

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Bejaia
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle
en vue d'obtention du diplôme de master professionnel en informatique
option : Génie logiciel

Thème

*La synchronisation d'une application desktop suivi d'une crèche
avec une application web*

Mémoire soutenu par :

M^{elle} Berkane Meriem
M^{elle} Ferhat Lynda

Devant le jury compose de :

Président : M^{me} H.EL BOUHISSI
Examineur : M^{me} S.GHANEM
Encadrant : M^{me} S.AIT KACI AZZOU

U.A/M Bejaia
U.A/M Bejaia
U.A/M Bejaia

Remerciements

*Tout d'abord, nous tenons à exprimer nos remerciements au « **Bon Dieu** ».*

*« **Le Généreux qui a enseigné à l'homme ce qu'il ne savait pas** »*

De nous avoir donné la volonté, la patience, la force, la foi et le courage pour réaliser ce travail. Merci Allah de nous avoir appris, protégées, guidées tout au long de notre vie.

*Nous tenons à remercier chaleureusement notre encadrant Mme **AIT KACI AZZOU Samira**, qui nous a orientées dans le bon sens quant à l'élaboration de ce projet...*

Nous remercions fermement les membres du jury pour nous avoir fait le grand honneur d'examiner et de juger ce travail.

Nous remercions d'autant que nous ne remercierons personne, nos très chers parents qui sont notre exemple de la réussite, que nous aimons tant et respectons, pour les précieux conseils, l'amour, la tendresse, la force qu'ils nous ont donnée.

Merci...

Liste des acronymes

2TUP: 2 Tracks Unified Process.

API: Application Programming Interface.

CMS: Content Management System.

CSS: Cascading Style Sheets.

EDI: Integrated Development Environment.

GIF: Graphics Interchange Format.

HTML: Hyper Text Markup Language.

HTTP: Hyper Text Transfer Protocol.

IHM: Interface Homme-Machine.

MySQL: My Structured Query Language.

PHP: Hypertext Preprocessor.

SDK: Software Development Kit.

SGBD: Système de Gestion de Base de Données.

SQL: Structured Query Language.

UML: Unified Modeling Language.

UP: Unified Process.

XML: Extensible Markup Language.

Liste de figures

Figure1 : Types de diagrammes d'UML	31
Figure2 : Diagramme de contexte du système.	34
Figure3 : Diagramme de cas d'utilisation « global de l'administrateur».....	34
Figure4 : Diagramme de cas d'utilisation « Gérer les enfants».....	35
Figure5 : Diagramme de cas d'utilisation « Gérer les éducatrices »	37
Figure6 : Diagramme de cas d'utilisation « Gérer les sections ».....	39
Figure7 : Diagramme de cas d'utilisation « suivi le payement».....	40
Figure8 : Diagramme de cas d'utilisation « Gestion des comptes».....	41
Figure9 : Diagramme de cas d'utilisation « Gestion des plannings».....	43
Figure10 : Diagramme de cas d'utilisation globale « Parent ».....	44
Figure11 : Diagramme de cas d'utilisation « Inscription »	44
Figure12 : Diagramme de cas d'utilisation «voir tout actualité»	45
Figure13 : Diagramme de cas d'utilisation « Globale éducatrice ».....	46
Figure14 : Diagramme de cas d'utilisation « Gestion des notes ».....	46
Figure15 : Diagramme de cas d'utilisation « Suivi état de présence»	48
Figure16 : Diagramme de séquence « Authentification ».....	49
Figure17 : Diagramme de séquence «Parent».....	50
Figure18 : Diagramme de séquence « Ajouter un dossier enfant »	51
Figure19 : Diagramme de séquence « Modifier un dossier enfant »	52
Figure20 : Diagramme de séquence « Supprimer un dossier enfant ».....	53
Figure21 : Diagramme de séquence « Gestion des comptes »	54
Figure22 : Diagramme d'interaction authentification d'un utilisateur	56
Figure23 : Diagramme d'interaction parent	57
Figure24 : Diagramme d'interaction ajouter un dossier enfant	58
Figure25 : Diagramme d'interaction modifier un dossier enfant	58

Figure26 : Diagramme d'interaction supprimer un dossier enfant	59
Figure27 : Diagramme de classe	63
Figure28 : Le design pattern MVC	67
Figure29 : Interface d'accueil.	69
Figure30 : Interface authentification.	70
Figure31 : Interface gestion des enfants.	71
Figure32 : Interface gestion des sections.	72
Figure33 : Interface gestion des plannings	72
Figure34 : Interface d'accueil de l'application Web... ..	73
Figure35 : Interface d'inscription en ligne... ..	74
Figure36 : Interface d'authentification en ligne.....	75

Liste des tableaux

Tableau 1: Identification des messages	33
Tableau 2: Tableau descriptif du cas d'utilisation gérer les enfants	35
Tableau 3: Tableau descriptif du cas d'utilisation gérer les éducatrices	37
Tableau 4: Tableau descriptif du cas d'utilisation gérer les sections.	39
Tableau 5: Tableau descriptif du cas d'utilisation suivi le paiement	40
Tableau 6: Tableau descriptif du cas d'utilisation gestion des comptes	41
Tableau 7: Tableau descriptif du cas d'utilisation gestion des plannings	43
Tableau 8: Tableau descriptif du cas d'utilisation inscription	44
Tableau 9 : Tableau descriptif du cas d'utilisation voir tout actualité	45
Tableau 10: Tableau descriptif du cas d'utilisation gestion des notes	46
Tableau 11: Tableau descriptif du cas d'utilisation suivi état de présence	49
Tableau 12: Description des classes et leurs attributs	60

Table des matières

Chapitre I

Partie 1: Généralités

1. Introduction.....	13
I.1.Application informatique	13
I.1.2.Types d'applications informatiques	13
I.1.2 .1.Application Desktop	13
I.1.2. 2.Application web	13
I.1.2.3. Application mobile	15

Partie 2:La Crèche

I.2.1. Présentation de l'organisme d'accueil	17
I.2.2 Les conditions d'accueil de l'enfant	17
I.2.3 Le rôle et la qualification du personnel	18
I.2.4. Activités proposées par les crèches (crèche Médine).....	19
I.2.5.Problématique	21
I.2.6.Les objectifs de la crèche	22
I.2.7. Présentation de notre projet	22
I.2.8. Spécification des besoins	23
I.2.8.1.Besoins fonctionnels	23
I.2.8.2.Besoins non fonctionnels	23
conclusion	23

Chapitre II

Partie 1: Méthodologie

1.Introduction	25
II.1. Organisation et approche méthodologique	25
II.1.1. Cahier des charges	25
II.1.1.1. Problématique	25

II.1.1. 2.Objectifs	25
II.1.2. Les méthodes d'analyse et de conception	26
II.1.3. Méthodes agiles	26
II.1.4. Processus de développement	27
II.1.4.1.Processus UP (Unified Process).....	27
II.1.4.1.1.Principes fondamentaux du Processus Unifié(UP)	27
II.1.4.2.Processus 2TUP	28
II.1.4.2.1 Présentation du processus 2TUP	28
II.1.4.2.2 Fonctionnement du processus 2TUP	28
II.1.4.2.3. Caractéristiques du processus 2TUP	29
II.1.5. Un processus de modélisation avec UML	30
II.1.5. 1..Bases d'UML	30
II.1.5. 1.2..Différents types de diagrammes d'UML	30
II.1.6. Les réalisateurs du projet	32

Partie 2: Analyse des besoins

II.2.1.Acteurs et modélisation du contexte	32
II.2.1.1.Identification des acteurs	32
II.2.1.2. <i>Identification des messages</i>	33
I.2.3.Diagramme de contexte	34

Partie 3: Analyse et étude conceptuelle

II.3.1.Formalisation des besoins fonctionnels	34
II.3.1.1. Identification des cas d'utilisation	34
II.3.1.1.1.Cas d'utilisation Schématisés	34
II.3.2. Analyse des besoins	49
II.3.2.1. Diagrammes de séquence	49
II.3.2.2. Les diagrammes de séquence de l'application à réaliser	49

Conclusion	55
------------------	----

Partie04 : conception

1 .Introduction.....	56
II.4.1. Diagramme d'interaction.....	56
II.4.1.1.Diagramme d'interaction authentification	56
II.4.1.2.Diagramme d'interaction de parent	57
II.4.1.3.Diagramme d'interaction gérer un dossier enfant	58
II.4.2.Diagramme de classe	60
II.4.2.1. Présentation des classes et leurs attributs	60
II.4.3. Diagramme de classe de l'application à réaliser	62
II.4.4. Passage au modèle relationnel	63
II.4.4.1. Règles de passage du diagramme de classe au modèle relationnel	63
Conclusion	64

Chapitre III

Réalisation

1.Introduction	66
III.1. L'environnement de développement	66
III.2. Langages utilisés	67
III.2.1. HTML (Hyper Text Markup Language)	67
III.2.2. CSS	67
III.2.3. Bootstrap	67
III.3. Outils et logiciels utilisés	68
III.3.1. MySQL (My Structured Query Language)	68
III.3.2. L'IDE NetBeans	68
III.3.3. WampServer	68
III.4. Présentation du système.....	69
III.4.1. Interfaces de l'application Desktop	69
III.4.2. Interfaces de l'application Web	73
conclusion	75

Conclusion générale

Bibliographie

Webographie

Introduction générale

Dans un monde de plus en plus moderne, l'informatique devient presque indispensable dans tous les domaines de vie à savoir, la science, l'industrie, l'administration, le commerce. . . etc.

Aujourd'hui, plusieurs sociétés essaient de profiter au maximum possible de cette informatique afin d'améliorer leurs productivités et de faire face à quelques problèmes pénibles qui peuvent constituer un obstacle de progression. En effet, la présence des applications Desktop et web est devenue indispensable en société, car c'est une bonne façon d'innover et de gérer ses activités.

Notre principal objectif dans ce projet de fin Master est la conception et la réalisation d'une application web synchronisée avec une application desktop, permettant de gérer une crèche d'une manière très facile. Ceci pour répondre à un ensemble des besoins .

Pour mener à bien le développement de ce projet, nous avons utilisé UML comme langage de modélisation et «2TUP » comme méthode de développement agile, pour l'implémentation, le choix du langage de programmation s'est porté sur le langage orienté objet JAVA, Java Entreprise Edition "Java EE" ; et la base de données sera implémentée avec MySQL.

Notre mémoire est constitué de trois chapitres:

- ✓ Le premier est consacré aux généralités et définitions.
- ✓ Le deuxième présente la méthodologie, l'étude préliminaire, et l'analyse et conception.
- ✓ Et le dernier, présente les outils employés pour la réalisation du système, et les interfaces obtenues.

Nous terminons par une conclusion générale et des perspectives.

Chapitre I

Partie 1:Généralités

Introduction

Un ensemble de facteurs est nécessaire pour la mise en œuvre d'une application web. Présenter l'organisme d'accueil, établir une problématique s'avèrent être l'étude préalable d'un projet. Ceci conduit à spécifier les besoins dans un cahier des charges détaillé.

I.1.Application informatique

Une application est un ensemble de programmes informatiques conçus de façon à fournir un service à l'utilisateur, dans un but précis. Elle est utilisée dans les entreprises pour faciliter le travail et augmenter la productivité des employés [W1].

I.1.2.Types d'applications informatiques

De nos jours, on voit que la technologie ne cesse de se développer, et ce n'est que pour la satisfaction de la population. Un simple exemple sur ça est l'évolution des applications de différents types: personnel, bureau, local, universel.

I.1.2.1. Application Desktop

Une application de bureau désigne tout logiciel qui peut être installé sur un seul ordinateur (ordinateur portable ou un ordinateur de bureau) et utilisé pour effectuer des tâches spécifiques. Certaines applications de bureau peuvent également être utilisées par plusieurs utilisateurs dans un environnement en réseau. Cette technologie est apparue avec les premiers environnements de bureau en 1970[W2].

I.1.2.2. Application web

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoins de l'installer sur son ordinateur. Il suffit de se connecter à l'application à l'aide d'un navigateur web[W3]. Les types d'une application web sont:

➤ Application web statique

La première chose à savoir sur ce type d'application est qu'elle contient peu d'informations et, en général, son contenu n'évolue pas ou très peu. Le développement d'applications web se fait habituellement en HTML (Hyper TextMarkup Langage) et CSS (Cascading Style Sheets). Il peut, néanmoins y avoir des objets animés tels que bannières, GIF (**G**raphics **I**nterchange **F**ormat), vidéos, etc.

➤ **Application web dynamique**

Les applications web dynamiques sont plus complexes sur le plan technique. Elles utilisent des bases de données pour charger des informations, et le contenu est mis à jour à chaque fois que l'utilisateur se connecte à l'application. En général, elles ont un panneau d'administration (appelé CMS) depuis lequel l'administrateur peut corriger ou modifier le contenu. L'actualisation d'une web application dynamique est très simple, et il n'est même pas nécessaire d'entrer dans le serveur pour faire des modifications. En outre, il est possible de mettre en oeuvre de nombreuses fonctionnalités telles que des forums ou des bases de données. Le design, et non seulement le contenu de l'application, peut être modifié en fonction du goût de l'administrateur.

➤ **Application web de type e-shop ou e-commerce**

Si l'application web est un e-shop (commerce numérique), son développement sera plus complexe, car elle doit permettre les paiements électroniques par carte de crédit, PayPal ou autre mode de paiement. Le développeur doit également créer un panel de gestion pour l'administrateur afin que ce dernier puisse mettre en vente des produits, faire des mises à jour et gérer les commandes.

➤ **Application web portail**

Il s'agit d'une application dont la page d'accueil permet d'accéder aux différentes sections ou catégories. Son contenu peut être très varié : forums, chats, e-mail, moteurs de recherche, formulaire d'enregistrement, contenu le plus récent, etc.

➤ **Application web animée**

La technologie FLASH est indispensable pour le développement d'applications web animées. Elle sert à créer le contenu avec des effets d'animation. Permettant un design plus créatif et moderne, FLASH est l'une des technologies les plus utilisées par les designers. L'inconvénient des applications web animées est le risque d'un référencement faible, car la technologie utilisée empêche les moteurs de recherche de lire correctement les informations.

➤ **Application web de type « content manager »**

Pour les applications web dont le contenu doit être souvent mis à jour, vous pouvez installer un système de gestion de contenu (Content Management System, CMS) à travers lequel l'administrateur aura la possibilité d'apporter des modifications. Ces systèmes de gestion sont intuitifs et très faciles à gérer. Les CMS les plus connus sont: WordPress est le CMS le plus répandu. Le site d'hébergement propose beaucoup d'informations sur le réseau, des didacticiels et guides qui expliquent comment l'utiliser et le personnaliser ; en plus, il est gratuit. Joomla arrive à la deuxième position des CMS les plus utilisés. Bien qu'il n'ait pas autant d'utilisateurs que WordPress, sa communauté est importante, et il est également très intuitif. Drupal est un logiciel CMS libre et open-source. Il est très facile d'usage et parfaitement adapté à la gestion de communautés. Ce type de web application est utilisé avant tout pour développer les blogs personnels et professionnels, les reportages, ... etc [W4].

I.1.2.3. Application mobile

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un « smart-phone », un baladeur numérique, une tablette tactile. Nous aborderons dans ce qui suit les différents types d'applications mobiles les plus populaires dans le monde [1].

- **Applications natives** Ce type d'applications correspond à des logiciels créés uniquement pour une plateforme mobile. Le développement de ces logiciels se fait au travers du SDK ou software développement kit de la plateforme mobile choisie. Le nom de ces applications vient du fait qu'elles sont développées avec l'utilisation de langages « natifs » comme par exemple le langage JAVA ou le langage Objective- C. Les natives applications sont téléchargées à partir d'une plateforme de téléchargement qui est souvent une application Store. C'est par exemple le cas pour l'Apple store ou encore Google Play.

- **Applications de type hybride**

Ces applications sont considérées comme un mix à la fois entre les applications natives et les applications web. En effet, elles sont compatibles avec toutes les plateformes mobiles. Ces applications sont principalement développées à l'aide d'HTML5 aujourd'hui qui est très performant mais utilisent aussi d'autres langages web comme CSS et JavaScript. Ainsi, une application hybride, contrairement à une application native, n'est pas dépendante d'une plateforme mobile en particulier. De la même manière, contrairement aux applications web, les applications hybrides peuvent accéder à toutes les fonctions présentes sur le mobile.

Cela est rendu possible par des liens faits entre le langage natif et la technologie web présents dans l'application hybride, en combinant les points forts des applications natives et web, les applications hybrides suscitent de plus en plus l'intérêt des développeurs d'applications mobiles.

Conclusion

Nous avons décidé de concevoir une application web qui va faciliter les interactions entre la crèche et les parents et une application desktop qui va gérer la gestion de la crèche.

Partie 2:La Crèche

I.2.1. Présentation de l'organisme d'accueil

Les crèches sont des établissements d'accueil des enfants durant la journée, qui ont pour vocation le bien être psychique de l'enfant, sa santé, son éducation, sa sécurité, disposant en principe d'un personnel qualifié.

Ce sont des structures pour les enfants bien portants qui pendant une grande partie de la journée, ne peuvent pas être gardés dans leurs familles, elles assurent le garde des nourrissons, d'enfants d'âge préscolaire et parfois d'enfants d'âge scolaire et fonctionnent plus de cinq heures par jours.

Les crèches doivent être aménagées de manière à répondre aux exigences des enfants, sur les plans physique et psychique. Elles sont désormais un facteur important dans la manière dont beaucoup d'enfants vivent, apprennent et se développent.

Leur mission est de veiller à la santé, la sécurité, au bien être de l'enfant et à son développement. Comme elle a pour fonction d'organiser des activités, à favoriser l'éducation et la socialisation des enfants. Aider les parents dans l'éducation de leurs enfants et leur permettre de concilier leur vie familiale, leur vie professionnelle et sociale.

La fréquentation de la crèche est réputée pour avoir un effet bénéfique sur le développement intellectuel, grâce au nombre de sollicitations et à la régularité des stimulations offertes et sur la socialisation grâce à l'extension du milieu, à l'entourage de temps contemporaine, à la sécurité que l'enfant trouve dans les substituts maternels. Par ailleurs, les crèches jouent aussi un rôle dans la préparation de l'enfant pour sa rentrée à l'école.

I.2.2 Les conditions d'accueil de l'enfant

Une crèche doit respecter des règles de fonctionnement strictes. Si le personnel est suffisamment nombreux, la crèche peut accueillir jusqu'à soixante enfants. Dans la plupart des crèches collectives, les enfants sont répartis par sections, chaque section correspondant à une tranche d'âge. Cela permet une meilleure entente, le rythme des enfants peut être mieux respecté et les personnes qui les encadrent sont

en mesure de proposer des activités adaptées. En règle générale, les crèches collectives sont ouvertes du dimanche à jeudi, de 7h30 à 18h30. Elles ferment le week-end et les jours fériés, ainsi que cinq semaines par an en vue de permettre au personnel de prendre ses congés. Certaines crèches ont des structures privées notamment proposent des horaires d'ouverture plus étendues pour s'adapter aux emplois du temps des parents.

I.2.3 Le rôle et la qualification du personnel

La qualification et les compétences du personnel constituent une garantie d'accueil de qualité. Il se compose de cinq catégories de personnels internes et de deux types de personnels externes qui interviennent au sein de la crèche.

Le personnel de la crèche est composé :

- Du responsable de la structure
 - Du personnel chargé de l'encadrement des enfants
 - Des agents
- **Le responsable de la structure : le directeur (ou la directrice)**
- assure la direction, l'organisation et la gestion de la crèche. Il fait appliquer les dispositions du règlement intérieur.
 - garant de la qualité du travail de son équipe auprès des enfants et coordonne les actions.
- **Le personnel chargé de l'encadrement des enfants :** ce sont des éducateurs de jeunes enfants, leur formations se situent à trois niveaux ; éducation, prévention et coordination, ils ont une influence importante sur l'enfant ;

Les éducatrices de jeune enfant ont pour mission de favoriser l'éveil et le développement global des enfants en collaboration avec les auxiliaires de puériculture et les agents placés auprès des enfants. C'est eux qui organisent et accompagnent l'enfant dans les différentes activités ludiques ou éducatives, leur mission est de favoriser le développement et l'épanouissement des enfants en l'absence de leurs parents. Elles assurent la surveillance et les soins

- **Les agents placés au près des enfants** : ils assurent l'accueil quotidien des enfants
- **Les agents de service intérieur** : ils assurent la préparation des repas des enfants, l'entretien du linge et la propreté des locaux.

- **Le personnel extérieur à la structure**

Deux intervenants extérieurs à la crèche collaborent à son fonctionnement.

- ❖ **Le psychologue**

Le psychologue contribue, avec la collaboration de l'équipe et des parents, à la bonne adaptation des enfants et à leur épanouissement. Il apporte aux équipes et aux parents conseils, soutien et orientation.

- ❖ **Le médecin et son rôle** : qu'il soit généraliste ou spécialiste en pédiatrie, il est indispensable dans une crèche, celui-ci doit assurer une action d'éducation et des promotions de la santé auprès du personnel et auprès des parents, il veille à l'application des mesures préventives d'hygiène générales et des dispositions à apprendre en cas de maladie contagieuse, il organise les conditions de recours aux services d'aide médicale d'urgence et il donne son avis lors de l'admission d'un enfant après un examen médical.

I.2.4. Activités proposées par les crèches (crèche Médine)

Dans la crèche Médine, on trouve trois sections principales :

Petite section : reçoit des enfants de 3 ans.

Moyenne section : réservée aux enfants de 4 ans.

Grande section (préscolaire) : ouverte aux enfants de cinq ans.

Le personnel de la crèche Médine doit respecter le programme pédagogique proposé par le ministère de l'éducation nationale pour les enfants de cet âge , les livres ainsi que les activités prévues dans les manuels.

- ✓ **Jeu libre** :

Souvent considéré comme une simple occupation, le jeu est un facteur

essentiel pour le développement de l'enfant, et un réel besoin par lequel il façonne son identité.

Par le jeu est stimulé le développement de l'enfant : psychomoteur (jeu de ballon, tricycle,...etc.), l'éveil cognitif (puzzles,...etc.), la créativité et l'imagination (jeux d'expression artistique) et la socialisation (poupées, marionnettes,...etc.). Il existe deux types de jeux :

A caractère individuel : petites voitures, jeux de construction, camions, poupées etc....

A caractère collectif : balançoires, ballons, toboggans etc....

L'enfant est appelé à choisir le jeu qu'il veut, il est libre et autonome, mais il faut qu'il soit toujours sous la surveillance de l'éducatrice qui intervient seulement pour garder une ambiance de jeu calme et aider l'enfant si cela est nécessaire. C'est aussi l'occasion pour l'observation des enfants surtout par le psychologue de l'établissement. En effet, cela va lui permettre de distinguer les enfants sociables des enfants qui s'isolent, agressifs de ceux qui sont calmes...etc.

L'objectif est de comprendre le vécu affectif du petit et de l'aider à ajuster ses émotions, à gérer ses angoisses et à s'adapter avec le groupe.

Une bonne observation de l'enfant (le jeu et la manière avec laquelle il joue) va permettre de mesurer le niveau de développement moteur, intellectuel, affectif et social .

✓ **Activités dirigées :**

Le reste des activités doit être dirigé par l'équipe pédagogique et l'enfant sera accompagné par son éducatrice qui l'oriente et qui lui explique les tâches à accomplir.

✓ **Jeux éducatifs :**

C'est l'association de l'éducation et de la distraction : l'enfant s'amuse avec son jeu dans lequel l'adulte a introduit des leçons et des contenus éducatifs.

Exemple :

- jeux de construction pour apprendre les couleurs.
- Camion et voiture pour apprendre à l'enfant de faire attention en traversant la route, qu'il ne doit pas se mettre à l'avant du véhicule avec le conducteur (père, mère) pour sa sécurité.
- Une histoire inventée pour apprendre à l'enfant à dire la vérité

✓ **Activités liées au langage :**

Le jardin d'enfants va stimuler l'apprentissage d'un vocabulaire varié et la mise en place progressive d'une syntaxe correcte. L'enfant commence à former son dictionnaire personnel.

✓ **Activités artistiques :**

L'enfant trouve un plaisir immense dans les activités artistiques à travers lesquelles il va découvrir un nouveau moyen d'expression (chants, danses, peinture, dessins...), les différents instruments de musique, sont des activités qui stimulent la créativité et l'imagination des petits.

✓ **Travaux manuels :**

L'ensemble des activités permettant de coller, de modeler, d'assembler, d'emboîter, de déchiqeter et de découper stimulent l'imagination créative de l'enfant. Les bricolages permettent d'explorer des matériaux très divers qui offrent un moyen de développer la coordination sensori-motrice. C'est également une manière très efficace qui permet à l'enfant de découvrir ses capacités et surtout à former son estime de soi.

✓ **Activités rythmiques :**

Ce sont des activités très amusantes pour l'enfant, le but est de l'aider à :

- Ressentir plus de confiance à exécuter des activités en groupe,
- Développer son imagination,
- Prendre conscience de l'autre,
- Améliorer sa concentration et à favoriser le phénomène de l'imitation, base de tout apprentissage.

I.2.5.Problématique

La crèche désire synchroniser ses services d'une manière performante, facile et efficace afin de satisfaire les parents et garantir le maintien de l'activité pour détecter les problèmes existants. Et après une observation continuelle de la crèche , nous avons pu observer les insuffisances suivantes :

1. Recherche difficile sur les registres qui engendre une perte de temps.
2. Insécurité des informations.
3. Possibilité d'erreur dans le remplissage des différents documents et registres.

4. Possibilité d'erreur dans le calcul des statistiques.
5. Nombre important des archives qui engendre une difficulté de stockage.

Nous constatons que la solution informatique est la plus adéquate puisqu'elle répond mieux aux anomalies souvent rencontrées dans la gestion manuelle. Ainsi nous avons décidé de concevoir une application web avec une application desktop qui va gérer les activités de cette crèche et qui va permettre par la suite de minimiser le support papier et d'améliorer la rapidité de l'accès à l'information et surtout aider les parents à avoir toutes les informations concernant leur enfant et faire une pré-inscription sans pouvoir se déplacer.

I.2.6. Les objectifs de la crèche

- Installer un espace « parents » au cœur de la crèche.
- Mettre en place différents moyens de communication avec les familles.
- Mettre en place un site informatique.
- Savoir traiter les demandes d'accès aux dossiers des enfants.
- Améliorer la gestion des informations en les centralisant
- Améliorer la disponibilité du dossier de l'enfant et des informations qu'il contient, y compris en cas d'urgence.

I.2.7. Présentation de notre projet

Pour remédier aux difficultés présentées dans la section précédente, nous avons proposé de concevoir et mettre en place une application web synchronisée avec une application desktop pour la gestion de la crèche. Cette solution doit fournir un accès à distance aux notes, à l'emploi du temps, aux absences, aux retards et d'autres. Le nouveau produit concernera aussi bien les parents pour la consultation et l'interaction, que la crèche pour la mise à disposition des informations et l'utilisation du système.

Ce dernier doit être considéré comme une extension du système actuel et comme premier pas vers la modernisation de l'existant.

Cette extension proposera aux parents, l'accès à un espace personnel dans lequel, ils auront accès en consultation à toutes les informations concernant leurs enfants inscrits dans l'établissement. Pour cela la crèche devra permettre à l'ensemble du personnel

(directeur, éducatrices) de renseigner les données nécessaires concernant les enfants et les cours suivis par ces derniers. Cette application a pour but de faciliter la communication parent-crèche et de fournir tous les outils nécessaires pour s'impliquer dans la scolarité de leurs enfants.

I.2.8. Spécification des besoins :

Cette phase consiste à comprendre le contexte du système informatique, il s'agit de déterminer quelles sont les fonctionnalités les plus pertinentes répondant aux besoins afin d'établir un cahier des charges.

I.2.8.1. Besoins fonctionnels

Les besoins fondamentaux que l'application doit assurer sont:

- ✓ S'inscrire au centre en tant que client.
- ✓ Créer des espaces personnalisés pour les parents auxquels ils accèdent avec le mot de passe de leur choix.
- ✓ Créer les différents profils d'administrateurs et d'utilisateurs avec des statuts spécifiques.
- ✓ Le système va permettre à l'administrateur de pouvoir modifier supprimer ajouter des informations concernant la gestion des éducatrices ainsi la gestion des enfants, gestion de groupes, gestion des sections et des emplois de temps.
- ✓ Le système doit avoir un historique des paiements des éducatrices et des versement des parents.

I.2.8.2. Besoins non fonctionnels

Les besoins non fonctionnels représentent les exigences implicites auxquelles le système doit répondre. Parmi ces besoins, on cite:

- ✓ Simplicité d'utilisation.
- ✓ L'application doit être performante, c'est-à-dire, à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.
- ✓ Compatibilité avec n'importe quel navigateur.
- ✓ l'IHM doit être simple et conviviale.

Conclusion

Ce chapitre nous a permis de donner un aperçu général sur l'organisme d'accueil et de mettre en évidence la problématique, élaborer un cahier des charges à suivre pour la résolution du problème.

Chapitre II

Partie 1: Méthodologie

Introduction

Dans ce chapitre, nous présenterons notre travail en commençant par la définition des étapes du processus de développement logiciel 2TUP, et la présentation du langage de modélisation UML (Unified Modeling Language), ensuite nous entamerons la modélisation de notre système en mettant en avant les diagrammes des cas d'utilisation, les diagrammes de séquence.

Nous avons choisi de modéliser notre système avec le langage de modélisation Unifié UML pour les avantages qu'il présente.

II.1. Organisation et approche méthodologique

Afin de maîtriser le cycle de développement du système à réaliser, nous nous sommes appuyés sur un ensemble de méthodes, de techniques et d'outils appropriés. Cette partie décrit la méthode Agile [2] que nous avons adoptée pour la réalisation de notre projet. Le processus 2TUP pour expliquer la séquence des étapes qui nous ont permis de développer notre produit, le langage de conception de l'analyse UML.

II.1.1. Cahier des charges

II.1.1.1. Problématique

La crèche désire synchroniser ses services d'une manière performante, facile et efficace afin de satisfaire le client et garantir le maintien de l'activité pour détecter les problèmes existants. Et après une observation continue de la crèche, nous avons pu observer les insuffisances suivantes :

1. Recherche difficile sur les registres qui engendre une perte de temps.
2. Insécurité des informations.
3. Possibilité d'erreur dans le remplissage des différents documents et registres.
4. Possibilité d'erreur dans le calcul des statistiques.
5. Nombre important des archives qui engendre une difficulté de stockage.

II.1.1. 2.Objectifs

Notre objectif est de faciliter la possibilité aux familles d'effectuer leur inscription en ligne et toutes les informations concernant la crèche et leurs enfants sans se déplacer.

Pour ce, nous avons pensé à créer une application web synchroniser avec une application mobile ayant pour but de :

1. Faciliter la recherche et l'accès aux informations.
2. Stocker les informations sur des supports informatiques ce qui assurera leur sécurité.
3. Gagner du temps dans le calcul des statistiques.
4. Automatiser les tâches qui se traitent manuellement.
5. Proposer une bonne codification.

II.1.2. Les méthodes d'analyse et de conception

Une méthode d'analyse et de conception a pour objectif de permettre la formalisation des étapes préliminaires du développement d'un système, afin de rendre ce développement plus fidèle aux besoins du client.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité, etc.

La phase de conception permet de décrire de manière non ambiguë, en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

Il existe plusieurs méthodes d'analyses, tel que 2TUP (TwoTrackUnifiedProcess) et UP (UnifiedProcess) [2].

II.1.3. Méthodes agiles

Les méthodes agiles sont des groupes de pratiques de pilotage et de réalisation de projets. Elles ont pour origine le manifeste Agile¹, rédigé en 2001, qui consacre le terme d'« agile » pour référencer de multiples méthodes existantes [W1].

Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles, impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles reposent sur un cycle de développement

itératif, incrémental et adaptatif et doivent respecter quatre valeurs fondamentales, qui sont:

- Les Individus et leurs interactions plus que les processus et les outils
- Des logiciels opérationnels plus qu'une documentation exhaustive
- La collaboration avec les clients plus que la négociation contractuelle
- L'adaptation au changement plus que le suivi d'un plan.

II.1.4. Processus de développement

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système ou à l'évaluation d'un système existant.

L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

Généralement, il existe plusieurs processus de développement d'applications, après avoir parcouru ces processus et leurs caractéristiques, et en ayant connaissance de notre projet et ses besoins, nous avons opté pour le processus UP (UnifiedProcess) qui semble adéquat pour ce genre d'application [2] .

II.1.4.1.Processus UP (Unified Process)

Le processus Unifié est un processus de réalisation ou d'évolution de logiciel entièrement basé sur UML, d'où l'intérêt de le mettre en œuvre dans notre projet. Le processus Unifié est constitué d'un ensemble de directives afin de produire un logiciel à partir des exigences. Chaque directive définit qui fait quoi et à quel moment. Il permet donc de structurer les différentes étapes d'un projet informatique [2] .

II.1.4.1.1.Principes fondamentaux du Processus Unifié(UP)

Le processus Unifié (UP) est un processus de développement logiciel "itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques" [3].

- **Itératif et incrémental** : Le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. A la fin de

chaque itération, une partie exécutable du système final est produite, de façon incrémentale [3].

- **Centré sur l'architecture** : Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte [3].
- **Piloté par les risques** : Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations [3].
- **Conduit par les cas d'utilisation** : Le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés [3].

II.1.4.2.Processus 2TUP

II.1.4.2.1 Présentation du processus 2TUP

Ce processus de développement logiciel qui met en œuvre la méthode du processus unifié, est créé par valtech, le processus 2TUP permet de répondre aux contraintes de changement continu imposées aux systèmes d'informations des entreprises. Un processus définit une séquence d'étapes ordonnées qui permet de produire un système logiciel ou de faire évoluer un système existant. Un processus est décomposé suivant l'axe de développement technique et l'axe de gestion du développement. L'axe de développement technique se préoccupe de la production tandis que la gestion du développement mesure, puis prévoit les coûts et les délais.

II.1.4.2.2 Fonctionnement du processus 2TUP

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte. Le processus s'articule ensuite autour de trois phases essentielles : une branche technique, une branche fonctionnelle, une phase de réalisation .[4]

- **La branche fonctionnelle**

Capture des besoins fonctionnels : elle aboutit à un modèle des besoins focalisé sur le métier des utilisateurs. Elle minimise le risque de produire un système inadéquat avec les besoins des utilisateurs de cette capture.

Analyse : étude des spécifications afin de savoir ce que le système va réellement réaliser en termes de métier. Découpage en composants.

- **La branche technique**

Capture des besoins techniques : recensement des outils, des matériels et des technologies à utiliser, ce qui permet d'aboutir à une première conception de l'architecture technique.

Conception générique : Découpage en composants nécessaires à la construction de l'architecture technique. Il est généralement conseillé de réaliser un prototype pour assurer la validité de l'architecture. Cette étape permet de minimiser l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles.

- **La phase de réalisation**

Conception préliminaire : étape délicate durant laquelle on intègre le modèle d'analyse dans l'architecture technique.

Conception détaillée : conception de chaque fonctionnalité.

Etape de codage : phase de programmation des fonctionnalités, avec des tests au fur et à mesure.

Etape de recette : phase de validation des fonctions du système développé.

II.1.4.2.3. Caractéristiques du processus 2TUP

- **Incrémental** : définir des incréments de réalisation est en effet la meilleure pratique de gestion des risques d'ordre à la fois technique et fonctionnel. Chaque incrément confirme la preuve de faisabilité auprès de l'équipe de développement et du client. De plus, le suivi des incréments constitue un excellent contrôle des coûts et délais.

- **Itératif** : non seulement à chaque cycle on ajoute une fonctionnalité mais de plus on améliore les fonctionnalités précédentes.

- **Piloté par les risques** : Lors du développement logiciel, plusieurs risques peuvent survenir. On peut citer par exemple : inadéquation aux besoins des utilisateurs, le non respect des coûts et délais.

- **Orienté composant** : Un composant est un module indépendant, qui pourrait servir pour d'autres projets. Le découpage en modules de ce type de processus se fait aussi bien en modélisation qu'en production, et permet la réutilisation logicielle.

- **Orienté utilisateur** : Les utilisateurs sont à l'origine du développement.

II.1.5. Un processus de modélisation avec UML

Le processus UP est basé sur UML comme langage de modélisation.

➤ **UML (Unified Modeling Language) :**

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue[2].

II.1.5. 1.Bases d'UML

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. Ce langage est certes issu du développement logiciel mais pourrait être appliqué à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la spécification jusqu'au codage. Dans ce cadre, un concept appartenant aux besoins des utilisateurs projette sa réalité dans le modèle de conception et dans le codage.

II.1.5. 1.2.Différents types de diagrammes d'UML

UML s'articule maintenant autour de 13 diagrammes différents, dont quatre nouveaux diagrammes introduits par UML 2.0. Chacun d'eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier. Par ailleurs, UML modélise le système suivant deux modes de représentation : l'un concerne la structure du système pris " au repos", l'autre concerne sa dynamique de fonctionnement. Les deux représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composants fonctionnent entre elles.

La figure suivant présente les différents types de diagramme de l'UML[2].

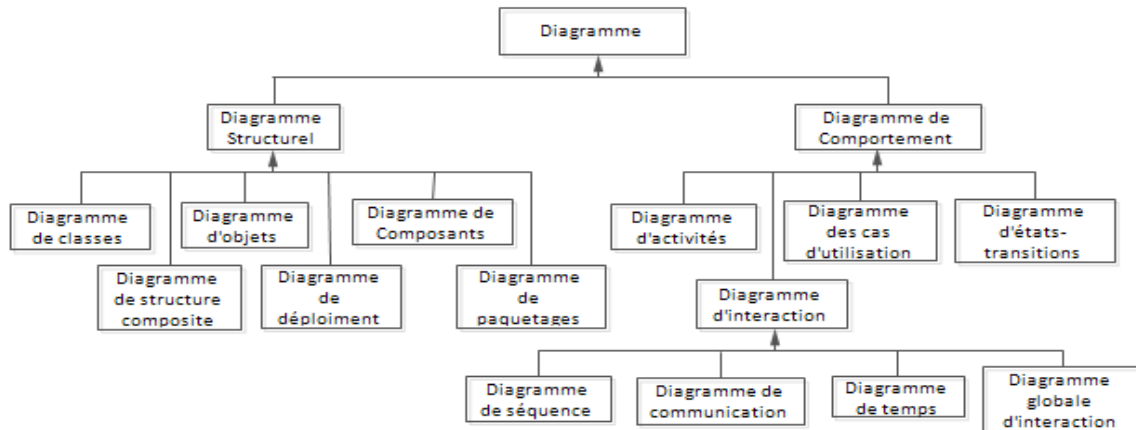


Figure1 :Types de diagrammes d’UML.

Dans notre projet nous allons utiliser seulement les diagrammes suivant :

- **Diagramme de cas d’utilisation** : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est utilisé dans les deux étapes de capture des besoins fonctionnels et techniques.
- **Diagramme de séquence** : est un diagramme d’interaction, il représente les échanges de messages entre objets, dans le cadre d’un fonctionnement particulier du système. Ils servent ensuite à développer en analyse les scénarios d’utilisation du système.
- **Diagramme de classe** : a toujours été le plus important dans toutes les méthodes orientés objet. C’est également celui qui contient la plus grande gamme de notations et de variantes centralise l’organisation des classes de conception, c’est lui qui se transforme le plus aisément en code.
- **Diagramme d’interaction** :

Le diagramme d’interaction est un diagramme de séquence détaillé.

Nous présentons dans cette section les diagrammes d’interactions, en remplaçant le système représenté par une boite noire dans le diagramme de séquences par les trois types de classes suivantes :

1. L’objet interface (dialogue)(<D>) : représente l’interface entre l’acteur et le système,
2. L’objet contrôle (<Ctrl>) : représente un traitement du system déclenché par un acteur,
3. L’objet entité (<E>) : représente des objets décrits dans le cas d’utilisation [5].

Nous respectons également les règles que nous avons fixées sur les relations entre les classes d'analyses, mais en nous intéressant cette fois-ci aux interactions dynamiques entre objets :

- Les acteurs ne peuvent interagir (envoyer des messages) qu'avec les dialogues ;
- Les dialogues peuvent interagir avec les contrôles ;
- Les contrôles peuvent interagir avec les dialogues, les entités, ou d'autres contrôles ;
- Les entités ne peuvent interagir qu'entre elles.

II.1.6. Les réalisateurs du projet

Les réalisateurs impliqués dans cette étude sont repartis dans différents groupes de travail comme suit :

Groupe de pilotage: oriente sur les choix stratégiques et techniques. Il met à disposition les moyens pour la réalisation du projet.

Groupe de projet ou de réalisation: est chargé de l'exécution du projet. Il réalise les différentes tâches d'analyse, de conception, de développement et de déploiement. Il est tenu de rendre compte au groupe de pilotage pour la validation de ses activités.

Groupe des utilisateurs: il a un rôle consultatif. Il est sollicité lors des interviews pour fournir des informations nécessaires à la bonne conduite du projet.

Partie 2: Analyse des besoins

II.2.1. Acteurs et modélisation du contexte

II.2.1.1. Identification des acteurs

La première étape de modélisation consiste à définir le périmètre du système, le contour de l'organisation. La seconde sert à modéliser et identifier les entités qui interagissent avec le système qui sont les acteurs.

Définition : Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

- ❖ **Le parent:** C'est la personne qui s'inscrit et devient adhérent, dans le but de profiter à des services du système.
- ❖ **Administrateur:** C'est un employé de la crèche /directeur , qui gère le système et tout ce qu'il a relation avec.
- ❖ **L'éducatrice :** c'est une personne de la crèche qui favorise l'éveil et le développement global des enfants.

II.2.1.2. Identification des messages

Un message représente la spécification d'une communication unidirectionnelle entre objets et qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur [2] .

Un message est normalement associé à deux occurrences d'évènement : un évènement d'envoi et un évènement de réception.

Acteurs	Messages émis par l'acteur	Messages émis par le système
Parent	M1 : Demande d'authentification et d'ouverture de l'application. M2 : Demande de mises à jour (Ajouts, modifications et suppressions). M3 : Demande d'inscriptions	MR1 : Demande de remplir les champs donnés. MR2 : Demande de confirmation. MR3 : Affichage des résultats.
Educatrices	M1 : S'authentifier M2 : Demande l'accès à l'un des gestions (gestion des notes et la gestion des présences). M3 : Ajouter/ Supprimer/ Modifier/ Consulter au niveau des gestions.	MR1 : Interface d'authentification MR2 : Interface de la gestion demandée MR3 : Valider le formulaire
Administrateur	M1 : S'authentifier M2 : Demande l'accès à l'un	MR1 : Interface d'authentification

	des gestions(enfants, groupes sections et emplois de temps. M3 : Ajouter/ Supprimer/ Modifier/Consulter au niveau des gestions	MR2 : Interface de la gestion demandée MR3 : Valider le formulaire
--	---	---

Tableau 1:Identification des messages.

I.2.3.Diagramme de contexte

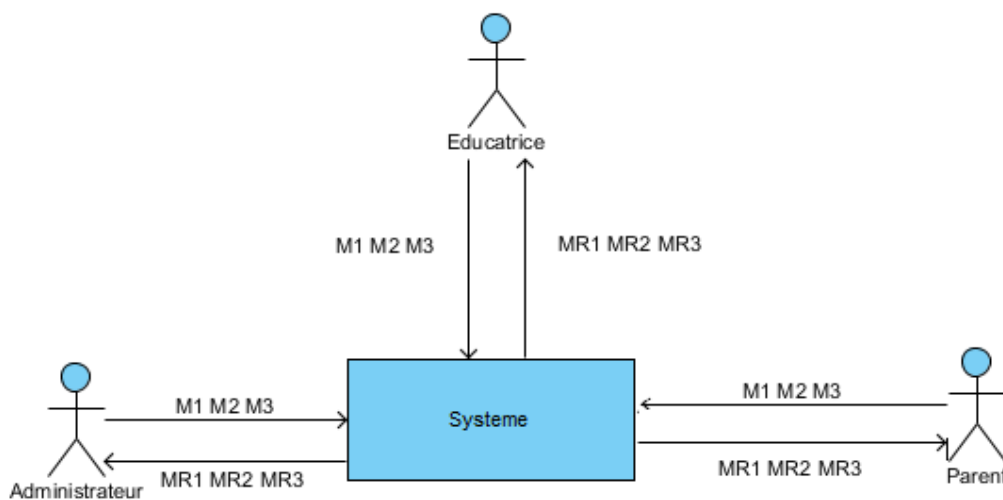


Figure 2:Diagramme de contexte du système.

Partie 3: Analyse et étude conceptuelle

Cette partie nous permettra de faire l'analyse de notre système. Cette analyse consiste dans un premier temps à rechercher et à décrire les cas d'utilisation de notre système à travers les diagrammes de cas d'utilisation, les diagrammes de séquences et les diagrammes d'activités. Cette analyse nous conduira aussi à la production de diagramme de classes et à la conception du modèle de stockage des données.

II.3.1.Formalisation des besoins fonctionnels

II.3.1.1.Cas d'utilisation Schématisés

- ✓ Cas d'utilisation global pour l'administrateur:

Cette figure (**Figure 3**) représente toutes les responsabilités de l'administrateur à la fois:

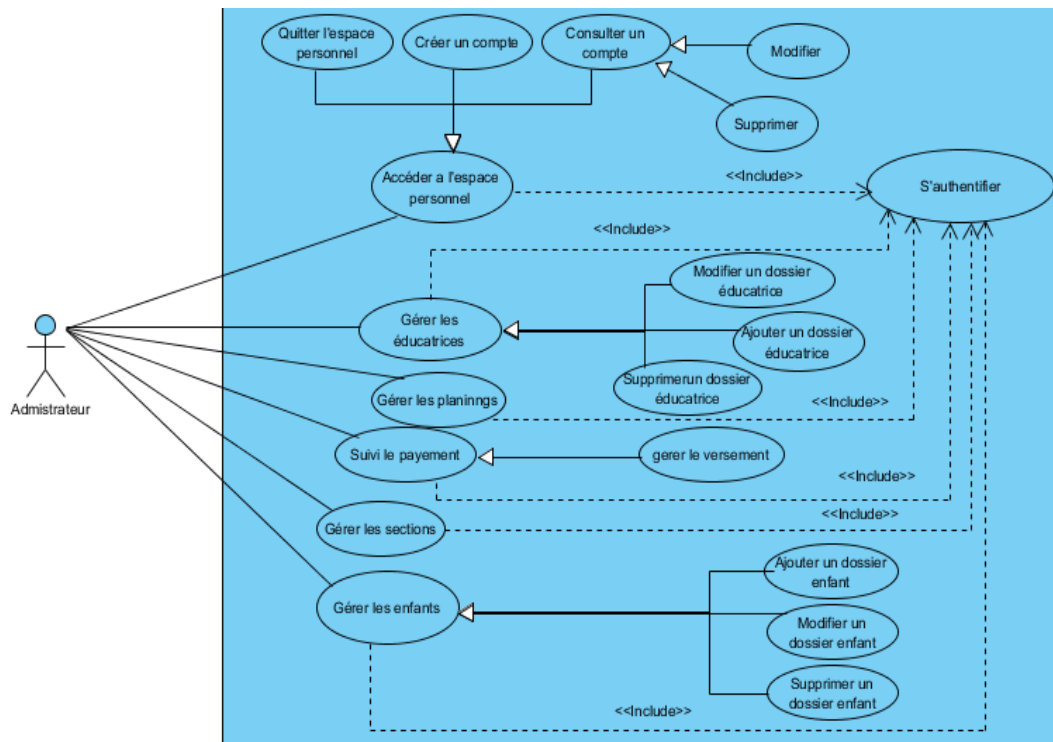


Figure 3:Diagramme de cas d'utilisation « global de l'administrateur».

✓ Cas d'utilisation pour gérer les enfants:

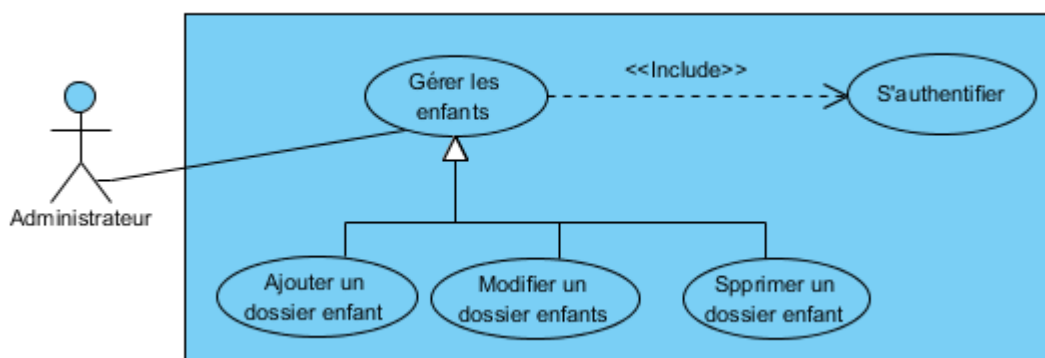


Figure 4:Diagramme de cas d'utilisation « Gérer les enfants».

Description	
Acteur	Administrateur
Objectif	Pouvoir ajouter, modifier, supprimer un

	dossier enfant.
Scénario nominal	<p>cas 1 : Ajouter un dossier enfant</p> <ol style="list-style-type: none">1. l'administrateur choisit d'ajouter un dossier enfant.2. Le système affiche le formulaire à remplir.3. l'administrateur remplit et valide le formulaire.4. Le système Crée les informations dans la base.5. Le système actualise la liste des enfants et l'affiche. <p>cas 2 : Modifier un dossier enfant</p> <ol style="list-style-type: none">1. l'administrateur choisit l'enfant à modifier.2. Le système affiche le formulaire de modification.3. Il modifie les champs voulus.4. Le système met _à jour les informations dans la base.5. Le système actualise la liste des enfants et l'affiche. <p>cas 3 : Supprimer un dossier enfant</p> <ol style="list-style-type: none">1. l'administrateur choisit l'enfant à supprimer.2. Le système demande une confirmation.3. l'administrateur confirme ou annule la suppression.4. Le système supprime le dossier d'un enfant de la base.5. Le système actualise la liste des enfants et l'affiche.

<p>Scénario alternatif</p>	<p>cas 1 :</p> <p>1. Enfant existe déjà a ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.</p> <p>cas 2 :</p> <p>1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.</p> <p>cas 3 :</p> <p>1. Enfant inexistant : un message d'erreur sera affiché.</p>
-----------------------------------	--

Tableau 2: Tableau descriptif du cas d'utilisation gérer les enfants.

✓ Cas d'utilisation pour gérer les éducatrices :

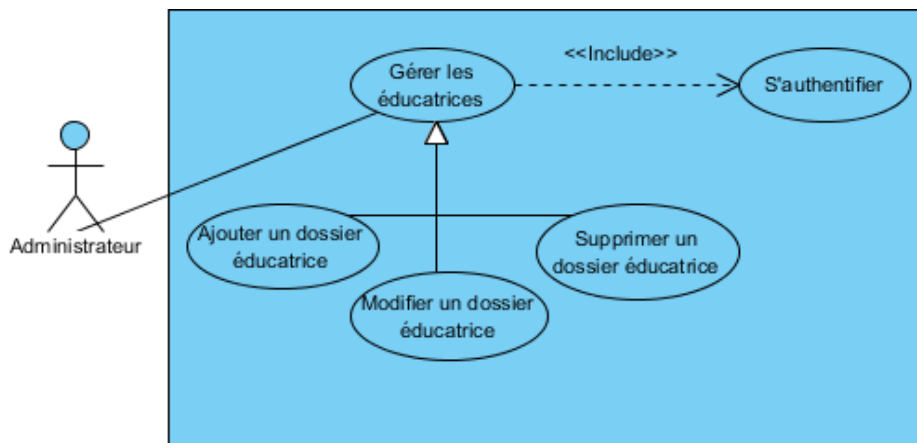


Figure 5:Diagramme de cas d'utilisation « Gérer les éducatrices » .

Description	
Acteur	Administrateur
Objectif	Pouvoir ajouter, modifier, supprimer un dossier éducatrice.
Scénario nominal	<p>cas 1 : Ajouter un dossier éducatrice</p> <p>1. l'administrateur choisit d'ajouter un dossier éducatrice.</p> <p>2. Le système affiche le formulaire à remplir.</p>

	<p>3. l'administrateur remplit et valide le formulaire.</p> <p>4. Le système ajoute les informations dans la base.</p> <p>5. Le système actualise la liste des éducatrices et l'affiche.</p> <p>cas 2 : Modifier un dossier éducatrice</p> <p>1. l'administrateur choisit le dossier à modifier.</p> <p>2. Le système affiche le formulaire de modification.</p> <p>3. Il modifie les champs voulus.</p> <p>4. Le système met _a jour les informations dans la base.</p> <p>5. Le système actualise la liste des éducatrices et l'affiche.</p> <p>cas 3 : Supprimer un dossier éducatrice</p> <p>1. l'administrateur choisit le dossier à supprimer.</p> <p>2. Le système demande une confirmation.</p> <p>3. l'administrateur confirme ou annule la suppression.</p> <p>4. Le système supprime le dossier de la base.</p> <p>5. Le système actualise la liste des éducatrices et l'affiche.</p>
<p>Scénario alternatif</p>	<p>cas 1 :</p> <p>1. Educatrice existe déjà a ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.</p>

	<p>cas 2 :</p> <p>1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.</p> <p>cas 3 :</p> <p>1. Educatrice inexistant : un message d'erreur sera affiché.</p>
--	---

Tableau 3: Tableau descriptif du cas d'utilisation gérer les éducatrices.

✓ Cas d'utilisation pour gérer les sections:

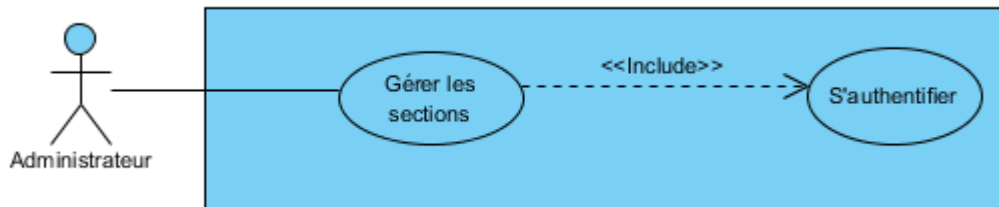


Figure 6:Diagramme de cas d'utilisation « Gérer les sections » .

Description	
Acteur	Administrateur
Objectif	Pouvoir gérer la liste des sections.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur saisi son login et son mot de passe. 2. Le système affiche la page d'accueil. 3. l'administrateur sélectionne la liste des sections. 4. Affichage de la liste de sections.
Scénario alternatif	<ol style="list-style-type: none"> 1. Le système n'arrive pas à identifier l'administrateur et affiche un message d'erreur.

Tableau 4: Tableau descriptif du cas d'utilisation gérer les sections.

✓ Cas d'utilisation pour suivi le paiement des parents:

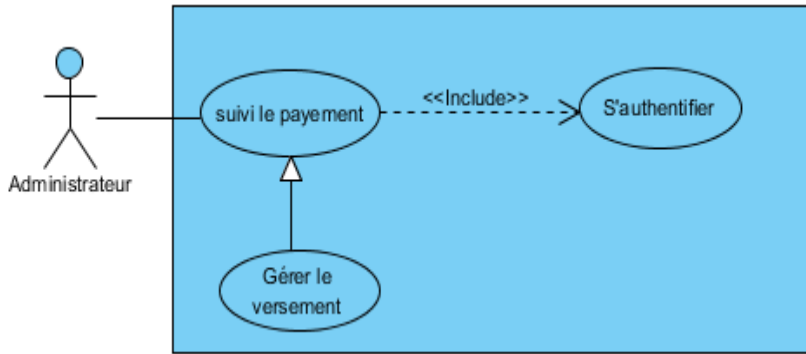


Figure 7: Diagramme de cas d'utilisation « suivi le paiement ».

Description	
Acteur	Administrateur
Objectif	Avoir un historique de tous les paiements effectués par les parents.
Scénario nominal	<ol style="list-style-type: none"> 1. l'administrateur saisi son login et son mot de passe. 2. Le système affiche la page d'accueil. 3. l'administrateur choisit d'ajouter le versement effectué par le parent. 4. Le système affiche la liste des versements effectués.
Scénario alternatif	<ol style="list-style-type: none"> 1. Le système n'arrive pas à identifier l'administrateur et affiche un message d'erreur.

Tableau 5: Tableau descriptif du cas d'utilisation suivi le paiement.

- ✓ Cas d'utilisation pour la gestion des comptes:

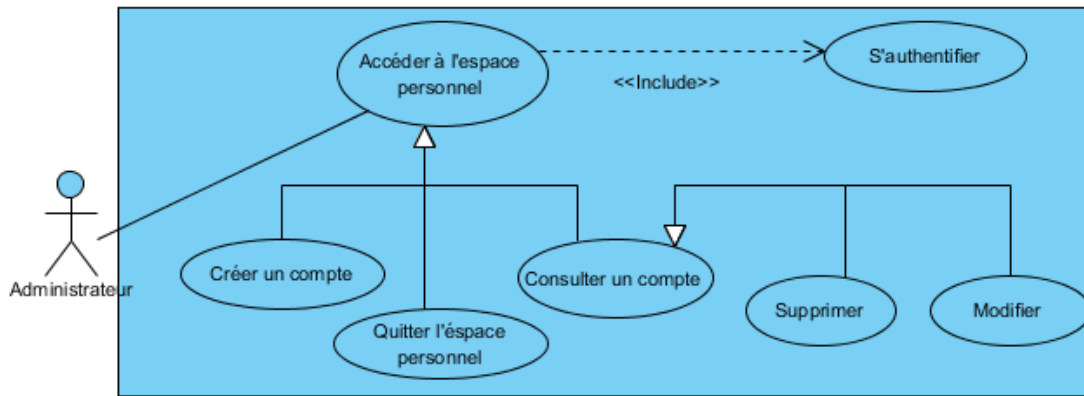


Figure 8:Diagramme de cas d'utilisation « Gestion des comptes».

Description	
Acteur	Administrateur
Objectif	Pouvoir créer, modifier et consulter un compte.
Scénario nominal	<p>Cas 1:Créer un compte</p> <ol style="list-style-type: none"> 1. l'administrateur choisit de créer un compte. 2. Affichage de la création d'un compte Internet avec la liste des familles ne possédant pas de compte. 3. Il choisissent un type dans la liste Proposée. 4. Génération d'un login et d'un mot de passe. 5. Enregistrement du compte dans le système. 6. Affichage du login et du mot de passe. <p>Cas2 : Consulter un compte</p> <ol style="list-style-type: none"> 1. L'administrateur accède à la page d'accueil après l'authentification. 2. Affichage d'un tableau avec les comptes (Les données

	<p>affichées sont : le nom, prénom, type de compte,...).</p> <p>Cas3 : Modifier un compte</p> <ol style="list-style-type: none"> 1. l'administrateur choisi de modifier un compte. 2. Affichage de la liste des comptes (les données affichées sont le login, le nom du propriétaire et le type de compte). 3. Il choisit un compte dans la liste. 4 .Affichage des informations reliées au compte. 5 .Il modifie les champs souhaités et valide. 6 .Vérification des données. 7. Enregistrement des modifications.
<p>Scénario alternatif</p>	<p>cas 1 :</p> <ol style="list-style-type: none"> 1. le compte existe déjà a ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché. <p>Cas2 :</p> <ol style="list-style-type: none"> 1. Le système n'arrive pas à identifier l'administrateur r et affiche un message d'erreur. <p>cas 3 :</p> <ol style="list-style-type: none"> 1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.

Tableau 6 :Tableau descriptif du cas d'utilisation gestion des comptes.

✓ Cas d'utilisation pour la gestion des plannings :

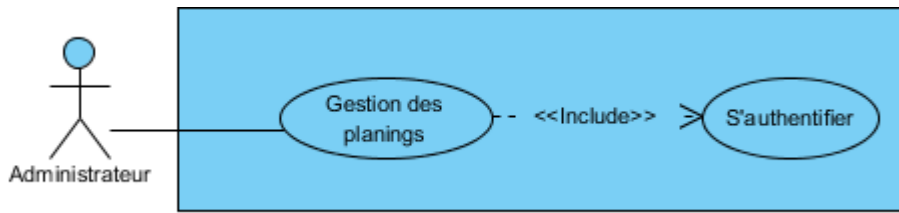


Figure 9:Diagramme de cas d'utilisation « Gestion des plannings».

Description	
Acteur	Administrateur
Objectif	Consulter l'emploi du temps.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur saisi son login et son mot de passe. 2. Le système affiche la page d'accueil. 3. l'administrateur sélectionne la section emploi du temps. 4. Affichage de l'emploi de temps.
Scénario alternatif	<ol style="list-style-type: none"> 1. Le système n'arrive pas à identifier l'administrateur et affiche un message d'erreur.

Tableau 7: Tableau descriptif du cas d'utilisation gestion des plannings.

✓ Cas d'utilisation global pour le Parent:

Cette figure (**Figure10**) montre toutes les tâches avec lesquelles le parent peut interagir avec le système:

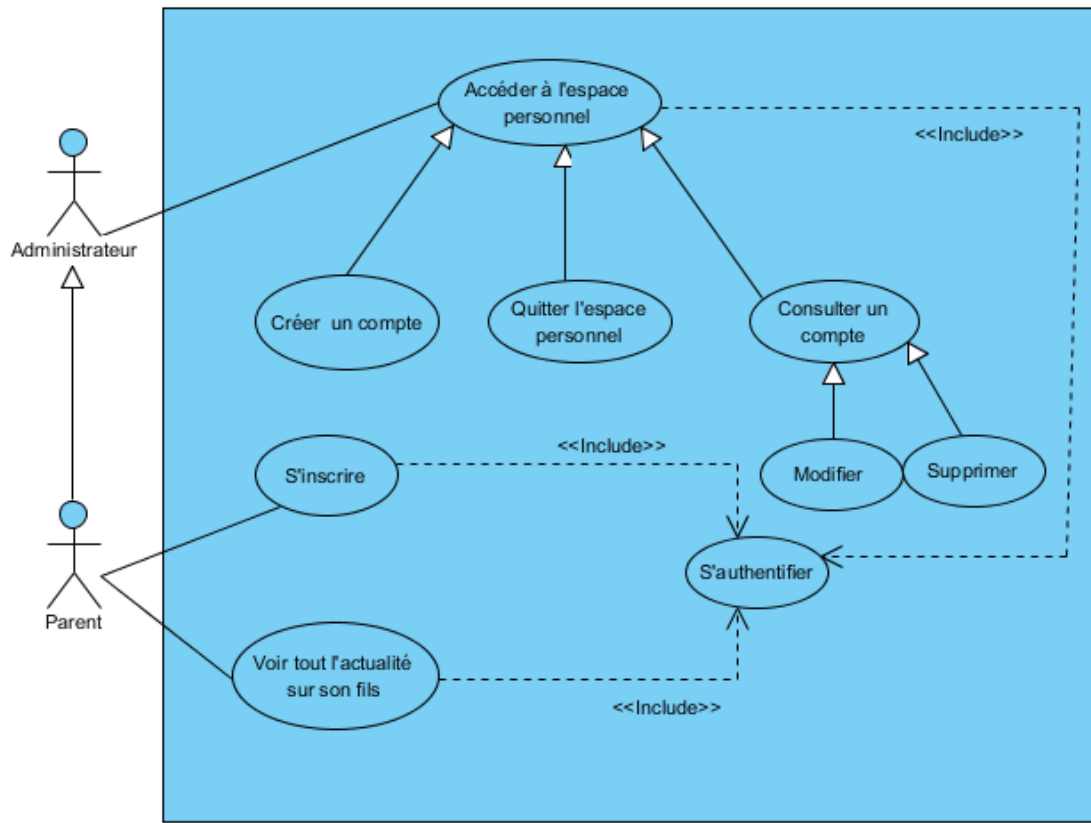


Figure 10:Diagramme de cas d'utilisation globale « Parent ».

✓ Cas d'utilisation pour l'inscription:



Figure11:Diagramme de cas d'utilisation « Inscription ».

Description	
Acteur	Parent
Objectif	Création d'un compte d'adhérent
Scénario nominal	1. Le parent demande la création d'un compte utilisateur

	<p>2- Le système affiche un formulaire de création de compte</p> <p>3. Le parent remplit les champs du formulaire (nom d'utilisateur + mot de passe, et autres).</p> <p>4- Le système affiche un message pour informer le parent que son compte a bien été créé.</p>
Scénario alternatif	<p>1. Le système n'arrive pas à identifier le parent et affiche un message d'erreur.</p>

Tableau 8 :Tableau descriptif du cas d'utilisation inscription.

✓ Cas d'utilisation pour voir tout actualité:

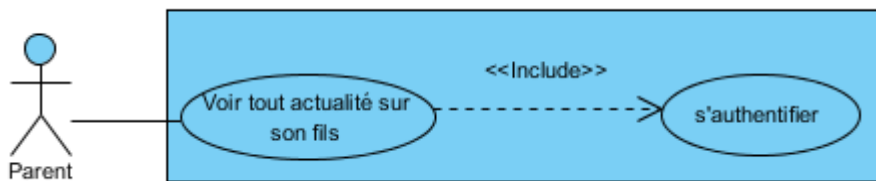


Figure 12 :Diagramme de cas d'utilisation «voir tout actualité».

Description	
Acteur	Parent
Objectif	Voir toute l'actualité de son fils
Scénario nominal	<p>1. le parent saisi son login et son mot de passe.</p> <p>2. Le système affiche la page d'accueil.</p> <p>3. le parent voit ce qu'il fait son fils pendant tout la journée.</p>
Scénario alternatif	<p>1. Le système n'arrive pas à identifier le parent et affiche un message d'erreur.</p>

Tableau 9: Tableau descriptif du cas d'utilisation voir tout actualité.

✓ Cas d'utilisation global pour l'éducatrice:

Cette figure (**Figure13**) représente toutes les responsabilités de l'éducatrice à la fois:

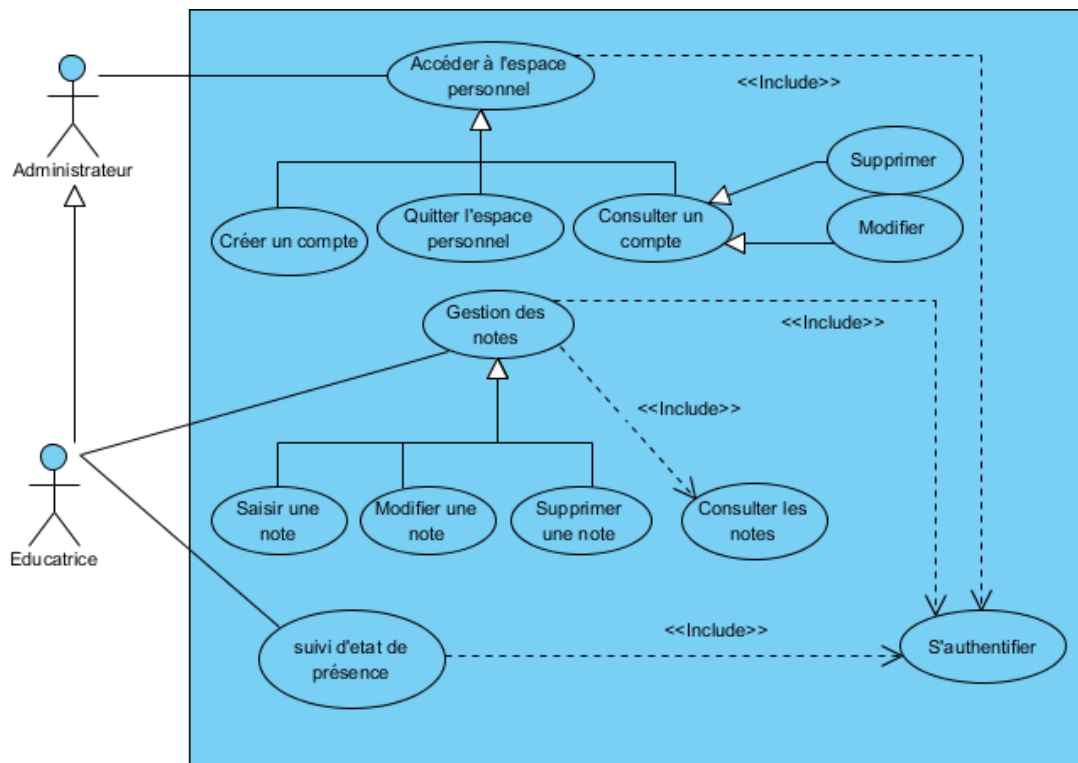


Figure 13: Diagramme de cas d'utilisation « Globale éducatrice ».

✓ Cas d'utilisation Gestion des notes:

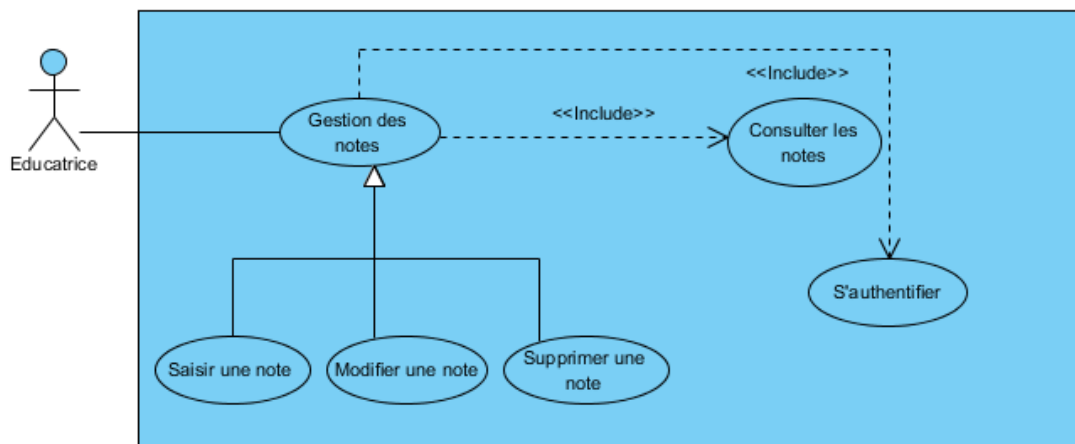


Figure 14: Diagramme de cas d'utilisation « Gestion des notes ».

Description	
Acteur	Educatrice.
Objectif	Pouvoir saisir, modifier, supprimer et consulter une note.
Scénario nominal	<p>cas 1 : Saisir une note</p> <p>1. l'éducatrice choisit de saisir une note à un enfant.</p>

	<ol style="list-style-type: none">2. Le système affiche le formulaire à remplir.3. l'éducatrice remplit et valide le formulaire.4. Le système ajoute les informations dans la base.5. Le système actualise la liste des notes et l'affiche. <p>cas 2 : Modifier une note</p> <ol style="list-style-type: none">1. l'éducatrice choisit la note de l'enfant à modifier.2. Le système affiche le formulaire de modification.3. Il modifie les champs voulus.4. Le système met _à jour les informations dans la base.5. Le système actualise la liste des notes et l'affiche. <p>cas 3 : Supprimer une note</p> <ol style="list-style-type: none">1. l'éducatrice choisit la note de l'enfant à supprimer2. Le système demande une confirmation.3. l'éducatrice confirme ou annule la suppression.4. Le système supprime la note de la base.5. Le système actualise la liste des notes et l'affiche. <p>Cas4 : consulter une note</p> <ol style="list-style-type: none">1. l'éducatrice saisit son login et son mot de passe.
--	--

	<ol style="list-style-type: none">2. Le système affiche la page d'accueil.3. l'éducatrice choisissent de consulter la note d'un enfant.4 .Affichage de la liste des notes avec leurs informations relatives.
Scénario alternatif	<p>cas 1 :</p> <ol style="list-style-type: none">1. la note existe déjà a ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché. <p>cas 2 :</p> <ol style="list-style-type: none">1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché. <p>cas 3 :</p> <ol style="list-style-type: none">1. la note inexistante : un message d'erreur sera affiché. <p>Cas4 :</p> <ol style="list-style-type: none">1. Le système n'arrive pas à identifier le l'éducatrice et affiche un message d'erreur.

Tableau 10: Tableau descriptif du cas d'utilisation gestion des notes.

✓ Cas d'utilisation suivi état de présence:

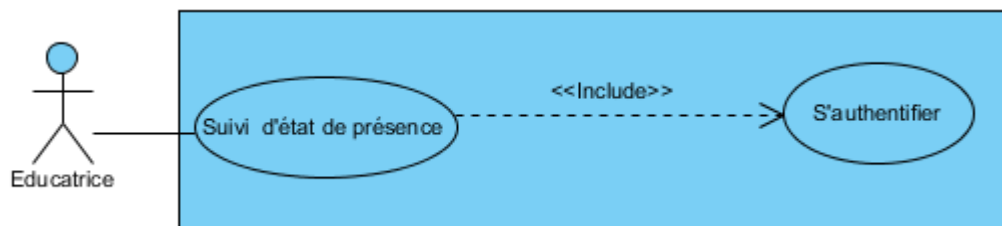


Figure 15:Diagramme de cas d'utilisation « Suivi état de présence ».

Description	
Acteur	Educatrice
Objectif	Consulter la liste des présences des enfants.
Scénario nominal	<ol style="list-style-type: none">1. l'éducatrice saisi son login et son mot de passe.2. Le système affiche la page d'accueil.3. l'éducatrice sélectionne les enfants qui sont présents.4. Le système affiche la liste des présences.
Scénario alternatif	<ol style="list-style-type: none">1. Le système n'arrive pas à identifier l'éducatrice et affiche un message d'erreur.

Tableau 11: Tableau descriptif du cas d'utilisation suivi état de présence.

II.3.2. Analyse des besoins

La phase analyse a pour objectif d'accéder à une compréhension des besoins et exigences du client. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme facilitant la compréhension, la préparation, la médication et la maintenance du système à réaliser.

II.3.2.1. Diagrammes de séquence

II.3.2.2. Les diagrammes de séquence de l'application à réaliser

Dans ce qui suit, nous représenterons les diagrammes de séquences de tous les cas d'utilisations.

- ✓ Diagramme de séquence du cas d'utilisation "S'authentifier":

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification des informations associées à un gestionnaire de point d'intérêt (généralement un login et un mot de passe). Ces informations sont préétablies dans une base de données. Lorsqu'un utilisateur (administrateur, Parent ou Educatrice) veut s'authentifier, trois cas peuvent se présenter : données correctes, données incorrectes ou une inscription qui n'est pas encore validée. C'est pourquoi on utilise l'opérateur «

Alt» (fragments multiple alternatifs "Si, Sinon"). En effet, si les données d'authentification fournies par l'utilisateur sont correctes alors le système accorde l'accès à l'interface appropriée. Dans le cas contraire, un message d'erreur est généré et la page d'authentification est réaffichée.

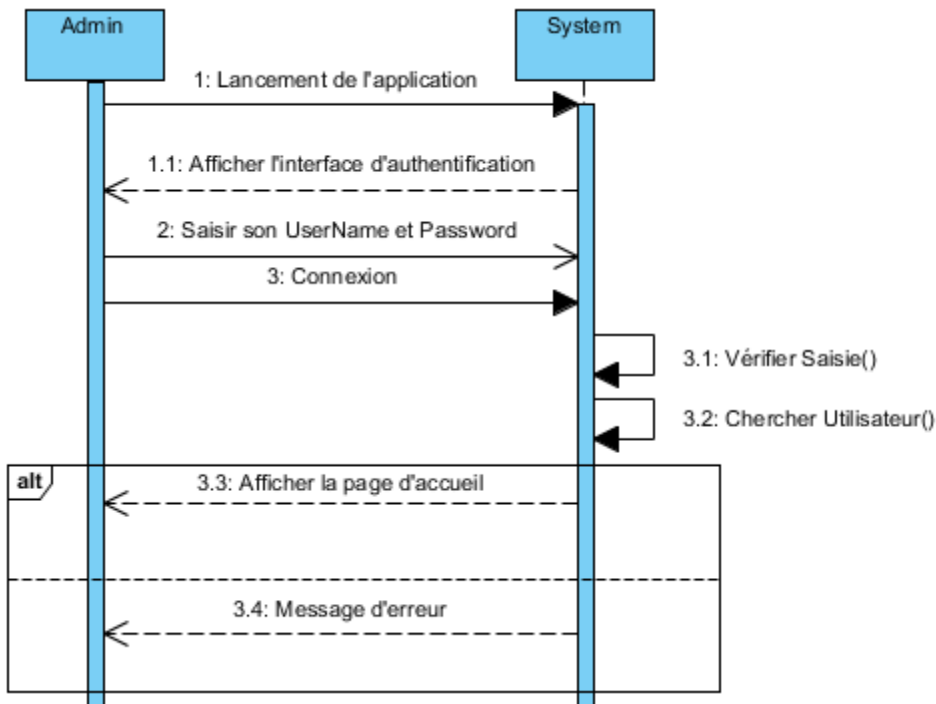


Figure16:Diagramme de séquence « Authentification ».

✓ Diagramme de séquence du cas d'utilisation Parent:

Ce diagramme représente toutes les tâches que peuvent être réalisées par le Parent via le système:

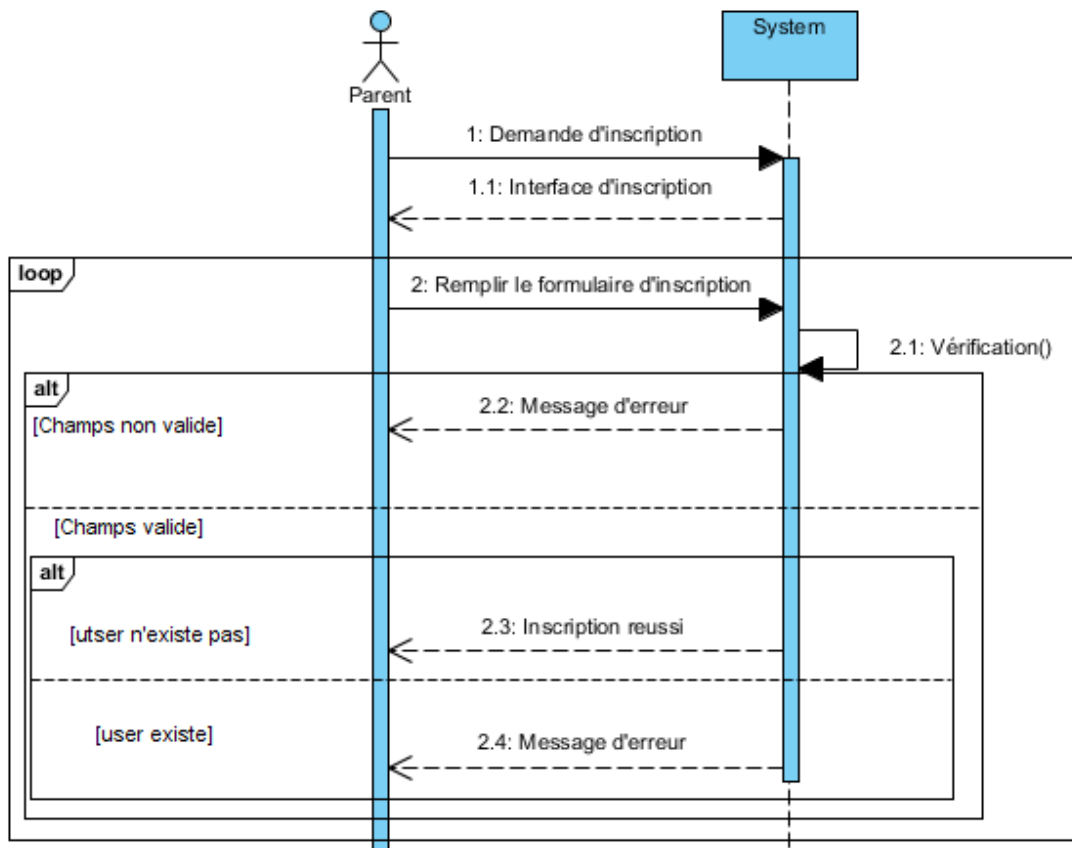


Figure 17: Diagramme de séquence «Parent».

- ✓ Diagramme de séquence du cas d'utilisation « Ajouter un dossier enfant ».

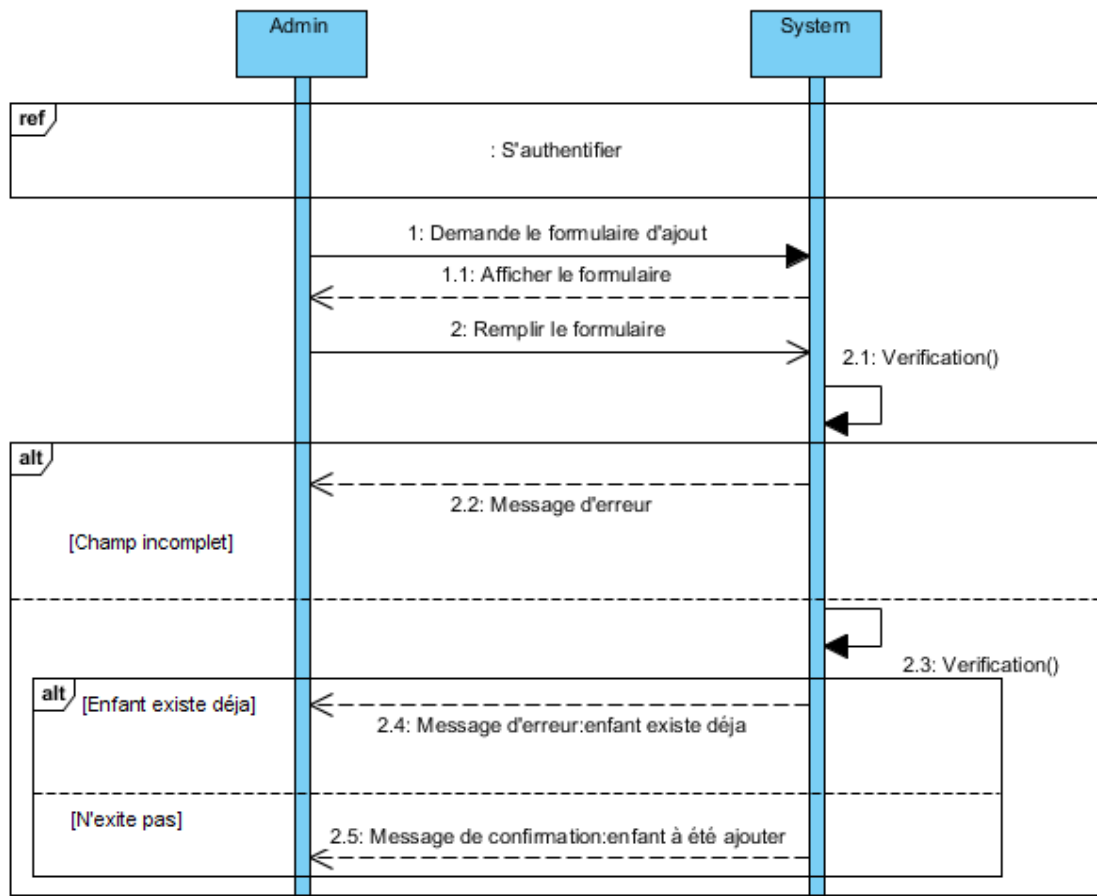


Figure18:Diagramme de séquence « Ajouter un dossier enfant ».

- ✓ Diagramme de séquence du cas d'utilisation « Modifier un dossier enfant ».

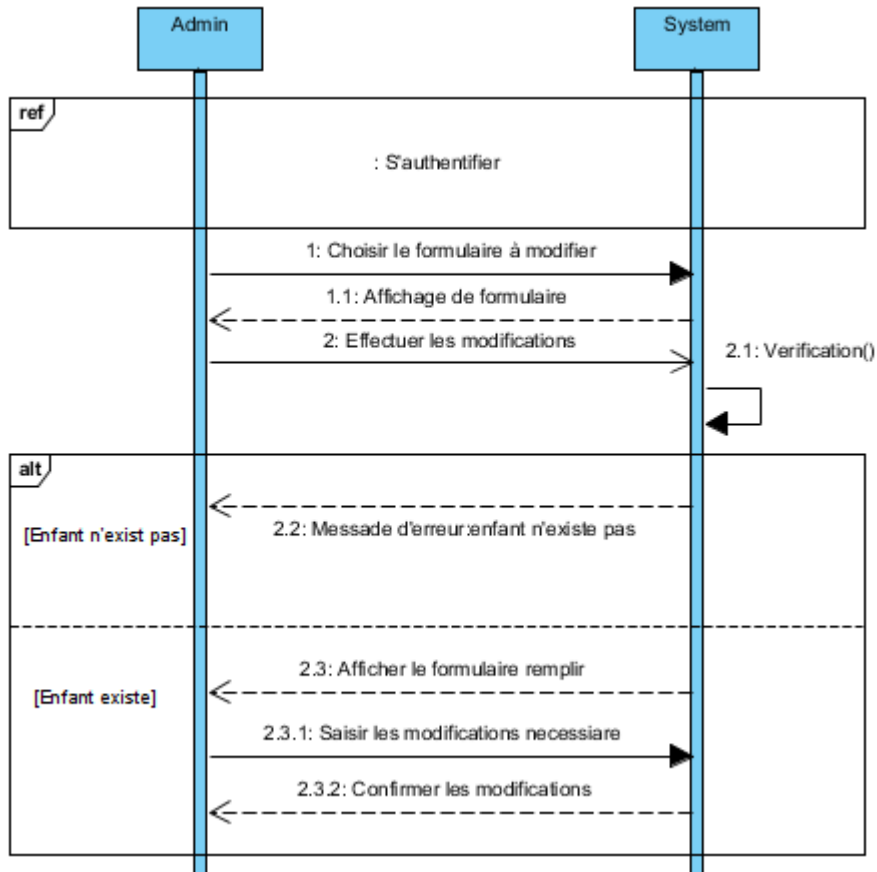


Figure 19: Diagramme de séquence « Modifier un dossier enfant ».

- ✓ Diagramme de séquence du cas d'utilisation « Supprimer un dossier enfant ».

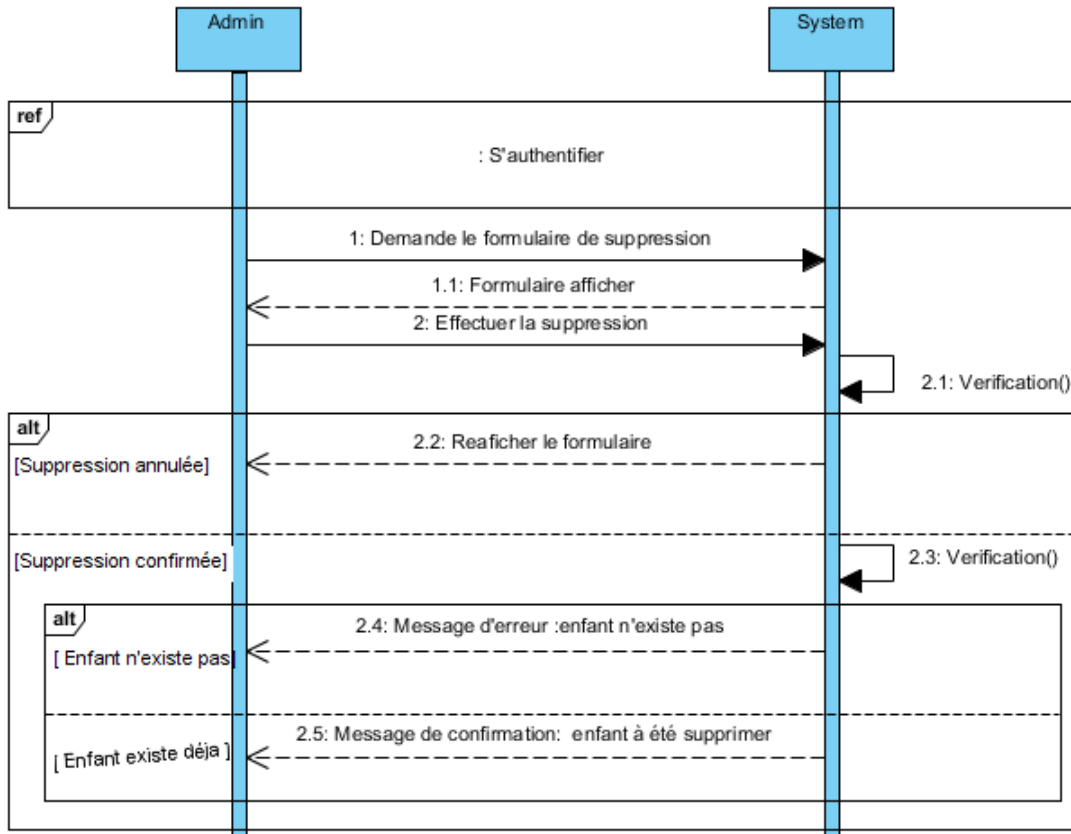


Figure20:Diagramme de séquence « Supprimer un dossier enfant ».

✓ Diagramme de séquence du cas d'utilisation « Gestion des comptes».

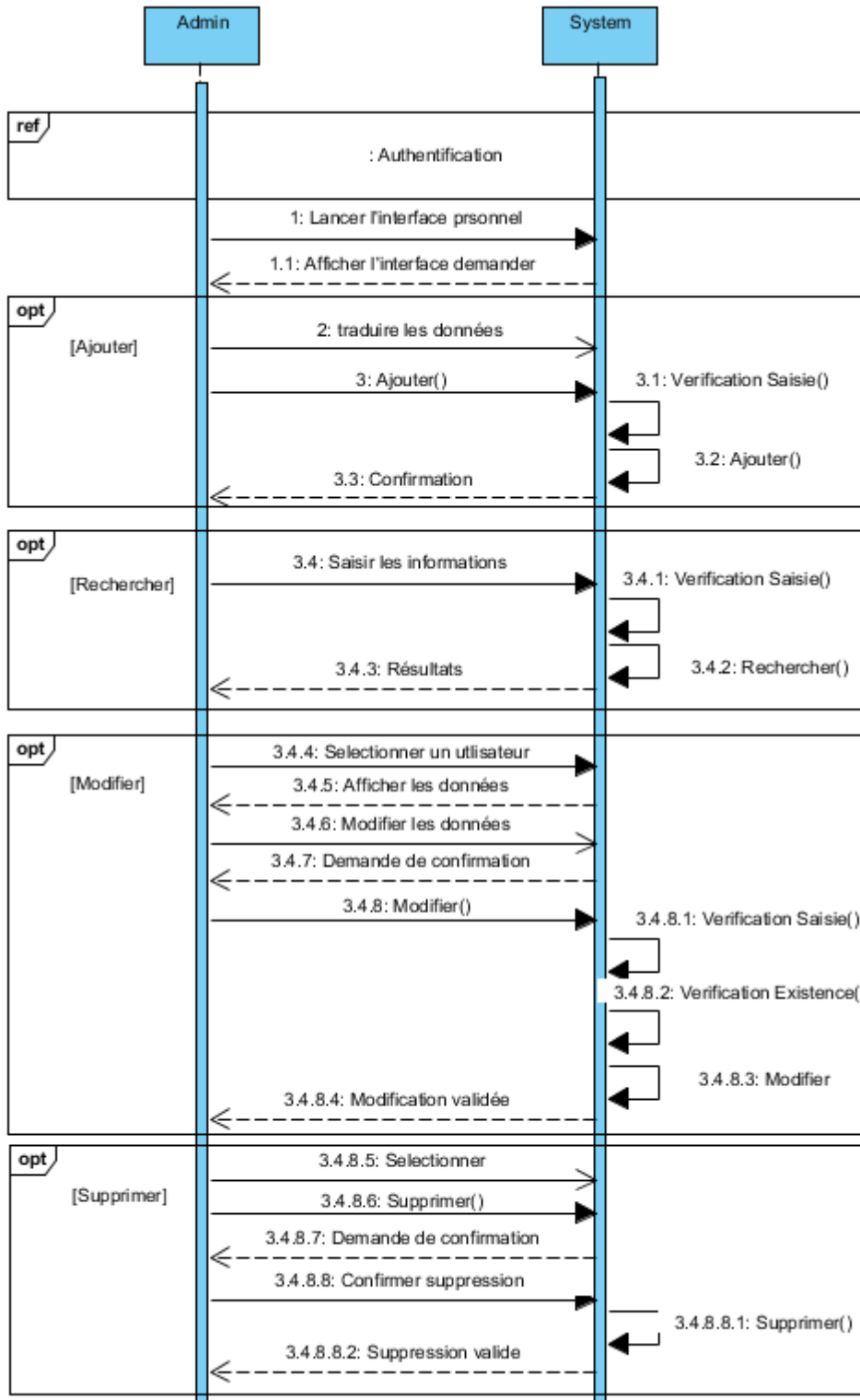


Figure 21 :Diagramme de séquence « Gestion des comptes ».

Conclusion

Dans ce chapitre, nous avons décrit le contexte de notre projet puis nous avons posé la problématique et tracé quelques objectifs.

Ensuite, nous nous sommes intéressés à analyser les besoins de notre application et à spécifier les différents cas d'utilisations en abordant le diagramme des cas d'utilisations et les diagrammes des séquences.

Dans ce qui suit nous allons entamer la conception en présentant les diagrammes des interactions et le diagramme des classes.

Partie04 : conception

II. 4 .Introduction

Dans le cycle de vie de notre projet, la conception représente une phase primordiale et déterminante pour produire une application de haute qualité.

Dans ce qui suit nous présentons les diagrammes d'interactions, puis nous passerons à la description des classes participantes dans le diagramme des classes, et nous construirons le modèle relationnel à partir des classes de l'application, et sa en suivant les règles du passage au relationnel.

II.4.1. Diagramme d'interaction

II.4.1.1. Diagramme d'interaction authentification

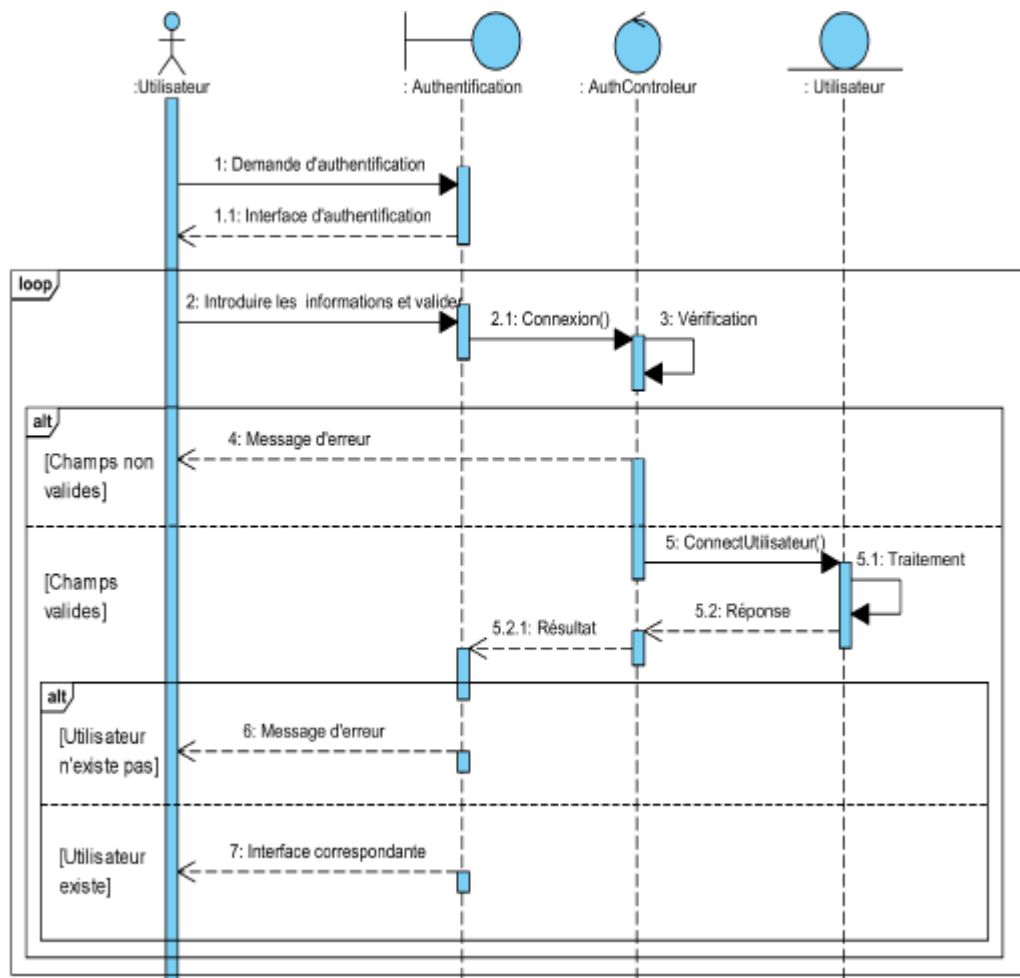


Figure 22 : Diagramme d'interaction authentification d'un utilisateur.

II.4.1.2. Diagramme d'interaction de parent

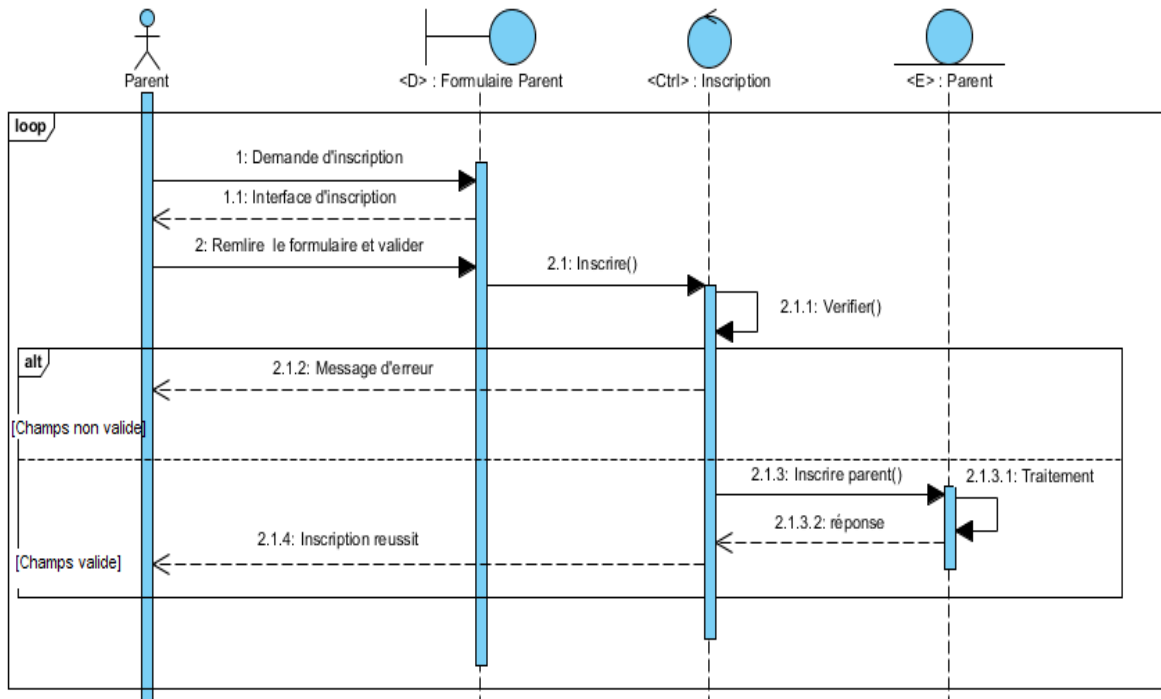


Figure 23 : Diagramme d'interaction parent.

II.4.1.3. Diagramme d'interaction gérer un dossier enfant

cas : ajouter un dossier enfant

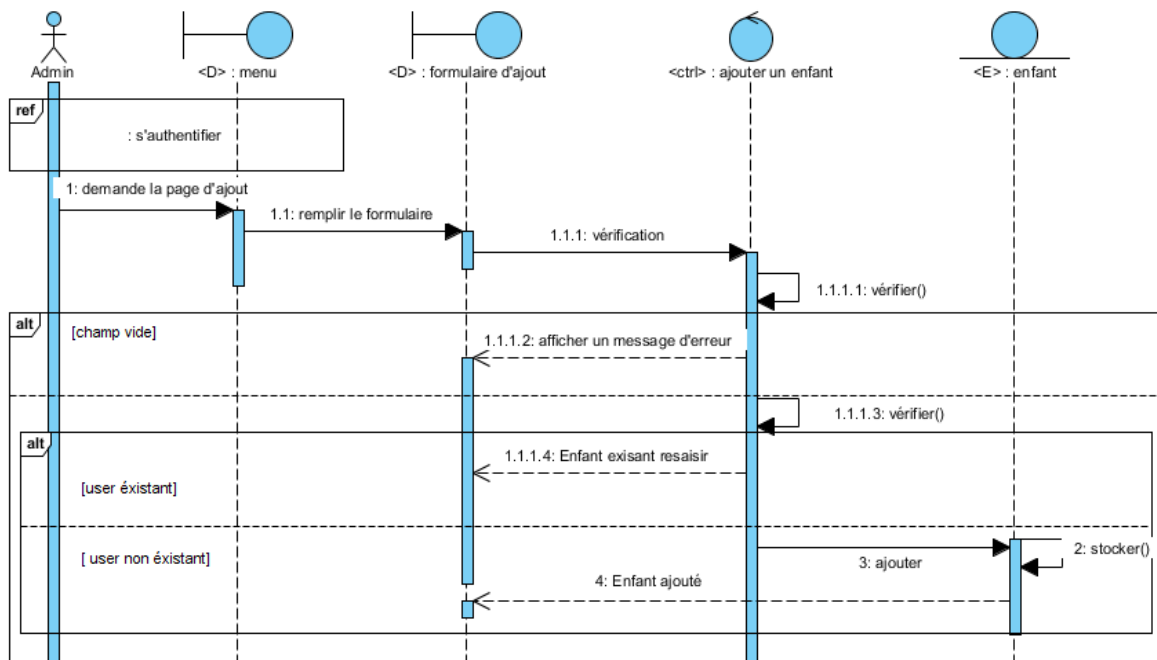


Figure 24 : Diagramme d'interaction ajouter un dossier enfant.

cas : modifier dossier enfant

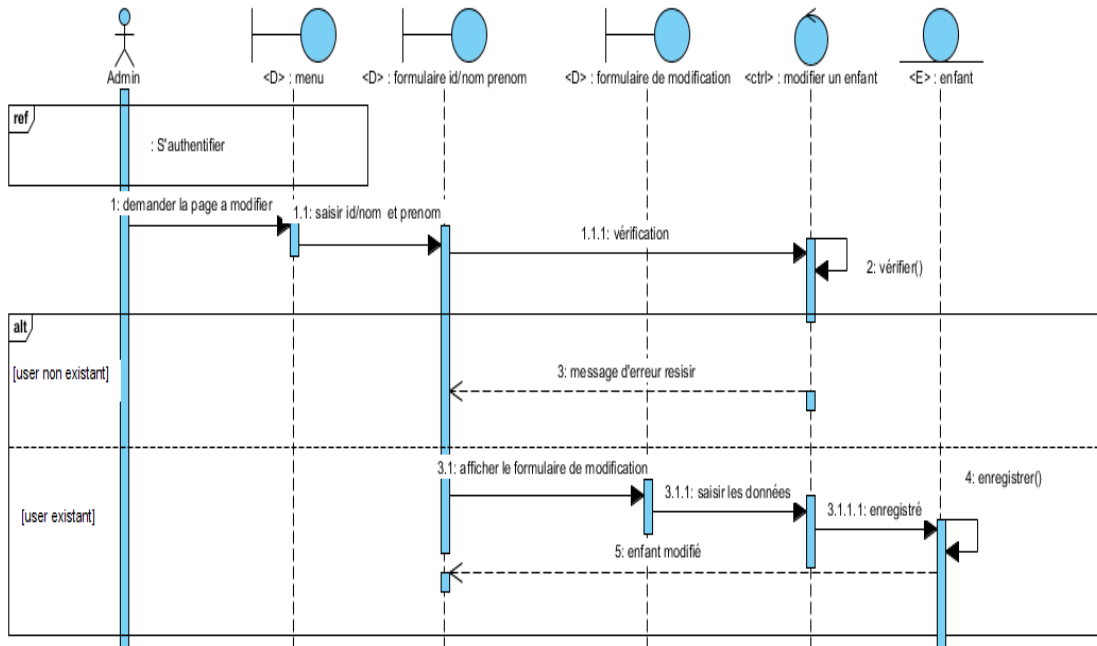


Figure25 : Diagramme d'interaction modifier un dossier enfant.

cas : supprimer un dossier enfant

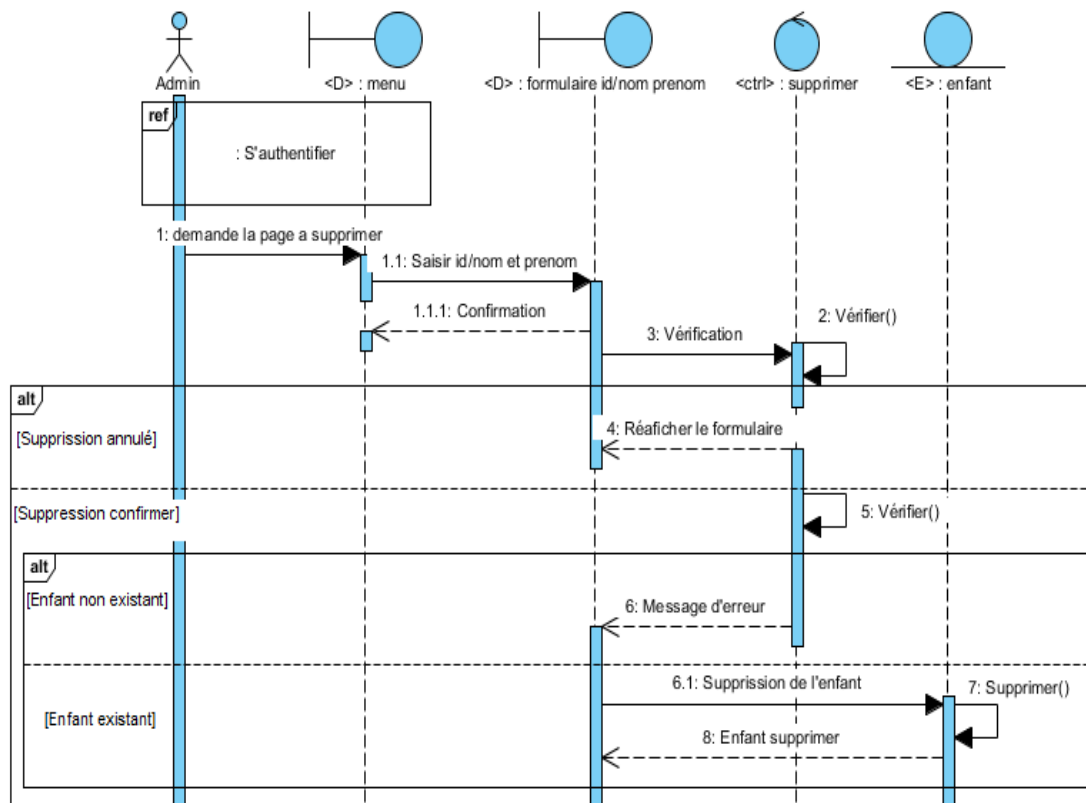


Figure26 : Diagramme d'interaction supprimer un dossier enfant.

Note : c'est les mêmes étapes pour les diagrammes d'interactions de la gestion des éducatrices.

Après avoir spécifié les cas d'utilisations de notre application, il nous reste à décrire les classes participantes dans notre application et à présenter notre diagramme des classes.

II.4.2. Diagramme de classe

II.4.2.1. Présentation des classes et leurs attributs

Classes	Attributs	Définition de l'attribut	Type
Utilisateur	nomuti	Le nom de l'utilisateur	String
	prenom	Le prénom de l'utilisateur	String
	datenaiss	La date de naissance de l'utilisateur	Date
	adresse	L'adresse de l'utilisateur	String
	Tel	Le numéro de téléphone de l'utilisateur	String
	nationalité	nationalité de l'utilisateur	String
	Mdp	Le mot de passe de l'utilisateur	String
	typeuyi	Le type de l'utilisateur soit l'administrateur ou l'éducatrice	String
Administrateur	idAdmin	L'identifiant de l'administrateur	String
Parent	idparent	L'identifiant de parent	Integer
	profession	La profession de père	String

	situationmarital	La situation maritale de père	String
	responenfant	Pour les parent divorcé, qui a la responsabilité de l'enfant	String
	nojeunefillemere	Le nom de jeune fille et prénom de la mère	String
	professionmere	La profession de la mère	String
Educatrice	ideducatrice	L'identifiant de l'éducatrice	Integer
	versement	Le versement de l'éducatrice	Integer
Enfant	numinscription	le numéro d'inscription de l'enfant	String
	nomenf	Le nom de l'enfant	String
	preenf	Le prénom de l'enfant	String
	datenaisse	La date de naissance de l'enfant	Date
	lieunaiss	Le lieu de naissance d'un enfant	String
	adresse	L'adresse de l'enfant	String
	Sexe	Le sexe de l'enfant	String
	dateentre	La date d'entrée de l'enfant	Date
	prblmsante	Problème particuliers concernant la santé	String
	nomprenrecup	Le nom et le prénom	String

		des personnes qui ont la possibilité de récupérer l'enfant en cas d'absence des parents	
	path	L'emplacement de code de la photo	String
Section	Codes	Code de section	String
	Types	Type de section	String
Groupe	Idgroupe	L'identifiant de groupe	String
	numgroupe	Le numéro de groupe	Integer
Note	Idnote	L'identifiant de la note	Integer
	Note	La note obtenue par l'enfant	Float
	Matiere	Les matières à étudier par l'enfant	String
Planning	Date	La date	Date
	Jour	Le jour	String
	Heur	L'heure	Integer

Tableau 12: Description des classes et leurs attributs.

II.4.3. Diagramme de classe de l'application à réaliser

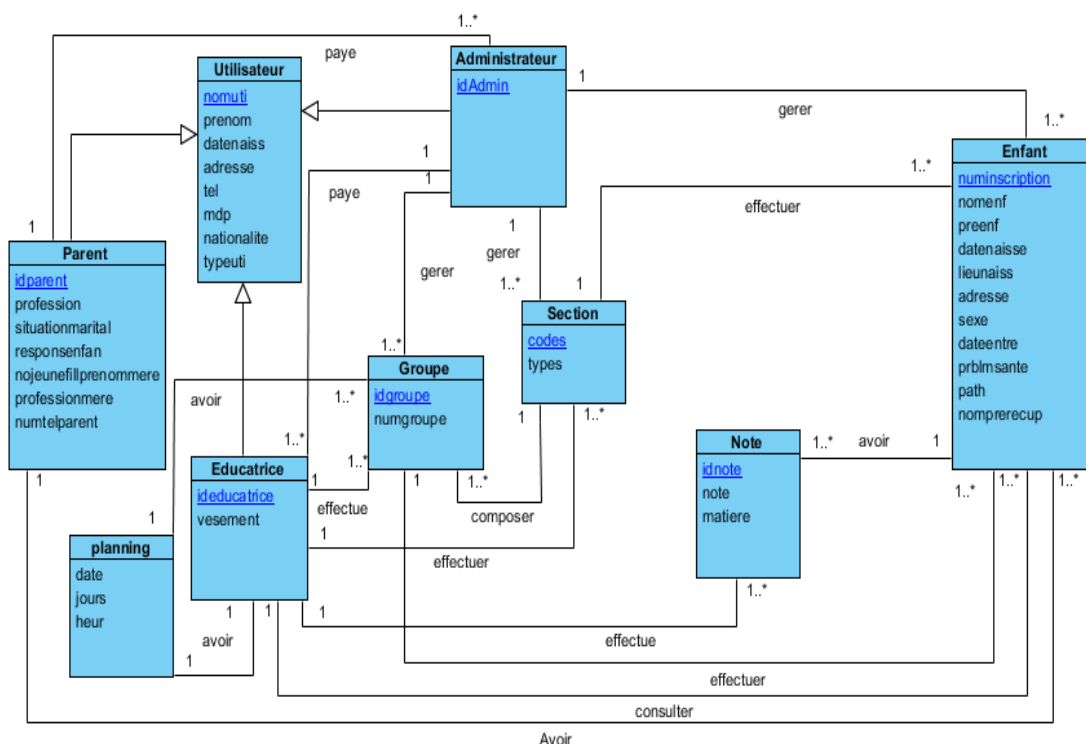


Figure27:Diagramme de classe.

II.4.4. Passage au modèle relationnel

Après avoir élaboré le diagramme de classe, ce dernier peut être traduit en modèle relationnel, c'est à dire en un ensemble de relations en troisième forme normale. Cet ensemble de relations peut être implémenté directement, sous forme d'une base de donnée, moyennant un système de gestion de base de données relationnelle.

II.4.4.1. Règles de passage du diagramme de classe au modèle relationnel

Les règles utilisées pour le passage du modèle des classes au modèle relationnel sont les suivantes:

- **Règle 1: Transformation de classe**

Chaque entité devient une relation. L'identifiant de l'entité devient une clé primaire pour la relation.

- **Règle 2: Association un à plusieurs**

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

• **Règle 3: Association plusieurs à plusieurs**

L'association devient une relation, dont la clé primaire est composée par la concaténation des identifiants de classes connectées à l'association. Le modèle relationnel équivalent à notre diagramme de classe est le suivant:

Utilisateur(nomuti, prenom, datenaiss, adresse, tel, nationalite, mdp, typeuti, idparent, profession, situationmarital, responenfant, nojeunefillemere, professinmere, ideducatrice, versement) ;

Enfant(Numinscription, nomencl, preenf, datenaiss, lieunaiss, adresse, sexe, dateentre, prblmsante, nomprenrecup, path, #codes, #idgroup, #nomuti,) ;

Section(codes, type, #nomuti) ;

Groupe (idgroupe, numgroupe, #codes, #nomuti) ;

Note(idnote, note,, matiere, #numinscription, #nomuti,) ;

Planning(date, jour, heure);

Conclusion

Dans ce chapitre nous avons défini les différentes relations entre les acteurs en se basant sur les diagrammes de cas d'utilisation. Nous avons également présenté le système de fonctionnement avec les diagrammes de séquences ; et réalisé un diagramme de classes pour définir toutes les classes participantes à la base de données. Aussi, la partie conceptuelle est très importante pour réaliser n'importe quelle application, car elle permet de bien voir l'architecture de cette dernière. Une fois que la phase de conception est terminée, nous pouvons passer à la phase de réalisation qui fera l'objet du chapitre suivant.

Chapitre III

1 .Introduction

Dans ce chapitre, nous présentons la partie réalisation qui est la phase finale de notre projet. Nous décrirons l'environnement ainsi que les outils qui nous ont servis pour la réalisation de l'application et nous terminerons enfin par la présentation de l'application. Nous nous sommes appuyés sur le Java Enterprise Edition comme environnement de développement, et sur différents langages pour l'implémentation, ceux qu'on va définir juste après.

III.1. L'environnement de développement

Le Java Enterprise Edition "Java EE" possède un grand nombre de bibliothèque remplissant différentes fonctionnalités ; il facilite le développement d'applications web robustes et distribuées.

Dans le cadre de notre application, nous avons choisi d'utiliser les API suivants:

- ✓ **Servlet:** C'est la classe Java, sur laquelle nous avons implémenter le code Java de l'application (Programmation).
- ✓ **JSP (Java Server Pages):** C'est un simple fichier, qui utilise comme langage le HTML dans le but de réaliser les vues dynamiques (Interfaces).
- ✓ **JDBC (Java DataBase Connectivity):** C'est un code écrit en Java, permettant la connexion à la base de données, et l'accès aux données stockées sous forme de tables. Ceci en utilisant un pilote, comme le "mysql-connector-java-5", dans notre cas.
- ✓ **JavaBean:** C'est un objet java, réutilisable, qu'on associe à une table de la base de données pour mieux représenter les données réelles.
- ✓ **JavaEE(Java Enterprise Edition) :**

La figure (**Figure 28**) représente le patron de conception MVC (Modèle, Vue, Control), associé aux API définis ci-dessus:

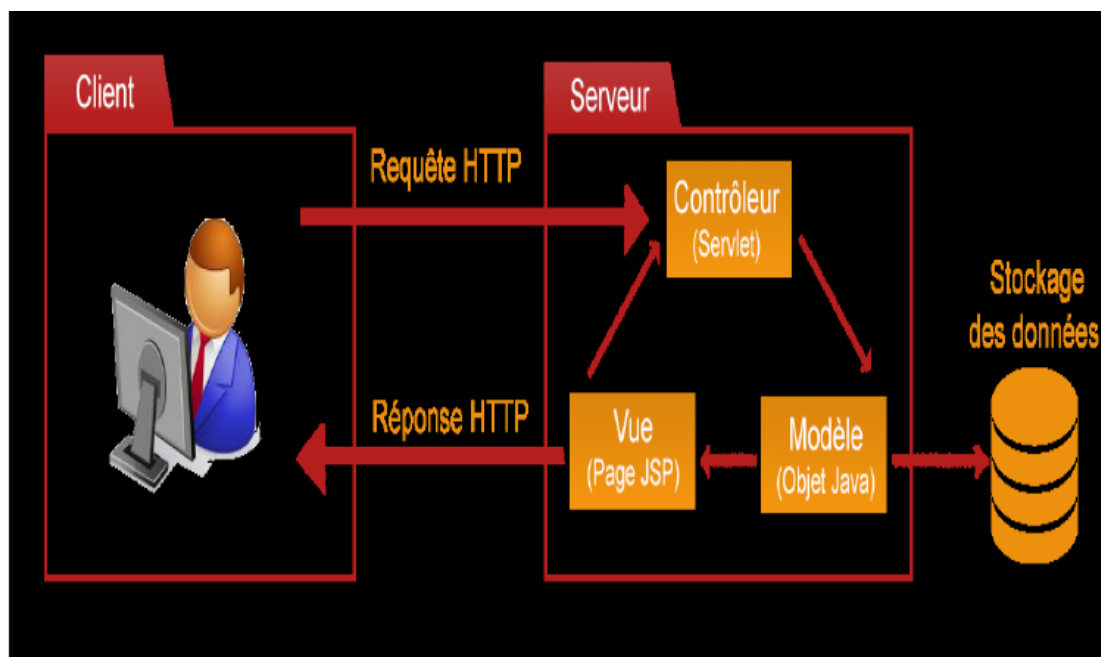


Figure 28: Le design pattern MVC.

III.2. Langages utilisés

Au cours de la réalisation de l'application web, une multitude de langages de programmation ont été mis en œuvre afin d'implémenter ce qui a été défini lors de la phase de conception. Pour cela, nous avons utilisé:

III.2.1. HTML (Hyper Text Markup Language)

Le langage HTML est le langage universel utilisé sur les pages web lisibles par tous les navigateurs web (Internet Explorer, Netscape, Mozilla, etc...). Ce langage fonctionne suivant l'assemblage et la combinaison de balises permettant de structurer et donner l'apparence voulue aux données textes, images et multimédias suivant la mise en page voulue. C'est un langage qui permet de créer des hyperliens, à savoir des liens d'un document à un autre ou d'un endroit d'un document à un autre endroit du même document (identificateur de fragment) [6].

III.2.2. CSS

Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML [W6].

III.2.3. Bootstrap

Kit CSS créé par les développeurs de Twitter, il permet de construire rapidement et facilement des sites web esthétiques et responsives. Bootstrap offre aussi des plugins jQuery de qualité pour enrichir les pages du site [W7].

III.3. Outils et logiciels utilisés

Les langages de programmations seuls, ne peuvent pas nous permettre d'avancer rapidement, pour cela nous allons nous servir de quelques logiciels : On a choisi NetBeans pour une bonne implémentation et une visibilité du code, ainsi que sa gratuité. Comme nous le savons, Java ne peut pas s'exécuter sans un environnement, donc nous avons opté pour WAMP serveur pour que nous puissions travailler en local.

III.3.1. MySQL (My Structured Query Language)

MySQL est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données [W8].

III.3.2. L'IDE NetBeans

NetBeans est un Environnement de Développement Intégré (EDI) ou bien en anglais Integrated Development Environment (IDE), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License), en plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le PHP et le HTML. NetBeans constitue et s'appuie sur une plateforme, qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)) [W9].

III.3.3. WampServer

WampServer (anciennement WAMP5) est une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de

script (PHP), ainsi que PHPMysqlAdmin pour l'administration web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icône (icône près de l'horloge de Windows).

Le serveur web « Apache est l'un des serveurs les plus réponsus sur internet et il est basé sur le protocole HTTP. L'objectif est de fournir, à un maximum de personnes et d'organisations, une plate-forme solide pour réaliser des tests et des applications de productions, son grand avantage est le fait qu'il fonctionne sur de nombreux systèmes d'exploitation dont Microsoft Windows, Linux,... MySQL et PHP sont respectivement un Système de Gestion de Bases de Données (SGBD et un langage impératif orienté objet »[W10].

III.4. Présentation du système

Notre système du crèche est constitué de deux applications, une est de type Desktop pour la gestion de la crèche, et l'autre, c'est une application Web, pour le parent.

III.4.1. Interfaces de l'application Desktop

➤ Interface d'accueil

C'est la première interface, où l'administrateur sélectionne une rubrique pour la fonctionnalité à gérer:



Figure 29: Interface d'accueil.

➤ Interface d'authentification

L'utilisateur (Administrateur / Educatrice) doit remplir les deux champs d'authentification en saisissant les coordonnées correctes au qu'il puisse accéder a la page d'accueil.

The image shows a software window titled 'Mère éducatrice'. At the top, there is a logo consisting of a stylized 'M' inside a circle, followed by the text 'Mère éducatrice'. Below the logo is a cartoon illustration of four children's faces. The main area of the window contains a login form with three fields: 'Type utilisateur' with a dropdown menu currently set to 'Administrateur', 'Nom utilisateur' with an empty text box, and 'Mot de passe' with an empty text box. At the bottom of the form area, there are two buttons: 'Connexion' and 'Annuler'.

Figure 30 : Interface authentification.

➤ Interface gestion des enfants :

L'administrateur inscrit les enfants et consulte les notes de chacun .Il peut aussi modifier, ajouter ou supprimer un enfant au niveau de cette liste.



Consulter les notes

N° inscription: **GS42020**













Gestion des enfants

Veuillez renseigner les champs:

Nom de l'enfant:	<input type="text" value="adam"/>	adresse:	<input type="text" value="semaoune"/>	lieu de naissance:	<input type="text" value="sidi aich"/>
Prénom de l'enfant:	<input type="text" value="djerroud"/>	Nom prénom de récupération de l'enfant:	<input type="text" value="djerroud samy"/>		
date naissance :	<input type="text" value="04/01/2015"/>	sexe:	<input checked="" type="radio"/> Garçon <input type="radio"/> Fille		
date d'entrée:	<input type="text" value="10/10/2020"/>	N° Groupe:	<input type="text" value="GS1"/>		
problème particuliers concernat sa santé:	<input type="text" value="aucun"/>				

Age: 5 Section attribuée: GS:Grande Section



Charger la photo

s enfant5ans (1).jpeg

N° inscri...	Nom	Prénom	Date nai...	Lieu de ...	Adresse	sexe	Date d'e...	Problem...	photo	codes	path	nom rec...	idgroup
GS42020	adam	djerroud	04/01/20...	sidi aich	semaou...	Garçon	10/10/2...	aucun	[B@783...	GS	C:\User...	djerroud...	GS1
GS52020	aylan	mahidine	06/06/20...	sidi aich	aidi aich	Garçon	23/09/2...	aucun	[B@345...	GS	C:\User...	mahidin...	GS1
GS72020	mayliss	debbou	23/03/20...	sidi aich	sidi aich	Fille	04/04/2...	aucun	[B@368...	GS	C:\User...	debou ...	GS1
GS82020	axel	berkane	05/05/20...	amizour	amizour	Garçon	12/12/2...	aucun	[B@11b...	GS	C:\User...	berkane ...	GS2
MS12020	wassim	faid	12/12/20...	sidi aich	sidi aich	Garçon	03/02/2...	aucun	[B@2c7...	MS	C:\User...	faid mat...	MS1
MS32020	lina	ferhat	07/02/20...	alger	alger	Fille	02/09/2...	aucun	[B@3ae...	MS	C:\User...	faid zahir	MS1
PS22020	yasmine	faid	05/04/20...	sidi aich	sidi aich	Fille	09/09/2...	aucun	[B@5a1...	PS	C:\User...	faid zahir	PS1
PS62020	dassine	djabar	04/04/20...	sidi aich	timzrit	Fille	12/09/2...	aucun	[B@520...	PS	C:\User...	djerroud...	PS1



Figure 31 : Interface gestion des enfants.

➤ Interface gestion des sections

Cette interface est utilisée pour consulter la liste des sections et des groupes des enfant qu'ils appartiennent.

Gestion des sections

Saisir la section :

Groupe:

Nom	Prénom	codes	N°Groupe
adam	djerroud	GS	GS1
aylan	mahidine	GS	GS1
mayliss	debbou	GS	GS1
axel	berkane	GS	GS2
wassim	faid	MS	MS1
lina	ferhat	MS	MS1
yasmine	faid	PS	PS1
dassine	djabar	PS	PS1

Figure 32 : Interface gestion des sections.

➤ **Interface d'emploi de temps :**

La réalisation de l'emploi du temps se fait automatiquement par le logicielle 'fet'.

Exemple d'interface emploi du temps avec logiciel 'fet'.

Gestion d'emploi du temps

EMPLOI DU TEMPS

	LUNDI	MARDI	MERCREDI	JEUDI	VENDREDI	SAMEDI	DIMANCHE	
08:00								08:00
09:00								09:00
10:00								10:00
11:00								11:00
12:00								12:00
13:00								13:00
14:00								14:00
15:00								15:00
16:00								16:00
17:00								17:00
18:00								18:00
19:00								19:00
20:00								20:00

Figure 33 : Interface gestion des emplois de temps.

III.4.2. Interfaces de l'application Web:

➤ Interface d'accueil

C'est la première interface que trouvera le parent après avoir taper le lien du site:

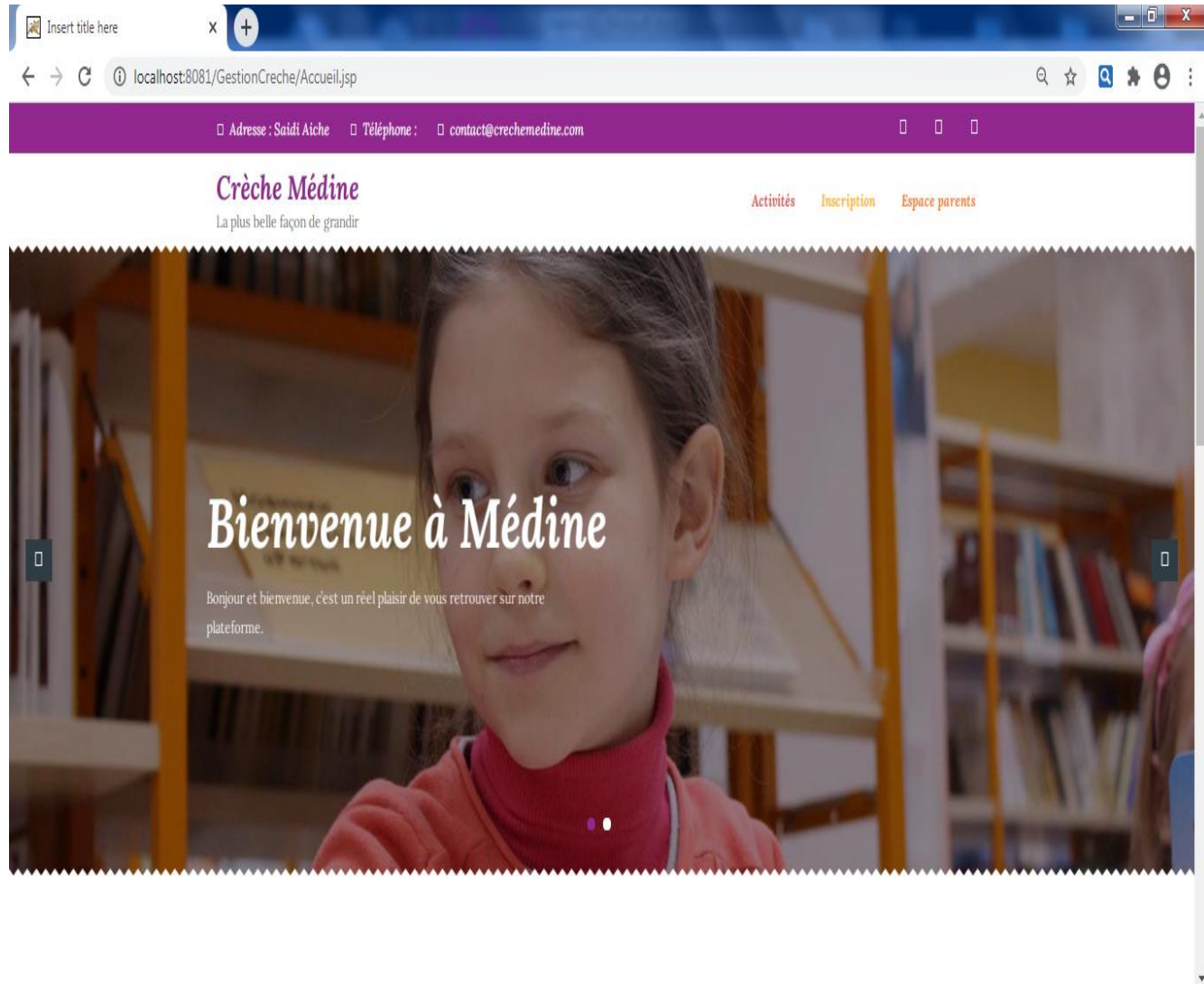


Figure 34 : Interface d'accueil de l'application Web.

➤ Interface d'inscription

Pour bénéficier des fonctionnalités de notre application, le parent doit s'inscrire et avoir un compte-utilisateur:

Réalisation

The image shows a web browser window with the address bar displaying 'localhost:8081/GestionCreche/Inscription.jsp'. The main content is a registration form titled 'FICHE D'INSCRIPTION'. The form contains the following fields and controls:

- Instruction: 'Veuillez Renseigner les champs'
- Nom de l'enfant :
- Prénom de l'enfant:
- Date de naissance :
- Lieu de naissance:
- adresse:
- Sexe: Garçon Fille
- Date d'entrer:
- Problème particuliers concernant sa santé:
- Nom du père:
- Prénom du père:
- Profession du père:
- Nom de jeune fille et Prénom de la mère :
- Profession de la mère :
- Situation maritale des parent :
- Pour les parents divorcés, qui a la responsabilité de l'enfants? :
- Numéro de telephone:
- Nom et Prénom des personnes ayant habilité à récupérer l'enfant en l'absence des parents:
- Submit button: 's'inscrire'

Figure 35 : Interface d'inscription en ligne

➤ Interface d'authentification

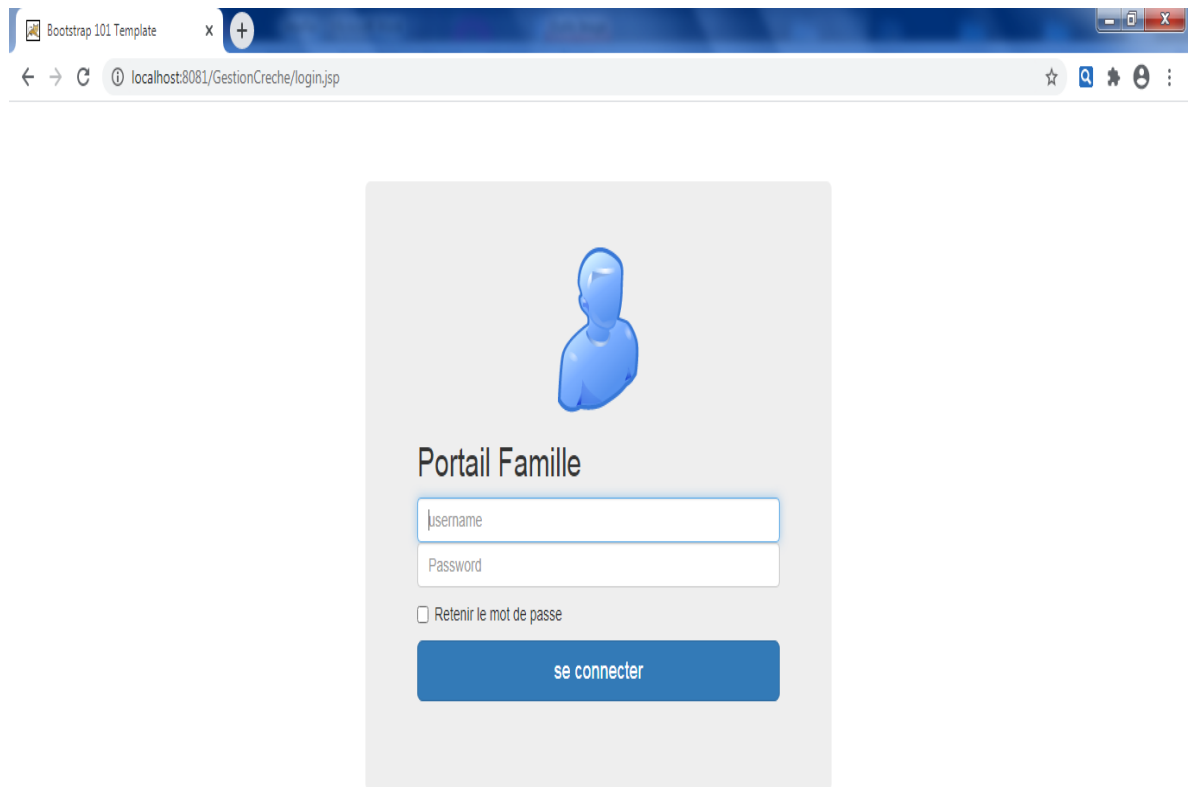


Figure 36: Interface d'authentification en ligne.

Conclusion

Ce chapitre a été consacré à la phase de réalisation. Cette phase est le fruit de nos efforts tout au long de la durée du projet. Dans ce chapitre, nous avons présenté les outils et langages utilisés pour la réalisation de notre application, montré plusieurs interfaces Homme-Machine (IHM) de cette dernière.

Conclusion générale

L'objectif de notre travail est la gestion d'une crèche pour cela nous avons utilisé pour la conception UML comme langage de modélisation ; et «2TUP » comme méthode de développement Agile.

Nous avons réalisé deux applications une web et une autre desktop, gérées par une seule base de donnée en commun.

L'application web que nous avons développée permet :

- aux parents de faire des inscriptions en ligne
- de consulter les notes de leurs enfants
- de communiquer avec les parents
- d'avoir des informations sur la crèche.

Par contre l'application desktop permet la gestion automatique de la crèche.

Elle permet, au responsable

- Le suivi pédagogique à savoir : les plannings d'enseignement, les sections, les éducatrices et les notes des enfants.
- De gérer la comptabilité de l'établissement qui consiste en le paiement des parents et les salaires des éducatrices

Notre application peut évoluer et en guise de perspectives, nous proposons:

- ✓ Une application mobile, pour des notifications en temps réel
- ✓ Intégration du paiement en ligne, pour que la procédure soit purement automatique
- ✓ Inclure des cours, des activités et même des évaluations à distance

Bibliographie

- [1] YAKOUBEN Djaafar et MESSOUAF Sara, *Conception et réalisation d'une application Androïde : Localisation des boutiques, hôtels, restaurants... etc. Se trouvant à proximité*, mémoire en informatique, université de Béjaia. 2015.
- [2] P. Roques et Franck Vallée, "*UML 2 en action, de l'analyse des besoins à la conception*", Editions Eyrolles, 4e Edition, 382 pages, 2007.
- [3] P. Roques, "*UML2 par la pratique, Editions Eyrolles*", 5e Edition, 2006. 5
- [4] P. Roques, "*UML2 Modéliser une application Web*", Editions Eyrolles, 4e Edition, 2008.
- [5] Pascal ROQUES. Les Cahiers du Programmeur UML 2 Modéliser une application web. ISBN : 978-2-212-12389-0. 4eme édition, Groupe Eyrolles, 2008.
- [6] HTML et JavaScript, Daniel CHARNAY, Philippe CHALEAT, 2014.

Webographie

- [W1] www.ideematic.com/dictionnaire-web, consulter le 15/11/2019.
- [W2] http://www.streetdirectory.com/travel_guide/114448/programming/desktop_applications_vs_web_applications.html, consulter le 15/11/2019.
- [W3] [tps://www.google.com/search?q=application+desktop&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi0ka_Nj_DfAhWbAWMBHV6JB8gQ_AUIDigB&biw=1366&bih=657#imgdii=3L_V7R3kwHDVxM:&imgsrc=G8teoFR1KYA5jM](https://www.google.com/search?q=application+desktop&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi0ka_Nj_DfAhWbAWMBHV6JB8gQ_AUIDigB&biw=1366&bih=657#imgdii=3L_V7R3kwHDVxM:&imgsrc=G8teoFR1KYA5jM), consulté le 10/01/2020.
- [W4] <https://fr.yeeply.com/blog/6-types-developpement-d-applications-web/> consulter le 05/02/2020.
- [W5] <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/sequence-diagram> ,consulter le 05/06/2020.
- [W6] <https://www.w3.org/Style/CSS>, consulté le 09/08/2020.
- [W7] <http://openclassrooms.com/courses/prenez-en-main-bootstrap>, consulter le 12/09/2020.
- [W8] <http://www.futura-sciences.com/magazines/hightech/infos/dico/d/internet-mysql-4640>, consulter le 20/09/2020
- [W9] <https://fr.wikipedia.org/wiki/NetBeans>, consulter le 20/09/2020.
- [W10] www.wampserver.com, consulter le 21/09/2020.

Résumé

L'utilisation des applications informatiques devient indispensable dans tous les domaines, notamment dans le secteur des crèches qui rencontrent des difficultés pour gérer les différentes tâches et ainsi avoir un accès rapide aux informations et optimiser le temps et l'espace de stockage de données.

Notre projet de fin d'étude consiste à la conception et l'implémentation d'une application desktop synchronisée avec une application web pour une crèche.

Pour concevoir notre système, nous avons utilisé UML comme langage de modélisation et «2TUP » comme méthode de développement agile. Nous avons réalisé nos applications, afin de les valider, et procédé à un ensemble de tests.

Abstrac

The use of computer applications is becoming essential in all areas, particularly in the nurse sector, which has difficulties in managing the various tasks and thus has fast access to informations and optimizes the time and data storage space.

Indeed, our graduation project consists in the design and the implementation of a desktop application synchronize with a web application.

To design our system, we used UML as a modeling language and "2TUP" as an agile development method.

We realized our applications, to validate them, and carried out a set of tests.