

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene



Faculté d'Electronique et d'Informatique

Département Informatique

Mémoire de Licence

Spécialité ACAD

Thème

Application pour la gestion d'une école
primaire

Présenté par : OUSSADI Yacine
HOUANTI Narimene

Encadré par : Mme. OUAZAR
Fatiha

Année universitaire : 2020/2021

Résumé

La gestion des données est un critère essentiel pour toute entreprise ou établissement, il se peut que ça soit une gestion de stock, une gestion des ressources humaines... Pour ce projet, nous avons opté pour le cas d'une école qui aurait besoin de gérer l'ensemble de ses élèves et ses enseignants à travers une interface simple et pratique, donc la gestion informatisée de ce secteur apporte vraiment une valeur ajoutée à l'établissement.

Dans ce travail, nous nous intéressons à la réalisation d'une interface graphique pour la gestion d'une école primaire ; la gestion de cette structure comporte les modules suivants :

- La gestion des élèves.
- La gestion du staff administratif.
- La gestion des fournitures.
- La gestion financière.

Mots clés : interface graphique, gestion informatisée, école.

Remerciements

Nous tenons à remercier, notre directrice de mémoire Mme. Ouazar, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à la réalisation de ce mémoire.

Nous remercions également les membres du jury pour avoir accepté d'examiner notre travail et de l'enrichir par leurs propositions.

Nous tenons à exprimer notre reconnaissance à toutes les personnes qui nous ont aidé lors de la rédaction de ce mémoire.

Table des figures

1.1	Exemple d'une interface graphique	4
1.2	Schéma représentatif d'une API	5
2.1	Diagramme de cas d'utilisation général	11
2.2	Diagramme de cas d'utilisation de la gestion des élèves	12
2.3	Diagramme de cas de la gestion du staff administratif	13
2.4	Diagramme de cas d'utilisation de la gestion des fournitures	14
2.5	Diagramme de cas d'utilisation de la gestion financière	14
2.6	Diagramme des classes	15
2.7	Diagramme de séquence d'ajout d'un élève	18
2.8	Diagramme de séquence de modification d'un élève	20
2.9	Diagramme de séquence de connexion	22
3.1	Interface de connexion	28
3.2	Interface de gestion de staff	28
3.3	Interface de gestion d'emplois du temps	29
3.4	Interface d'emplois du temps des enseignants	29
3.5	Interface d'ajout, suppression et modification des informations des enseignants	29
3.6	Interface de gestion des fournitures	30
3.7	Interface de gestion financière	30

Table des matières

Introduction générale	1
1 Étude de l'existant et Analyse des besoins	3
1.1 Introduction	3
1.2 Quelques notions	3
1.2.1 Interface graphique	3
1.2.2 Gestion informatisée	4
1.2.3 API	4
1.3 Étude de l'existant	5
1.3.1 AirmesSchool	5
1.3.2 Fonctionnalités	5
1.3.3 Points forts	6
1.4 Analyse des besoins	6
1.4.1 Besoin fonctionnel	7
1.4.2 Besoin non fonctionnel	8
1.5 Conclusion	9
2 Étude Conceptuelle	10
2.1 Introduction	10
2.2 Modélisation UML	10
2.3 Identification des acteurs	10
2.3.1 Diagramme des classes	15
2.3.2 Diagramme de séquence	17
2.3.3 Schéma relationnel	23
2.4 Conclusion	25
3 Implémentations et tests	26
3.1 Introduction	26
3.2 Outils et langage de développement	26
3.2.1 Outils utilisés	26
3.2.2 Langage de développment	27
3.3 Système de gestion de base de donnée	27
3.4 Architecture Model-View-Controller	28
3.5 Présentation de l'application	28
3.5.1 Interfaces graphiques	28
3.6 Conclusion	30
Conclusion général	31

Introduction générale

Le but des institutions Algériennes est de réduire la bureaucratie dans les administrations publiques, à l'image de la possibilité d'obtenir certains documents et certificats officiels, directement sur Internet, a entraîné une informatisation de leurs systèmes administratifs.

Notre travail a pour ambition la réalisation d'une application de gestion d'école primaire qui aura pour but de proposer à ses utilisateurs la possibilité de gérer la totalité de l'école de manière informatisée et automatisée.

Notre manuscrit est structuré en trois chapitres :

Dans le premier chapitre "Étude de l'existant" : nous allons définir quelques notions de base qui représentent les mots clés de notre projet. Ensuite, nous étudierons un cas similaire qui décrit une application de gestion d'école tout en mettant en avant quelques une de ses fonctionnalités. Enfin, nous établirons les objectifs de notre système qui consiste à faire une analyse des besoins de manière détaillée.

Dans le deuxième chapitre "Étude conceptuelle" : nous allons aborder la conception UML de notre système de gestion ; pour cela, nous mettrons en avant les principaux acteurs qui vont interagir avec le système. Ensuite nous présenterons les diagrammes explicatifs de cas d'utilisation, de classe, et de séquence avec leurs descriptions. Enfin nous établirons le schéma relationnel lié à la base de donnée utilisée.

Dans le troisième chapitre "Implémentation et tests" : nous allons présenter

les outils matériels de logiciels ainsi que les langages de programmations et le système de gestion de base de donnée, suivi de l'architecture Model-View-Controller. En dernier, nous déroulerons une suite de figures montrant quelques exemples d'exécution de l'application.

Chapitre 1

Étude de l'existant et Analyse des besoins

1.1 Introduction

Depuis la fin de la crise du logiciel et, par conséquent, le développement du génie logiciel, de nombreux domaines publics et privés axés sur la gestion d'informations, de personnel et d'inventaire, ont informatisé leurs systèmes de gestion. La gestion informatisée a permis, entre autres, la facilitation de la recherche d'informations et son transport vers d'autres supports aisément accessibles et partageables comme un serveur internet ou bien un système de stockage externe tel que les disques durs et les clés USB.

Dans le cas présent, nous nous intéressons à l'informatisation d'un système pour la gestion d'une école primaire. L'un des nombreux avantages à cette méthode de gestion par ordinateur est la rapidité d'exécution des nombreuses tâches administratives et logistiques en comparaison avec un système de gestion classique dit manuel qui est bien plus sujet à l'erreur humaine et à la perte potentielle de données.

1.2 Quelques notions

1.2.1 Interface graphique

L'interface utilisateur est un dispositif basé sur l'emploi d'éléments graphiques tels que les fenêtres, les icônes et les menus [17], qui permettent le dialogue homme-machine, visant la simplicité d'utilisation en créant un environnement de travail convivial [11] (voir FIGURE 1.1).

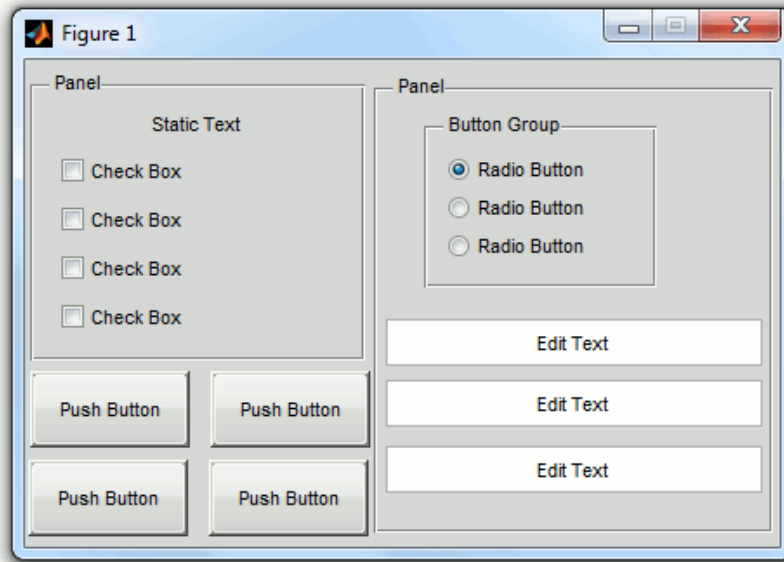


FIGURE 1.1 – Exemple d'une interface graphique

1.2.2 Gestion informatisée

Consiste à relier le domaine d'informatique à des activités industrielles, de gestion ou de documentation. Le but est en effet d'organiser des informations, de les vérifier et de les collecter, de manière à être plus optimal tout en apportant un gain de temps [10].

1.2.3 API

Une API (Application Programming Interface) est un ensemble de protocoles, de méthodes ainsi que de classes contribuant à faire communiquer des applications entre elles et de s'échanger mutuellement des services ou des données [19]. Elle sert d'intermédiaire entre l'utilisateur et le système source (voir FIGURE 1.2).

Il existe tous type d'API pour divers domaines notamment la gestion qui fera l'objet de cette présente étude.

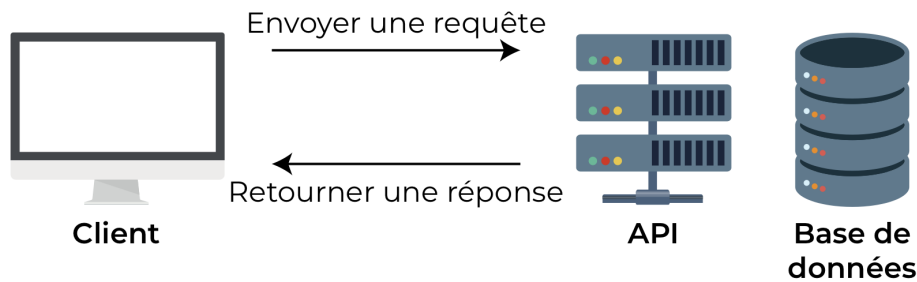


FIGURE 1.2 – Schéma représentatif d'une API

1.3 Étude de l'existant

La gestion manuelle d'un système peut s'avérer complexe, peu optimale, et nécessitant beaucoup de temps. En effet, elle requiert la prise en charge de plusieurs spécificités, notamment pour les tâches administratives. Pour cette problématique, nous constatons que l'informatisation de cette gestion prend de plus en plus d'ampleur au fil des années afin de simplifier le pilotage des systèmes et être mieux efficient en terme d'exploitation des ressources humaines de ce dernier. Dans cette optique et parmi la panoplie de ces systèmes existants, nous allons présenter l'un de ces logiciels.

1.3.1 AirmesSchool

AirmesSchool est une marque d'Euro Tech Conseil [16] qui accompagne les chefs d'établissements scolaires dans leurs missions et facilite leurs tâches à travers son logiciel gestion scolaire. Ce logiciel est développé exclusivement pour gérer la totalité des opérations des institutions scolaires. En effet, il est conçu à la base des exigences et des besoins réels de tout type d'école et quelle que soit la taille [2].

1.3.2 Fonctionnalités

- La gestion des factures et avoirs : consiste à mettre à la disposition des administrateurs une seule solution apte d'effectuer convenablement la procédure de la facturation d'une école. En cas de changement des besoins, le logiciel de gestion scolaire est extensible et modifiable selon les préférences.

- La gestion d'école : offre une panoplie de fonctionnalités utiles pour gérer les sites et les départements de leurs établissements.
- La gestion des absences scolaire : permet de créer, imprimer et exporter les feuilles de présence, d'interpréter des statistiques sur les taux d'absentéisme des élèves. La liste des absences est envoyée en vie scolaire et sera ainsi traitée en temps réel pour chaque séance de la journée.
- Planning des salles : établie un calendrier pour les différentes planifications au cours de l'année. Il permet de répartir les élèves dans les salles, d'intégrer les enseignants directement à votre programmation annuelle et d'organiser les emplois du temps.
- La gestion des notes scolaire : Chaque enseignant peut avoir un compte, Ils peuvent se connecter et corriger des examens et les moyennes se calculent automatiquement par étudiant.
- Statistique scolaire : Permet de visualiser graphiquement les statistiques des inscriptions, des rapports annuels et des progressions des frais de scolarité des élevés d'une classe et de suivre la progression des compétences à maîtriser.

1.3.3 Points forts

- Mise à jour évolutive gratuite.
- Développement sur mesure et personnalisation.
- Assistance et maintenance incluse.

1.4 Analyse des besoins

L'analyse du projet est une démarche à suivre pour spécifier les besoins et modéliser le système selon les critères du client afin de satisfaire sa demande. L'analyse est constituée de deux parties ; les besoins fonctionnels et non fonctionnels [2]

1.4.1 Besoin fonctionnel

Vise à décortiquer les éléments du système pour décrire le rôle de ses derniers afin de mettre en évidence de manière principale, son fonctionnement et ses conditions d'utilisation.

- Gestion des élèves qui consiste à sauvegarder les données de chaque élève et les mettre à jour après chaque examen, l'historique des résultats obtenus avec les dates de chaque élève.
- Gestion du staff administratif qui consiste en la gestion des emplois du temps des enseignants et à mettre à jour les informations des enseignants, agents de sécurité, directeur et administrateur, avec la possibilité de rajout, suppression et modification.
- Gestion des fournitures qui consiste à avoir la possibilité de rajouter ou supprimer des fournitures.
- Gestion financière qui consiste en la gestion des fiches de paie du staff administratif et enseignant.

Description détaillée des différentes fonctionnalités

Gestion des élèves

Consiste à :

- Affichage de la liste d'élève de toute l'école.
- Ajouter, modifier, supprimer ou consulter les donnée d'un élève.
- Ajouter, modifier, supprimer ou consulter les bulletins d'un élève.

Gestion du Staff Administratif

Englobe la gestion des enseignants, des concierges ainsi que les gardiens de l'école :

- Gestion du programme du personnel en ajoutant, modifiant, supprimant ou consultant un emploi du temps propre à chacun.
- Gestion des informations du personnel en ajoutant, modifiant, supprimant ou consultant les données propre à chacun.

Gestion des fournitures

Description de l'inventaire de l'école qui concernera les salles de classe et le bureau administratif en :

- Ajoutant, modifiant, supprimant ou consultant les fournitures.

Gestion financière

Permet d'établir les fiches de paie de chaque salarié au sein du staff administratif en :

- Ajoutant, modifiant, supprimant ou consultant une fiche.

1.4.2 Besoin non fonctionnel

Consiste à établir les contraintes liées au logiciel, assurant le bon fonctionnement de ce dernier ; dans notre cas, on distingue les besoins non fonctionnels suivants :

- Sécurité : l'utilisateur aura le choix entre : créer un mot de passe en s'authentifiant à chaque fois qu'il accède à la plate-forme, et continuer sans mot de passe.
- Ergonomie : facile à manipuler, compréhensible et convivial.
- Maintenabilité : détection des erreurs, extensibilité.
- Portabilité : compatible avec diverses plate-formes, nécessite une installation du SGBD (MySQL).

1.5 Conclusion

Dans un premier temps, on a défini quelques notions qui représentent les mots clés de notre projet.

Suite à cela, on a abordé l'étude de l'existant qui consiste à décrire un système de gestion nommé AirmesSchool ainsi que ses composants qui sont plus ou moins similaires aux nôtres.

Enfin, on a mis en évidence de manière détaillée les acteurs ainsi que les différentes fonctionnalités de l'interface, ce qui nous a permis de mieux cerner les contraintes pour une amélioration et une adaptation pour nos besoins et à ceux du client.

Le chapitre prochain aura pour but la modélisation de notre approche théorique de manière explicative.

Chapitre 2

Étude Conceptuelle

2.1 Introduction

Afin de concevoir l'application qui nous aidera à résoudre la problématique de la gestion manuelle des écoles primaires du système éducatif algérien, il est nécessaire de passer par une modélisation UML du fonctionnement des écoles primaires.

2.2 Modélisation UML

Le langage UML (Unified Modeling Language) est un langage de modélisation graphique semi-formel aidant à visualiser le système et à spécifier la structure de ses données, permettant ainsi de mieux comprendre son mécanisme. [6]

2.3 Identification des acteurs

Dans notre cas, la directrice de l'école représente l'utilisateur, c'est donc le seul acteur. En effet, elle a les droits d'accès à la base de donnée et peut ainsi administrer la totalité du système.

Diagramme des cas d'utilisation**Définition**

Il met en évidence les interactions fonctionnelles entre l'utilisateur et le système, il est utilisé dans les spécifications des besoins. [3]

Il est à noter que la description des diagrammes suivants a été faite dans la partie analyse des besoins fonctionnel du premier chapitre.

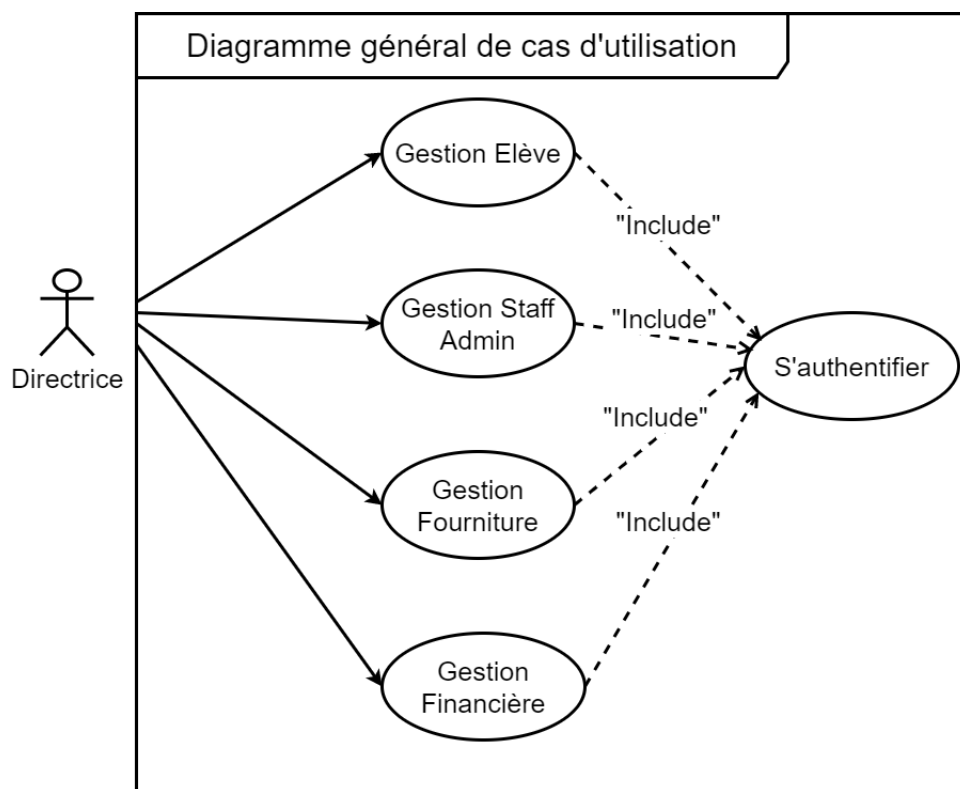
Diagramme général

FIGURE 2.1 – Diagramme de cas d'utilisation général

Diagramme de la gestion des élèves

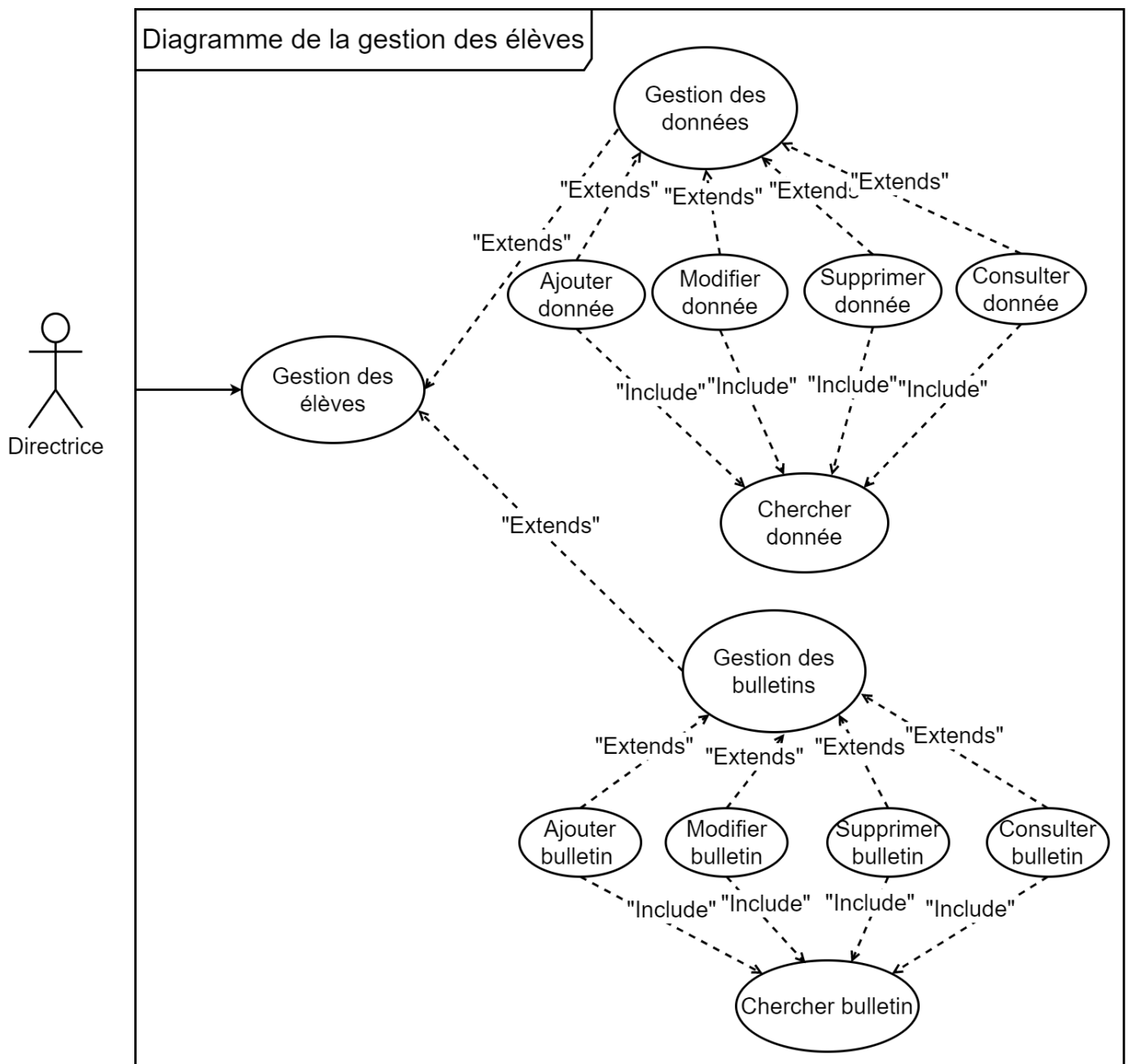


FIGURE 2.2 – Diagramme de cas d'utilisation de la gestion des élèves

Diagramme de cas d'utilisation de la gestion du staff administratif

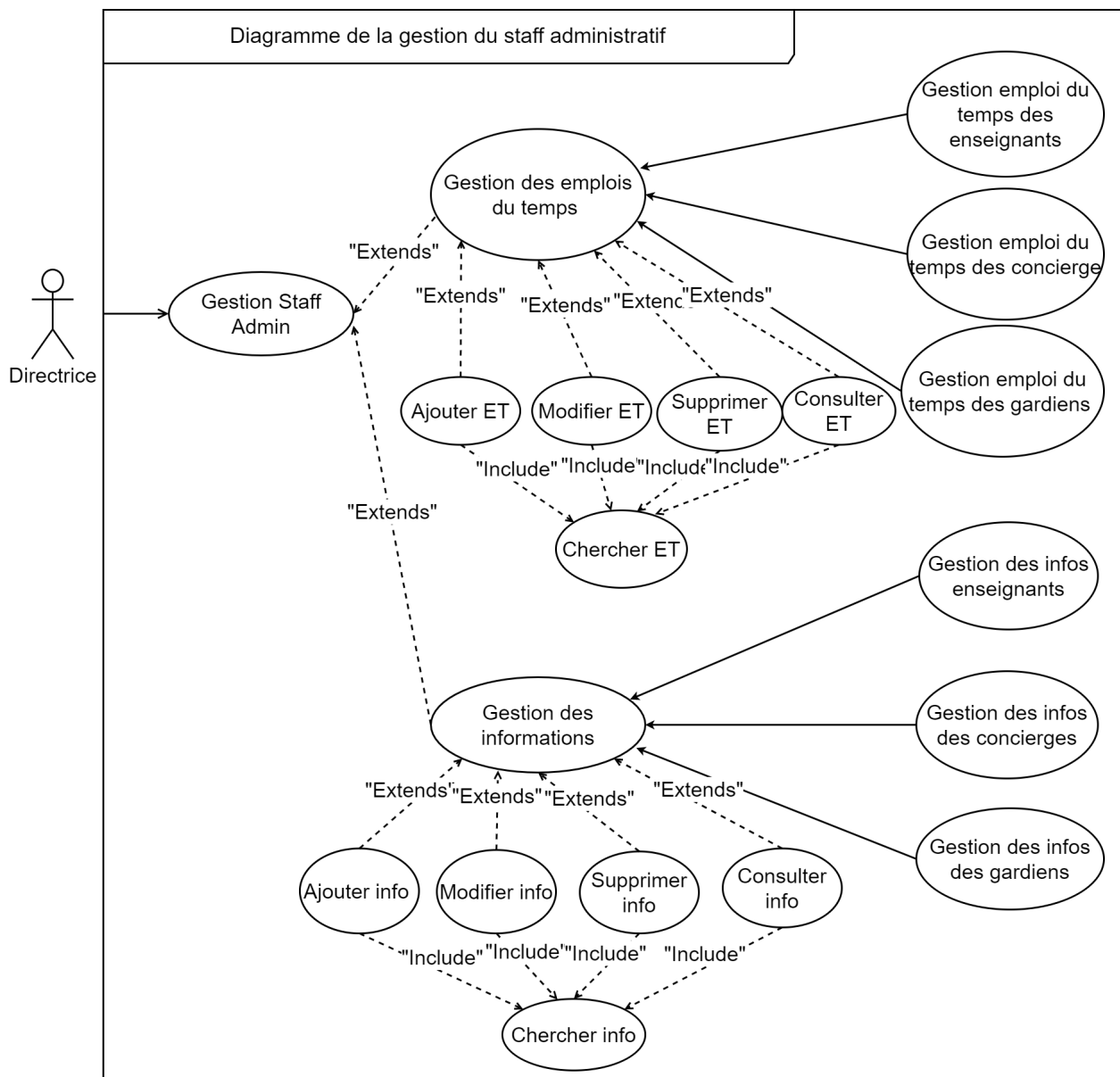


FIGURE 2.3 – Diagramme de cas de la gestion du staff administratif

Diagramme de la gestion des fournitures

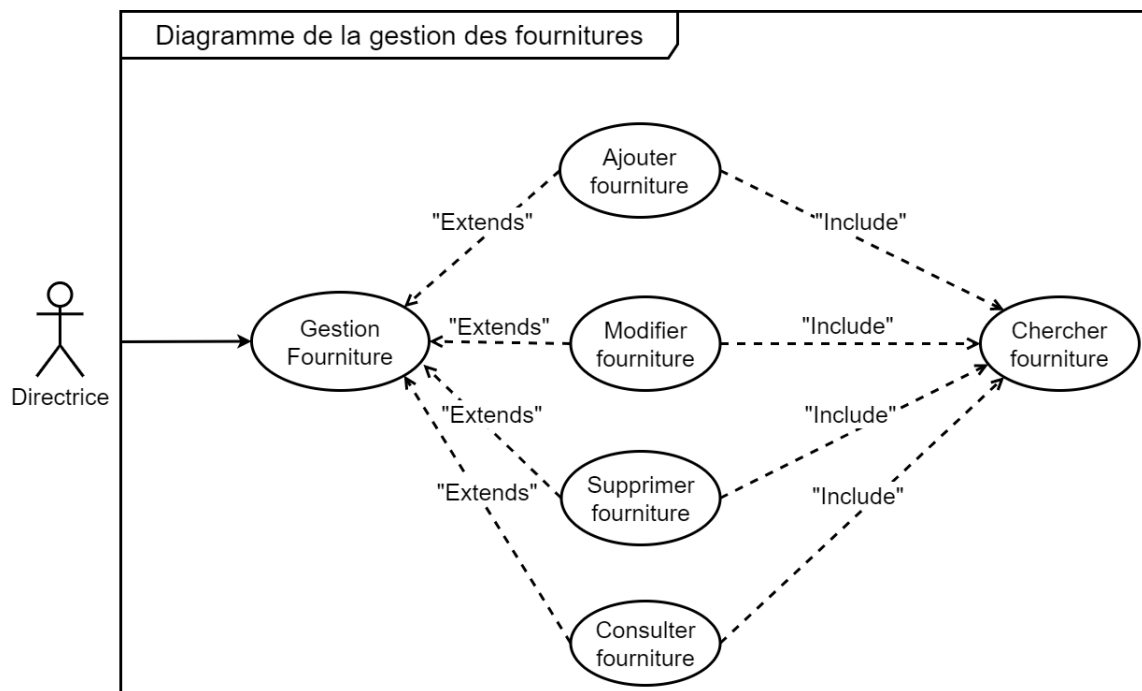


FIGURE 2.4 – Diagramme de cas d'utilisation de la gestion des fournitures

Diagramme de la gestion financière

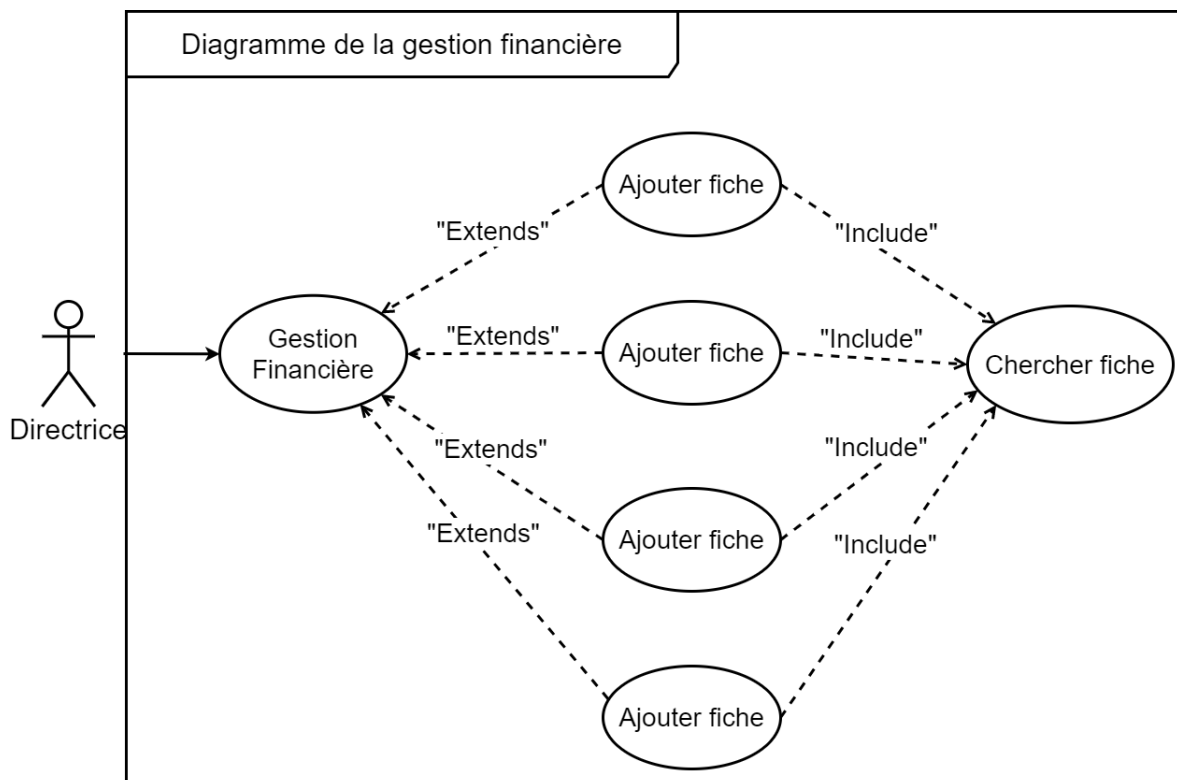


FIGURE 2.5 – Diagramme de cas d'utilisation de la gestion financière

Description du diagramme

- École : représente la classe mère du système, elle est gérée par une seule directrice et elle est composée de :
 - Un ou plusieurs salariés.
 - Une ou plusieurs zones qui représentent une salle ou des salles.
 - Une ou plusieurs classes.
 - Un ou plusieurs élèves.
- Classe : représente l'entité classe de cours qui est tenue par un enseignant responsable et qui englobe un ou plusieurs élèves. La répartition de chaque classe reste flexible (1 à N groupes).
- Élève : Chaque élève appartient à une seule école, à une seule classe et possède 1 à 3 bulletins durant son année scolaire. On peut ajouter, modifier, supprimer et/ou consulter les données d'un élève.
- Bulletin : Concerne 1 ou plusieurs élèves et est rédigé par la directrice. On peut ajouter, modifier, supprimer et/ou consulter les bulletins qui regroupent les notes des élèves.
- Directrice : Gère une seule école et est responsable de :
 - Rédiger un ou plusieurs bulletins.
 - Rédiger un ou plusieurs documents administratifs.
 - Gère une ou plusieurs fournitures.
- Salarié : est une classe mère qui regroupe les employés ; enseignants, gardiens et concierges. Chaque salarié appartient à une seule école et est concerné par un ou plusieurs documents administratifs. On peut ajouter, modifier, supprimer et/ou consulter les données d'un salarié ainsi que son emploi du temps.
- Enseignant : classe fille de la classe salarié. Un enseignant peut enseigner une ou plusieurs classes.
- Gardien : classe fille de la classe salarié. Un gardien est chargé de garder une seule école.
- Concierge : classe fille de la classe salarié. Une concierge est chargée de nettoyer une ou plusieurs zones (compartiments de l'école).

- Document Administratif : classe mère qui regroupe fiche de paie et emplois du temps. Un document est rédigé par la directrice et concerne un ou plusieurs salariés. On peut ajouter, modifier, supprimer et/ou consulter un document ou plusieurs documents.
- Fiche de paie : classe fille de document administratif, elle hérite des méthodes de cette dernière.
- Fiche d'info : classe fille de document. Une fiche d'information concerne un seul élève.
- Emplois du temps : classe fille de document administratif, elle hérite des méthodes de cette dernière.
- Fourniture : est gérée par la directrice.
- Zone : classe mère qui représente les différents compartiments de l'école ; WC, salles de classe et l'administration.
- WC : classe fille de zone.
- Salle de classe : classe fille de zone.
- Administration : classe fille de zone.

2.3.2 Diagramme de séquence

Définition du diagramme

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs, montrant les interactions entre les objets selon un point de vue temporel, tout en décrivant les scénarios [4].

Diagramme d'ajout d'un élève

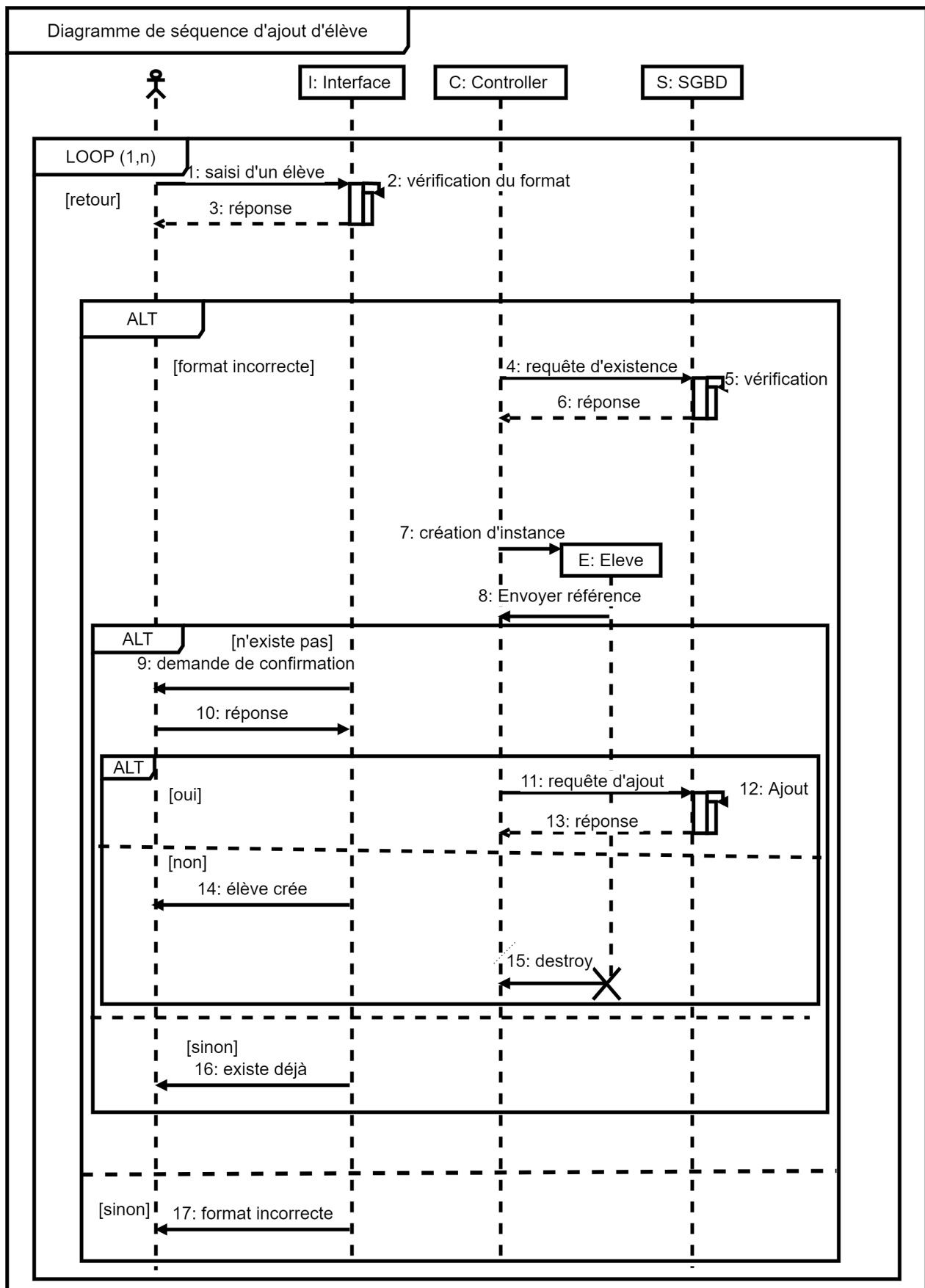


FIGURE 2.7 – Diagramme de séquence d'ajout d'un élève

Description du diagramme

L'utilisateur peut ajouter un nouvel élève en saisissant ses informations, une vérification du format des champs saisis sera faite par l'interface, dans le cas où c'est correct le contrôleur lancera une requête de demande de vérification de l'existence de l'élève ajouté au SGBD et il créera une instance d'un objet élève, qui, devra lui envoyer sa référence. Si l'élève n'existe pas, l'interface envoie une demande de confirmation de l'ajout à l'utilisateur, si ce dernier confirme, le contrôleur va envoyer une requête d'ajout au SGBD qui sauvegardera les informations de l'élève, l'interface informera l'utilisateur que ce dernier a bien été ajouté. Dans le cas où l'utilisateur ne confirme pas l'ajout, l'instance de l'objet élève sera détruite.

Si l'élève existe déjà ou le format des champs sont incorrectes, l'interface informera l'utilisateur de ces deux cas d'erreurs.

L'utilisateur peut ajouter un ou plusieurs élèves, le traitement sera le même à chaque ajout.

Diagramme de modification d'un élève

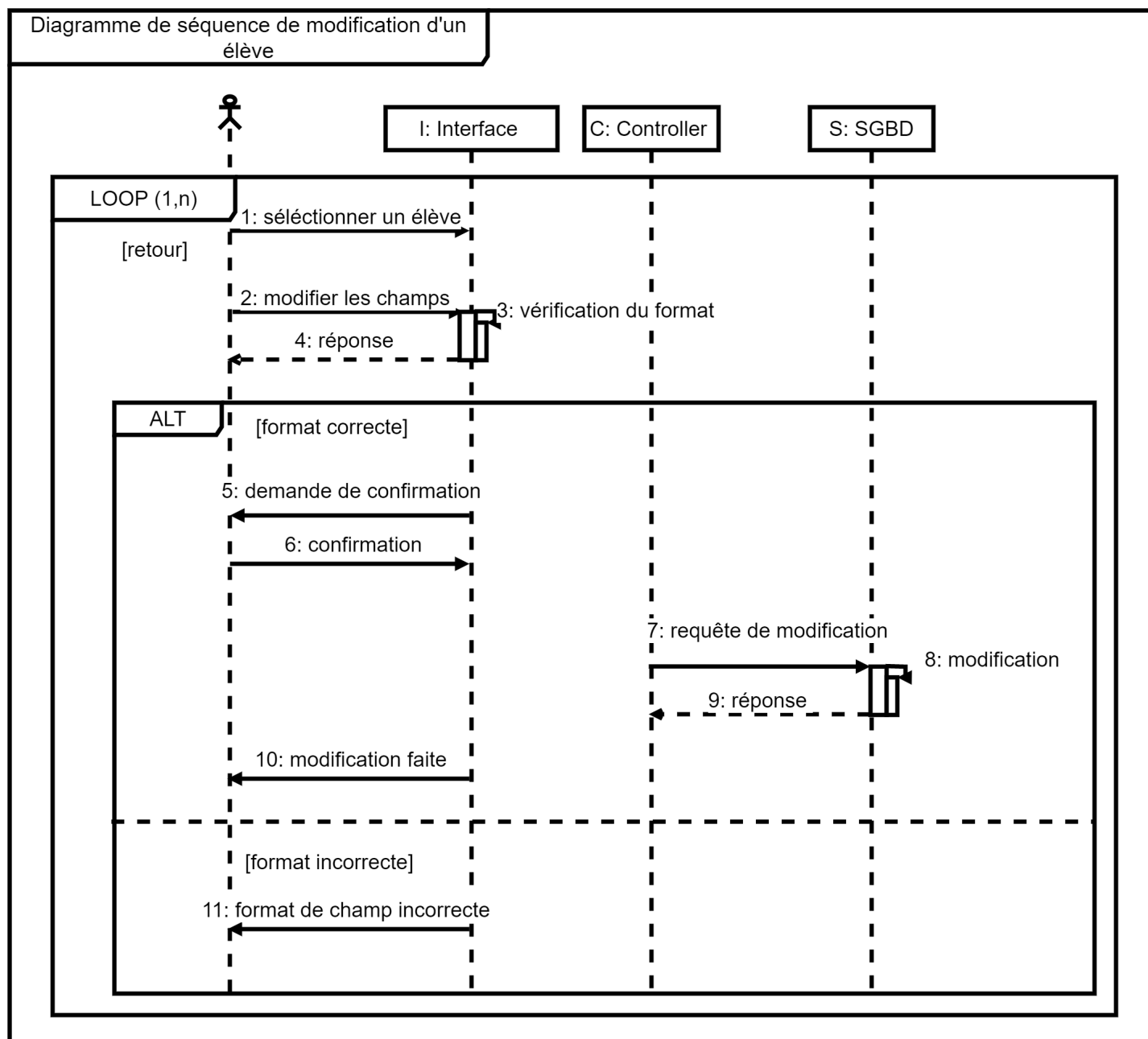


FIGURE 2.8 – Diagramme de séquence de modification d'un élève

Description du diagramme

Afin de modifier les informations personnelles de l'élève, il faut le sélectionner, une fois que la saisie des nouvelles informations sera faite, l'interface vérifiera le format des champs. Si le format est correcte l'utilisateur doit confirmer sa modification, le contrôleur va envoyer une requête de demande de modification au SGBD qui modifiera les informations de l'élève sélectionné. Une fois que la mise à jour a été effectuée, l'interface informera l'utilisateur que la modification a été faite avec succès. Dans le cas où le format des champs n'est pas correct un message d'erreur sera envoyé à l'utilisateur par l'interface.

L'utilisateur pourra modifier les informations d'un ou plusieurs élèves autant de fois qu'il le désire.

Diagramme de connexion

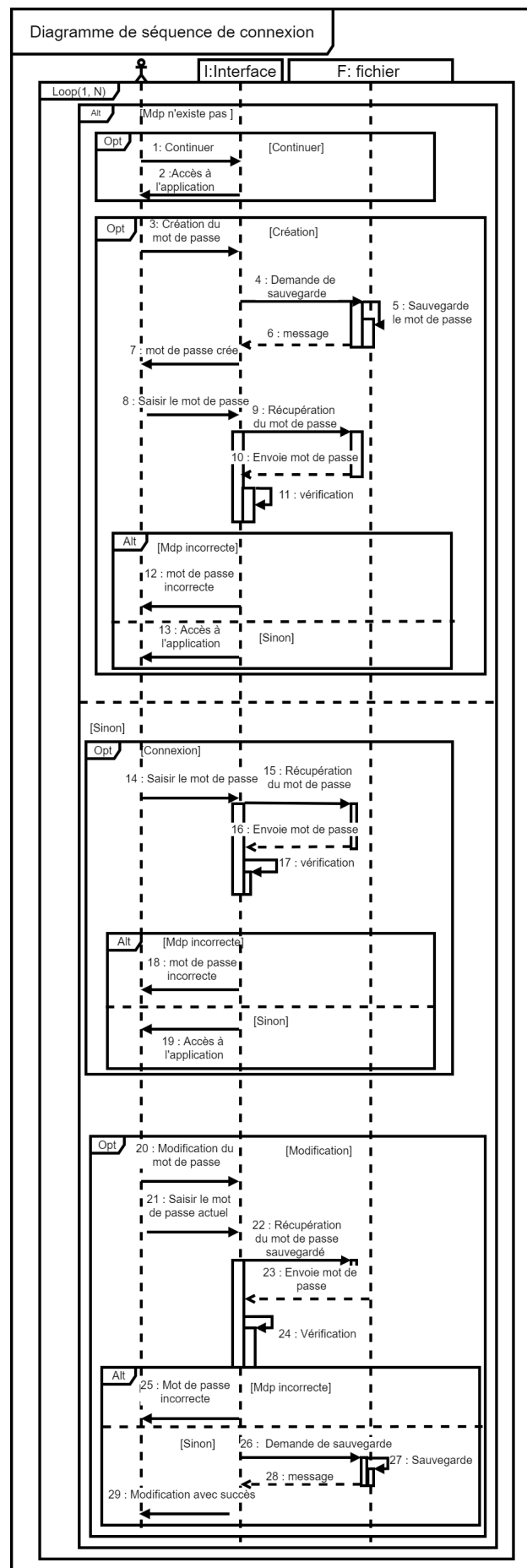


FIGURE 2.9 – Diagramme de séquence de connexion

Description du diagramme

Pour la connexion à l'application, il existe deux cas ; mot de passe inexistant mot de passe existant. Dans le premier cas l'utilisateur pourra continuer sans se connecter, il pourra donc accéder directement à l'application ou bien créer un mot de passe, l'interface va envoyer une demande de sauvegarde au fichier de configuration, ce dernier enregistrera le mot de passe saisi par l'utilisateur qui se connectera ensuite, l'interface lancera une demande de récupération du mot de passe au fichier, une fois reçu, elle vérifiera s'il est correct ou pas, elle donnera l'accès à l'application à l'utilisateur s'il est correct sinon elle envoie un message d'erreur.

Dans le deuxième cas, l'utilisateur peut se connecter (le traitement est le même que celui cité avant) ou modifier son mot de passe, pour cela il faut saisir le mot de passe actuel s'il est correct l'utilisateur pourra entrer un nouveau mot de passe, ce dernier sera enregistré dans le fichier sous la demande de l'interface.

2.3.3 Schéma relationnel**Définition**

Notre système exploite une base de données relationnelles, ce qui implique la description des différentes tables et les relations entre elles.

Notions de base

- La clé primaire qui est unique, est soulignée dans le schéma.
- La clé étrangère qui représente la clé primaire d'une autre table, est suivie par une étoile * dans le schéma.

Description

- Classe (N, IdC, NomE, PrenomE)
- Enseignant (NumE, NomE, PrenomE, Date_Naiss_E, Lieu_Naiss_E , Cadre, StatutE , Matiere_E, Grade, Date_Debut, Diplome)

- Gardien (NumG, NomG ,PrenomG ,StatutG ,Heure_Deb, Heure_Fin)
- Concierge (NumC, NomC, PrenomC ,StatutC)
- Salle (IdS, Nom_Salle)
- Eleve (NumL, IdL, NomL , PrenomL, Date_Naiss_L ,Lieu_Naiss_L ,Annee, Groupe)
- Bulletin (id_bulletin, id_eleve*, nom_Eleve, prenom_Eleve, annee, trimestre, note_Arabe, note_Math, note_Edu_civile, note_Edu_Islamique, note_Edu_Scientifique, note_His_Geo, note_Francais)
- Fourniture (idFourniture, Type_F, Ref_F, Qte_F)
- Emploi_concierge (id_emploi_con, jour_con, num_con*, nom_con, heure_debut_emploi_con, heure_fin_con)
- Emploi_enseignant (idFourniture, Type_F, Ref_F, Qte_F)
- Emploi_concierge (id_emploi_ens, jour_ens, num_ens*, nom_ens, heure_debut_emploi_ens, classe_ens, heure_fin_ens)
- Emploi_gardien (id_emploi_gar, jour_gar, num_gar*, nom_gar, heure_debut_emploi_gar, heure_fin_gar)
- Prog_concierge (idprog_concierge, zone_affectee*, Num_C*)
- fiche_info (id_fiche, id_eleve, nom_pere, prenom_pere, fonction_pere, nom_mere, prenom_mere, fonction_mere, adresse, num_tel_parents)

2.4 Conclusion

Dans un premier temps, nous avons défini le langage UML.

Ensuite, nous avons illustré les diagrammes de cas d'utilisation, le diagramme de classe en donnant une description explicative qui présente les attributs et les méthodes de chaque classe, et enfin, les diagrammes de séquence pour quelques-unes des principales opérations. Nous avons en outre présenté le schéma relationnel.

Le prochain chapitre présentera l'implémentation de notre système.

Chapitre 3

Implémentations et tests

3.1 Introduction

Ce chapitre présentera la réalisation du projet. Dans un premier temps nous allons exposer les différents outils utilisés qui regroupent les outils matériels et logiciels, ainsi que le langage de développement et les modules. Puis nous allons présenter l'architecture modèle view controller. Enfin nous allons parcourir quelques scénarios d'exécution.

3.2 Outils et langage de développement

3.2.1 Outils utilisés

Outils matériels

Pour réaliser notre projet, nous avons utilisé :

- La machine lenovo ThinkPad T480s 8GB, intel core i5 vpro 8th Gen.
- La machine lenovo Thinkpad T460 8GB, intel core i5 vpro 6th Gen.

Outils logiciels

- Visual Studio Code : est un éditeur de texte de code extensible, multi plate-forme, open source et gratuit, supportant une multitude de langages [15].
- Java Development Kit 16 : est un environnement de développement logiciel utilisé pour développer des applications. Il comprend le Java Runtime Environment (JRE), un interpréteur/chargeur (java), un compilateur

(javac), un archiveur (jar), un générateur de documentation (javadoc) et d'autres outils nécessaires au développement Java [14].

- SceneBuilder : est un logiciel de type outil interactif permettant de créer des interfaces graphiques FXML [13].
- Overleaf : est un éditeur et compilateur Latex en ligne [1].
- Draw.io : est un logiciel en ligne permettant de créer tous types de diagrammes [8].

3.2.2 Langage de développement

- Java : est un langage de programmation open source orienté objet, fortement inspiré des langages C et C++. [18].
- FXML : est un format de données textuelles dérivé du format XML, qui permet de décrire une interface utilisateur pour des applications conçues avec JavaFX [12].
- CSS : Cascading Style Sheet est un langage de programmation servant à décrire la présentation des documents HTML et XML [7].

Module externe

- JavaFX :est un framework et un ensemble de packages graphiques et multimédias servant à concevoir, créer, tester, déboguer et déployer des applications clientes, qui fonctionnent de manière cohérente sur diverses plateformes. [20].

3.3 Système de gestion de base de donnée

Nous avons utilisé le logiciel MySql qui est une base de donnée relationnelles open source.

3.4 Architecture Model-View-Controller

Le modèle-vue-contrôleur est un motif d'architecture logicielle utilisé pour la programmation client/serveur d'interface graphique. plus précisément, la partie vue est représentée par les documents FXML, les contrôleurs et les modèles sont des classes java mise respectivement dans les packages "intefaces" et "model".
[9]

3.5 Présentation de l'application

3.5.1 Interfaces graphiques



FIGURE 3.1 – Interface de connexion

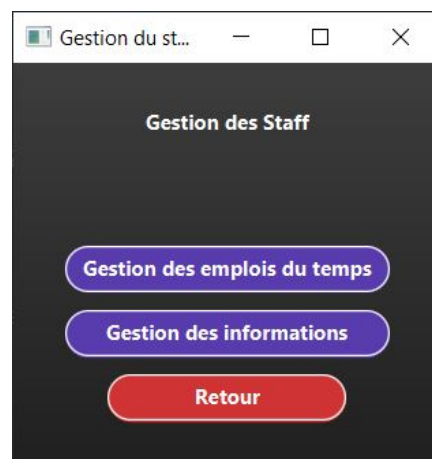


FIGURE 3.2 – Interface de gestion de staff



FIGURE 3.3 – Interface de gestion d’emplois du temps

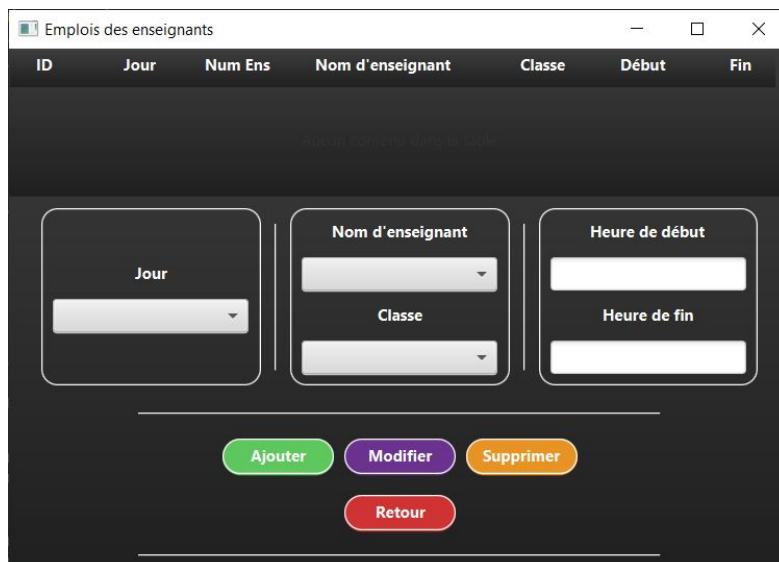


FIGURE 3.4 – Interface d’emplois du temps des enseignants

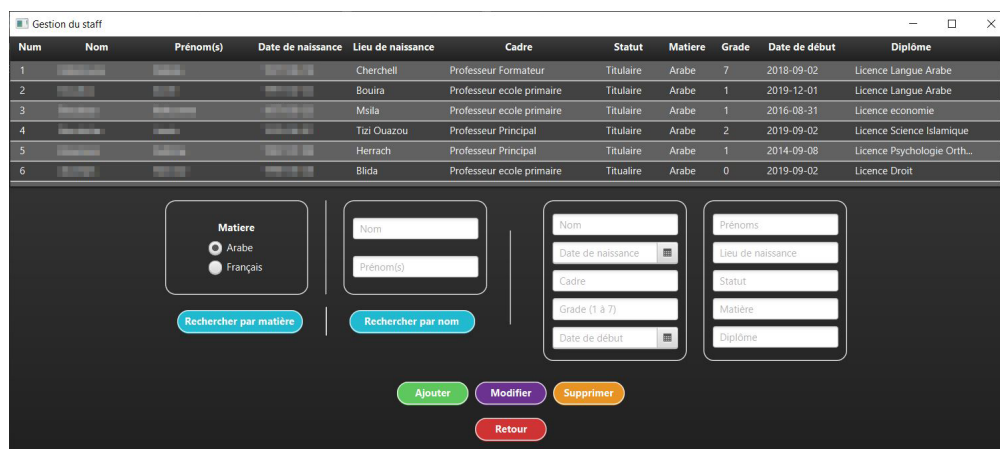
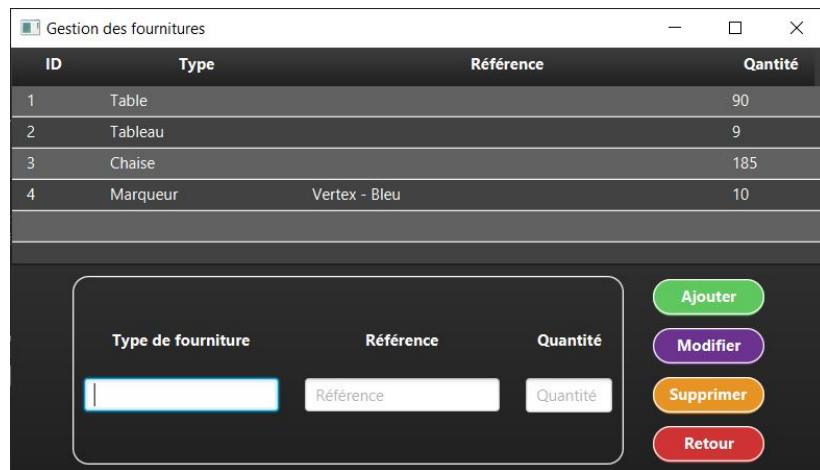


FIGURE 3.5 – Interface d’ajout, suppression et modification des informations des enseignants

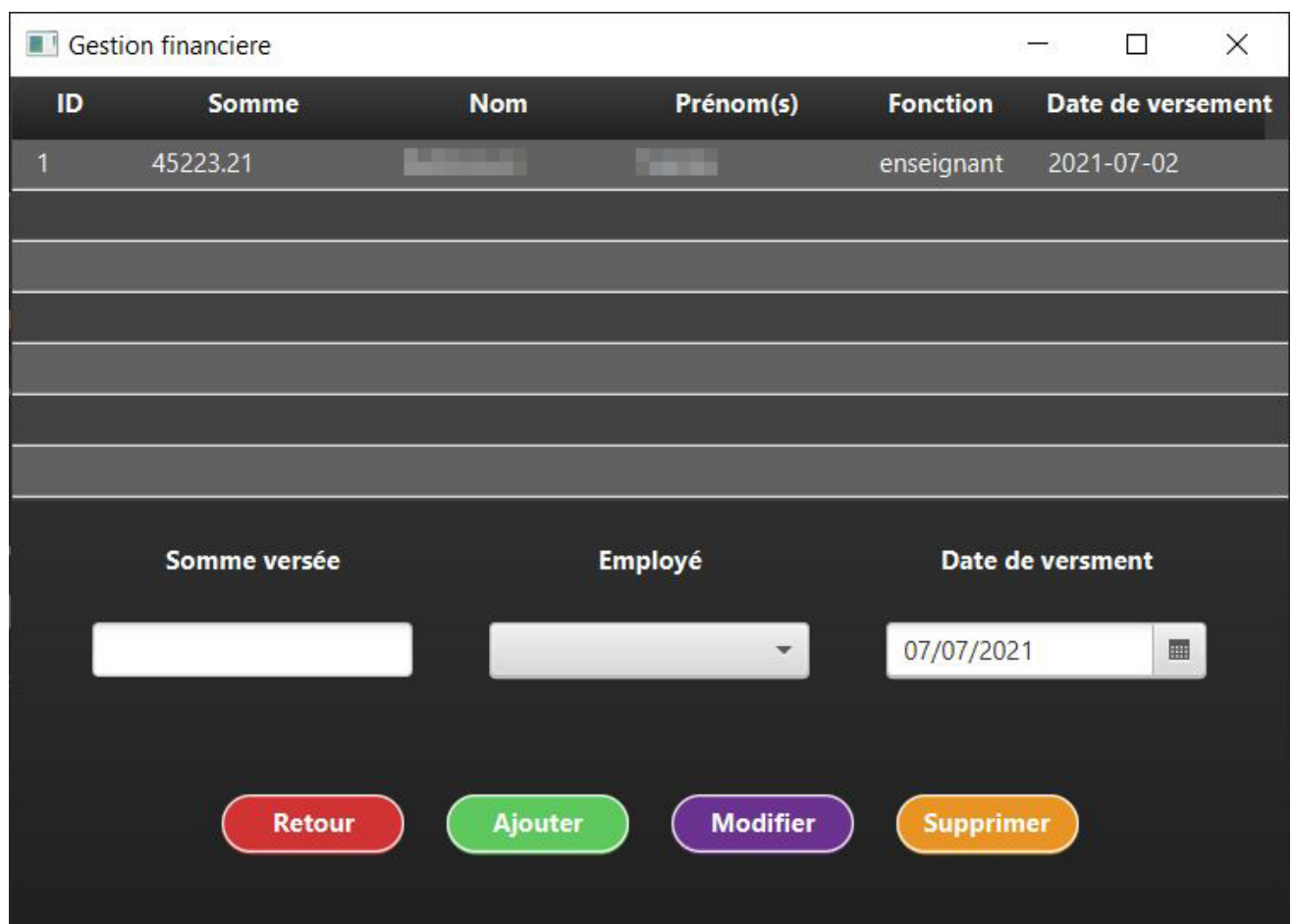


ID	Type	Référence	Quantité
1	Table		90
2	Tableau		9
3	Chaise		185
4	Marqueur	Vertex - Bleu	10

Form fields: Type de fourniture, Référence, Quantité

Buttons: Ajouter, Modifier, Supprimer, Retour

FIGURE 3.6 – Interface de gestion des fournitures



ID	Somme	Nom	Prénom(s)	Fonction	Date de versement
1	45223.21			enseignant	2021-07-02

Form fields: Somme versée, Employé, Date de versement

Buttons: Retour, Ajouter, Modifier, Supprimer

FIGURE 3.7 – Interface de gestion financière

3.6 Conclusion

Dans ce chapitre nous avons présenté la réalisation de notre application de gestion. Pour ce faire, nous avons cité les différents outils et langages de programmation, ainsi que des exemples de scénarios d'exécution.

Conclusion général

Le principal objectif de ce projet est de réaliser une application permettant la gestion automatique et informatisée des différents aspects administratifs qui, jusqu'à présent, étaient gérés manuellement par le directeur ou directrice de l'établissement. Ces aspects sont ; la gestion des élèves, la gestion du staff, la gestion de l'inventaire de fournitures ainsi que la gestion financière.

Dans notre cas, nous avons décidé de contacter une école primaire pour nous permettre de mieux comprendre le fonctionnement de l'administration scolaire de niveau primaire, plus particulièrement la gestion des élèves et des effectifs salariés.

Premièrement, nous avons exposé les principales fonctionnalités de l'application de manière théorique ainsi qu'un brève étude de l'existant. Ensuite, on a présenté la conception des fonctionnalités en utilisant le langage UML. Enfin, nous avons conclu ce rapport en présentant les outils de production utilisés lors de la réalisation de ce projet et en démontrant quelques exemples d'exécution.

L'application pourrait évoluer vers une version plus aboutie et diversifié en proposant plus de services et d'options comme par exemple ; création d'un archive permettant d'avoir accès à tous les élèves et salariés qui ne sont plus actifs dans l'école, établissement d'un plan de salle lors des examens de fin de cycle, et le passage automatique basé sur les notes de chaque élève en fin d'année.

Bibliographie

- [1] *About us*. URL : <https://www.overleaf.com/about>.
- [2] BERKANI. “Cycle de vie des logiciels”. In : (2020).
- [3] BERKANI. “Diagramme de cas d’utilisation”. In : (2020).
- [4] BERKANI. “Diagramme de séquence”. In : (2020).
- [5] BERKANI. “Diagramme des classes”. In : (2020).
- [6] BERKANI. “Langage UML”. In : (2020).
- [7] *CSS : qu’est-ce que c’est ?* URL : <https://www.futura-sciences.com/tech/definitions/internet-css-4050/>.
- [8] *draw.io*. 2020. URL : <https://www.computerhope.com/jargon/d/drawio.htm#:~:text=Designed%20by%20Seibert%20Media%2C%20draw,for%20making%20diagrams%20and%20charts.&text=They%20have%20a%20large%20selection,great%20looking%20diagram%20or%20chart>.
- [9] F. GAYRAL. “Architecture modèle-vue-contrôleur (MVC)”. In : (). URL : <https://lipn.univ-paris13.fr/~gayral/CoursGraphique/coursMVC.pdf>.
- [10] *Informatisé*. URL : <https://cnrtl.fr/definition/informatis%C3%A9>.
- [11] *interface utilisateur graphique*. URL : http://gdt.oqlf.gouv.qc.ca/fiche0qlf.aspx?Id_Fiche=2080261.
- [12] *Introduction to FXML*. URL : https://openjfx.io/javadoc/11/javafx.fxml/javafx/fxml/doc-files/introduction_to_fxml.html.
- [13] W. JACKSON. *Beginning Java 8 Games Development*. Apress, 2014. ISBN : 9781484204160. URL : <https://books.google.dz/books?id=RFXLoQEACAAJ>.
- [14] Paul KRILL. “JDK 16 : The new features in Java 16”. In : *InfoWorld* (2021). URL : <https://www.infoworld.com/article/3569150/jdk-16-the-new-features-in-java-16.html>.
- [15] Frederic LARDINOIS. “Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows”. In : *techcrunch* (2015). URL : <https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/>.
- [16] *Logiciel scolaire*. URL : <https://airmessoft.fr/logiciels/logiciel-scolaire/>.
- [17] *Qu’est-ce qu’une interface graphique ?* URL : https://www.c2i-revision.fr/complement.php?id_con=173.
- [18] *Qu’est-ce que la technologie Java et pourquoi en ai-je besoin ?* URL : https://www.java.com/fr/download/help/whatis_java.html.
- [19] *Qu’est-ce que l’informatique de gestion ?* URL : <https://www.computerland.fr/quest-ce-que-linformatique-de-gestion/>.

- [20] *What Is JavaFX?* URL : <https://docs.oracle.com/javafx/2/overview/jfxpub-overview.html>.