

REPUBLIQUE TOGOLAISE

Travail – Liberté – Patrie

MINISTERE DE LA PLANIFICATION
DU DEVELOPPEMENT ET DE LA
COOPERATION



Institut Africain d'Informatique
Représentation du TOGO
(IAI-TOGO)

07 BP 12456 Lomé 07

Tel : (+228) 22 20 47 00

E-mail : iaitogo@gmail.com

Site-Web : www.iai-togo.com



06 B.P. 62301 Lomé - Togo

Tel : (+228) 92026860 / 92286127

Adresse : Agoè-Nyivé à Coté de CEG Cacavéli Route de la CEET.

RAPPORT DE STAGE PRATIQUE EN ENTREPRISE

Type de stage : Programmation

GESTION DU RECOUVREMENT DES FRAIS SCOLAIRES CAS DES ETABLISSEMENTS DU TOGO

Période : Du 06 Juin au 05 Août 2022

Rédigé par :
BAGNA Tarik
Etudiant en deuxième année Tronc Commun
Année Universitaire : 2021 – 2022

Superviseur :

M. TETTEH Amen

Enseignant à l'IAI-Togo

Maître de stage :

M. BILANTE Addo

Directeur de Odda
Technology

REMERCIEMENTS

Nous remercions Dieu le tout puissant de nous avoir donné la santé et la détermination tout au long de ce stage. Nous tenons à remercier et à exprimer notre profonde gratitude à toutes ces personnes qui ont contribué de loin ou de proche à l'accomplissement de ce travail. Nos remerciements vont également :

- A M. **BILANTE Addo**, Directeur de Odda Technology et mon maître de stage, pour son soutien et ses conseils ;
- A M. **AGBETI Kodjo, Directeur Général du CENETI et Représentant National de l'IAI-TOGO, pour ses conseils et l'intérêt incontestable qu'il porte à tous les étudiants ;**
- A M. **AMEYIKPO Kossi Nicolas**, Directeur des Affaires Académiques et de la Scolarité de l'IAI-TOGO ;
- A M. **BILANTE David**, Adjoint du directeur de Odda Technology, pour toute sa disponibilité et ses apports dans la réalisation de ce stage ;
- A M. **TETTEH K. Amen**, Enseignant à l'IAI-TOGO, pour sa disponibilité, ses précieux conseils et les suivis qu'il m'a prodigué pendant toute la durée de mon stage, en tant que superviseur ;
- Aux membres de l'administration de l'IAI-TOGO, pour l'encadrement rigoureux et en même temps bénéfique ;
- A mes parents, frères et sœurs pour leur amour, leur soutien, leurs encouragements et surtout leur présence.

Merci à tous.

SOMMAIRE

REMERCIEMENTS	I
SOMMAIRE	II
LISTE DES FIGURES.....	IV
LISTE DES TABLEAUX	VI
INTRODUCTION	1
PARTIE 1 : CAHIER DES CHARGES	3
1 Cahier des charges	3
1.1 Présentation de l’Institut Africain d'Informatique, Représentation du TOGO.....	3
1.2 Présentation de Odda Technology	4
1.3 Présentation du sujet	4
1.4 Problématique du sujet	5
1.5 Intérêt du sujet.....	5
1.5.1 Objectifs.....	5
1.5.2 Résultats	6
PARTIE 2 : PRE-PROGRAMMATION	8
2 Préprogrammation.....	8
2.1 Etude de l’existant.....	8
2.2 Critique de l’existant	8
2.3 Propositions de solutions	8
2.3.1 Se procurer un abonnement sur le site myscol.com.....	9
2.3.2 Réaliser une application web personnalisée	9
2.4 Evaluation financière des solutions.....	10
2.4.1 Solution : myscol	10
2.5 Choix de la solution	10
2.6 Planning prévisionnel de réalisation	10
2.6.1 Tableau résumant le panning prévisonnel de réalisation	10
2.6.2 Rédaction du rapport de stage	11
2.7 Présentation de la méthode d’analyse.....	11
2.7.1 Le langage UML	11
2.7.2 Le processus 2TUP	15
2.8 Présentation de l’outil de modélisation : Power AMC	16
2.9 Etude détaillée de la solution	17

2.9.1	Les acteurs.....	17
2.9.2	Capture des besoins fonctionnels	18
PARTIE 3 : REALISATION ET MISE EN ŒUVRE.....		32
3	Réalisation et mise en œuvre	32
3.1	Matériels et logiciels utilisés	32
3.1.1	Matériels.....	32
3.1.2	Langages et logiciels	33
3.2	Sécurité de l'application.....	43
3.2.1	Authentification.....	43
3.2.2	Contrôle d'accès	43
3.2.3	L'intégrité des données	44
3.2.4	Confidentialité	44
3.2.5	Non répudiation des données	44
3.3	Evaluation Financière de la solution	44
3.3.1	Coût matériel de la solution	44
3.3.2	Développement de la solution	45
3.3.3	Coût total de la solution	45
3.4	Présentation de l'application	46
3.4.1	Présentation (Squelette de l'application : enchainement des menus).....	46
3.4.2	Architecture de l'application	47
3.4.3	Mise en place de la base de données.....	48
3.4.4	Plan de navigation	51
3.4.5	Quelques masques de saisi.....	52
3.4.6	Quelques états et statistiques.....	54
CONCLUSION		56
BIBLIOGRAPHIE INDICATIVE		VI
WEBOGRAPHIE INDICATIVE.....		VII
DOCUMENTS ANNEXES		VIII
TABLES DES MATIERES		IX

LISTE DES FIGURES

FIGURE 1 : LOCALISATION DE L'IAI-TOGO.....	3
FIGURE 2 : LOCALISATION ODDA TECHNOLOGY.	4
FIGURE 3 : LOGO DE UML.	11
FIGURE 4 : ORGANIGRAMME DES DIAGRAMMES UML.	13
FIGURE 5 : SCHEMA DU CYCLE DE VIE EN Y DU PROCESSUS 2TUP.	16
FIGURE 6 : LOGO DE POWER AMC.	17
FIGURE 7 : DIAGRAMME DE CONTEXTE.	18
FIGURE 8 : DIAGRAMME DE CAS D'UTILISATION VUE DE LA SECRETAIRE.....	20
FIGURE 9 : DIAGRAMME DE CAS D'UTILISATION VUE DU DIRECTEUR.	21
FIGURE 10 : DIAGRAMME DE CAS D'UTILISATION VUE GLOBALE.	22
FIGURE 11 : DIAGRAMME DE CLASSES.....	26
FIGURE 12 : DIAGRAMME DE SEQUENCE : « S'AUTHTENTIFIER ».....	27
FIGURE 13 : DIAGRAMME DE SEQUENCE : « ENREGISTRER UN PAYEMENT ».....	28
FIGURE 14 : DIAGRAMME D'ACTIVITE : « S'AUTHTENTIFIER ».....	29
FIGURE 15 : DIAGRAMME D'ACTIVITE : « ENREGISTRER UN PAYEMENT ».....	30
FIGURE 16 : LOCALISATION DE L'IAI-TOGO.....	32
FIGURE 17 : LOGO DE HTML5.....	33
FIGURE 18 : LOGO DE CSS3.....	34
FIGURE 19 : LOGO DE SASS.	34
FIGURE 20 : LOGO DE JAVASCRIPT.	35
FIGURE 21 : LOGO DE TYPESCRIPT.	36
FIGURE 22 : LOGO DE JASPERSTUDIO.....	37
FIGURE 23 : LOGO DE JAVA.....	37
FIGURE 24 : LOGO DE INTELLIJ IDEA.....	38
FIGURE 25 : LOGO DE WAMPSEVER.	39
FIGURE 26 : LOGO DE PHPMYADMIN.	39
FIGURE 27 : LOGO DE BOOTSTRAP.....	40
FIGURE 28 : LOGO DE JQUERY.	41
FIGURE 29 : LOGO DE SPRING.....	41
FIGURE 30 : LOGO DE ANGULAR.....	42
FIGURE 31 : LOGO DE SPRING SECURITY.....	43
FIGURE 32 : GENERATION DE LA BASE DE DONNEES ET DES TABLES.	48
FIGURE 33 : CREATION DE LA TABLE CLASSE.....	49
FIGURE 34 : CREATION DE LA TABLE INSCRIPTION.....	49
FIGURE 35 : CREATION DE LA TABLE PAYEMENT.....	50

FIGURE 36 : PLAN DE NAVIGATION.....	51
FIGURE 37 : ENREGISTREMENT D'UNE INSCRIPTION.	52
FIGURE 38 : ENREGISTREMENT D'UNE CLASSE.	52
FIGURE 39 : ENREGISTREMENT D'UNE ANNEE.....	53
FIGURE 40 : ENREGISTREMENT D'UN PAYEMENT.....	53
FIGURE 41 : LISTE DES ELEVES D'UNE CLASSE.....	54
FIGURE 42 : REÇU DE PAYEMENT.	55

LISTE DES TABLEAUX

TABLEAU 1 : COUT D'UTILISATION DE L'APPLICATION MYSCOL	10
TABLEAU 2 : PLANNING PREVISIONNEL DE REALISATION	10
TABLEAU 3 : RECAPITULATIF DES CAS D'UTILISATION	19
TABLEAU 4 : CAS D'UTILISATION « S'AUTHTENTIFIER »	23
TABLEAU 5 : CAS D'UTILISATION « ENREGISTRER UN PAYEMENT »	24
TABLEAU 6 : CAS D'UTILISATION « S'INSCRIRE »	25
TABLEAU 7 : COUT MATERIEL DE LA SOLUTION	44
TABLEAU 8 : COUT DE DEVELOPPEMENT DE LA SOLUTION	45
TABLEAU 9 : COUT TOTAL	45

INTRODUCTION

Le monde d'aujourd'hui fait face à une grande évolution technologique et ceci à cause du besoin qui ne cesse d'accroître. D'une part nous avons les grandes structures qui cherchent à basculer progressivement du système traditionnel, donc manuel, vers le système informatique et d'autre part, les petites entreprises naissantes qui aspirent également à informatiser leurs activités. De ce fait, l'informatique trouve toute sa place et son importance dans notre société non seulement en ce qui concerne l'accès, le traitement, la diffusion des données mais aussi la gestion facile des activités de la population.

C'est dans le souci de rendre accessible l'informatique et tous ses services en Afrique que plusieurs écoles supérieures et écoles de formation professionnelle spécialisés dans ce domaine furent créées. Parmi elles, nous pouvons citer l'Institut Africain d'Informatique (IAI) qui depuis sa création en 1971, forme des ingénieurs informatiques. En vue de favoriser l'acquisition des connaissances pratiques pour la suite de leur cycle, l'IAI intègre dans sa formation pour les étudiants en deuxième(2e) année, un stage en entreprise de deux (2) mois ; Cela leur facilitera aussi l'intégration professionnelle avec une formation complète et adaptée aux réalités locales. Nous avons justement, dans cet optique réalisé notre stage du 06 juin 2022 au 05 août 2022 sur le thème « **Gestion du recouvrement des frais scolaires** ». Ce projet nous est soumis dans le but de faciliter la gestion des paiements des frais scolaires dans une école.

Ce document constitue le rapport de stage effectué pendant deux (2) mois en entreprise à **Odda technology**. Il est subdivisé en trois (3) grandes parties : Le cahier de charge dans lequel nous développerons le thème étudié, le rapport de pré programmation dans lequel nous étudierons le domaine de la gestion des mobiliers et la partie réalisation et mise en œuvre qui contient les différentes étapes de la mise en œuvre de l'application.

PARTIE 1 :

CAHIER DES CHARGES

PARTIE 1 : CAHIER DES CHARGES

1 Cahier des charges

1.1 Présentation de l'Institut Africain d'Informatique, Représentation du TOGO

En janvier 1971, la conférence des chefs d'Etats et de gouvernements de l'OCAM adopta et signa la convention portant création de l'Institut Africain d'Informatique à Libreville au Gabon. Dès lors elle a des représentations dans 11 pays de l'Afrique. Ainsi, sa représentation au Togo fut créée en 2002 pour faciliter la formation par l'institut, d'ingénieurs de travaux informatiques dans le pays. L'IAI-Togo forme ses étudiants durant 3 années au cours desquelles ils acquièrent des connaissances en informatique. En troisième année ils choisissent une spécialité parmi celles proposées par l'institut (Génie logiciel, systèmes et réseaux et Multimédia, technologies web et infographie).

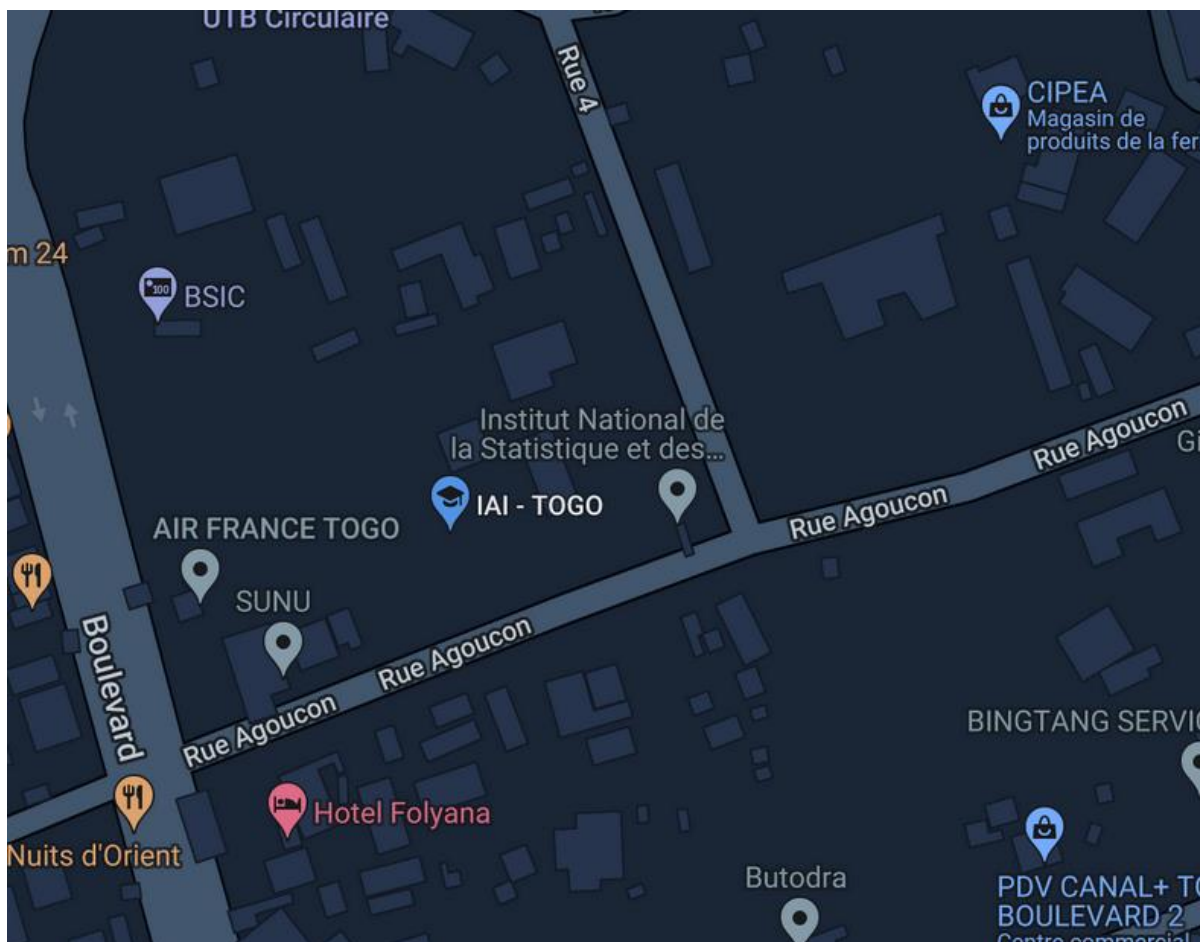


Figure 1 : Localisation de l'IAI-TOGO.

1.2 Présentation de Odda Technology

Entreprise spécialisée dans le développement d'applications web. Avec plus de 12 ans d'expérience, elle gère une application SM-WEB 2.0 de gestion des écoles avec certains modules à l'appui tels que : la gestion de la scolarité, la gestion des bulletins, le module de gestion de paye.

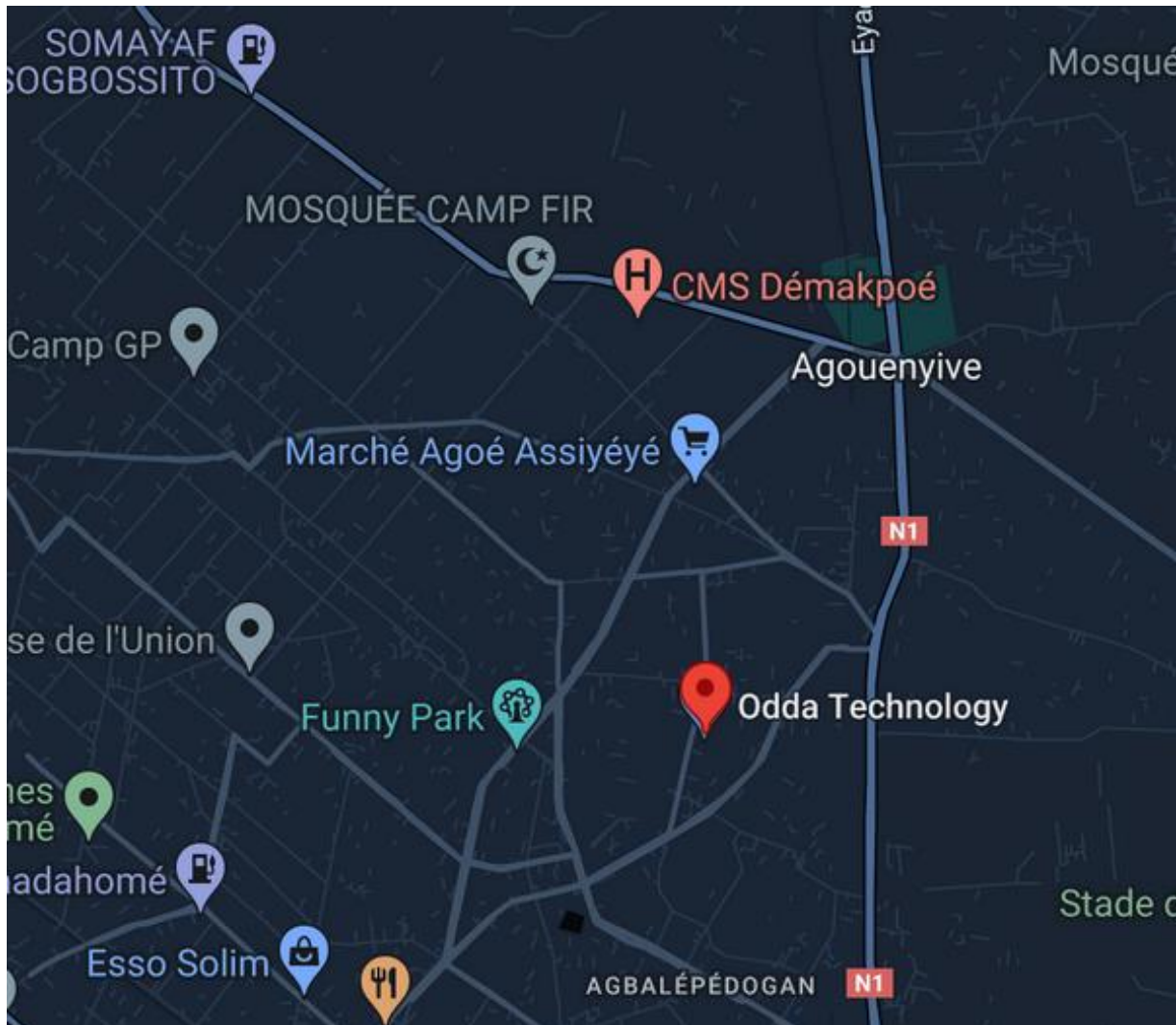


Figure 2 : Localisation Odda Technology.

1.3 Présentation du sujet

Notre thème fait l'objet d'un projet dont la réalisation vise à mettre à la disposition de l'entreprise, désireux d'améliorer les procédures de recouvrement de frais scolaire dans certaines écoles et faciliter les paiements.

L'informatique est de plus en plus présente dans notre quotidien avec l'avènement des Nouvelles Technologies de l'Information et de la Communication (NTIC). Cependant bon nombre de structures ne disposent pas d'outils pouvant leur permettre la gestion facile et efficace de leurs activités.

A cet effet, il s'avère important d'améliorer et de faciliter les tâches manuelles et répétitives par les différents utilisateurs. Ainsi pour y arriver, il nous a été demandé de travailler sur le projet intitulé : « **Gestion du recouvrement des frais scolaires** ».

1.4 Problématique du sujet

D'une part, dans le souci de gérer le recouvrement des frais scolaires, un constat se dresse : dans nombre d'établissements scolaires du Togo, le secrétariat à prime abord doit disposer de fiches pré-imprimées. Ensuite, lors du recouvrement les remplit manuellement. D'autre part nous constatons que le suivi des engagements, transactions est difficile et non sécurisé ainsi qu'une difficulté d'appréciation des statistiques. Au regard de ce qui précède, des questions se posent :

- Comment gérer un recouvrement facilement ?
- Comment gérer les engagements ?
- Comment permettre un bon suivi des transactions ?
- Comment sécuriser les transactions ?
- Comment permettre un bon rendu des statistiques ?

1.5 Intérêt du sujet

1.5.1 Objectifs

1.5.1.1 Objectifs généraux

Mettre en place un système permettant aux écoles une bonne gestion du recouvrement et le suivi puis la sécurité des transactions.

1.5.1.2 Objectifs spécifiques

Le système permettra de :

- Gérer les inscriptions ;
- Gérer les frais d'écologie ;
- Gérer les autres frais scolaires ;
- Elaborer des statistiques ;

- Gérer les engagements ;
- Sécuriser les données manipulées et traitées.

1.5.2 Résultats

L'application devra embarquer des fonctionnalités suivantes :

- La gestion des inscriptions ;
- La gestion des frais d'écolage ;
- La gestion des autres frais scolaires ;
- L'élaboration des statistiques ;
- La gestion des engagements ;
- La sécurité des données manipulées et traitées.

PARTIE 2 :

PRE-PROGRAMMATION

PARTIE 2 : PRE-PROGRAMMATION

2 Préprogrammation

Cette partie consistera à l'analyse du système et processus existants, ce qui permettra de mettre à nus ces faiblesses.

2.1 Etude de l'existant

Dans la perspective d'atteindre les objectifs fixés par le projet, les enquêtes ont été menées pour recueillir des informations et avoir une idée réelle de ce qui se fait. Ces enquêtes ont abouti aux constats suivants :

- Lors du paiement des frais scolaires, la secrétaire fait une recherche dans le registre des élèves inscrits puis récupère le montant du client et délivre à ce dernier un reçu qu'elle remplit de façon manuscrite ;
- Lors de l'inscription d'un élève, la secrétaire vérifie si ce dernier se trouve dans le registre sinon il l'ajoute dans le registre puis passe au rendu du reçu de paiement d'inscription en échange du montant des inscriptions ;
- Lors du paiement d'autres frais, elle vérifie la présence du dit élève dans l'établissement puis passe au rendu du reçu de paiement en échange du montant de paiement ;
- A la fin de chaque opération, elle remplit de manière manuscrite un registre d'opérations de paiement.

2.2 Critique de l'existant

Bien que l'existant pour le moment n'ait l'air d'être un souci, il présente plusieurs inconvénients. Le principal inconvénient se trouve dans la gestion manuelle des paiements. En effet, cette gestion est une perte de temps et pour l'école et pour le client (élève, parent, tuteur). Outre ce fait on peut aussi noter que les statistiques sont parfois faux dû à une erreur humaine. Aussi, chaque opération doit être renseignée dans le registre d'opérations. En définitive, le système actuel présente des failles et insuffisances, d'où la nécessité de proposer des solutions qui permettront de remédier à ces problèmes.

2.3 Propositions de solutions

Au vu des insuffisances du système actuel, deux solutions se présentent à nous :

2.3.1 Se procurer un abonnement sur le site myscol.com

Il s'agit d'une plateforme de gestion scolaire qui englobe plusieurs modules dont le recouvrement des frais scolaires. Il permet non seulement une bonne gestion des transactions mais aussi d'autres modules tels que : la vie scolaire, le partage des documents, le E-learning, etc.

2.3.1.1 Avantages

- L'application est déjà disponible donc prête à l'utilisation ;
- Elle est facile d'utilisation ;
- Elle permet une bonne gestion du recouvrement

2.3.1.2 Inconvénients

- Elle est très couteuse et pas forcément avantageux au cabinet qui a pour objectif des proposer des solutions à des prix intéressants à ces clients ;
- Elle n'est pas maintenable sur le long terme car elle ne peut qu'être utilisée et non acquise ;

2.3.2 Réaliser une application web personnalisée

2.3.2.1 Avantages

- L'application sera personnalisée, sur-mesure et respectera à la lettre les demandes du client ;
- L'application sera facilement maintenable et adaptable aux besoins fréquents du client ;
- Sur le long terme l'application pourra rapporter des bénéfices ;
- Nous aurons accès à toutes les données de statistiques dont on aura besoin pour améliorer la plateforme ;
- Nous avons l'opportunité de mettre en œuvre nos connaissances acquises à l'école à travers la mise en place de la solution, nous-mêmes.

2.3.2.2 Inconvénients

L'application n'existe pas encore donc il faudra du temps pour la réaliser ;

2.4 Evaluation financière des solutions

2.4.1 Solution : myscol

Couts d'utilisation : Les écoles sont facturées au nombre d'élèves voulu et au type d'abonnement voulu. Le tableau ci-dessous récapitule les offres disponibles (d'après le site <https://myscol.com/>).

Tableau 1 : Coût d'utilisation de l'application myscol

Type d'abonnement	Pack	Nombre d'élèves	Montant
Mensuel	SMALL	100	55 366,39 FCFA
	MEDIUM	500	158 189,69 FCFA
	LARGE	1 000	237 284,54 FCFA
Annuel	SMALL	100	332 198,35 FCFA
	MEDIUM	500	949 138,14 FCFA
	LARGE	1 000	1 423 707,21 FCFA

2.5 Choix de la solution

En comparant les deux solutions tout en pesant les avantages et les inconvénients de chacune par rapport à l'autre, le choix fut porté sur la deuxième solution (Réaliser un site web personnalisé). Ce choix a été fait non seulement dans un but pédagogique afin de gagner de l'expérience mais aussi car l'application élaborée sera adaptée aux besoins du client qui sont de gérer le paiement des frais scolaires. Elle sera aussi facilement maintenable.

2.6 Planning prévisionnel de réalisation

2.6.1 Tableau résumant le panning prévisonnel de réalisation

Tableau 2 : Planning prévisionnel de réalisation

Phases du projet	Date de Début	Date de Fin	Durées en jours
Phase d'initialisation	07Jun 2022	02 Juillet 2022	26 jours
Connaissance du thème	07 Juin 2022	25 juin 2022	20 jours

Enquête et collecte des informations	26 Juin 2022	07 Juillet 2022	14 jours
Prise de contact avec le superviseur	1 ^{er} Juillet 2022	01 Juillet 2022	1 jour
Phase de conception	08 Juillet 2022	15 Juillet 2022	7 jours
Maquette et design de l'application	08 Juillet 2022	12 Juillet 2022	5 jours
Phase de programmation	12 juillet 2022	1 ^{er} Aout 2022	17 jours

2.6.2 Rédaction du rapport de stage

Ces deux mois de stage nous ont permis d'approfondir nos connaissances en conception et analyse d'un système d'information. Nous avons eu dans la première semaine d'analyse à cogiter sur le système d'information, ce qui nous a permis d'en tirer les besoins et fonctionnalités du projet. Nous avons ensuite eu à suivre des formations dans les langages de programmation suivants : Java, TypeScript, HTML, JavaScript, CSS. Les trois dernières semaines étaient consacrées à la programmation proprement dite. Nous avons eu à rencontrer des bugs qui nous ont pris des heures avant la résolution. Deux jours avant la date limite, nous avons effectué des tests afin de s'assurer du bon fonctionnement du logiciel.

2.7 Présentation de la méthode d'analyse

2.7.1 Le langage UML



Figure 3 : logo de UML.

Pour la réalisation de notre projet, nous avons utilisé un langage de modélisation afin de simplifier l'abstraction du système. Le langage de modélisation que nous avons choisi est UML. UML (Unified Modeling Language ou Langage de Modélisation Unifié) est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). UML 1.0 a été normalisé en janvier 1997 ; UML 2.0 a été adopté par l'OMG en juillet 2005. La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017).

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet.

UML offre un standard de modélisation, pour représenter l'architecture logicielle.

Les différents éléments représentables sont :

- Activité d'un objet/logiciel ;
- Acteurs ;
- Processus ;
- Schéma de base de données ;
- Composants logiciels ;
- Réutilisation de composants.

UML 2 a de nombreux types de diagrammes, qui sont divisés en deux catégories. Certains types représentent des informations structurales, et les autres représentent des types généraux de comportement, dont quelques-uns représentent différents aspects des interactions. Ces diagrammes peuvent être classés hiérarchiquement comme indiqué dans le diagramme de classes suivant :

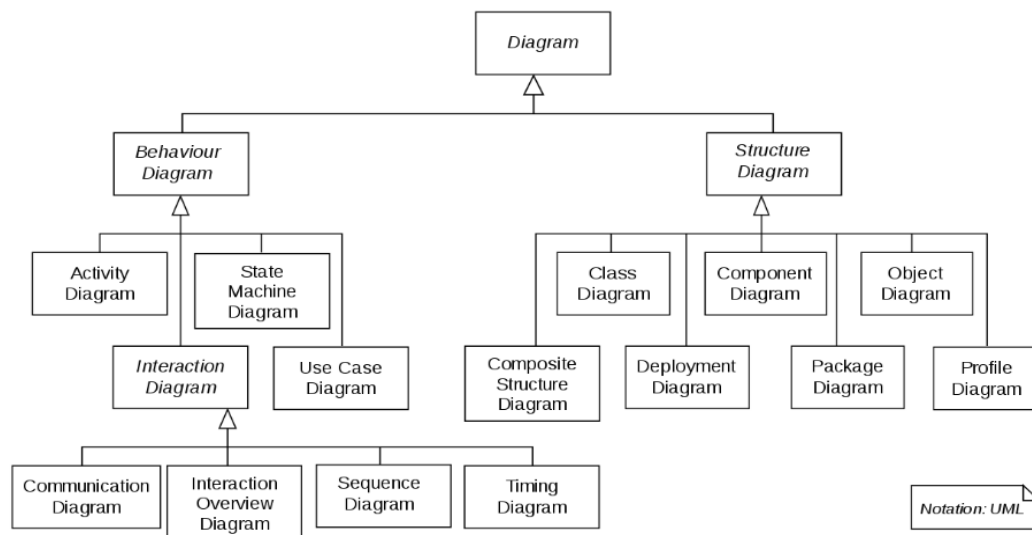


Figure 4 : Organigramme des diagrammes UML.

- **Les diagrammes structurels ou statiques (07)** rassemblent :
 - **Diagramme de classes** (Class Diagram) : représentation des classes intervenant dans le système ;
 - **Diagramme d'objets** (Object Diagram) : représentation des instances de classes (objets) utilisées dans le système ;
 - **Diagramme de composants** (Component Diagram) : représentation des composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...) ;
 - **Diagramme de déploiement** (Deployment Diagram) : représentation des éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux ;
 - **Diagramme des paquets** (Package Diagram) : représentation des dépendances entre les paquets (un paquet étant un conteneur logique

- permettant de regrouper et d'organiser les éléments dans le modèle UML), c'est-à-dire entre les ensembles de définitions ;
- **Diagramme de structure composite** (Composite Structure Diagram) : représentation sous forme de boîte blanche des relations entre composants d'une classe (depuis UML 2.x) ;
 - **Diagramme de profils** (Profile Diagram) : spécialisation et personnalisation pour un domaine particulier d'un métamodèle de référence d'UML (depuis UML 2.2).
 - **Les diagrammes comportementaux (03)** rassemblent :
 - **Diagramme de cas d'utilisation** (Use-case Diagram) : représentation des possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire de toutes les fonctionnalités que doit fournir le système ;
 - **Diagramme d'états-transitions** (State Machine Diagram) : représentation sous forme de machine à états finis du comportement du système ou de ses composants ;
 - **Diagramme d'activité** (Activity Diagram) : représentation sous forme de flux ou d'enchaînement d'activités du comportement du système ou de ses composants.
 - **Les diagrammes d'interaction ou dynamiques (04)** :
 - **Diagramme de séquence** (Sequence Diagram) : représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs ;
 - **Diagramme de communication** (Communication Diagram) : représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets (depuis UML 2.x) ;
 - **Diagramme global d'interaction** (Interaction Overview Diagram) : représentation des enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité) (depuis UML 2.x) ;
 - **Diagramme de temps** (Timing Diagram) : représentation des variations d'une donnée au cours du temps (depuis UML 2.3).

2.7.2 Le processus 2TUP

Pour élaborer le projet, nous avons couplé le langage UML au processus 2TUP. Le 2TUP est un processus de développement logiciel qui implémente la méthode du processus unifié. Il est subdivisé en plusieurs étapes dont chacune découle de la précédente. Son cycle de vie forme un Y, ce qui permet de dissocier les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire pour 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 qui sont :

- La branche fonctionnelle ;
- La branche technique ;
- La branche conception ou de réalisation.

La branche fonctionnelle : Elle capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.

La branche technique : Elle capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système le sont indépendamment des fonctions à réaliser.

La branche conception : Elle consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

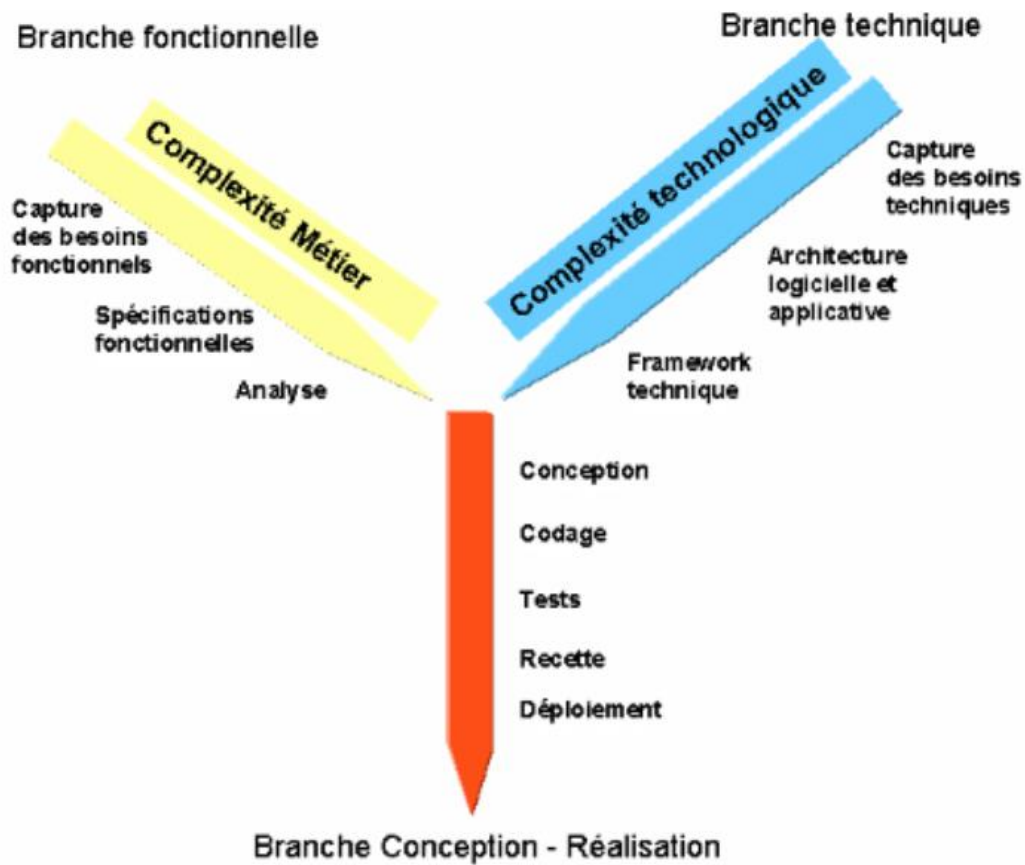


Figure 5 : Schéma du cycle de vie en Y du processus 2TUP.

2.8 Présentation de l'outil de modélisation : Power AMC

Pour modéliser notre système nous avons eu recours à un logiciel de modélisation. De nos jours plusieurs logiciels permettent d'accomplir cette tâche :

- StarUML logiciel développé par MKLab ;
- BOUML développé par Bruno Pages ;
- Power AMC développé par SAP.

Le logiciel utilisé dans ce projet est Power AMC

POWER AMC est l'un des premiers outils qui permet d'élaborer des modèles de données que cela soit MERISE, UML ou autre, de manière graphique et de les implémenter quel que soit le SGBD et ce de manière automatique. De même, l'outil permet de modéliser les processus métiers. Le lien entre la modélisation des

données et la modélisation des processus peut être effectué, offrant ainsi aux entreprises qui possèdent POWER AMC / AMC Designor l'opportunité de mettre en œuvre un référentiel unique des développements et des processus que ceux-ci soient informatisés ou non.

Aussi Power AMC est une force dans tout nouveau projet d'entreprise car il permet d'identifier avec précision quels processus, quelles personnes et/ou quelles données seront impactés. L'estimation et maîtrise des coûts en est grandie.



Figure 6 : Logo de POWER AMC.

2.9 Etude détaillée de la solution

2.9.1 Les acteurs

En UML, un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé. Les acteurs apparaissent dans les diagrammes de cas d'utilisation. Dans notre projet, Les acteurs sont :

- La secrétaire : elle se charge de l'inscription d'un élève, de la perception des frais scolaires d'un élève.

- Le directeur : se charge de la mise en place de l'année académique, de la gestion des élèves, de la gestion des frais, des types de frais. Il consulte aussi les rapports d'inscription et de paiement en vue du recouvrement.
- Le Fondateur ou Promoteur, acteur secondaire qui fixe les différents frais et consulte les différents rapports.
- L'administrateur : s'occupe de la gestion des comptes (création de compte, désactivation de compte) et de toutes les fonctionnalités des autres acteurs.

2.9.2 Capture des besoins fonctionnels

Dans cette partie, nous aurons à détailler les fonctions spécifiques au système à l'aide de quelques diagrammes notamment ceux de cas d'utilisation, d'activités et de séquences.

2.9.2.1 Diagramme de contexte

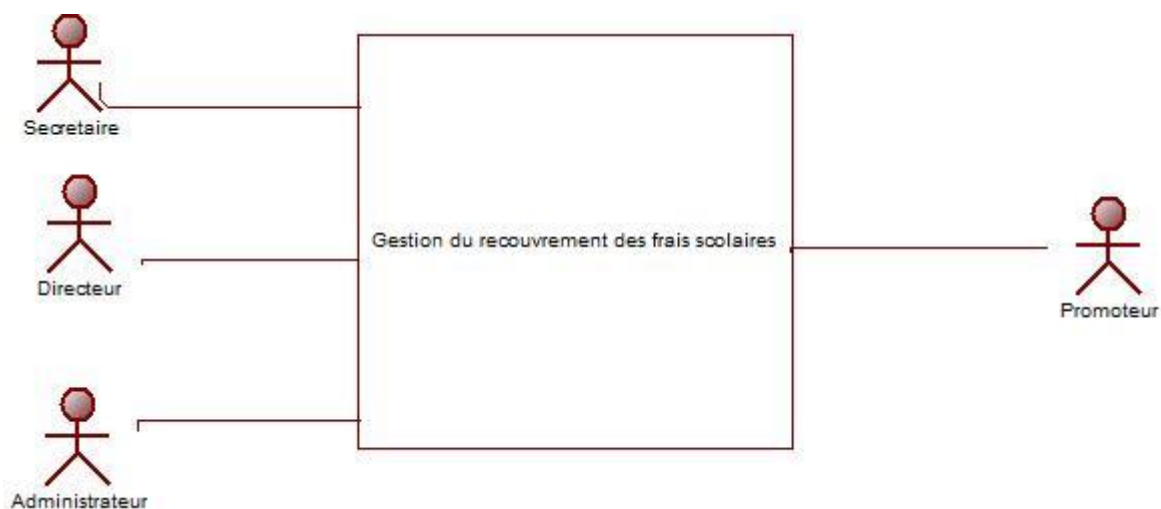


Figure 7 : Diagramme de contexte.

2.9.2.2 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils permettent d'illustrer ce que le système fait et comment les différents acteurs l'utilisent.

2.9.2.2.1 Les cas d'utilisation

Le tableau ci-dessous récapitule les cas d'utilisation recensés dans notre système.

Tableau 3 : Récapitulatif des cas d'utilisation

Cas d'utilisation		Acteurs
Gérer une année	Editer une année	Administrateur, Directeur
	Supprimer une année	
	Définir l'année en cours	
Consulter la liste des années dans le système		
Gérer une classe	Refondre une année	
	Annuler une année	
	Définir une année par défaut	
Consulter la liste des classes		
Gérer un type de Frais	Fixer un type de frais	
	Editer un type de frais	
	Supprimer un type de frais	
Consulter la liste des types de frais		
Gérer un Frais	Fixer un frais	
	Editer un frais	
	Supprimer un frais	
Consulter la liste des frais		
Gérer une adresse	Enregistrer une adresse	Administrateur, Secrétaire
	Editer une adresse	
	Bannir une adresse	
Consulter la liste des adresses		
Gérer une inscription	Enregistrer une inscription	
	Editer une inscription	
	Annuler une inscription	
Consulter la liste des inscriptions d'une classe		
Consulter la liste des inscriptions		
Gérer un élève	Editer un élève	Administrateur, Secrétaire, Directeur
Consulter la liste des élèves d'une classe		
Consulter la liste de tous les élèves		
Gérer un engagement		Administrateur, Directeur
Consulter la liste des engagements		
Gérer un élève spécial		
Consulter la liste des élèves spéciaux		
Gérer un paiement	Enregistrer un paiement	Administrateur, Secrétaire
	Editer un paiement	
	Annuler un paiement	
Consulter les paiements d'un élève		

Gérer les comptes	Administrateur
Consulter tous les utilisateurs	
S'authentifier	Administrateur, Secrétaire, Directeur
Consulter les statistiques	

2.9.2.2.2 Les diagrammes de cas d'utilisation

- Diagramme de cas d'utilisation vu de la secrétaire

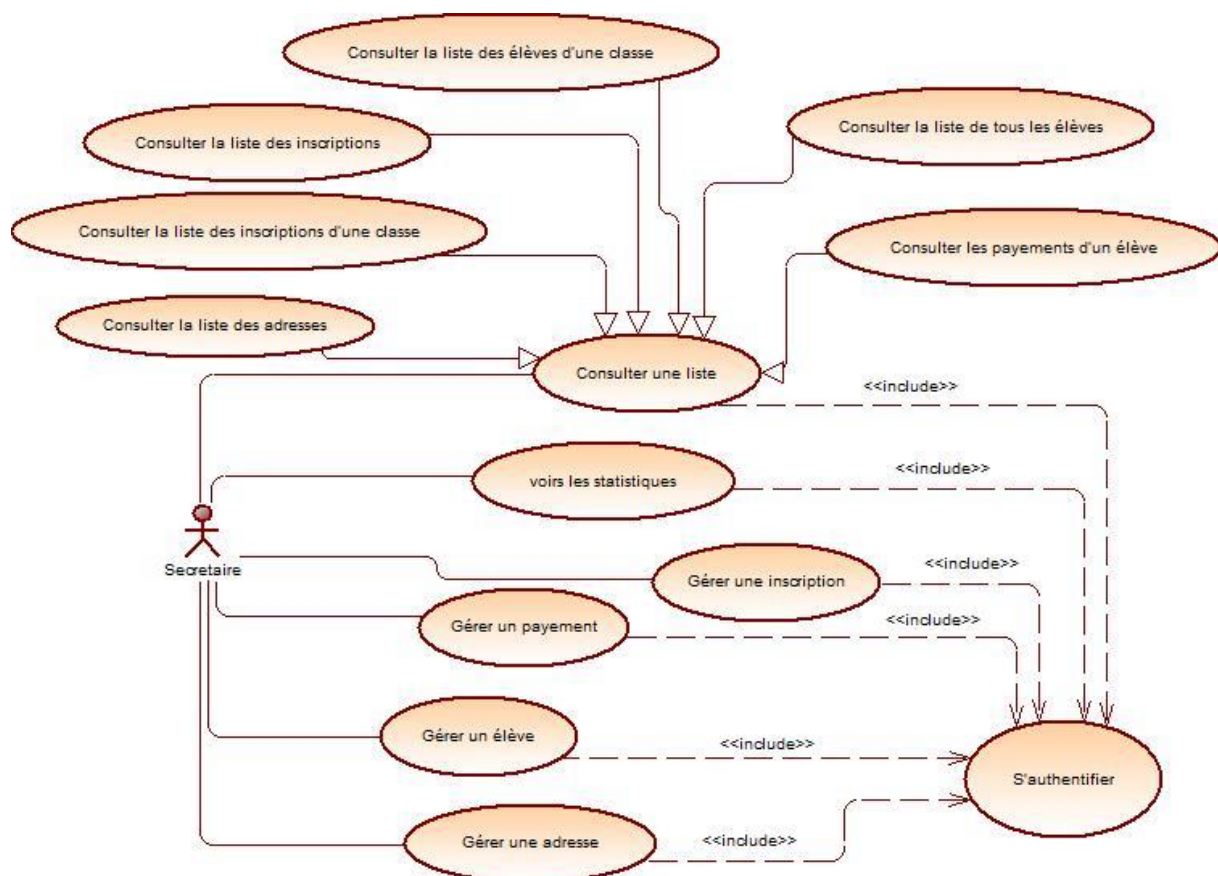


Figure 8 : Diagramme de cas d'utilisation vue de la secrétaire.

- **Diagramme de cas d'utilisation vu du directeur**

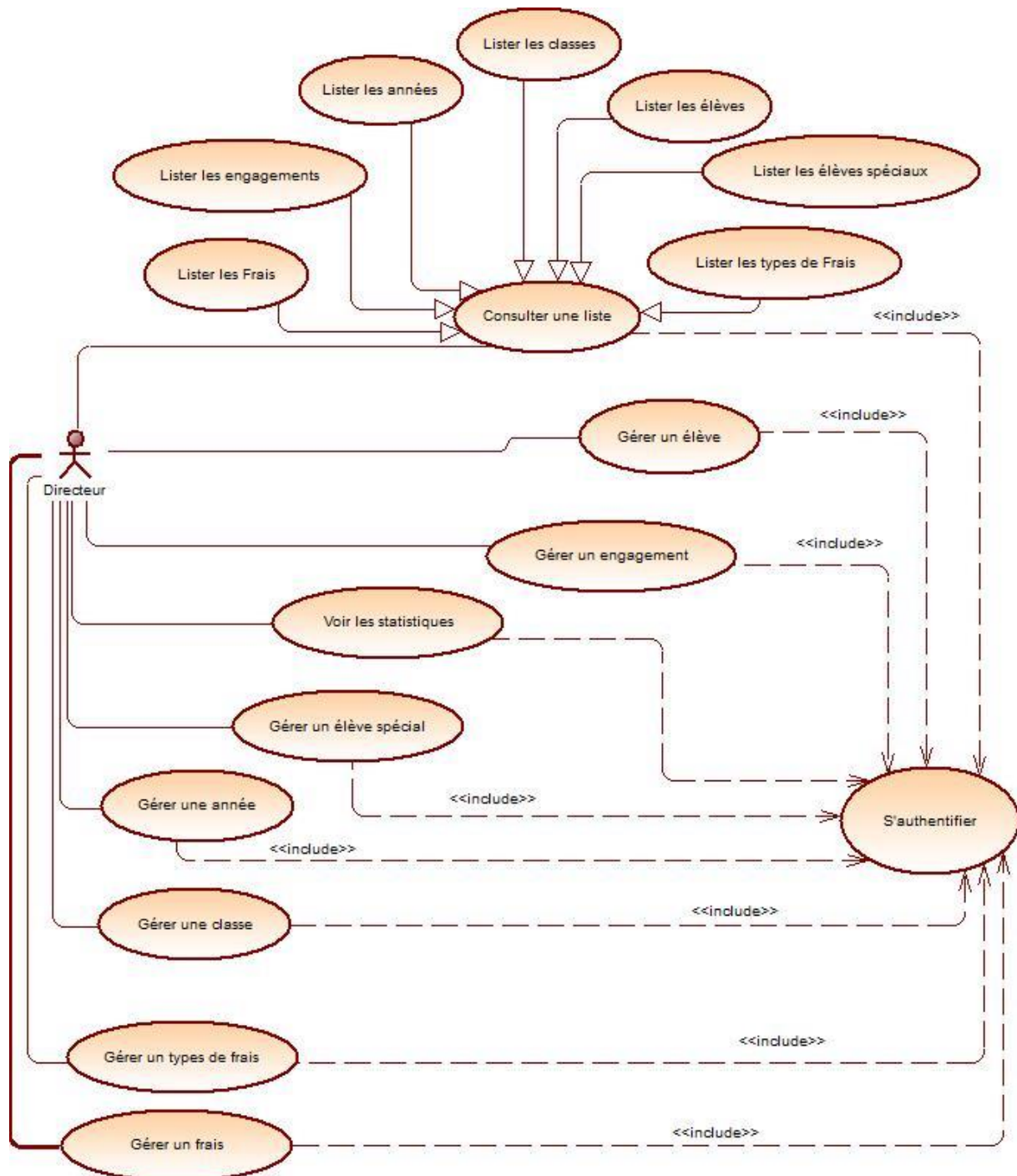


Figure 9 : Diagramme de cas d'utilisation vue du directeur.

- Diagramme de cas d'utilisation global et vu de l'administrateur

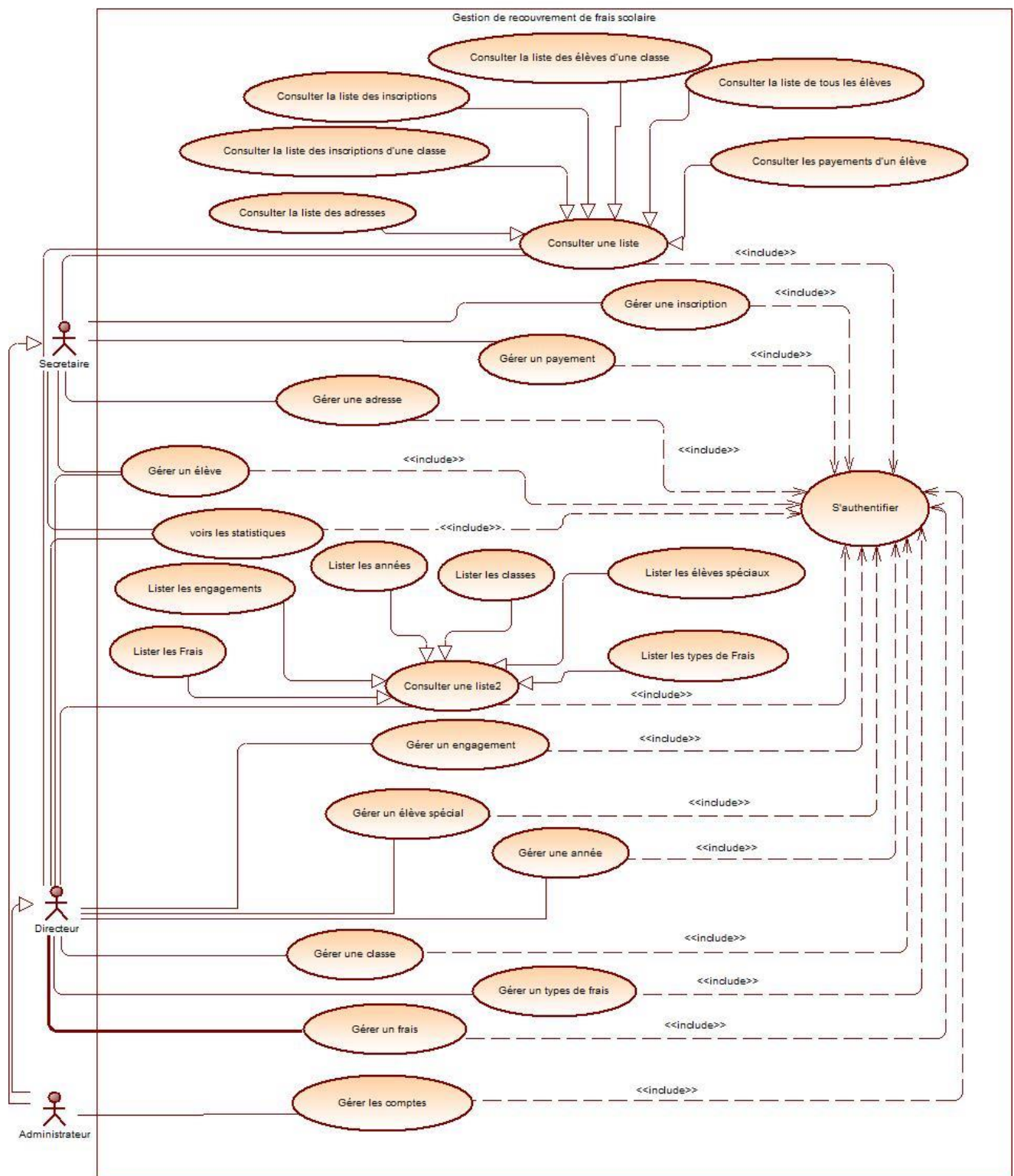


Figure 10 : Diagramme de cas d'utilisation vue globale.

2.9.2.2.3 Description textuelle de certains cas d'utilisation**- Cas d'utilisation « S'authentifier »****Tableau 4 : cas d'utilisation « s'authentifier »**

Sommaire d'identification
Résumé : Permet à tout utilisateur de se connecter à l'application pour avoir accès à ses données
Acteurs : Administrateur, Directeur, Secrétaire
Auteur : BAGNA Tarik
Date de création : 30/07/2022
Version : 3.0
Date de dernière modification : 03/08/2022
Description des scénarii
Préconditions
- Fonctionnalité optimale de l'application
- Le non accès direct aux fonctionnalités de l'application
Scénario nominal
1- L'Utilisateur saisit les informations associées à son compte (identifiant de connexion et mot de passe)
2- Le Système vérifie la validité des informations saisies
3- Les informations saisies sont validées par le système
4- Le Système affiche la page d'accueil à l'Utilisateur
Scénarios alternatifs
SA1 : Les informations saisies sont incorrectes. SA1 débute au point 2 du scénario nominal. 3- Le Système affiche une erreur d'authentification Le scénario reprend au point 1 du scénario nominal.
Postconditions
- L'utilisateur s'authentifie avec succès

- **Cas d'utilisation « Enregistrer un paiement »**

Tableau 5 : cas d'utilisation « enregistrer un paiement »

Sommaire d'identification
Résumé : Permet à tout utilisateur d'enregistrer un paiement effectué par un élève.
Acteur principal : Secrétaire
Auteur : BAGNA Tarik
Date de création : 30/07/2022
Version : 3.0
Date de dernière modification : 03/08/2022
Description des scénarii
Préconditions
<ul style="list-style-type: none">- Fonctionnalité optimale de l'application- La Secrétaire a accès à ses fonctionnalités dans l'application
Scénario nominal
1- La Secrétaire choisit une salle
2- Elle sélectionne un élève
3- Elle choisit le frais que cet élève peut payer
4- Elle saisit le montant que l'élève veut verser
5- L'application vérifie la validité des cases remplies et sélectionnées
6- Les informations sont validées
7- L'application enregistre le paiement
Scénarios alternatifs
SA1 : Le montant saisi est supérieur au montant du frais. SA1 débute au point 5 du scénario nominal. 6- L'application affiche une erreur Le scénario reprend au point 4 du scénario nominal.
Postconditions
<ul style="list-style-type: none">- Le paiement a été enregistré avec succès- La Secrétaire délivre un reçu de paiement

- **Cas d'utilisation « s'inscrire »**

Tableau 6 : cas d'utilisation « s'inscrire »

Sommaire d'identification
Résumé : Permet à tout élève de s'inscrire au secrétariat de l'école.
Acteur principal : Secrétaire
Auteur : BAGNA Tarik
Date de création : 30/07/2022
Version : 3.0
Date de dernière modification : 08/08/2022
Description des scénarii
Préconditions
<ul style="list-style-type: none">- Fonctionnalité optimale de l'application- La Secrétaire a accès à ses fonctionnalités dans l'application
Scénario nominal
1- La Secrétaire saisie les informations de l'élève
2- Elle choisit une salle
3- Le système vérifie les informations saisies et choisies
4- L'élève est pré enregistrer
5- La secrétaire valide l'inscription après la réception du montant d'inscription
6- Le système enregistre l'inscription
Scénarios alternatifs
SA1 :Les informations saisies sont erronées. SA1 débute au point 3 du scénario nominal. 6- L'application affiche une erreur Le scénario reprend au point 1 du scénario nominal.
Postconditions
<ul style="list-style-type: none">- L'élève est préinscrit et inscrit avec succès avec succès- La Secrétaire délivre un reçu de payement

2.9.2.3 Diagramme de classes

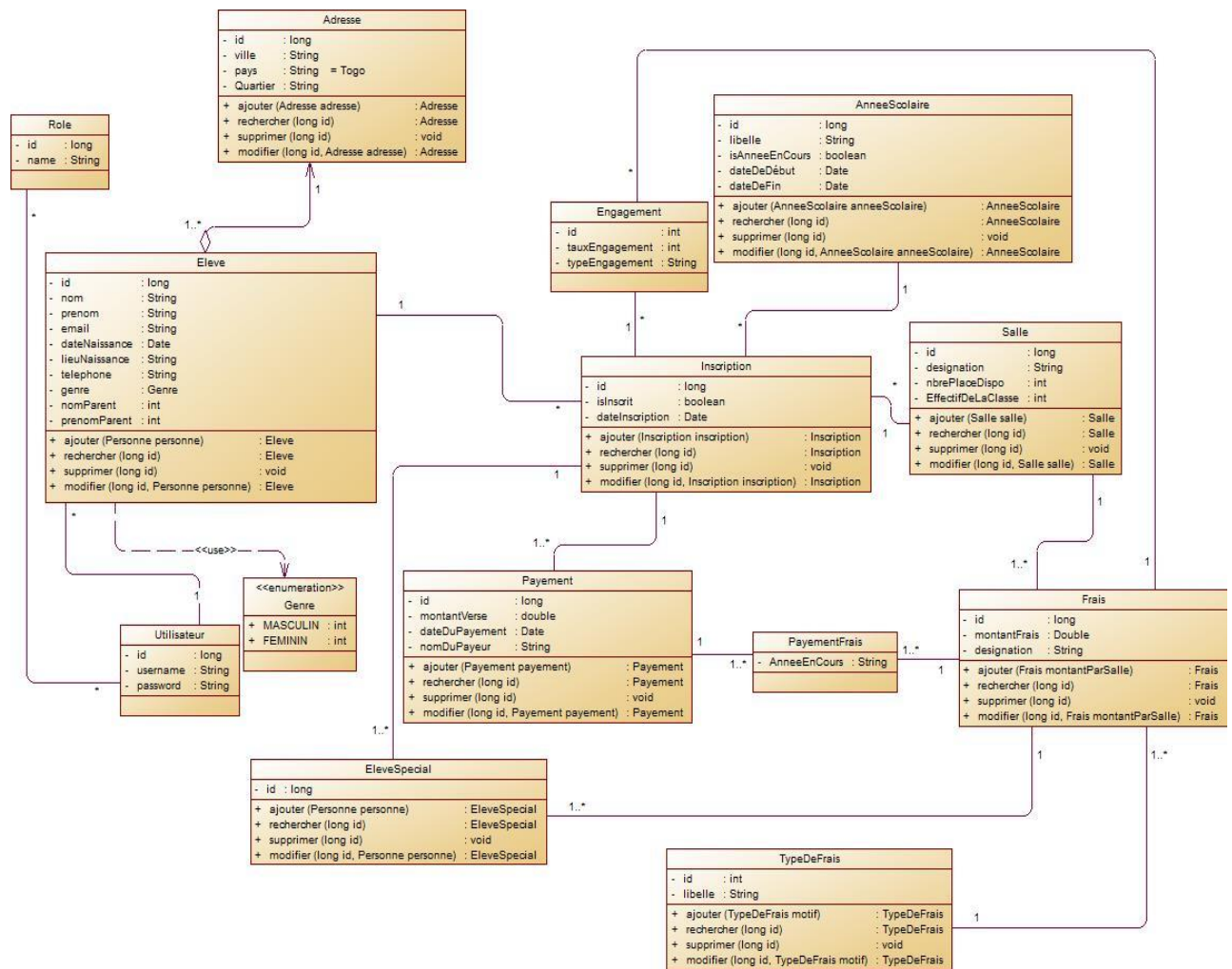


Figure 11 : Diagramme de classes.

2.9.2.4 Diagrammes de séquence - Authentification

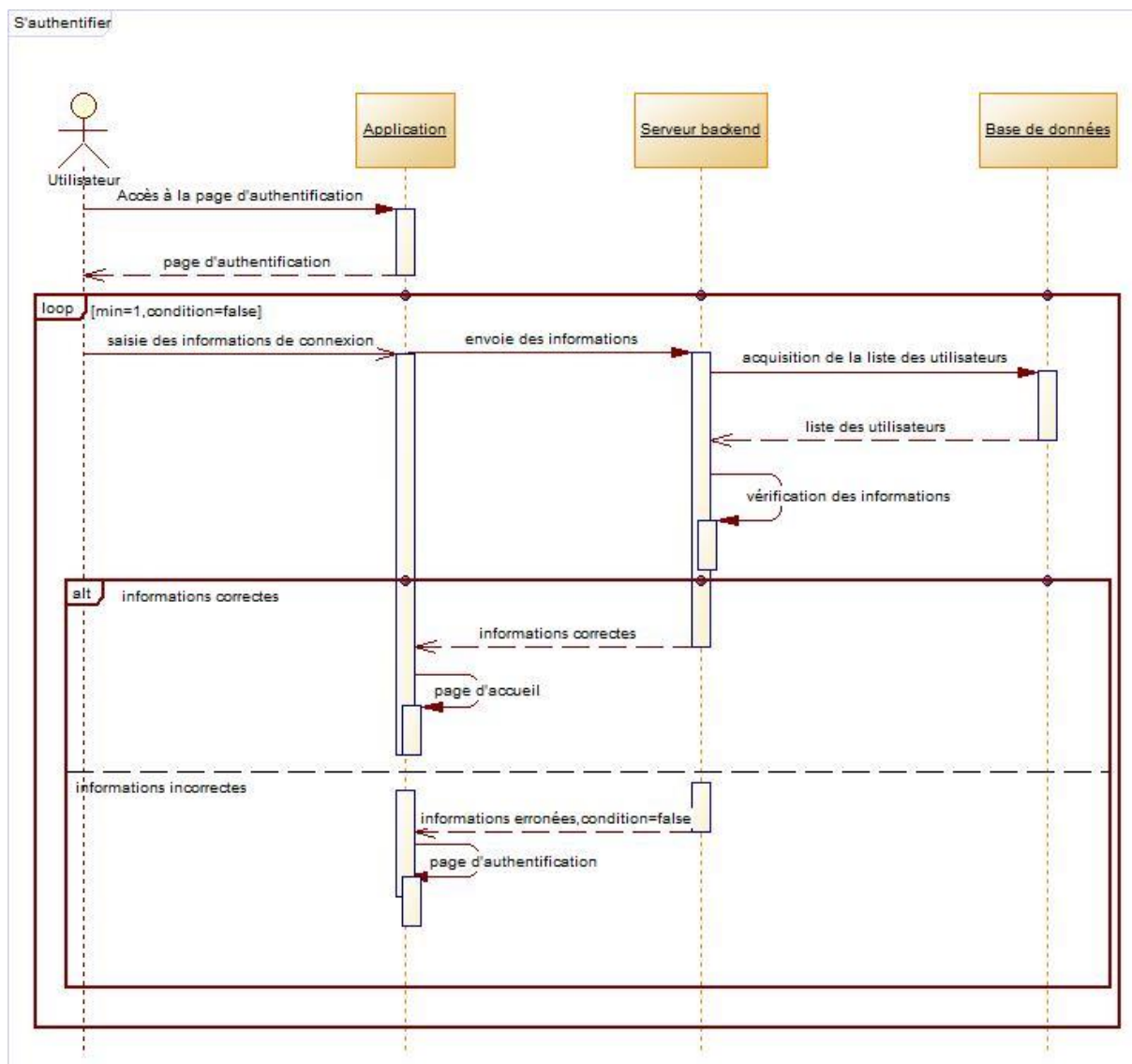


Figure 12 : diagramme de séquence : « s'authentifier ».

- Enregistrer un paiement

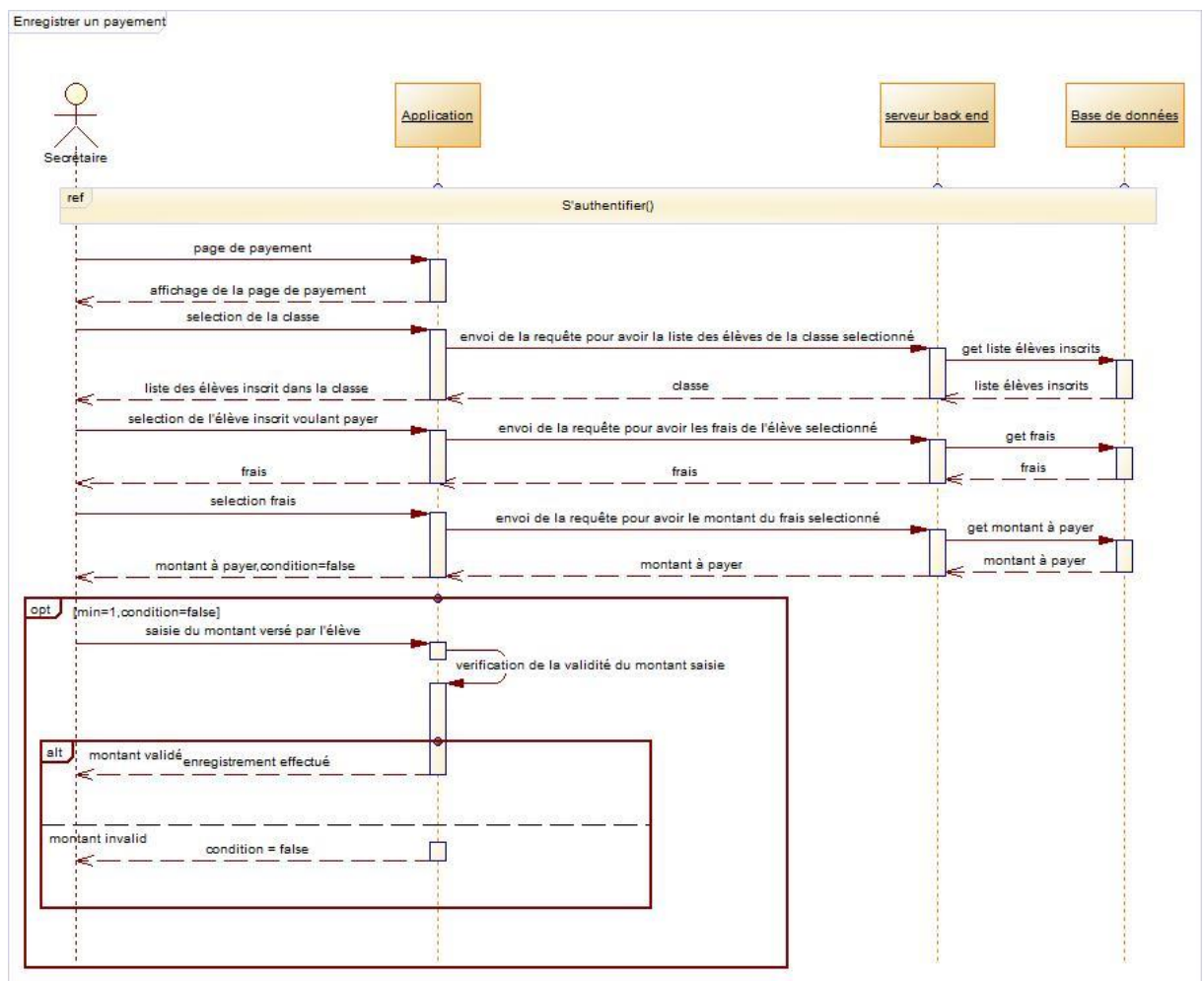


Figure 13 : diagramme de séquence : « enregistrer un paiement ».

2.9.2.5 Diagrammes d'activités

-S'authentifier

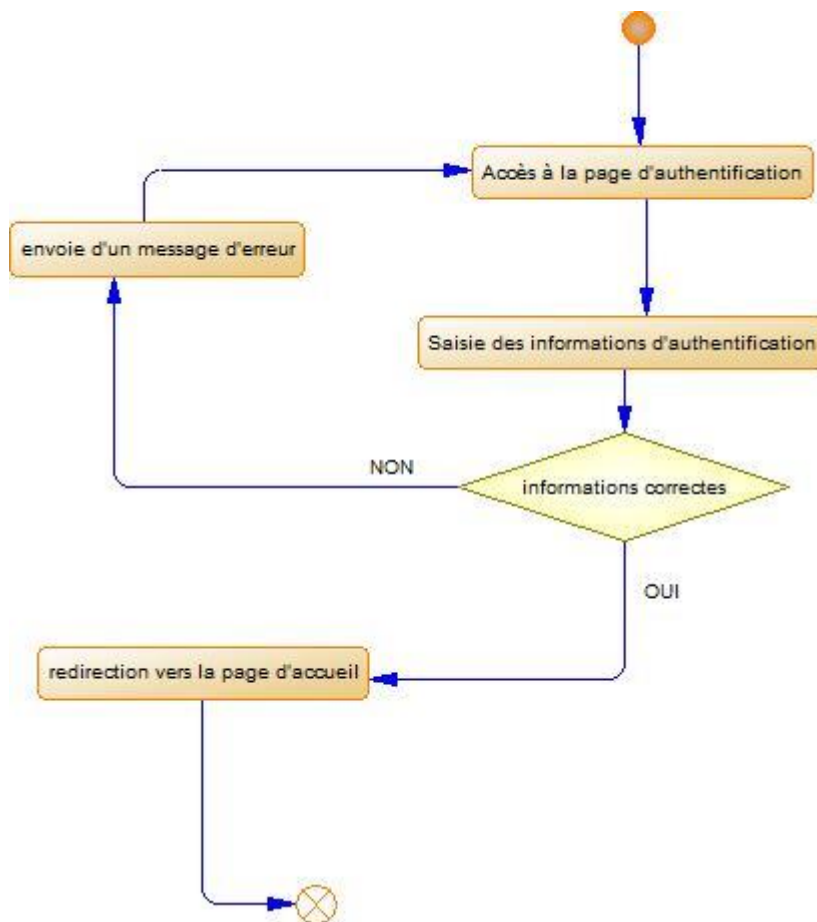


Figure 14 : diagramme d'activité : « s'authentifier ».

-Enregistrer un paiement

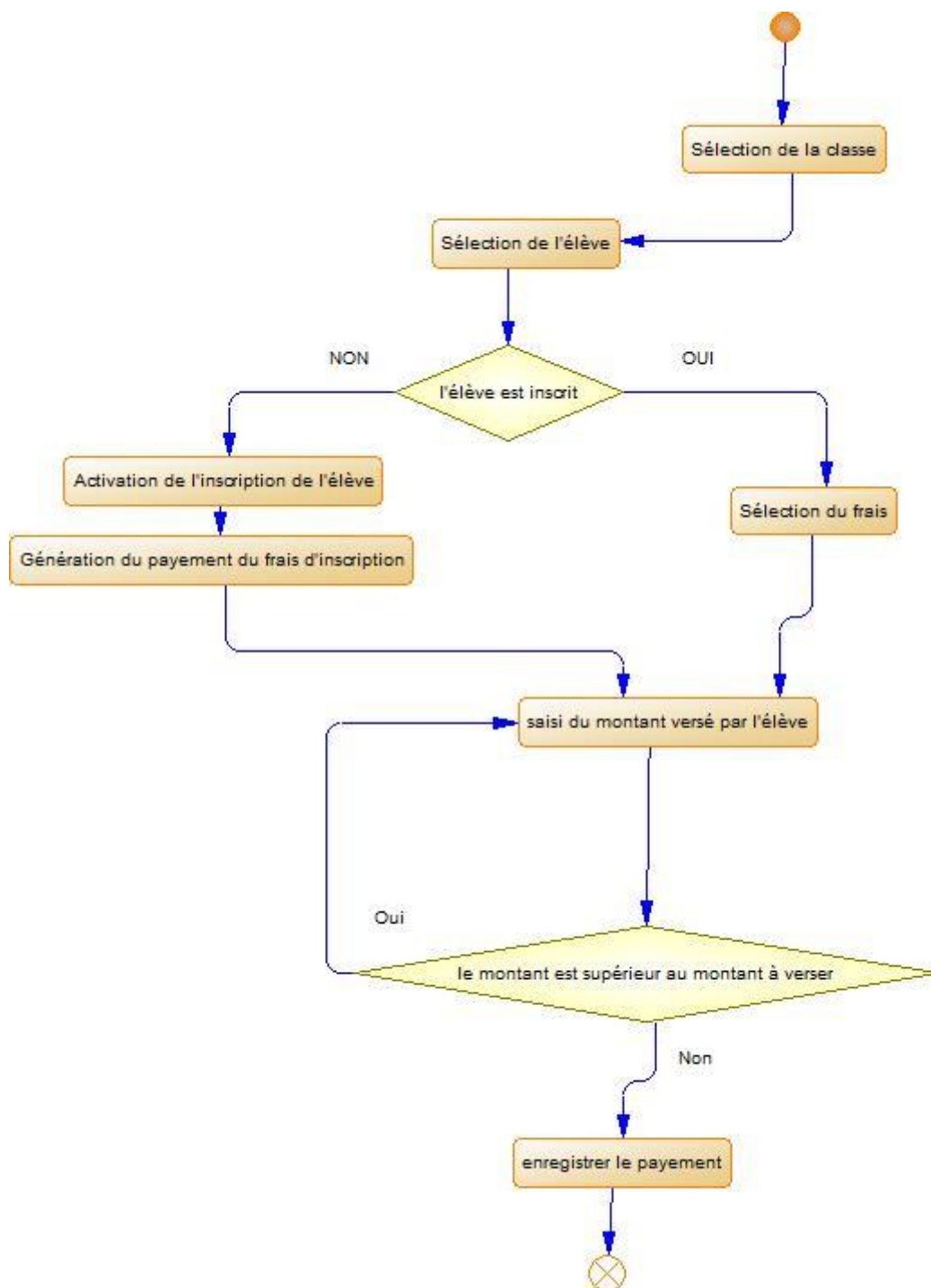


Figure 15 : diagramme d'activité : « enregistrer un paiement ».

PARTIE 3 :

REALISATION ET MISE EN ŒUVRE

PARTIE 3 : REALISATION ET MISE EN ŒUVRE

3 Réalisation et mise en œuvre

3.1 Matériels et logiciels utilisés

3.1.1 Matériels

Pour la conception de notre application, nous avons eu à utiliser un ordinateur portable dont les caractéristiques sont ci-dessous :

- **Marque** : HP ;
- **Modèle** : HP ZBook 15 G1 15,6 pouces ;
- **Stockage**: 250 Go SSD;
- **Mémoire ram**: 16 Go;
- **Processeur** : Intel(R) Core (TM) i7-4810MQ CPU @ 2.80GHz ;
- **Système d'exploitation** : Windows 10 Professionnel.

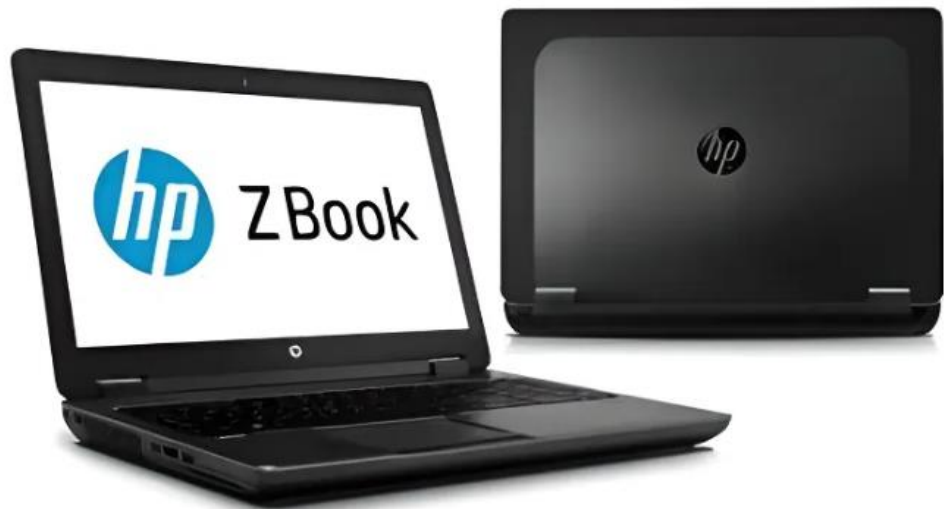


Figure 16 : Localisation de l'IAI-TOGO.

3.1.2 Langages et logiciels

3.1.2.1 Langages de programmation

3.1.2.1.1 HTML 5



Figure 17 : Logo de HTML5.

HTML5 (HyperText Markup Language 5) est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM : HTML5 et XHTML5. Le langage comprend également une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme. Le travail a été repris par le W3C en mars 2007 après avoir été lancé par le WHATWG. Les deux organisations travaillent en parallèle sur le même document afin de maintenir une version unique de la technologie. Le W3C clôt les ajouts de fonctionnalités le 22 mai 2011, annonçant une finalisation de la spécification en 2014¹, et encourage les développeurs Web à utiliser HTML 5 dès ce moment. Fin 2016, la version 5.1 est officiellement publiée et présente plusieurs nouveautés qui doivent faciliter le travail des développeurs d'applications Web.

3.1.2.1.2 CSS 3



Figure 18 : Logo de CSS3.

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

3.1.2.1.3 SASS



Figure 19 : Logo de SASS.

Sass (Syntactically Awesome Stylesheets) est un langage dynamique de génération de feuilles de style en cascade initialement développé par Hampton Catlin et Nathalie Weizenbaum. Sass est un préprocesseur CSS. C'est un langage de description compilé en CSS. SassScript est un langage de script pouvant être utilisé à l'intérieur du code Sass. Deux syntaxes existent. La syntaxe originale, nommée « syntaxe indentée », est proche de Haml. La nouvelle syntaxe se nomme SCSS. Elle

a un formalisme proche de CSS. Sass peut être étoffé avec Compass (pratique pour les préfixes des différents navigateurs).

3.1.2.1.4 JavaScript



Figure 20 : Logo de JavaScript.

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser le cas échéant. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet de créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm, avec environ 500 000 paquets en août 2017.

3.1.2.1.5 TypeScript



Figure 21 : Logo de TypeScript.

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript). Le code TypeScript est transcompilé en JavaScript, et peut ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript. TypeScript permet un typage statique optionnel des variables et des fonctions, la création de classes et d'interfaces, l'import de modules, tout en conservant l'approche non-contrainante de JavaScript.

3.1.2.1.6 JASPER STUDIO



Figure 22 : Logo de JasperStudio

JasperReports est un outil de reporting open source, offert sous forme d'une bibliothèque qui peut être embarquée dans tous types d'applications Java. JasperReports se base sur des fichiers XML (dont l'extension est en général .jrxml) pour la présentation des états. Il peut être couplé à iReport (outil WYSIWYG) ou JasperStudio (plugin Eclipse équivalent) pour faciliter sa mise en œuvre dans une application Java, classique ou orientée web.

3.1.2.1.7 JAVA



Figure 23 : Logo de Java.

Java est une technique informatique développée initialement par Sun Microsystems puis acquise par Oracle à la suite du rachat de l'entreprise. Défini à l'origine comme un langage de programmation, Java a évolué pour devenir un ensemble cohérent d'éléments techniques et non techniques. Ainsi, la technologie Java regroupe :

- Des standards (la plate-forme Java) définis sous forme de spécification par le Java Community Process (JCP), en trois éditions :

- Java SE (*standard edition*),
- Java EE (*enterprise edition*), s'appuyant sur Java SE,
- Java ME (*micro edition*), indépendante des deux précédentes ;
- Des logiciels (langages informatiques, bibliothèques, frameworks, serveurs d'application, outils d'aide au développement), dont :
- Des implémentations (concurrentes) de ces spécifications,
- Un écosystème d'autres logiciels s'appuyant sur tout ou partie de ces standards, voire leur faisant concurrence ;
- Des communautés d'entreprises, organisations à but non lucratif (fondations, Java User Groups, universités) et indépendants, membres ou non du JCP, possédant tout ou partie des marques, brevets, parts de marché liés à la technologie Java.

3.1.2.2 Logiciels

3.1.2.2.1 IDE : IntelliJ IDEA



Figure 24 : Logo de IntelliJ IDEA.

IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré (en anglais Integrated Development Environment – IDE) destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

3.1.2.2.2 Serveur local : WampServer



Figure 25 : Logo de wampServer.

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin 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 icon (icône près de l'horloge de Windows).

3.1.2.2.3 Base de données : phpMyAdmin



Figure 26 : Logo de phpMyAdmin.

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL. Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer. Cette interface pratique permet d'exécuter, très facilement et sans grandes

connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export.

3.1.2.3 Librairies

3.1.2.3.1 NPM

npm est le gestionnaire de paquets par défaut pour l'environnement d'exécution JavaScript Node.js de Node.js. npm se compose d'un client en ligne de commande, également appelé npm, et d'une base de données en ligne de paquets publics et privés payants, appelée le registre npm. Le registre est accessible via le client, et les paquets disponibles peuvent être parcourus et recherchés via le site Web de npm. Le gestionnaire de paquets et le registre sont gérés par npm, Inc. Depuis la version 0.6.3 de Node.js, npm fait partie de l'environnement et est donc automatiquement installé par défaut³. Npm fonctionne avec un terminal et gère les dépendances pour une application. Il permet également d'installer des applications Node.js disponibles sur le dépôt npm. En avril 2020, GitHub annonce l'acquisition de npm.

3.1.2.3.2 Bootstrap



Figure 27 : Logo de Bootstrap.

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

3.1.2.3.3 JQuery



Figure 28 : Logo de jQuery.

jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. La première version est lancée en janvier 2006 par John Resig. Le but de la bibliothèque étant le parcours et la modification du DOM (y compris le support des sélecteurs CSS 1 à 3 et un support basique de Xpath), elle contient de nombreuses fonctionnalités ; notamment des animations, la manipulation des feuilles de style en cascade (accessibilité des classes et attributs), la gestion des événements, etc. L'utilisation d'Ajax est facilitée et de nombreux plugins sont présents.

3.1.2.4 Framework

3.1.2.4.1 Spring



Figure 29 : Logo de spring

Spring est un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests. Spring s'appuie principalement sur l'intégration de trois concepts clés : l'inversion de contrôle, la programmation orientée aspect, une couche d'abstraction. La couche d'abstraction permet d'intégrer d'autres frameworks et bibliothèques avec une plus grande facilité.

Cela se fait par l'apport ou non de couches d'abstraction spécifiques à des frameworks particuliers. Il est ainsi possible d'intégrer un module d'envoi de mails plus facilement.

3.1.2.4.2 Angular



Figure 30 : Logo de Angular.

Angular (communément appelé « Angular 2+ » ou « Angular v2 et plus »)^{2,3} est un framework côté client, open source, basé sur TypeScript, et codirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète d'AngularJS, cadriciel construit par la même équipe. Il permet la création d'applications Web et plus particulièrement d'applications web monopage : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.

3.2 Sécurité de l'application

La sécurité constitue un enjeu majeur dans l'utilisation mais également dans le développement des applications web. Ainsi, spring par son framework spring security, permet de garantir 5 aspects de la sécurité d'une application.



Figure 31 : Logo de spring security.

3.2.1 Authentification

Pour accomplir certaines actions spécifiques sur notre application telles que l'inscription d'un élève, le paiement de la scolarité, il est nécessaire de se connecter. L'authentification sur le site se déroule à travers plusieurs étapes qui garantissent la sécurité du site.

3.2.1.1 Chiffrement

A l'authentification tout comme à l'ajout d'un utilisateur par l'administrateur, il est nécessaire de chiffrer les informations soumises (tels que les mots de passe). Les mots de passe du site qui sont enregistrés dans la base de données sont chiffrés par la Classe BCryptPasswordEncoder de spring framework.

3.2.1.2 JWT token

JSON Web Token (JWT) est un standard ouvert défini dans la RFC 7519. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité et de l'authenticité des données. Elle s'effectue par l'algorithme HMAC ou RSA.

3.2.2 Contrôle d'accès

Une fois authentifié, l'utilisateur souhaite accéder à des fonctionnalités offertes par l'application. Au préalable, il faut contrôler s'il a le droit d'y accéder. Cette partie est assurée par le framework spring security qui dispose des classes et

fonctions de configurations d'accès à certaines fonctionnalités d'une application spring boot.

3.2.3 L'intégrité des données

Il s'agit ici de prévenir l'altération volontaire ou accidentelle d'une donnée ou des services d'un système. Elle s'applique à la phase de communication entre composants, au flux, au stockage des données (altérations de contenu) et au système (détection d'intrusion).

3.2.4 Confidentialité

La confidentialité des données doit être assurée lors d'échange de données sensibles (mot de passe, données bancaires ou médicales.) Il s'agit de garantir que des données acquises illégalement soient inutilisables. Spring security gère cette partie.

3.2.5 Non répudiation des données

Cette fonction consiste à s'assurer que l'envoi et la réception d'un message sont incontestables. En d'autres termes, l'émetteur ou le récepteur d'une donnée ne doit pas être en mesure de nier son implication en cas de litige. Le moyen technologique repose sur les certificats. La non répudiation des données sur notre application est assurée par la journalisation de toutes les tâches effectuées.

3.3 Evaluation Financière de la solution

3.3.1 Coût matériel de la solution

Tableau 7 : Coût matériel de la solution

Désignation	Description	Prix Unitaire	Quantité	Montant Total
Ordinateur	Achat d'un ordinateur	432 335,71 FCFA ¹	1	432 335,71 FCFA

¹ Source : <https://www.cddiscount.com/informatique/ordinateurs-pc-portables/ordinateur-portable-hp-zbook-15-g2-core-i7-ram/f-1070992-rcdhp00603ec.html#mpos=0|mp>

Connexion	La connexion internet	30 000 FCFA/mois ²	12 mois	360 000,00 CFA
------------------	-----------------------	-------------------------------	---------	----------------

3.3.2 Développement de la solution

Tableau 8 : Coût de développement de la solution

Désignation	Description	Prix Unitaire	Quantité	Montant Total
Développement de la solution	Analyse, conception et développement de la solution	5 000 CFA ³	400 H (8H*50 jours)	2 000 000 CFA

3.3.3 Coût total de la solution

Tableau 9 : coût total.

Désignation	Description	Prix Unitaire	Quantité	Montant Total
Ordinateur	Achat d'un ordinateur	432 335,71 FCFA ¹	1	432 335,71 FCFA
Connexion	La connexion internet	30000 CFA /mois ²	12 mois	360 000 FCFA
Développement de la solution	Analyse, conception et développement de la solution	5 000 FCFA ³	400 H (8H*50 jours)	2 000 000 FCFA
Total				2 792 335,71 FCFA

² Source : <https://togocom.tg/la-fibre/>

³ Source : <https://www.axxiss.be/index.php/2020/02/01/cout-logiciel-sur-mesure/>

3.4 Présentation de l'application

3.4.1 Présentation (Squelette de l'application : enchainement des menus)

Notre application nommée **FS Management** est une plateforme web réalisée pour faciliter l'inscription des élèves et la bonne gestion du paiement des frais scolaires. Son accès est contrôlé par une authentification.

L'application est divisée en 3 modules : le module Secrétaire, le module Directeur, le module administrateur.

- **Le module Secrétaire** : module à travers lequel il est permis aux secrétaires de gérer les inscriptions et les paiements des frais. Ce module regroupe des fonctionnalités telles que :

- Lister les inscriptions d'une classe ;
- Lister les adresses ;
- Lister les élèves d'une classe ;
- Lister tous les élèves ;
- Lister les paiements relatifs à un élève ;
- Gérer une inscription ;
- Gérer un paiement ;
- Gérer un élève ;
- Gérer une adresse.

- **Le module directeur** : module à travers lequel il est permis au directeur de gérer les frais et les types de frais pouvant être payés ainsi que les engagements. Ce module regroupe des fonctionnalités telles que :

- Lister les frais ;
- Lister les années ;
- Lister les engagements ;
- Lister les classes ;
- Lister les élèves ;
- Lister les élèves spéciaux ;
- Lister les types de frais ;
- Gérer les années ;
- Voir les statistiques ;
- Gérer un engagement ;

- Gérer un élève ;
 - Gérer un élève spécial ;
 - Gérer une classe ;
 - Gérer un type de frais ;
 - Gérer un frais.
- **Le module administrateur** : C'est le module à travers lequel les administrateurs (ceux qui y ont accès) gèrent les différentes composantes du site web. Son accès est contrôlé par une page d'authentification. Ce module intègre en plus des fonctionnalités des autres modules, une fonctionnalité administrateur : la gestion des comptes utilisateurs.

3.4.2 Architecture de l'application

3.4.2.1 Architecture MVVM(Model-View-ViewModel) côté front-end (Angular)

Il s'agit d'une variante du modèle MVC. Elle présente quelques différences essentielles, dont la plus importante est la séparation de l'interface utilisateur et de la logique métier. Cela permet au concepteur et au développeur de travailler indépendamment, avec moins de va-et-vient et de temps perdu à essayer de comprendre ce que fait l'autre. Cela permet également de tester l'interface utilisateur indépendamment du reste de l'application. Le ViewModel est une classe à laquelle l'interface utilisateur se lie. Elle est chargée de coordonner les données envoyées à la vue et les commandes envoyées au contrôleur. Cela permet de découpler la vue des données, et le contrôleur des données et de la vue. Le ViewModel est responsable de la gestion des données et des commandes permettant de les manipuler.

3.4.2.2 Architecture de microservices

L'architecture de microservices désigne un style d'architecture utilisé dans le développement d'applications. Elle permet de décomposer une application volumineuse en composants indépendants, chaque élément ayant ses propres responsabilités. Pour diffuser la requête d'un utilisateur unique, une application basée sur des microservices peut appeler plusieurs microservices internes pour composer sa réponse.

3.4.3 Mise en place de la base de données

Pour générer la base de données, nous avons eu recours au code First qui consiste à créer des classes d'abord afin de générer la base de données grâce aux migrations. Voici ci-dessous quelques exemples de codes de création de la base de données :

```

Hibernate: create table adresse (id bigint not null auto_increment, pays varchar(255), quartier varchar(255), ville varchar(255), primary key (id)) engine=InnoDB
Hibernate: create table annee (id bigint not null auto_increment, date_debut datetime(6), date_fin datetime(6), is_annee_en_cours bit not null, libelle varchar(255), primary key (id)) engine=InnoDB
Hibernate: create table classe (id bigint not null auto_increment, effectif_de_la_classe integer not null, designation varchar(255), nombre_de_place_disponible integer not null, pri
Hibernate: create table eleve (id bigint not null auto_increment, date_de_naissance datetime(6), email varchar(255), fonction_parent_tuteur varchar(255), genre varchar(255), lieu_de
Hibernate: create table eleve_special (id bigint not null auto_increment, frais_id bigint, inscription_id bigint, primary key (id)) engine=InnoDB
Hibernate: create table engagement (id bigint not null, taux bigint not null, type_engagement varchar(255), frais_id bigint, inscription_id bigint, primary key (id)) engine=InnoDB
Hibernate: create table frais (id bigint not null auto_increment, designation varchar(255), montant_frais double precision not null, classe_id bigint, type_frais_id bigint, primary
Hibernate: create table inscription (id bigint not null auto_increment, date_inscription datetime(6), is_inscrit bit not null, annee_id bigint, classe_id bigint, eleve_id bigint, pr
Hibernate: create table paiement (id bigint not null auto_increment, date_du_paiement datetime(6), montant_verse double precision not null, non_payeur varchar(255), reste_a_payer dou
Hibernate: create table roles (id bigint not null auto_increment, name varchar(255), primary key (id)) engine=InnoDB
Hibernate: create table type_frais (id bigint not null auto_increment, libelle varchar(255), primary key (id)) engine=InnoDB
Hibernate: create table users (id bigint not null auto_increment, is_enable bit not null, is_not_locked bit not null, password varchar(255), username varchar(255), primary key (id))
Hibernate: create table users_roles (user_id bigint not null, roles_id bigint not null) engine=InnoDB
Hibernate: alter table eleve add constraint FKsr9J8lhq8yux8vwwa9c14wh1 foreign key (id_adresse) references adresse (id)
Hibernate: alter table eleve_special add constraint FKgxfqb8jxns8nen3c39afjwe foreign key (frais_id) references frais (id)
Hibernate: alter table eleve_special add constraint FKqkh2td5grkat2u4wgjuqhndh foreign key (inscription_id) references inscription (id)
Hibernate: alter table engagement add constraint FK7df4ngmhf3pae9ul988vj4hf1 foreign key (frais_id) references frais (id)
Hibernate: alter table engagement add constraint FKq15hakbey39dlyyrk9fx22g3t foreign key (inscription_id) references inscription (id)
Hibernate: alter table frais add constraint FKqgt1iffkos12bb9osuocj8mpi foreign key (classe_id) references classe (id)
Hibernate: alter table frais add constraint FK3bycl65bh7vnc9r2124o8alpv foreign key (type_frais_id) references type_frais (id)
Hibernate: alter table inscription add constraint FK257iufpjbith4qh8rbh344ut foreign key (annee_id) references annee (id)
Hibernate: alter table inscription add constraint FKo6s2ahy3b4e0bkschg3lr7fmr foreign key (classe_id) references classe (id)
Hibernate: alter table inscription add constraint FKcttrk9ja646r33g42sxlwruic foreign key (eleve_id) references eleve (id)
Hibernate: alter table paiement add constraint FK6lwc2crgifpsqknayxrtgpxv foreign key (frais_id) references frais (id)
Hibernate: alter table paiement add constraint FKler186yxavaq9x7a5loipdo2t foreign key (inscription_id) references inscription (id)
Hibernate: alter table users_roles add constraint FKa62j07k5mhg1fpp955h37ponj foreign key (roles_id) references roles (id)
Hibernate: alter table users_roles add constraint FK2o0jvgh89leuvvo17cbqvdxaa foreign key (user_id) references users (id)

```

Figure 32 : Génération de la base de données et des tables.

```

0      private static final long serialVersionUID = 1L;
1
2      @Id
3      @GeneratedValue(strategy = GenerationType.IDENTITY)
4      private long id;
5
6      private String designation;
7
8      private int nombreDePlaceDisponible;
9
10     private int EffectifDeLaClasse;
11
12     @OneToMany(mappedBy = "classe", fetch = FetchType.LAZY, cascade = CascadeType.ALL)
13     private List<Inscription> inscriptions;
14
15     @OneToMany(mappedBy = "classe", fetch = FetchType.LAZY, cascade = CascadeType.ALL)
16     private List<Frais> frais;
17
18     kirat_2003
19     public Classe(){
20
21     }

```

Figure 33 : création de la table classe

```

public class Inscription implements Serializable {

    private static final long serialVersionUID = 1L;

    2 usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    2 usages
    private boolean isInscrit;
    2 usages
    @Transient
    private String statut;
    2 usages
    private Date dateInscription;

    2 usages
    @ManyToOne(fetch = FetchType.EAGER)
    private Eleve eleve;
    2 usages
    @ManyToOne(fetch = FetchType.EAGER)
    private Annee annee;
    2 usages
    @ManyToOne(fetch = FetchType.EAGER)
    private Classe classe;
    2 usages
    @OneToMany(mappedBy = "inscription", fetch = FetchType.LAZY, cascade = CascadeType.ALL)
    private List<EleveSpecial> eleveSpecials;
    @OneToMany(mappedBy = "inscription", fetch = FetchType.LAZY, cascade = CascadeType.ALL)
    private List<Engagement> engagements;
    
```

Figure 34 : création de la table inscription.


```
import ...

29 usages kirat_2003 *
@Entity
public class Payement implements Serializable {

    private static final long serialVersionUID = 1L;

    2 usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    2 usages
    private double montantVerse;

    2 usages
    private double resteApayer;

    2 usages
    private Date dateDuPayement;

    2 usages
    private String nomPayeur;

    2 usages
    @ManyToOne(fetch = FetchType.EAGER)
    private Inscription inscription;

    2 usages
    @ManyToOne(fetch = FetchType.EAGER)
    private Frais frais;
```

Figure 35 : création de la table payement.

3.4.4 Plan de navigation

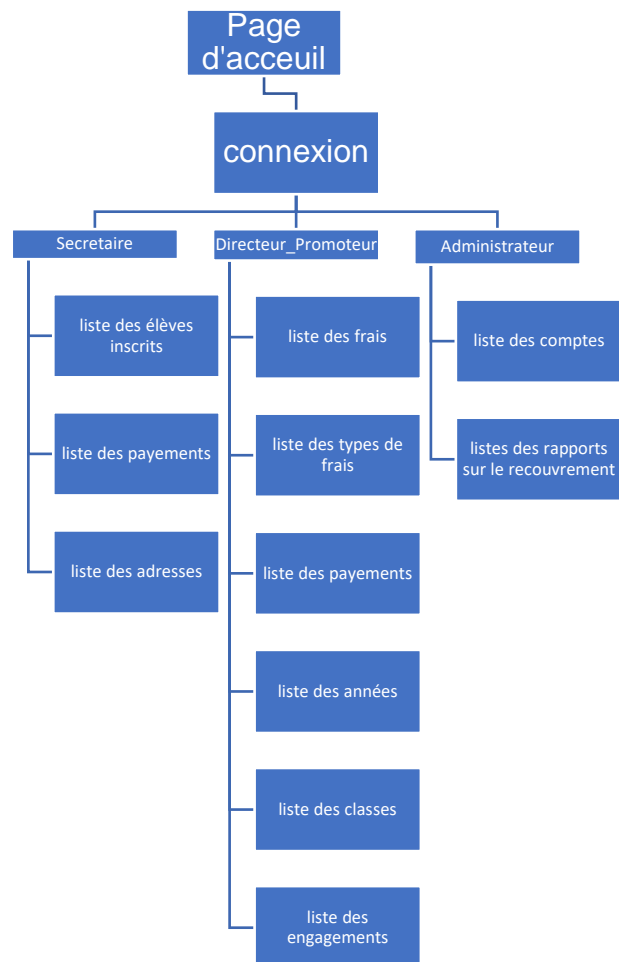
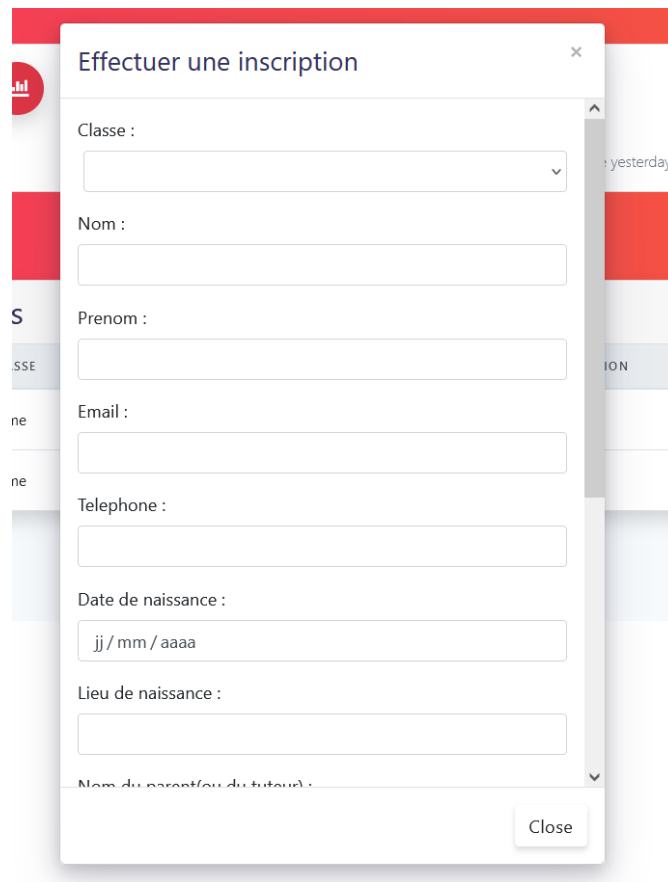


Figure 36 : plan de navigation.

3.4.5 Quelques masques de saisi

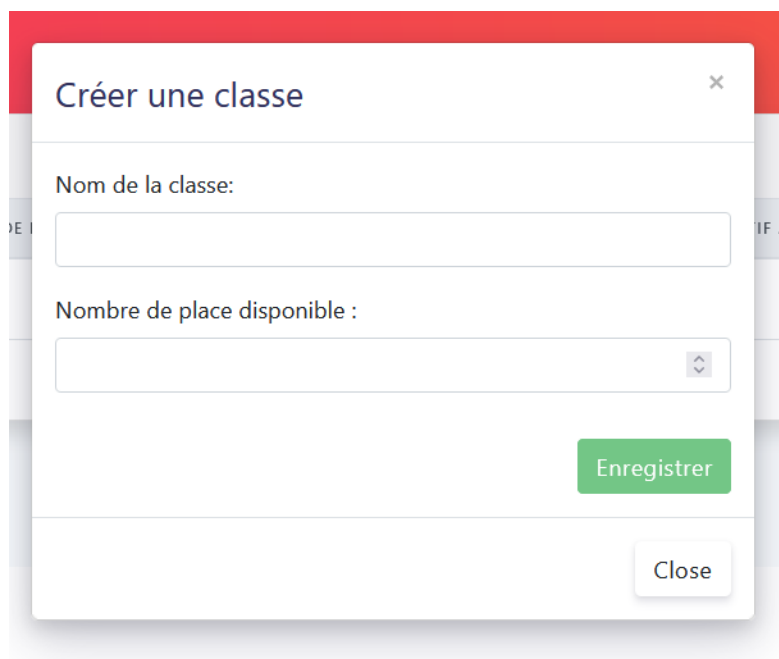


A modal form titled "Effectuer une inscription" with a close button (X) in the top right corner. The form contains the following fields:

- Classe : A dropdown menu.
- Nom : A text input field.
- Prenom : A text input field.
- Email : A text input field.
- Telephone : A text input field.
- Date de naissance : A text input field with a placeholder "jj / mm / aaaa".
- Lieu de naissance : A text input field.
- Nom du parent(ou du tuteur) : A text input field.

A "Close" button is located at the bottom right of the modal.

Figure 37 : Enregistrement d'une inscription.



A modal form titled "Créer une classe" with a close button (X) in the top right corner. The form contains the following fields:

- Nom de la classe: A text input field.
- Nombre de place disponible : A text input field with a small up/down arrow icon on the right.

At the bottom right, there is a green button labeled "Enregistrer" and a "Close" button below it.

Figure 38 : Enregistrement d'une classe.

A modal form titled "Créer une année" (Create a year) with a close button (X) in the top right corner. It contains two date input fields: "Date de début d'année scolaire" (Start of school year) and "Date de fin d'année scolaire" (End of school year), both with a placeholder "jj / mm / aaaa". At the bottom right, there is a green "Enregistrer" (Save) button and a white "Close" button.

Figure 39 : Enregistrement d'une année.

A modal form titled "Effectuer un paiement" (Make a payment) with a close button (X) in the top right corner. It contains several input fields: "Nom&Prenom :" (Name & Surname) with the value "BAGNA Tarik", "Classe :" (Class) with the value "6eme", "Frais :" (Fees) with a dropdown arrow, "Montant Versé :" (Amount Paid) with the value "0" and a spinner, and "Auteur du paiement :" (Payment Author) which is empty. At the bottom right, there is a green "Enregistrer" (Save) button and a white "Close" button.

Figure 40 : Enregistrement d'un paiement.

3.4.6 Quelques états et statistiques

Liste des élèves				
Classe : 6eme				
FS Management				
Nom&Prenoms	Genre	Adresse	Date d'inscription	Statut
BAGNA Tarik	MASCULIN	TOGO, Lomé, Addidogomé	27/07/2022 01:47	Inscrit
string string	MASCULIN	TOGO, Lomé, Adéwui	30/07/2022 12:58	Inscrit

Figure 41 : Liste des élèves d'une classe.


	SF MANAGEMENT	07/08/2022 13:14 2022-2023
<u>Reçu de paiement N°</u> 18		
Nom :	BAGNA	
Prénom(s) :	Tarik	
Classe :	6eme	
Motif :	Ecolage 6eme	
La somme de : (FCFA)	10000.0	
Reste à payer :	65000.0	

Figure 42 : Reçu de paiement.

CONCLUSION

Notre stage en interne, réalisé sous la supervision de Mr. **TETTEH Kodjo Amen** a pour thème : « **Gestion du recouvrement des frais scolaires** ». Il s'agit de faciliter la gestion des recouvrements au sein d'une école grâce à une application web. Nous avons donc réalisé et mis en place une plateforme de suivi et de gestion des paiements des frais scolaires. L'administrateur enregistre des comptes utilisateurs à la secrétaire, au directeur. Les utilisateurs se connectent et ont accès à leurs fonctionnalités spécifiques. Une inscription aussitôt effectuée est enregistré dans la base de données de même qu'un paiement. Ce stage nous a permis d'appréhender de nouvelles notions et d'approfondir nos connaissances théoriques et techniques dans la réalisation et l'implémentation d'une plateforme web mais aussi d'apprendre les framework Spring et Angular. Nous en avons aussi appris plus sur la gestion des recouvrements dans une école afin de pouvoir implémenter un système de gestion des paiements des inscriptions.

BIBLIOGRAPHIE INDICATIVE

NOTES DE COURS :

- 1- Cours de UML 2^{ème} année IAI-TOGO (2021-2022) : Mr **AMEVOR** Kossi Amédomé ;
- 2- Cours de conception des bases de données 2^{ème} année IAI-TOGO (2021-2022) : Mr **GBODUI** Roland-Joseph ;
- 3- Cours sur spring boot et spring security: **Mohamed youssfi**;
- 4- Cours sur angular : **Mohamed youssfi**.

WEBOGRAPHIE INDICATIVE

Sites web visités :

- <https://fr.m.wikipedia.org/> (dernière date de visite : 06/08/2022)
- <https://stackoverflow.com/> (dernière date de visite : 09/08/2022)
- <https://getbootstrap.com/docs/5.1/getting-started/introduction/> (dernière date de visite : 31/07/2022)
- <https://mvnrepository.com/> (dernière date de visite : 08/08/2022)
- <https://www.youtube.com/> (dernière date de visite : 07/08/2022)
- <https://www.google.com/> (dernière date de visite : 09/08/2022)

DOCUMENTS ANNEXES

- 1- Rapport de stage de M TRAORE Abdel-Qahar (2019-2020) sur le thème : « Réalisation d'une plateforme d'achats et de réservation de places pour des évènements » ;
- 2- Rapport de stage de Mlle YERIMA Boucharatou (2020-2021) sur le thème : « Gestion informatisée des mobiliers de CENETI et de IAI-TOGO ».

TABLES DES MATIERES

REMERCIEMENTS	I
SOMMAIRE	II
LISTE DES FIGURES.....	IV
LISTE DES TABLEAUX	VI
INTRODUCTION	1
PARTIE 1 : CAHIER DES CHARGES	3
1 Cahier des charges	3
1.1 Présentation de l’Institut Africain d’Informatique, Représentation du TOGO.....	3
1.2 Présentation de Odda Technology	4
1.3 Présentation du sujet	4
1.4 Problématique du sujet.....	5
1.5 Intérêt du sujet.....	5
1.5.1 Objectifs.....	5
1.5.2 Résultats	6
PARTIE 2 : PRE-PROGRAMMATION	8
2 Préprogrammation.....	8
2.1 Etude de l’existant.....	8
2.2 Critique de l’existant	8
2.3 Propositions de solutions	8
2.3.1 Se procurer un abonnement sur le site myscol.com.....	9
2.3.2 Réaliser une application web personnalisée	9
2.4 Evaluation financière des solutions.....	10
2.4.1 Solution : myscol	10
2.5 Choix de la solution	10
2.6 Planning prévisionnel de réalisation	10
2.6.1 Tableau résumant le panning prévisonnel de réalisation	10
2.6.2 Rédaction du rapport de stage	11
2.7 Présentation de la méthode d’analyse.....	11
2.7.1 Le langage UML	11
2.7.2 Le processus 2TUP	15
2.8 Présentation de l’outil de modélisation : Power AMC	16
2.9 Etude détaillée de la solution	17
2.9.1 Les acteurs.....	17

2.9.2	Capture des besoins fonctionnels	18
PARTIE 3 : REALISATION ET MISE EN ŒUVRE.....		32
3	Réalisation et mise en œuvre	32
3.1	Matériels et logiciels utilisés	32
3.1.1	Matériels.....	32
3.1.2	Langages et logiciels	33
3.2	Sécurité de l'application	43
3.2.1	Authentification.....	43
3.2.2	Contrôle d'accès	43
3.2.3	L'intégrité des données	44
3.2.4	Confidentialité	44
3.2.5	Non répudiation des données	44
3.3	Evaluation Financière de la solution	44
3.3.1	Coût matériel de la solution	44
3.3.2	Développement de la solution	45
3.3.3	Coût total de la solution	45
3.4	Présentation de l'application	46
3.4.1	Présentation (Squelette de l'application : enchainement des menus)	46
3.4.2	Architecture de l'application	47
3.4.3	Mise en place de la base de données.....	48
3.4.4	Plan de navigation	51
3.4.5	Quelques masques de saisi.....	52
3.4.6	Quelques états et statistiques.....	54
CONCLUSION		56
BIBLIOGRAPHIE INDICATIVE		VI
WEBOGRAPHIE INDICATIVE.....		VII
DOCUMENTS ANNEXES		VIII
TABLES DES MATIERES		IX