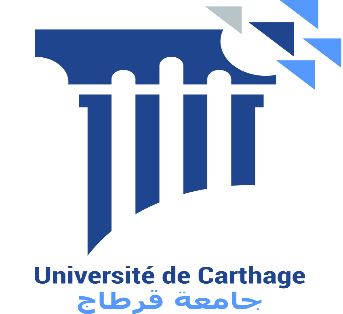
** MINISTERE DE L’ENSEIGNEMENT SUPERIEUR **

**ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE DE CARTHAGE**

**Ecole Nationale d'Ingénieurs de Carthage**

***Projet de conception informatique***

*Spécialité :*

***Génie des systèmes industriel et logistique***

***Application de gestion d’école primaire***

**Encadré par : Réalisé par :**

**Mme.Daassi Ola Houimel Firas**

**Encadrant académique Marnissi Mahjoub**

**Nasr Ichrak**

**Année Universitaire: 2020/2021**

**Table de tableaux**

[**Table 1 Identification des besoins fonctionnels** 8](#_Toc69854925)

[**Table 2 Description textuelle de cas d’utilisation s’authentifier** 13](#_Toc69854926)

[**Table 3 Description textuelle de cas d’utilisation gérer les élèves** 14](#_Toc69854927)

[**Table 4 Description textuelle de cas d’utilisation frais d’inscription** 15](#_Toc69854928)

[**Table 5 Description textuelle de cas d’utilisation imprimer emploi** 15](#_Toc69854929)

[**Table 6 Description textuelle de cas d’utilisation consulter emploi** 16](#_Toc69854930)

**Listes des figures**

[**Figure 1Diagramme des cas d’utilisation principal** 10](#_Toc70085452)

[**Figure 2 Diagramme des cas d’utilisation « Gérer les gestionnaires »** 11](#_Toc70085453)

[**Figure 3 Diagramme des cas d’utilisation « Gérer les élèves »** 11](#_Toc70085454)

[**Figure 4 Diagramme des cas d’utilisation « Gérer les professeur »** 12](#_Toc70085455)

[**Figure 5 Diagramme des cas d’utilisation « Gérer les emplois du temps »** 12](#_Toc70085456)

[**Figure 6 Diagramme des cas d’utilisation « Gérer les frais scolaires »** 13](#_Toc70085457)

[**Figure 7 Diagramme de séquence authentification** 17](#_Toc70085458)

[**Figure 8Diagramme de séquence Ajouter élève** 18](#_Toc70085459)

[**Figure 9Diagramme de séquence modifier élève** 18](#_Toc70085460)

[**Figure 10 Diagramme de séquence gérer élève** 19](#_Toc70085461)

[**Figure 11 Diagramme de séquence imprimer emploi** 19](#_Toc70085462)

[**Figure 12Diagramme de séquence gérer les frais scolaires** 20](#_Toc70085463)

[**Figure 13 Diagramme de classe** 24](#_Toc70085464)

[**Figure 14 Diagramme de séquence objet modifier élève** 25](#_Toc70085465)

[**Figure 15 Diagramme de séquence objet ajouter élève** 26](#_Toc70085466)

[**Figure 16 Diagramme de séquence frais scolaire** 27](#_Toc70085467)

[**Figure 17 Diagramme de séquence objet imprimer emploi** 28](#_Toc70085468)

[**Figure 18 Représentation graphique des nœuds d'activité** 29](#_Toc70085469)

[**Figure 19 Diagramme d'activité gérer les notes** 30](#_Toc70085470)

Table des matières

[Introduction générale 1](#_Toc69854931)

[Chapitre 1: 3](#_Toc69854932)

[**I.** **Introduction** 4](#_Toc69854933)

[**II.** **La méthodologie et le formalisme adopté** 4](#_Toc69854934)

[**III.** **Le formalisme adopté** 4](#_Toc69854935)

[**IV.** **Le Specification des besoins** 5](#_Toc69854936)

[**V.** **Conclusion** 5](#_Toc69854937)

[Chapitre 2: 6](#_Toc69854938)

[**I.** **Introduction** 7](#_Toc69854939)

[**II.** **L’identification des acteurs** 7](#_Toc69854940)

[**III.** **Les Besoins fonctionnels** 8](#_Toc69854941)

[**IV.** **Les Besoins non-fonctionnels** 9](#_Toc69854942)

[**V.** **Le diagramme de cas d’utilisation** 9](#_Toc69854943)

[**1.** **Diagramme de cas d’utilisation** 9](#_Toc69854944)

[**2.** **Diagramme du cas d'utilisation raffinée** 11](#_Toc69854945)

[**3.** **La description textuelle de cas d’utilisation** 13](#_Toc69854946)

[**VI.** **Diagramme de séquence système** 17](#_Toc69854947)

[**VII.** **Conclusion** 20](#_Toc69854948)

[Chapitre 3: 21](#_Toc69854949)

[**I.** **Introduction** 22](#_Toc69854950)

[**II.** **Diagramme de classes** 22](#_Toc69854951)

[**III.** **Diagramme de séquence objet** 25](#_Toc69854952)

[**IV.** **Diagramme d’activité** 27](#_Toc69854953)

[**V.** **Conclusion** 29](#_Toc69854954)

[Dans ce chapitre, nous avons conçu toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (ensemble de diagrammes) prêt à être codé. 29](#_Toc69854955)

[**Conclusion générale** 30](#_Toc69854956)

# Introduction générale

Actuellement, le monde connait un progrès technologique considérable dans tous les secteurs et cela à l’aide de l’informatique, qui joue un rôle important dans le développement de nombreuses entreprises et organisations.

À cause de volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans l´établissement des documents, Insécurité des informations et Possibilité d’erreur dans le remplissage des différents documents et registres. Ce pour c’est raison les écoles primaires ont besoin d’intégrer certain type de système d’informations pour gérer ses informations.

Le développement d’un application web présente certaines particularités, au niveau technique et ergonomique. Cette spécificité nous oblige d’utilise comme UML (Unified Modeling Language)

Le but essentiel de la gestion des étudiants est le suivi et l’historique, et l’intérêt de l’informatisation est :

– d’avoir une base de données pour les élèves, pour les enseignants et les autres intermédiaires.

– de gérer les emplois de temps et le calendrier

– de contrôler les payements des frais d’inscriptions.

L’objectif de notre projet présente dans ce rapport est la conception d’une application web simple de gestion des étudiants ainsi que les enseignants et tous les intermédiaires qui sont inter reliées avec l’application de gestion d’une école primaire. Et pour la réalisation de cette tâche, notre choix s’est porté sur UML (Unified Modeling Language) comme un langage de modélisation.

Pour programmer une application, il ne convient pas de se lancer tête baissée dans l’écriture du code : il faut d’abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et étapes de la réalisation. C’est cette démarche antérieure à l’écriture que l’on appelle modélisation : son produit est un modèle.

L’application va lui donner la possibilité de stocker les donnes de manière léger et facile et aussi va réduire le temps d’exécution d’une tache. Ce rapport est structuré en quatre chapitres dont le

* Le premier "Analyse et spécification des besoins “est consacré à la méthodologie et le formalisme qu’on va adopter pour la conception et la réalisation de la solution ainsi qu’à la spécification des besoins.
* Le deuxième présentera "Architecture et Conception de la Solution “approfondira la conception, cette conception sera effectuée en utilisant le langage de modélisation UML abordant les comportements du système.

# Chapitre 1:

**Contexte générale**

1. **Introduction**

Ce chapitre de spécification des besoins est obligatoire pour la conception et la réalisation de l’application pour comprendre dès le début les nécessités, la nature des utilisateurs cible ainsi que les scénarios d’utilisation du système à développer. Nous exposons dans ce dernier la méthodologie de travail à suivre.

1. **La méthodologie et le formalisme adopté**

Pour réussir un projet, il est essentiel de choisir la méthodologie adéquate. Pour le fait, le choix d’un modèle de cycle de vie pour notre projet constitue une étape essentielle pour assurer le bon fonctionnement et le succès de notre projet par la suite. Pour réaliser cette étape notre système sera mené par la méthode d’analyse et de conception UML.

1. **Le formalisme adopté**

UML est un langage de modélisation graphique orienté objet Il fournit une description complète, évolutive, publique d’un futur système. Parmi les nombreux avantages de langage UML est d'être le standard en termes de modélisation objet universellement reconnu, utilisé par AGL. Ce langage permet de représenter avec un moyen simple à manipuler plusieurs projections d’une même représentation grâce à ces différents diagrammes, ces projections représentent le système d’un point de vue dynamique et statique. Les vues dynamiques montrent le comportement du système, les interactions des objets et leur fonctionnement par contre les vues statiques représentent les squelettes et les classes intervenant dans le système.

Chaque vue offerte par UML est constituée par de multiples diagrammes, chacun étant consacré à la présentation d’un concept bien particulier pour un système logiciel. Pour bien comprendre notre projet, nous exposons quelques diagrammes que nous avons trouvés totalement utiles et suffisants Il existe plusieurs diagrammes tels que le diagramme de cas d’utilisation, le diagramme de classe, le diagramme de séquence ainsi que le diagramme de flux.

* Diagramme de cas d’utilisation: représente les fonctionnalités nécessaires aux utilisateurs.
* Diagramme de séquence: offre une représentation temporelle et comportementale permettant de décrire les différents scénarios d’utilisation du système.
* Diagramme de classes: ce diagramme permet une observation statique du système analysé, en représentant les classes et les différentes relations entre eux.

1. **Le Specification des besoins**

L’application considérée doit satisfaire principalement des besoins fonctionnels qui seront effectués par le système et des besoins non fonctionnels qui améliorent la qualité logicielle du système en précisant notre acteur.

1. **Conclusion**

Dans ce chapitre, nous avons abordé les objectifs de notre projet en critiquant l’existant, et nous avons fini par le travail demandé.

# Chapitre 2:

**Analyse et spécification des besoins**

1. **Introduction**

Dans ce deuxième chapitre, nous présentons l’étape de conception qui est la phase la plus importante, elle a pour but la réutilisation des composants engendrés. Au niveau de ce présent chapitre, nous révélons notre solution conceptuelle. Commençant par les patrons de conception adoptés, par la suite nous détaillons notre conception à travers le diagramme de classe et nous introduisons les diagrammes de séquences.

1. **L’identification des acteurs**

* **Administrateur**
* **L’authentification** (id & pwd)
* **Gestion du compte** (modifier son compte)
* **Gestion des gestionnaire** (ajout, suppression, modification)
* **Gestionnaire finnancière**
* **La gestion des paiements des frais de scolarité** (choisir l’élève qui va payer et introduire le montant et d’autre information par exemple le type de paiement soit chaque ou espèce.)
* **La gestion des frais continue** (ajouter)
* **La gestion les frais de transports (**choisir l’élève qui va payer le frais de transport pour monter au bus scolaire.)
* **Gestionnaire de Ressource humaine (RH)**
* **La gestion de dossier des élevés** (nom, prénom, Adresse, date naissance, on va donner la possibilité de modifier les informations d’un élevé ou bien supprimer totalement ses coordonnes)
* **La gestion professeurs** (ajouter des professeurs, lui spécifier la matière qui va enseigner)
* **Gestionnaire de plannfication**
* **La gestion des matière** (ajouter les matières qui vont être enseigner dans chaque niveau, supprimer une matière)
* **La gestion d’absences** (la possibilité de marquer 1’absence d’un élève)
* **La gestion des notes** (saisir la note d’un élève dans une matière et aussi calculer sa moyenne)
* **La gestion d’emplois du temps** (définir les séances du cours pour réaliser cela on va vérifier la disponibilité des salles dans une séance donnée aussi on va faire une vérification de la disponibilité des professeurs)
* **La gestion des niveaux** (ajouter les niveaux qui vont être existe dans cet établissement, Consulter la liste des élèves dans un niveau, Consulter la liste des matières enseignées dans un niveau.,)

1. **Les Besoins fonctionnels**

Les besoins fonctionnels permettent de caractériser les fonctions de service attendues. Afin de satisfaire les besoins de l’acteur principale. Notre application doit fournir les fonctionnalités suivantes aux utilisateurs:

**Table 1 Identification des besoins fonctionnels**

|  |  |
| --- | --- |
| **Acteurs** | **Description fonctionnelle** |
| **Gestionnaire RH** | * Gérer les professeurs * Gérer les élevés * Gérer les personnels |
| **Gestionnaire financière** | * Gérer les frais scolaires |
| **Gestionnaire plannificateur** | * Gérer les emplois * Gérer les notes * Gérer les matières * Gérer d’absences * Gérer les niveaux |
| **Adminstrateur** | * Gérer les gestionnaires * Gérer du compte |

1. **Les Besoins non-fonctionnels**

Les besoins non fonctionnels sont les propriétés du système et les contraintes associées à l’implémentation et à l’environnement à respecter pour la réalisation et qui se présente comme suit:

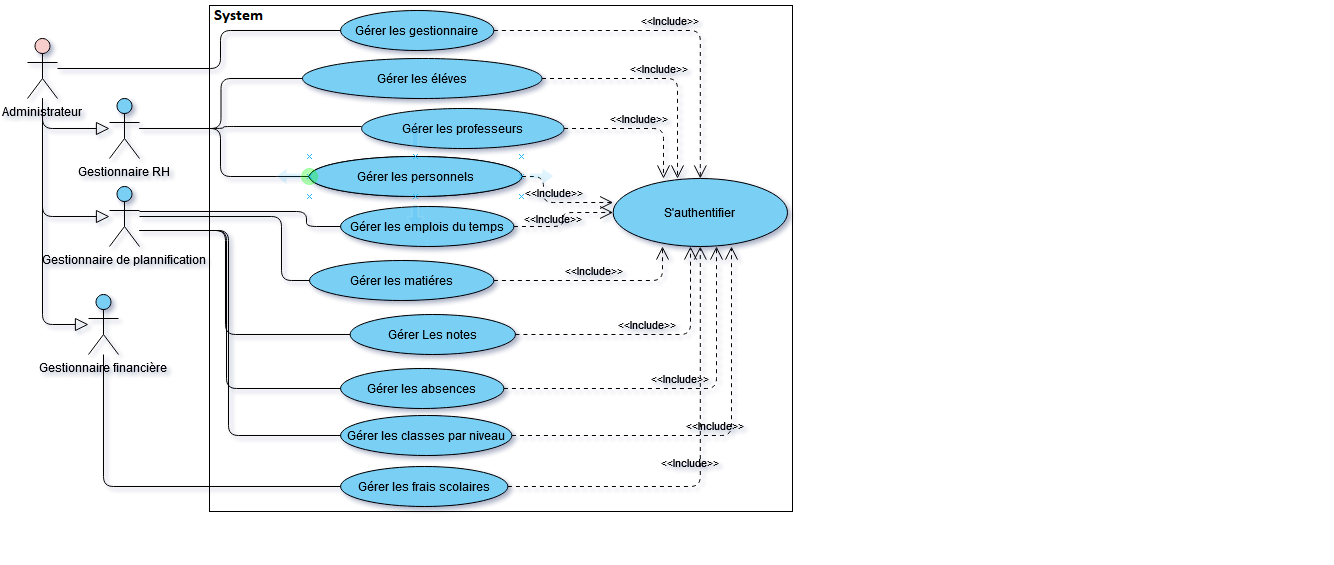
Un besoin non-fonctionnel est une condition qui spécifie les critères qui peuvent être utilisés pour juger du fonctionnement d’un système, plutôt que des comportements de celui-ci. Cela devrait être mis en contraste avec les exigences fonctionnelles qui définissent le comportement spécifique ou les fonctions désirés. Afin d’assurer un bon fonctionnement de l’application et de garantir la satisfaction de l’utilisateur, des contraintes doivent être prises en compte tout au long du développement du projet :

* Ergonomie et convivialité : L’application doit fournir une interface simple et élégante pour l’utilisateur afin de faciliter l’exploitation des services de l’application.
* Portabilité, maintenance et évolution : L’application doit être multiplateforme c’est à-dire fonctionnelle sur n’importe quel système d’exploitation. Le code doit être bien lisible, compréhensible et modulaire pour garantir la souplesse, l’évolution et la maintenance de la solution et ainsi répondre aux changements du marché.
* Rapidité et robustesse : L’application doit assurer la rapidité de réponse et la fiabilité du résultat procuré : les candidats doivent être capable de postuler facilement aux offres et l’administrateur doit être capable de sélectionner facilement, efficacement et d’une manière fiable les candidatures les plus pertinentes.
* Sécurité : L’application doit tenir compte de confidentialité des données des utilisateurs - Autonomie du système : le système s’exécute et fonctionne entièrement sans avoir recours à d’autres applications.
* Gestion des erreurs : les erreurs doivent être signalées par des messages d’erreur explicites.

1. **Le diagramme de cas d’utilisation**
2. **Diagramme de cas d’utilisation**

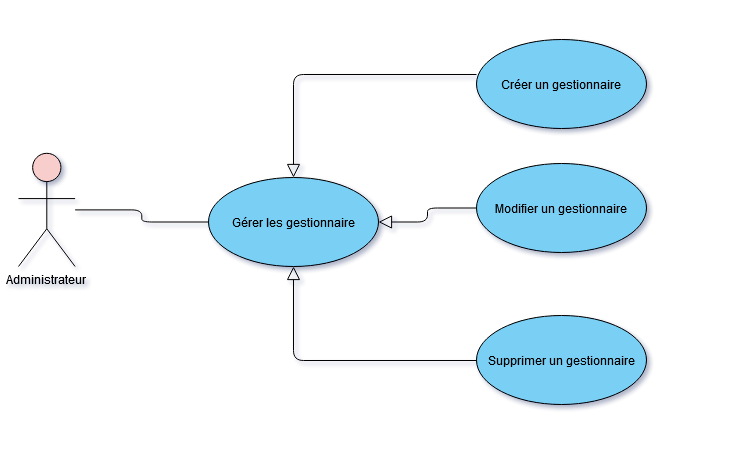
Le diagramme de cas d’utilisation est un diagramme UML qui est utilisé pour le but de donner une vue globale du comportement fonctionnel d’un système logiciel. Il permet de synthétiser les fonctionnalités offrir par le système. Les cas d’utilisation permettent d’expliquer d’une façon normalisée les liens entre système étudiée les acteurs. Ils sont spécifiés et modélisés par le diagramme de cas d’utilisation, qui représente l’interaction entre l’utilisateur et le système afin de produire le résultat. Dans cette partie, nous allons présenter et expliquer le diagramme de cas d’utilisation global de notre application.

Cette figure illustre le diagramme de cas d’utilisation global de notre application :

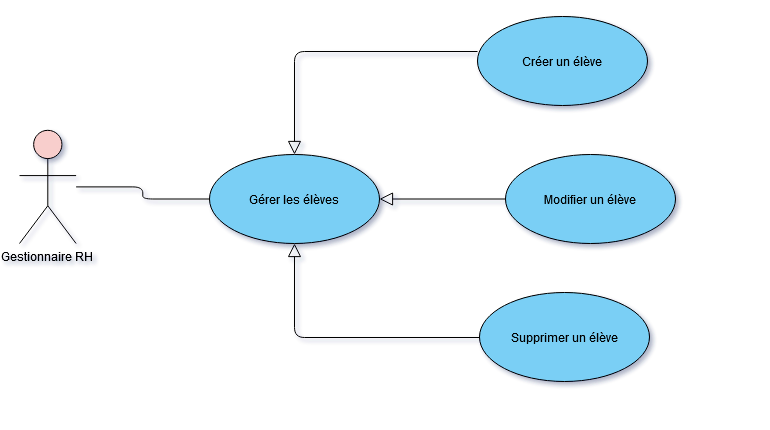


**Figure 1Diagramme des cas d’utilisation principal**

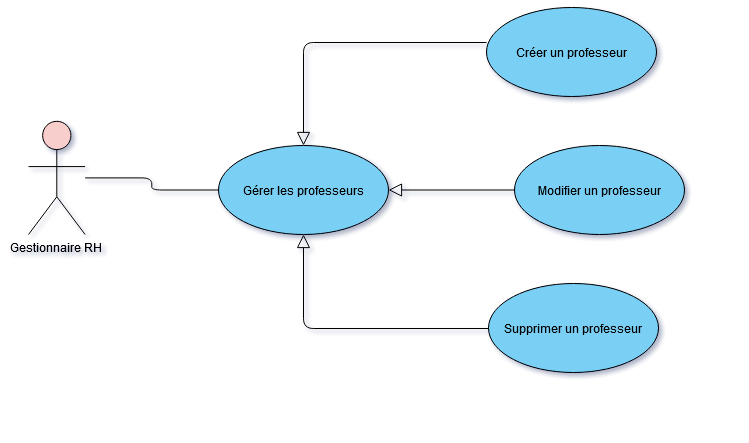
1. **Diagramme du cas d'utilisation raffinée**



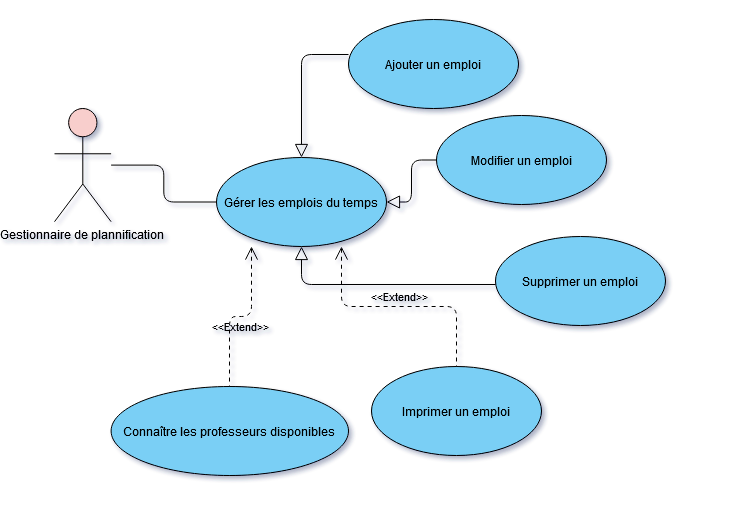
**Figure 2 Diagramme des cas d’utilisation « Gérer les gestionnaires »**



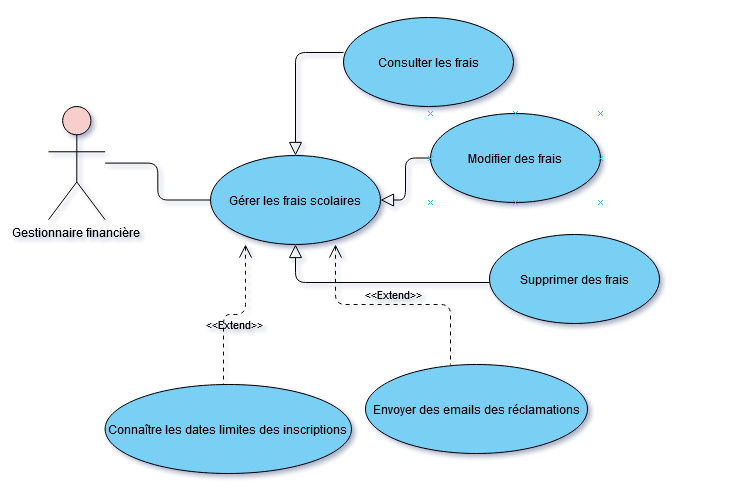
**Figure 3 Diagramme des cas d’utilisation « Gérer les élèves »**



**Figure 4 Diagramme des cas d’utilisation « Gérer les professeur »**



**Figure 5 Diagramme des cas d’utilisation « Gérer les emplois du temps »**



**Figure 6 Diagramme des cas d’utilisation « Gérer les frais scolaires »**

1. **La description textuelle de cas d’utilisation**

**Table 2 Description textuelle de cas d’utilisation s’authentifier**

|  |  |
| --- | --- |
| **Titre** | **S’authentifier** |
| **Résumé** | Authentification pour accéder à l’espace de travail |
| **Acteur** | Agent |
| **Pré-condition** | Utilisateur doit être enregistré dans la base de données |
| **Post-condition** | En cas de réussite d’authentification, l’utilisateur est redirigé vers la page d’accueil. |
| **Enchainements :Scénario nominal** | 1. L’utilisateur accède à l’application. 2. L’utilisateur doit saisir son login et son mot de passe. 3. Le système vérifie les coordonnées saisies. 4. Le système redirige l’utilisateur vers la page d’accueil. |
| **Scénario alternatif** | 1.A Le système affiche un message d’erreur si le login ou le mot de passe est incorrect |
| **Exception** | E1:L’utilisateur quitte l’application |

**Table 3 Description textuelle de cas d’utilisation gérer les élèves**

|  |  |
| --- | --- |
| **Titre** | **Gérer les élèves** |
| **Résumé** | Cas d’utilisation permet à l’utilisateur de modifier l’un de l’information personnel d’élèves sélectionnée. |
| **Acteur** | Gestionnaire RH |
| **Pré-condition** | Choisir l’élève qui doit modifier |
| **Post-condition** | Modification de l'étape générée.  Les modifications sont enregistrées dans la base de données |
| **Enchainements** | Cas 1 : Ajouter élève  1. L’utilisateur choisit d’ajouter un nouvel élève.  2. le système affiche un formulaire à remplir  3. l’acteur rempli et valide les informations de formulaire.  4. le système ajoute les informations dans la base.  5. le système analyse la liste des élèves est l’affiche.  Cas 2 Modifier élève   1. L’utilisateur choisir l’élève à modifier 2. Le système lui permet de le modifier 3. L’utilisateur modifie l’un de l’information de l’élève 4. Le système enregistre la modification   Cas3 : Supprimer élève  1. L’utilisateur choisit à supprimer un élève  2. il sélection la ligne qu’il veut le supprimer.  3. le système demande la confirmation de suppression, en affichant un message.  4. l’acteurs confirme ou annule la suppression.  5. le système supprime l’élève de la base  6. le système actualise la liste et l’affiché |
| **Scénario alternatif** | 1A. L’utilisateur sort avant d’avoir enregistré les modifications  4A. les modifications déjà existe. |
| **Exception** | E1 : Les champs obligatoires sont vides : Le système indique que les champs sont obligatoires en affichant un message. |

**Table 4 Description textuelle de cas d’utilisation frais d’inscription**

|  |  |
| --- | --- |
| **Titre** | **Gérer les frais scolaires** |
| **Résumé** | Cas d’utilisation permet à l’utilisateur de suivre le paiement de frais d’inscription |
| **Acteur** | Gestionnaire financier |
| **Pré-condition** | Utilisateur authentifié |
| **Post-condition** | Les modifications sont enregistrées dans la base de données |
| **Enchainements** | 1. L’utilisateur demande la page « Suivre les frais d’inscription ». 2. Le système lui affiche la page demandée. 3. L’utilisateur recherche l’élève à suivre à travers ses critères de recherche 4. L’utilisateur clique sur le bouton « rechercher » 5. Le système affiche l’élève a cherché 6. L'utilisateur peut modifier le statut soit il active, désactive ou bien envoyer un alerte à l’élève. 7. Le système enregistre les modifications |
| **Scénario alternatif** | 1A. L’utilisateur sort avant d’avoir enregistré les modifications |

**Table 5 Description textuelle de cas d’utilisation imprimer emploi**

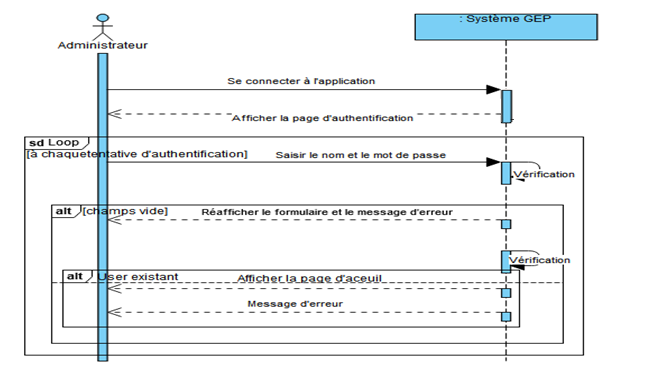
|  |  |
| --- | --- |
| **Titre** | **Imprimer un emploi** |
| **Résumé** | Cas d’utilisation permet à imprimer les emplois du temps |
| **Acteur** | Gestionnaire de planification |
| **Pré-condition** | Choisir l’emploi à imprimer |
| **Post-condition** | Les modifications sont enregistrées dans la base de données |
| **Enchainements** | 1. L’utilisateur demande la page « emploi du temps ». 2. Le système lui affiche la page demandée. 3. L’utilisateur recherche l’emploi à imprimer à travers ses critères de recherche 4. L’utilisateur clique sur le bouton « imprimer » 5. Le système imprime l’emploi selectionner |
| **Scénario alternatif** | 1A. L’utilisateur sort avant de clique sur le bouton imprimer  2A. Le système n’enregistre pas l’action. |
| **Exception** |  |

**Table 6 Description textuelle de cas d’utilisation consulter emploi**

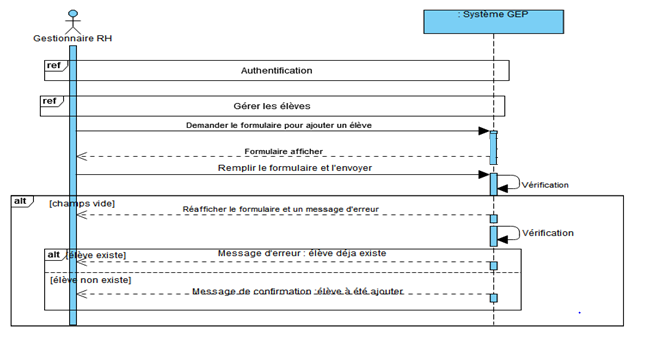
|  |  |
| --- | --- |
| **Titre** | **Consulter un emploi** |
| **Résumé** | Cas d’utilisation permet à l’utilisateur d’ajouter un emploi |
| **Acteur** | Gestionnaire de planification |
| **Pré-condition** | Utilisateur authentifié |
| **Post-condition** | Les enregistrements sont enregistrées dans la base de données |
| **Enchainements** | 1. L’utilisateur demande la page «emploi du temps». 2. Le système lui affiche la page demandée. 3. L’utilisateur cliquer sur le bouton « ajouter emploi » 4. Le système importe l’emploi 5. L'utilisateur enregistre l’action par cliquer sur le bouton « enregistrer » 6. Le système enregistre les modifications |
| **Scénario alternatif** | 1A. L’utilisateur ne sélectionne pas l’emploi qui va l’ajouter  2A. Le système n’enregistre pas les modifications. |
| **Exception** |  |

1. **Diagramme de séquence système**

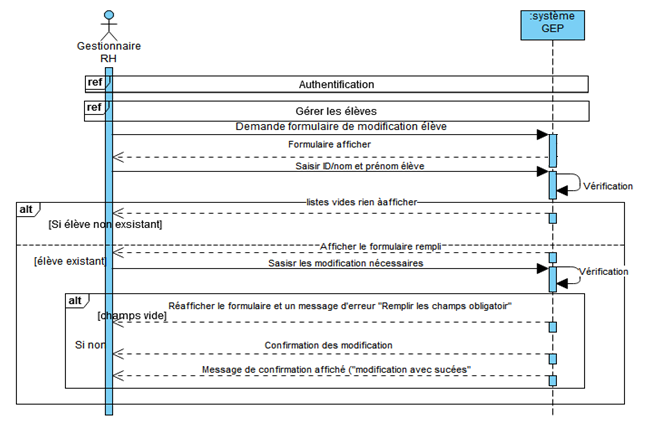
Pour mieux concrétiser les interactions entre les acteurs du système vis-à-vis de l’application et la base de données, nous sommes amenés à traduire nos scénarios en diagrammes de séquence. Les diagrammes de séquence servent à illustrer les cas d’utilisation. Ils permettent de représenter des collaborations entre les objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois des messages.



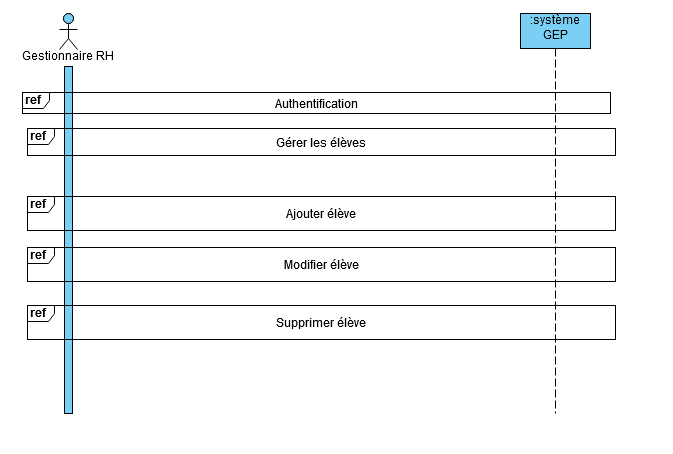
**Figure 7 Diagramme de séquence authentification**



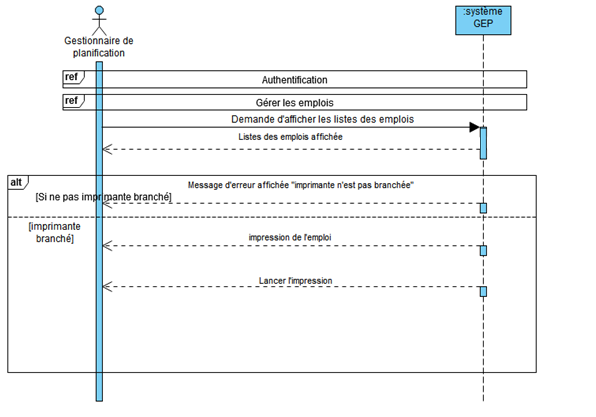
**Figure 8Diagramme de séquence Ajouter élève**



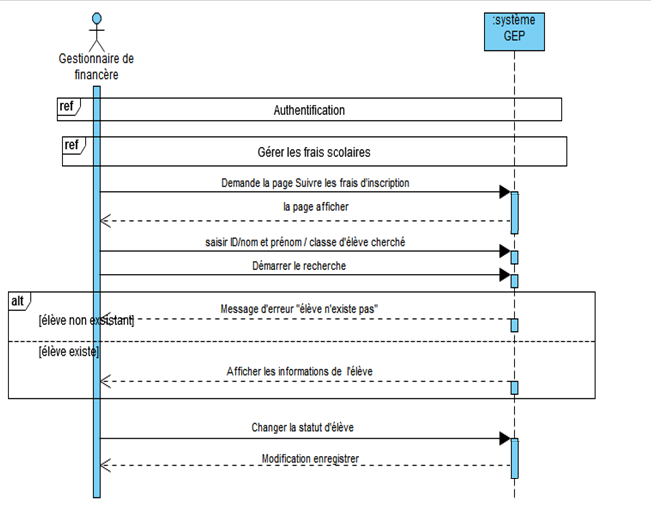
**Figure 9Diagramme de séquence modifier élève**



**Figure 10 Diagramme de séquence gérer élève**



**Figure 11 Diagramme de séquence imprimer emploi**



**Figure 12Diagramme de séquence gérer les frais scolaires**

1. **Conclusion**

Dans ce deuxième chapitre nous avons présente les fonctionnalités principales que nous avons illustrées à travers les différents diagrammes de cas d’utilisation.

# Chapitre 3:

**Conception de la Solution**

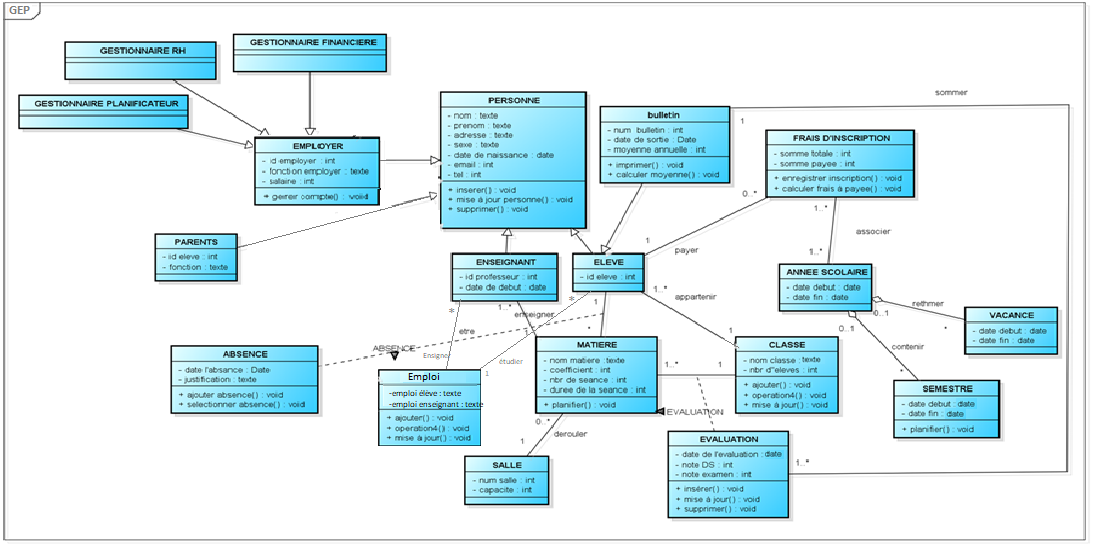
1. **Introduction**

Dans ce chapitre, on va présenter le diagramme de classes,les diagrammes de séquence objet et le diagramme d’activité

1. **Diagramme de classes**

Le diagramme de classe est une description statique du système focalisé sur le concept de classe et d’association. Une classe représente un ensemble d’objets qui possèdent des propriétés similaires et des comportements communs décrivant en terme d‘attributs et d’opérations. P a g e 64 | 92 Une association consiste à présenter les liens entre les instances de classe. Durant cette section, nous allons présenter les diagrammes de classes entités à notre application.

Cette figure présente le diagramme de classe :

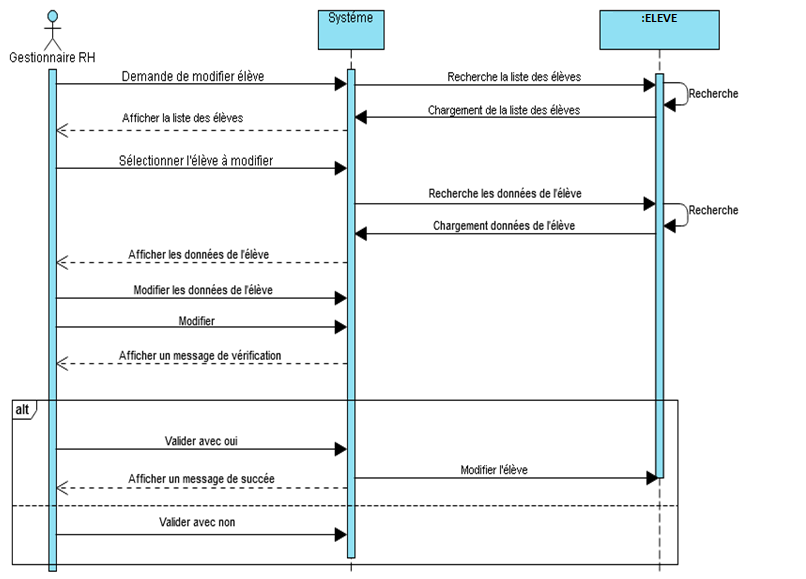


**Figure 13 Diagramme de classe**

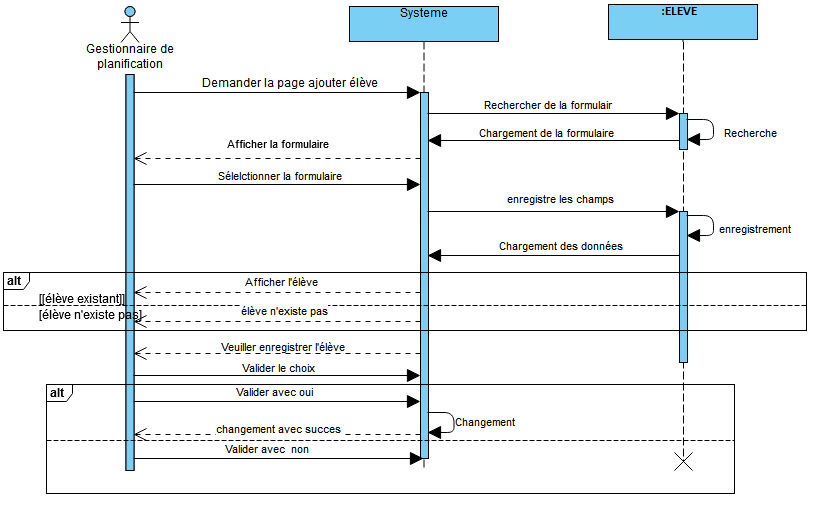
1. **Diagramme de séquence objet**

Le diagramme de séquence décrit comment les objets (et les instances) échange des messages dans un ordre particulier. Les objets sont les éléments de base des diagrammes UML. Selon le type de diagramme, ils représentent certaines caractéristiques d’un élément du système. Dans les interactions, les objets sont des lignes de vie.

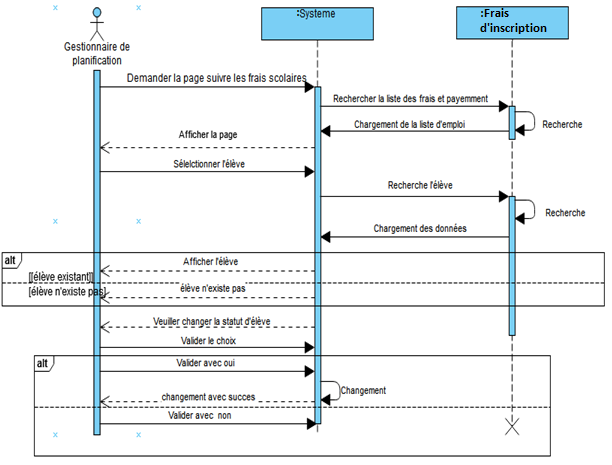
Ces dissous les diagrammes de séquences objet



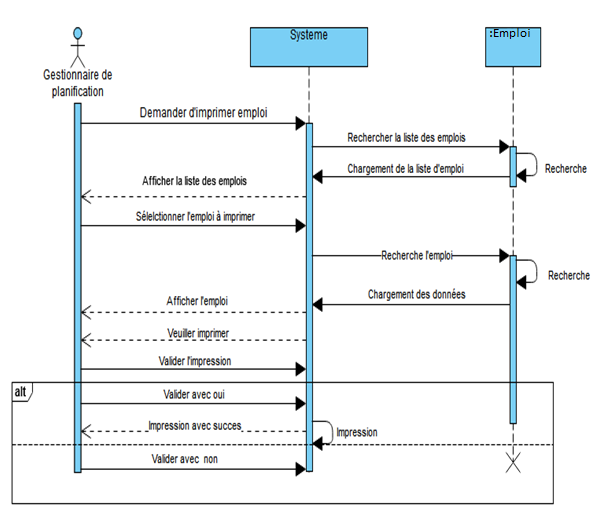
**Figure 14 Diagramme de séquence objet modifier élève**



**Figure 15 Diagramme de séquence objet ajouter élève**



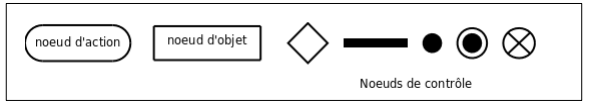
**Figure 16 Diagramme de séquence frais scolaire**



**Figure 17 Diagramme de séquence objet imprimer emploi**

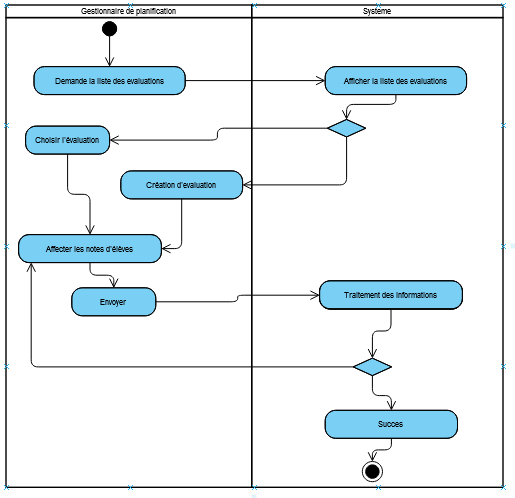
1. **Diagramme d’activité**

Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation. De plus, leur représentation sous forme d'organigrammes les rend facilement intelligibles et beaucoup plus accessibles que les diagrammes d'états-transitions. On parle généralement dans ce cas de modélisation de workflow. On se concentre ici sur les activités telles que les voient les acteurs qui collaborent avec le système dans le cadre d'un processus métier. • Nœuds d’activités : De la gauche vers la droite, on trouve : le nœud représentant une action, qui est une variété de nœud exécutable, un nœud objet, un nœud de décision ou de fusion, un nœud de bifurcation ou d'union, un nœud initial, un nœud final et un nœud final de flot.

****

**Figure 18 Représentation graphique des nœuds d'activité**

Cette figure présente le diagramme d’activité :

****

**Figure 19 Diagramme d'activité gérer les notes**

1. **Conclusion**

Dans ce chapitre, nous avons conçu toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (ensemble de diagrammes) prêt à être codé.

**Conclusion générale**

L’objectif de notre projet était de concevoir et implémenter une application de gestion d’école primaire.

Le point de départ de la réalisation de ce projet était une récolte d’informations nécessaires pour dresser un état de l’existant, présenter un aperçu sur la problématique ainsi que l’architecture utilisée au sein des réseaux des établissements.

Par la suite, nous nous sommes intéressés à l’analyse et la spécification des besoins qui nous a permis de distinguer les différents acteurs interagissant avec l’application visée.

L’objectif de la partie suivante était la conception détaillée, dans laquelle nous avons fixé la structure globale de l’application.

Le dernier volet de notre projet était la partie réalisation qui a été consacrée à la présentation des outils du travail et les interfaces les plus significatives de notre application.

L’apport de ce travail a été d’une importance très considérable. En effet, il nous a permis : de suivre une méthodologie de travail bien étudiée, d’approfondir nos connaissances dans le monde de développement des applications.