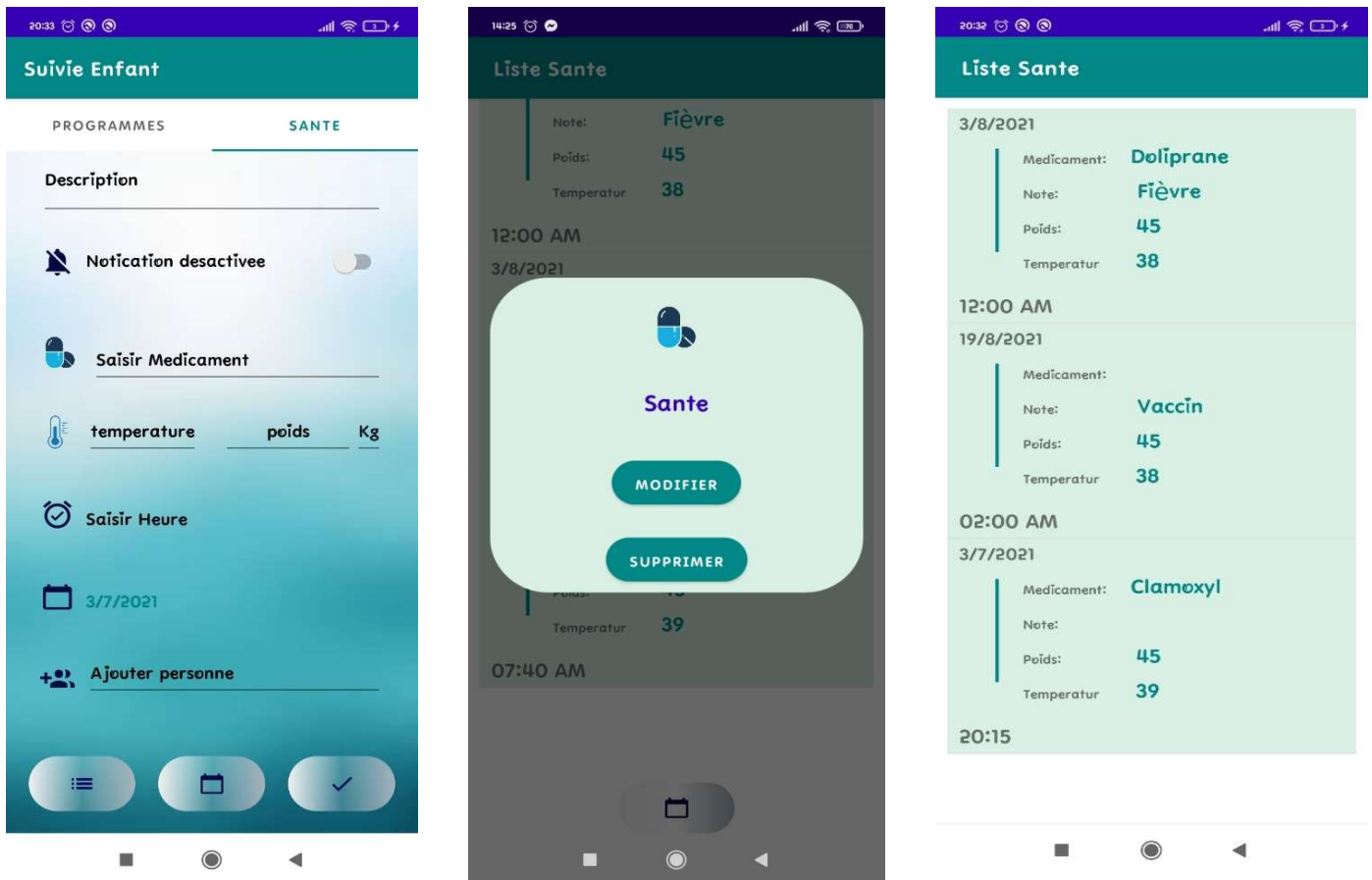


Figure 3. 11 Interface d'état de santé d'un enfant

## 3.4 Conclusion

Dans ce chapitre nous avons présenté les différents outils et langages utilisés pour l'implémentation de notre application, ainsi on a illustré les fonctionnalités principales par des captures d'écrans des interfaces réalisées.

## *Conclusion Générale*

L'objectif visé dans ce projet de fin d'étude est la mise en place d'une application mobile afin d'offrir un moyen qui permet à une personne d'avoir une meilleure organisation et plus de liberté d'esprit. Grâce à cette application chaque parent aura une grande possibilité d'améliorer son quotidien entre la vie personnel : le suivi des affaires de maison et vie de ses enfants, et professionnel : réunion et rendez-vous...etc. En effet notre application vise essentiellement la tranquillité d'esprit chez les parents.

Ce projet a fait l'objet d'une expérience intéressante qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation, faire des recherches, apprendre à gérer un projet, transformer les aspects théoriques en pratique et d'implémenter un projet en utilisant un langage de modélisation UML, XML, android studio, Firebase.

Pour atteindre cet objectif nous avons abordé en premier lieu une phase d'étude générale sur le domaine de notre application, en deuxième lieu on a réalisé la conception de cette dernière et on a spécifié ses différents fonctionnalités, en fin ce qui concerne la réalisation on a utilisé :

- Le langage JAVA sur android studio
- XML
- Le langage de modélisation UML
- Système de gestion de base de données Firebase

Notre application peut être améliorée en ajoutant d'autres fonctionnalités, par exemple :

- Un système chat (SMS, APPEL)
- Partage des vidéos
- Intégrée d'autres langues
- Développer l'application mobile en une application web
- Généraliser l'application pour par exemple le suivi des malades dans un centre pour personnes âgées (entre autres malades)

- [1]. Agenda. *Wikipédia*. [En ligne]  
[https://fr.wikipedia.org/wiki/Agenda?fbclid=IwAR1\\_jBh3MuIPau4EjT2RMdT-fChgx4ChnudBAD4xAatrT79QCr1AHD\\_8As](https://fr.wikipedia.org/wiki/Agenda?fbclid=IwAR1_jBh3MuIPau4EjT2RMdT-fChgx4ChnudBAD4xAatrT79QCr1AHD_8As).
- [2]. lagenda papier vs lagenda numerique. *helloootoday*. [En ligne]  
<https://helloootoday.com/2020/01/07/lhttps://helloootoday.com/2020/01/07/lagenda-papier-vs-lagenda-numerique>.
- [3]. comment choisir le meilleur agenda. *matthieudesroches*. [En ligne]  
<https://matthieudesroches.com/articles/comment-choisir-le-meilleur-agenda>.
- [4]. choix agenda. *www.magazine-racines*. [En ligne] <https://www.magazine-racines.fr/choix-agenda/>.
- [5]. avantage agenda ligne. *www.multi-planning*. [En ligne] <https://www.multi-planning.com/blog/avantage-agenda-ligne/>.
- [6]. bien choisir son agenda partage. *www.acollab*. [En ligne] <https://www.acollab.com/bien-choisir-son-agenda-partage/>.
- [7]. applications agenda calendrier. *www.codeur*. [En ligne]  
<https://www.codeur.com/blog/applications-agenda-calendrier/?fbclid=IwAR1p9MZXCP3UZf0erqzfF4A5sRekG0FR8pX9kmFM8QnJh5WPK0-4I6Fq3QE>.
- [8]. test jorte. *www.nextpit*. [En ligne] <https://www.nextpit.fr/test-jorte>.
- [9]. organisateur familial organisation familiale. *maitriser son temps*. [En ligne]  
<https://maitriser-son-temps.com/organisateur-familial-organisation-familiale/>.
- [10]. UML (informatique). *wikipedia*. [En ligne]  
[https://fr.wikipedia.org/wiki/UML\\_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)).
- [11]. Diagramme de cas d'utilisation. *wikipedia*. [En ligne]  
[https://fr.wikipedia.org/wiki/Diagramme\\_de\\_cas\\_d'utilisation](https://fr.wikipedia.org/wiki/Diagramme_de_cas_d'utilisation).
- [12]. Diagramme de séquence. *wikipedia*. [En ligne]  
[https://fr.wikipedia.org/wiki/Diagramme\\_de\\_s%C3%A9quence](https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence).
- [13]. Diagramme de classes. *wikipedia*. [En ligne]  
[https://fr.wikipedia.org/wiki/Diagramme\\_de\\_classes](https://fr.wikipedia.org/wiki/Diagramme_de_classes).
- [14]. 312.htm. *www.techno-science.ne*. [En ligne] <https://www.techno-science.net/definition/312.htm>.
- [15]. Android Studio. *wikipedia*. [En ligne] [https://fr.wikipedia.org/wiki/Android\\_Studio](https://fr.wikipedia.org/wiki/Android_Studio).
- [16]. 1203615 xml extensible markup language definition traduction. *www.journaldunet*. [En ligne] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction/>.
- [17]. internet java. *www.futura-sciences*. [En ligne] <https://www.futura-sciences.com/tech/definitions/internet-java-485/>.
- [18]. Firebase. *wikipedia*. [En ligne] <https://fr.wikipedia.org/wiki/Firebase>.
- [19]. sql nosql. *grafikart*. [En ligne] <https://grafikart.fr/blog/sql-nosql>.