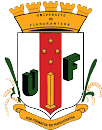
1. 

UNIVERSITÉ DE FIANARANTSOA

ECOLE DE MANAGEMENT ET D’INNOVATION TECHNOLOGIQUE

MINI-MÉMORE EN VUE DE PASSAGE EN TROISIÈME ANNÉE DE LICENCE

EN DÉVELOPPEMENT D’APPLICATION INTRANET ET INTERNET

Mention : Informatique

Conception et réalisation d’un système de gestion des courriers

Cas de la direction régional du budget Vakinankaratra

Présenté par : ANDRIANARIVELO Sitrakarinjaka Tovonirina

Sous la direction de : RAKOTOARISON Tsiorinantenaina René

Année universitaire : 2018-2019



**FORMATION**

2018-aujourd’hui En cours de Licence en développement d’application internet et intranet

Fianarantsoa*EMIT (Ecole de Management et d’Innovation Technologique)*

2017 Baccalauréat série C

Antsirabe*Gallo Junior*

2014  BEPC

Antsirabe *Les orchidées*

2010 CEPE

Antsirabe *Les orchidées*

Lot 0910 C 87 Mahafaly Vatofotsy

Tél : 034 93 605 22

Email: sitrakarinjakatovonirina@gmail.com

**Développeur informatique**

**ANDRIANARIVELO Sitrakarinjaka Tovonirina**

**EXPERIENCE PROFESSIONNELLE**

**CENTRES D’INTÉRÊT**

**Juillet 2018 – septembre 2019 EMIT**

***Fianarantsoa***

Tâches réalisées : Conception et réalisation d’un site e-commerce de location de vaisselle

**Logiciels:**Word, Excel, Powerpoint,Adobe Photoshop, Adobe illustrator.

**Languages:**Java, Python, C#, HTML 5, CSS 3, PHP, JavaScript.

**Frameworks:**React, React Native, Symfony.

Liste des figures

**Langues**  **Malagasy :**

***A*nglais :**

**Français :**

Voyager, Membre d’une association répondant au nom de SELMADA (Solidarité Etudiant Lyon-Madagascar).

Sport, hobbies: Basket-ball, velo, natation, …

[Figure 2.1. Les formations offertes en mention management niveau LICENCE 7](#_Toc127564077)

[Figure 2.2. Les formations offertes en mention management niveau MASTER 8](#_Toc127564078)

[Figure 2.3. Les formations offertes en mention RPM niveau LICENCE 9](#_Toc127564079)

[Figure 2.4. Les formations offertes en mention informatique niveau LICENCE 11](#_Toc127564080)

[Figure 2.5. Les formations offertes en mention informatique parcours SIGD niveau MASTER 12](#_Toc127564081)

[Figure 2.6. Les formations offertes en mention informatique parcours M2I niveau MASTER 13](#_Toc127564082)

[Figure 2.7. Organigramme de l’EMIT 14](#_Toc127564083)

[Figure 2.8. Organigramme du membre de bureau du présidence des étudiants de l’EMIT 18](#_Toc127564084)

[Figure 3.1. Organigramme du lycée privée « Les Orchidées » 23](#_Toc127564085)

[Figure 4.1. Comparaison des applications de bureau, web et mobile 46](#_Toc127564086)

[Figure 5.1. Cycle de développement selon la méthode Agile 51](#_Toc127564087)

[Figure 6.1. Illustration de l’organisation des inscription et embauche dans l’établissement 54](#_Toc127564088)

[Figure 6.2. Illustration de l’organisation des paiements de salaires au sein de l’établissement 55](#_Toc127564089)

[Figure 7.1. Diagramme de cas d’utilisation « Gestion des élèves » 67](#_Toc127564090)

[Figure 7.2. Diagramme du cas d’utilisation « Gestion cycle d’enseignement » 70](#_Toc127564091)

[Figure 7.3. Diagramme du cas d’utilisation « gestion du personnels administratifs et pédagogique » 72](#_Toc127564092)

[Figure 7.4. Diagramme d’activité du cas d’utilisation « Gestion élève » 76](#_Toc127564093)

[Figure 7.5. Diagramme d’activité du cas d’utilisation « ajouter cours » 77](#_Toc127564094)

[Figure 7.6. Diagramme de séquence du cas d’utilisation « Première connexion » 78](#_Toc127564095)

[Figure 7.7. Diagramme de séquence du cas d’utilisation « Paiement écolage » 79](#_Toc127564096)

[Figure 7.8. Diagramme de séquence du cas d’utilisation « Paiement salaires enseignants ». 80](#_Toc127564097)

[Figure 7.9. Diagramme de séquence du cas d’utilisation « Paiement salaire personnel » 81](#_Toc127564098)

[Figure 7.10. Diagramme de séquence du cas d’utilisation « ajouter une remarque à un élève » 82](#_Toc127564099)

[Figure 7.11. Diagramme de séquence du cas d’utilisation « ajouter historique sur un élève » 83](#_Toc127564100)

[Figure 7.12. Diagramme de packages du système 84](#_Toc127564101)

[Figure 7.13. Exemple d’un diagramme de classe 88](#_Toc127564102)

[Figure 7.14. Deux exemples de façons de modéliser une association. 88](#_Toc127564103)

[Figure 7.15. Diagramme de classe globale. 89](#_Toc127564104)

[Figure 9.1. Architecture MVC. 101](#_Toc127564105)

[Figure 9.2. figure d’un client lourd 102](#_Toc127564106)

[Figure 9.3. Code source pour le paiement de l’écolage 104](#_Toc127564107)

[Figure 9.4. Code source pour le paiement du salaire d’un enseignant 106](#_Toc127564108)

[Figure 10.1. Formulaire d’inscription des élèves 107](#_Toc127564109)

[Figure 10.2. Formulaire pour gérer du temps des différentes classes 108](#_Toc127564110)

[Figure 10.3. Formulaire lors de la première connexion 108](#_Toc127564111)

Liste des tableaux

[Tableau 5.1. Type de méthode de gestion de projet 49](#_Toc127563845)

[Tableau 6.1. Caractéristiques des ordinateurs au sein de l’établissement 56](#_Toc127563846)

[Tableau 6.2. Outils utilisés par l’établissement 56](#_Toc127563847)

[Tableau 6.3. Comparaison entre la solution de restructuration et celle de l’informatisation. 58](#_Toc127563848)

[Tableau 7.1. Description des données utilisées 66](#_Toc127563849)

[Tableau 7.2. Description du cas d’utilisation « lister les élèves » » 68](#_Toc127563850)

[Tableau 7.3. Description du cas d’utilisation « inscrire élève » 69](#_Toc127563851)

[Tableau 7.4. Description « Gérer cycle d’enseignement » 71](#_Toc127563852)

[Tableau 7.5. Description du cas d’utilisation « Créer cadre administratifs » 73](#_Toc127563853)

[Tableau 7.6. Description du cas d’utilisation « créer enseignant fixe » 74](#_Toc127563854)

[Tableau 7.7. Description du cas d’utilisation « Créer un enseignant chargé de cours » 75](#_Toc127563855)

[Tableau 8.1. Comparaison des outils de modélisation pour UML 94](#_Toc127563856)

[Tableau 8.2. Comparaison des différents langages 95](#_Toc127563857)

[Tableau 8.3. Comparaison des différent IDE de Java 97](#_Toc127563858)

[Tableau 8.4. Comparaison des technologies pour l’interface graphique en Java 97](#_Toc127563859)

[Tableau 8.5. Comparaison entre quelques SGBD relationnel. 98](#_Toc127563860)

Liste des abréviations

AN : Alpha Numérique

API: Application Programming Interface

CSS: Cascading StyleSheet

IDE: Integrated Development Environment

MVC: Model View Controller

N: Numeric

B: Boolean

SGBD : Système de Gestion des Bases de Données

UML: Unified Modeling Language

ORM: Object-Relational Mapping

POO : Programmation Orientée Objet

RPM : Relation Publique et Multiimédia

AES : Administration Economique et Sociale

SIGD : Système d’information Géomatique et Décisionnel

DA2I : Développement d’Application Inttranet et Internet

EMIT : Ecole de Management et d’Innovation Technologique

MERISE : Méthode d’Etude et de Réalisation Informatique pour les Systèmes d’Entreprise

DRB : Direction Régional du Budget

TABLE DES MATIERES

[Liste des figures ii](#_Toc127563969)

[Liste des tableaux iii](#_Toc127563970)

[Liste des abréviations iv](#_Toc127563971)

[TABLE DES MATIERES v](#_Toc127563972)

[AVANT-PROPOS ix](#_Toc127563973)

[REMERCIEMENTS x](#_Toc127563974)

[INTRODUCTION 1](#_Toc127563975)

[Partie II: Présentation générale 4](#_Toc127563976)

[Chapitre 2 : Présentation de l’EMIT 5](#_Toc127563977)

[2.1. Généralité 5](#_Toc127563978)

[2.2. Les formations 6](#_Toc127563979)

[2.2.1. SCIENCES DE LA SOCIETE 7](#_Toc127563980)

[2.2.2. SCIENCES ET TECHNOLOGIE 10](#_Toc127563981)

[2.3. Les personnels 14](#_Toc127563982)

[2.4. Les partenaires 15](#_Toc127563983)

[2.4.1. LABORATOIRES DE RECHERCHE 15](#_Toc127563984)

[2.4.2. ECOLES ET UNIVERSITÉS 15](#_Toc127563985)

[2.4.3. ENTREPRISES 16](#_Toc127563986)

[2.4.4. ORGANISMES PUBLIQUES 17](#_Toc127563987)

[2.5. Vie étudiante 17](#_Toc127563988)

[2.5.1. MEMBRES DE BUREAU 17](#_Toc127563989)

[2.5.2. ACTIVITÉS 18](#_Toc127563990)

[2.6. E.N.T. 19](#_Toc127563991)

[2.6.1. BIBLIOTHÈQUE NUMÉRIQUE 20](#_Toc127563992)

[2.6.2. FORMATION EN LIGNE 20](#_Toc127563993)

[Chapitre 3 : Présentation de l’Entreprise d’accueil 21](#_Toc127563994)

[3.1. Généralité 21](#_Toc127563995)

[3.1.1. Coordonnées de l’entreprise 21](#_Toc127563996)

[3.1.2. Historiques et performances passées 21](#_Toc127563997)

[3.1.3. Dirigeants 21](#_Toc127563998)

[3.1.4. Les locaux de l’entreprise 21](#_Toc127563999)

[3.1.5. L’emploi, la formation, le personnel 22](#_Toc127564000)

[3.1.6. Valeurs 22](#_Toc127564001)

[3.2. Organigramme 22](#_Toc127564002)

[Chapitre 4 : Présentation du projet 24](#_Toc127564003)

[4.1. Généralité 24](#_Toc127564004)

[4.2. Origine du projet 24](#_Toc127564005)

[4.3. Problématiques 24](#_Toc127564006)

[4.4. Cahier des charges 25](#_Toc127564007)

[4.4.1. Définition 25](#_Toc127564008)

[4.4.2. Introduction 25](#_Toc127564009)

[4.4.3. Description 25](#_Toc127564010)

[4.4.4. Formulation 42](#_Toc127564011)

[4.4.5. Objectifs du projet 42](#_Toc127564012)

[4.4.6. Enumération des besoins 43](#_Toc127564013)

[4.4.7. Fonctionnalités attendues 43](#_Toc127564014)

[4.4.8. Résultats attendus 44](#_Toc127564015)

[4.4.9. Améliorations proposées 44](#_Toc127564016)

[4.4.10. Type d’application à réalisé 44](#_Toc127564017)

[4.5. Conclusion 47](#_Toc127564018)

[Partie III: Analyse et Conception du projet 48](#_Toc127564019)

[Chapitre 5 : Méthodes et notations utilisées 49](#_Toc127564020)

[5.1. Introduction : 49](#_Toc127564021)

[5.2. Méthode de gestion de projet informatique adopté 49](#_Toc127564022)

[5.2.1. Méthode de développement de projet informatique 49](#_Toc127564023)

[5.2.2. Choix de la méthode Agile 50](#_Toc127564024)

[Chapitre 6 : Analyse du projet 52](#_Toc127564025)

[6.1.1. Analyse des besoins 52](#_Toc127564026)

[6.2. Analyse de l’existant 53](#_Toc127564027)

[6.2.1. Organigramme de fonctionnement 53](#_Toc127564028)

[6.2.2. Moyens matériels et logiciels 55](#_Toc127564029)

[6.3. Critique de l’existant 56](#_Toc127564030)

[6.3.1. Points forts 57](#_Toc127564031)

[6.3.2. Points faibles 57](#_Toc127564032)

[6.4. Solutions proposées 57](#_Toc127564033)

[6.4.1. La solution de restructuration 57](#_Toc127564034)

[6.4.2. Solution d’informatisation 57](#_Toc127564035)

[6.4.3. Comparaison de deux (02) solutions proposées 58](#_Toc127564036)

[6.5. Solution retenue 58](#_Toc127564037)

[Chapitre 7 : Conception du projet 59](#_Toc127564038)

[7.1. Définition de quelques concepts 59](#_Toc127564039)

[7.1.1. Etape conceptuelle 59](#_Toc127564040)

[7.1.2. Conception 59](#_Toc127564041)

[7.2. Identification des acteurs 59](#_Toc127564042)

[7.1. Dictionnaire de données 60](#_Toc127564043)

[7.2. Diagramme des cas d’utilisation 67](#_Toc127564044)

[7.2.1. Généralité 67](#_Toc127564045)

[7.2.2. Cas d’utilisation « Gestion des élèves » 67](#_Toc127564046)

[7.2.3. Cas d’utilisation « Gestion cycle d’enseignement » 70](#_Toc127564047)

[7.2.4. Cas d’utilisation « Gestion du personnels administratifs et pédagogique » 72](#_Toc127564048)

[7.3. Diagrammes d’activités 75](#_Toc127564049)

[7.3.1. Diagramme d’activité d’inscription d’étudiants 75](#_Toc127564050)

[7.3.1. Diagramme d’activité d’ajout d’un cours 77](#_Toc127564051)

[7.4. Diagramme de séquence 77](#_Toc127564052)

[7.5. Diagramme de paquetage 84](#_Toc127564053)

[7.6. Diagramme de classe 86](#_Toc127564054)

[7.6.1. Généralités 86](#_Toc127564055)

[7.6.2. Diagramme de classe globale 89](#_Toc127564056)

[7.7. Conclusion 90](#_Toc127564057)

[Partie IV: Réalisation du projet 91](#_Toc127564058)

[Chapitre 8 : Spécification des outils de réalisation 92](#_Toc127564059)

[8.1. Introduction 92](#_Toc127564060)

[8.2. Présentation des outils utilisés 92](#_Toc127564061)

[8.2.1. Le langage UML 92](#_Toc127564062)

[Chapitre 9 : Mise en œuvre et implémentation 100](#_Toc127564063)

[9.1. Architecture logicielle 100](#_Toc127564064)

[9.1.1. Architecture MVC (Model-View-Controller) 100](#_Toc127564065)

[9.2. Architecture matérielle 101](#_Toc127564066)

[9.2.1. Description 102](#_Toc127564067)

[9.3. Les différents modules de l’application 102](#_Toc127564068)

[9.3.1. Le paiement de l’écolage 103](#_Toc127564069)

[9.3.2. Le paiement des salaires des enseignants 105](#_Toc127564070)

[Chapitre 10 : Présentation de l’application développée 107](#_Toc127564071)

[10.2. Conclusion 109](#_Toc127564072)

[CONCLUSION GÉNÉRALE 110](#_Toc127564073)

[BIBLIOGRAPHIE 112](#_Toc127564074)

[RÉSUMÉ 113](#_Toc127564075)

[ABSTRACT 113](#_Toc127564076)

AVANT-PROPOS

Les Nouvelles Technologies de l'Information et de la Communication (NTIC) sont en constante évolution et leur utilisation dans le monde du travail est devenue indispensable. Cela se traduit par la conception et la maintenance de ces technologies ou encore par l'investissement des entreprises dans ces nouveaux outils. L'importance croissante des NTIC dans le monde du travail est indéniable et leur domaine d'application ne cesse de s'étendre. Toutefois, l'utilisation de ces outils nécessite une compréhension approfondie de leurs caractéristiques, de leurs avantages et de leur influence.

Dans ce contexte, la gestion des courriers est une fonction clé dans les organisations publiques et privées. La mise en place d'un système de gestion des courriers efficace peut contribuer à améliorer la productivité et la transparence de l'administration. Néanmoins, cette fonction peut également être une source de problèmes et de dysfonctionnements si elle n'est pas gérée de manière optimale.

L'objectif de ce mini-mémoire est d'étudier la conception et la réalisation d'un système de gestion des courriers pour la représentation régionale du Ministère des Finances et du Budget. Plus précisément, ce travail vise à répondre aux questions suivantes :

Comment concevoir et réaliser un système de gestion des courriers efficace dans un contexte organisationnel spécifique ?

Quels sont les outils et les techniques disponibles pour la conception et la réalisation d'un tel système ?

Quels sont les avantages et les limites de la mise en place d'un système de gestion des courriers dans une organisation publique ?

Comment évaluer l'impact d'un tel système sur la productivité, la transparence et la qualité de service de l'administration ?

Notre méthodologie consiste en une étude de cas appliquée à la représentation régionale du Ministère des Finances et du Budget. Cette étude de cas s'appuiera sur une analyse documentaire et des entretiens semi-directifs avec des acteurs clés de l'organisation. Les résultats de cette étude seront présentés et discutés dans la suite de ce travail.

On vise à fournir des informations et des recommandations pratiques pour la conception et la réalisation d'un système de gestion des courriers dans une organisation publique. Nous espérons que ce travail sera utile pour les praticiens de l'administration et pour les chercheurs intéressés par ce sujet.

REMERCIEMENTS

Nous remercions tout d’abord Dieu tout puissant de nous avoir permis de mener à terme ce projet qui présente pour nous le point de départ de notre carrière.

Il nous est agréable d’exprimer notre reconnaissance auprès de toutes les personnes, dont l’intervention au cours de ce projet, a favorisé son aboutissement.

Mes vifs remerciements s’adressent aussi à :

* Monsieur RAFAMANTANANTSOA Fontaine, Professeur, Président de l’université de Fianarantsoa de m’avoir permis d’effectuer mes études au sein l’Université ;
* Monsieur HAJALALAINA Aimé Richard, Docteur et Directeur de l’Ecole de Management et d’Innovation Technologique qui m’a donné l’opportunité d’effectuer mes études au sein de l’EMIT ;
* Madame **RABEZANAHARY Hobiniaina** responsable de la mention informatique en licence ;

Ainsi, nous tenons également à remercier tout le personnel du service régional du budget Vakinankaratra pour leurs précieuses directives et leurs conseils pertinents qui ont été d’un appui considérable au cours de ce projet.

INTRODUCTION GÉNÉRALE

La direction régional du budget dispose déjà de plusieurs solutions informatiques pour effectuer son travail efficacement, notamment grâce à un service informatique qui aide les gens dans leurs planifications financières et fiscales en utilisant des sites dédiés du ministère. Ce service forme les utilisateurs et crée des logins pour eux tout en les accompagnant dans le processus.

Cependant, malgré l'utilisation de ces outils, le service a identifié un besoin accru de fonctionnalités et d'outils de gestion pour répondre aux exigences croissantes de son travail. Il est donc nécessaire de développer une solution informatique sur mesure qui permettra au service d'optimiser ses activités et de fournir des services de qualité supérieure aux citoyens. Pour ce faire, le service dispose de plusieurs ordinateurs à la disposition des utilisateurs qui en ont besoin.

Notre objectif est de concevoir une solution qui aidera le service régional du budget à gérer plus efficacement leurs courriers, tout en améliorant l'expérience utilisateur et en offrant des fonctionnalités supplémentaires pour répondre aux besoins spécifiques du service.

La gestion des courriers est un élément essentiel de toute entreprise ou organisation, qu'il s'agisse de traiter les demandes des clients, de gérer la correspondance interne ou d'assurer le suivi des documents importants. Cependant, ce processus peut être fastidieux, chronophage et sujette aux erreurs humaines. C'est pourquoi la mise en place d'un système de gestion des courriers informatisé peut aider à simplifier et à rationaliser cette tâche. Dans ce texte, nous explorerons les étapes clés de la conception et de la réalisation d'un tel système, en nous concentrant sur les fonctionnalités clés, les avantages pour l'entreprise et les considérations techniques.

Notre projet de conception et réalisation d'un système de gestion des courriers sera organisé en trois grandes parties. Dans un premier temps, nous allons étudier le contexte général du projet pour mieux appréhender le cœur du travail, les besoins du client et les différentes fonctionnalités que notre application devra assurer. Nous exposerons également l'organisme d'accueil ainsi que notre établissement d'origine.

Par la suite, nous nous pencherons sur l'analyse des besoins fonctionnels du système futur. Nous commencerons par une étude préalable de l'existant afin de détailler les défaillances du système existant et de proposer des solutions adaptées. Nous concevrons ensuite les différents modules de notre projet en utilisant le Méthode d’Etude et de Réalisation Informatique pour les Systèmes d’Entreprise (MERISE).

Dans la troisième et dernière partie, nous nous concentrerons sur l'étude technique et la mise en œuvre de notre projet. Nous décrirons l'architecture technique, les outils et les langages utilisés pour la réalisation du nouveau système d'information. Nous expliquerons également la démarche de développement et illustrerons certaines fonctionnalités assurées par notre application à travers une présentation de l'application via des captures d'écran. Enfin, nous discuterons des résultats obtenus.

En conclusion, nous récapitulerons le travail accompli dans le cadre de ce projet de fin d'études. Nous présenterons les connaissances acquises au cours de ce projet ainsi que les possibilités à considérer pour raffiner davantage l'outil développé.

1. Présentation générale
   1. Présentation de l’EMIT
      1. Généralité

L'E.M.I.T. (Ecole de Management et d’Innovation Technologique) est une grande école publique au sein de l'Université de Fianarantsoa, qui a basculé totalement vers le système L.M.D. Elle se situe au cœur de l'Université de Fianarantsoa, à 3km de la gare routière sur la route qui mène vers Mahasoabe.

Anciennement connue sous le nom de « Centre Universitaire de Formation Professionnalisante » (CUFP), l’EMIT est un établissement public implanté au sein de l’Université de Fianarantsoa, avec le système LMD.

Notre établissement se démarque des autres par le fait que ses formations assurent aux étudiants une carrière professionnelle immédiate après le cursus universitaire.

En effet, à ce propos, bon nombre de sortants de l’Etablissement ont fait leurs preuves, occupant actuellement des postes importants, tant dans le secteur privé (des organisations et entreprises) que public.

L’Ecole dispose de huit (08) parcours répartis dans trois (03) mentions bien définies, à savoir le Management, l’Informatique et les Relations Publiques et Multimédia.

Au sein de l’EMIT, nous mettons également l’accent sur l’interaction des Enseignants-Chercheurs et des étudiants pour toutes recherches scientifiques et académiques afin d’atteindre des résultats positifs et satisfaisants, conformément aux normes et exigences internationales. Nos attentes ont été récompensées grâce à la collaboration avec des partenaires nationaux et internationaux (entreprises, laboratoires et universités).

* + 1. Les formations

E.M.I.T. vous propose huit parcours répartis dans trois mentions :

* Mention Management
  + Grade de Licence
    - [Administration Economique et Sociale (AES)](http://www.emit.mg/InnovationTechnologie/assets/img/AES-2.jpg)
  + Grade de Master
    - [Management Décisionnel (MD)](http://www.emit.mg/InnovationTechnologie/assets/img/MD-2.jpg)
* Mention Informatique
  + Grade de Licence
    - [Développement d'Application Internet/Intranet (DA2I)](http://www.emit.mg/InnovationTechnologie/assets/img/DAII-2.jpg)
    - [Conception, Intégration et Gestion des Systèmes d'Information (CIGSI)](http://www.emit.mg/InnovationTechnologie/assets/img/DAII-2.jpg)
  + Grade de Master
    - [Système d'Information, Géomatique et Décision (SIGD)](http://www.emit.mg/InnovationTechnologie/assets/img/SIGD-2.jpg)
    - [Modélisation et Ingénierie Informatique (M2I)](http://www.emit.mg/InnovationTechnologie/assets/img/M2I.jpg)
* Mention Relations Publiques et Multimédia
  + Grade de Licence
    - [Communication Multimédia (CM)](http://www.emit.mg/InnovationTechnologie/assets/img/RPM-2.jpg)
    - [Relations Publiques et Communication Organisationnelle (RPCO)](http://www.emit.mg/InnovationTechnologie/assets/img/RPM-2.jpg)
  + Grade de Master
    - [Communication Multimédia (CM)](http://www.emit.mg/InnovationTechnologie/assets/img/RPM-2.jpg)
    - [Relations Publiques et Communication Organisationnelle (RPCO)](http://www.emit.mg/InnovationTechnologie/assets/img/RPM-2.jpg)
      1. SCIENCES DE LA SOCIETE
* Mention : MANAGEMENT
* Parcours Licence
* ADMINISTRATION ECONOMIQUE ET SOCIALE
* Parcours Master
* MANAGEMENT DECISIONNEL

Les formations offertes en mention management niveau LICENCE



Les formations offertes en mention management niveau MASTER

* Mention : RELATIONS PUBLIQUES ET MULTIMEDIA
* Parcours Licence
* COMMUNICATION MULTIMEDIA
* RELATIONS PUBLIQUES ET COMMUNICATION ORGANISATIONNELLE

Les formations offertes en mention RPM niveau LICENCE

* + - 1. SCIENCES ET TECHNOLOGIE
* Mention : INFORMATIQUE
* Parcours Licence
* DEVELOPPEMENT D’APPLICATION
* CONCEPTION, INTEGRATION ET GESTION DES SYSTEMES D’INFORMATION
* Parcours Master
* SYSTEME D’INFORMATION, GEOMATIQUE ET DECISION INTERNET/INTRANET
* MODELISATION ET INGENIERIE INFORMATIQUE



Les formations offertes en mention informatique niveau LICENCE

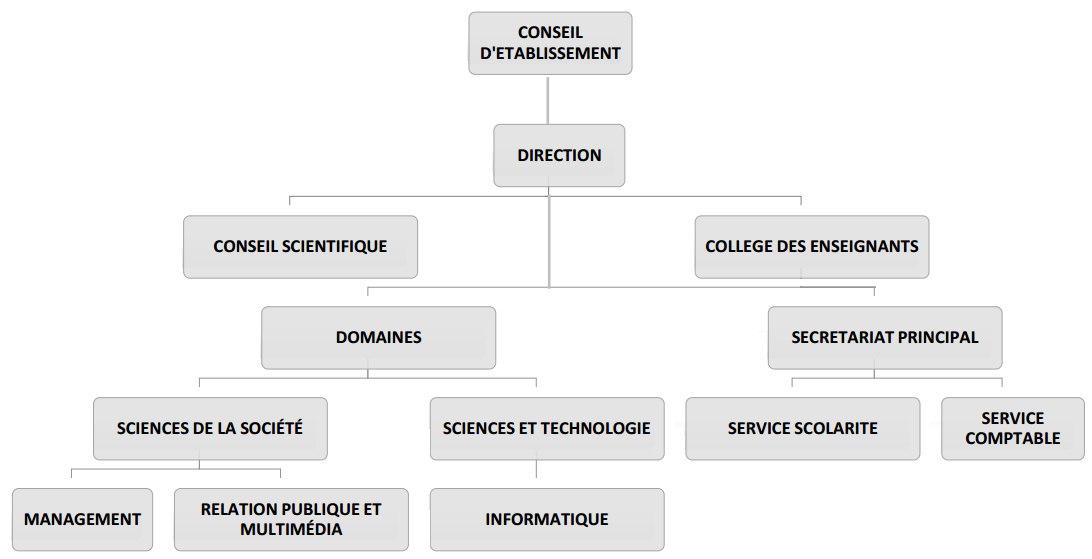


Les formations offertes en mention informatique parcours SIGD niveau MASTER



Les formations offertes en mention informatique parcours M2I niveau MASTER

* + 1. Les personnels

L’équipe pédagogique de l’Ecole compte actuellement 56 enseignants malgaches et étrangers qui se répartissent en Professeur titulaire, en Professeur, en Maître de conférences, en Maître assistant et en Professionnels, Consultants et Cadres supérieurs d'entreprise, Spécialistes des thématiques abordées. La formation est très ouverte à une participation des intervenants extérieurs. En raison de l’insuffisance des enseignants permanents, l’école a optée pour une politique d’ouverture envers d’autres établissements. La majorité des liens actuels se concentre sur des institutions de formation et de recherche locales. Actuellement, l’école est en train d’élargir ses relations vis-à-vis des établissements à l’international afin d’accueillir des étudiants rangers et de permettre à ces jeunes.

Organigramme de l’EMIT

* + 1. Les partenaires

L’objectif de l’école est actuellement de rechercher toutes les formes possibles de coopération, de favoriser le contact avec les différentes académies, d’inciter le monde de l’université et le monde de l’entreprise à se rencontrer, à mieux travailler ensemble et à s’enrichir mutuellement. Ces différentes collaborations permettront de valoriser les résultats des recherches, de profiter du savoir-faire des entreprises. Par ailleurs, ce partenariat permet également aux entreprises de s’impliquer dans les formations afin de permettre aux étudiants une vision plus large et approfondie.

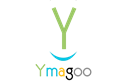
* + - 1. LABORATOIRES DE RECHERCHE

http://www.emit.mg/assets/img/partenaire/limad.png[http://www.emit.mg/assets/img/partenaire/ird.png](https://www.ird.fr/)[](http://www.cirad.fr/)[](http://www.espace-dev.fr/)http://www.emit.mg/assets/img/partenaire/cnre.png[](http://www.univ-antananarivo.mg/Laboratoire-des-Radio-Isotopes)[](https://limos.isima.fr/)

* + - 1. ECOLES ET UNIVERSITÉS

http://www.emit.mg/assets/img/partenaire/gouvsomu.png[](http://www.uca.fr/)[](http://www.esmia-mada.com/)http://www.emit.mg/assets/img/partenaire/isstm.pnghttp://www.emit.mg/assets/img/partenaire/lma.png

* + - 1. ENTREPRISES

[http://www.emit.mg/assets/img/partenaire/etech.png](https://www.etechconsulting-mg.com/)[](http://nelly-studio.ru/)[](http://www.manao.eu/)[](http://www.itdcmada.mg/)[](http://www.telma.mg/)[](https://www.orange.be/fr)[](http://www.agro-oi.com/)[http://www.emit.mg/assets/img/partenaire/star.png](http://www.star.fr/)[](https://www.alliancefr.org/)[](http://bfvsg.mg/)[http://www.emit.mg/assets/img/partenaire/bni.png](http://international.scholarvox.com/)[](https://www.premiya.mg/)[](http://www.ymagoo.com/)[](http://www.jirama.mg/)[](http://www.nyhavana.mg/)[](http://les-professionnels-de-madagascar.com/)[](http://www.aro.mg/)[](http://www.boa.mg/)[](http://international.scholarvox.com/)

* + - 1. ORGANISMES PUBLIQUES

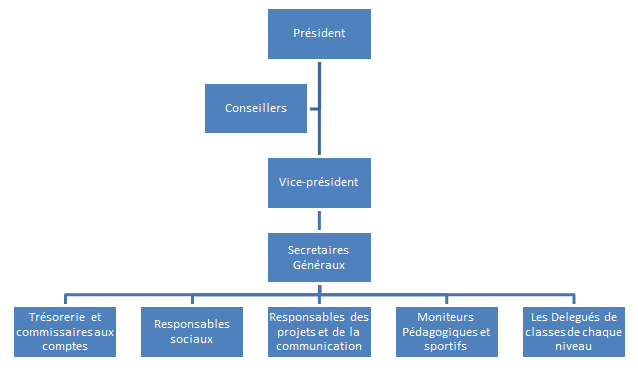
[](http://www.banque-centrale.mg/)[](http://www.meteomadagascar.mg/)[](https://www.instat.mg/)[http://www.emit.mg/assets/img/partenaire/ftm.gif](http://www.ftm.mg/)[](http://www.mefb.gov.mg/)[](http://www.education.gov.mg/)http://www.emit.mg/assets/img/partenaire/bng.png[](http://www.mtpm.gov.mg/)[](http://www.paositramalagasy.mg/)

* + 1. Vie étudiante

La vie associative des étudiants compose fondamentalement le projet pédagogique de l’école. Elle développe de véritables qualités humaines, organisationnelles et solidaires, en permettant ainsi aux étudiants :

* L’acquisition de l’expérience,
* La mise en pratique des théories enseignées en cours,
* L’amélioration de leurs habilités,
* L’élargissement de leur réseau de contact personnel,
* L’intégration dans la vie professionnelle.
  + - 1. MEMBRES DE BUREAU

Ce bureau est en charge de l’organisation de la relation de l’association avec l’administration de l’Ecole. Il est aussi en charge de l’amélioration de la vie étudiante en organisant des activités qui rehaussent l’esprit d’appartenance des étudiants et traduisent les valeurs de l’Ecole. Composé des représentants des étudiants, voici la structure formant cette entité.



Organigramme du membre de bureau du présidence des étudiants de l’EMIT

* + - 1. ACTIVITÉS
         1. Association sportive

« Dans un corps sain, l’esprit sain » dit-on.

Cette association sportive regroupe les passionnés du sport. Aspirants ou pratiquants, ils y développent ensemble l’esprit sportif et combatif tout en tenant compte du sens du travail d’équipe.

En visant l’équilibre entre l’intellectualité et le physique ces étudiants défendent et rendent fière l’Ecole dans des différents tournois internes (inter-niveaux) qu’externes (inter-établissement).

* + - * 1. Club de danse

Le club de danse de l’EMIT regroupe les amoureux de la danse de tout genre. Ici, les étudiants développent le sens du respect mutuel et du partage en équipe qui se matérialisent par des apprentissages et des démonstrations. Il performe donc le « temps danse » groupant amusement et développement personnel.

Les étudiants bougent autour de cette passion, au rythme des valeurs soufflées par l’Ecole.

* + - * 1. Club de musique

Ce club de musique regroupe les jeunes talents musicaux de l’EMIT. Il prône le sens de respect et de l’écoute mutuel en se communiquant autrement avec les instruments musicaux, harmonisant l’équilibre intellectuelle et culturelle des étudiants. Tout ceci en intégrant les grandes valeurs et idéologies de l’Ecole.

* + - * 1. Club des jeunes entrepreneurs

« Visons haut, entreprenons ensemble, car nous sommes l’avenir ! »

* Le CJE vise à développer l’esprit entrepreneurial des étudiants, basé sur le travail d’équipe.
* Formation, partage d’expérience et pratique entrepreneuriale composent la vie de ce club. Il développe le sens de leadership, de responsabilité et de créativité de chacun afin de garantir les valeurs de l’Ecole.
  + - * 1. English Club

« Let’s learn and perform together! »

L’English Club EMIT regroupe les passionnés de l’anglais. Avec diverses activités d’apprentissage et de pratique, il a pour but de promouvoir le développement de cette langue et la performance de la vie associative des étudiants.

* + - * 1. Jeux de Société

« L’échec n’est que le jeu, jouons avec et gagnons ! »

Ici, les étudiants se regroupent pour jouer et développer ensemble leur esprit créatif, compétitif et résolutif tout en s’amusant.

* + 1. E.N.T.

L'environnement numérique de travail (ENT) de l'EMIT permet aux enseignants et aux étudiants d'accéder aux services mis à disposition, depuis n'importe quel lieu et à n'importe quel moment. Il suffit de disposer d'un équipement connecté à l'internet.

* + - 1. BIBLIOTHÈQUE NUMÉRIQUE

L’école est également affiliée à une bibliothèque numérique communautaire multilingue scholarvox. Ceci étant, les étudiants et les enseignants ont accès en illimité à des trentaines de milliers d’ouvrages et d’articles spécialisés dans plusieurs disciplines.

Par ailleurs, l’université dispose de plusieurs places de lecture, dont des espaces informatiques, salles de travail ainsi qu’un espace de consultation spécifique aux fonds patrimoniaux.

* + - 1. FORMATION EN LIGNE

EMIT, cumul des avantages de la formation et de la recherche, et ce sont :

* Reconnaissance nationale du diplôme
* Adaptation à l’international de la formation
* Vaste réseau d’université partenaires
* Vaste réseau d’entreprises partenaires
* Des équipes enseignantes réputées pour leurs travaux de recherche et leur expertise dans leur discipline
* Des équipements pédagogiques modernes : espace numérique de travail, visioconférence,
  1. Présentation du service d’accueil
     1. Généralité
        1. Coordonnées du service

Le Lycée Privé Les Orchidées est, comme son nom l’indique, un lycée d’enseignement général situé dans la banlieue de la ville d’Antsirabe, lot 0910 A 101 Mahafaly, avec deux numéros d’autorisation d’ouverture distinct pour les niveaux primaires-collège et lycée, 036/2007-MENRS du 18 Juin 2007 pour le Collège et 046/2016-MEN du 05 Février 2016 pour le Lycée.

Téléphone : 033 12 333 09 - 034 11 333 09

Email : [lesorchideesabe@gmail.com](mailto:lesorchideesabe@gmail.com)

* + - 1. Historiques et performances passées

L’établissement scolaire a été créé dans l’année 1997 avec seulement les classes primaires. Ce n’était en 2006 que le niveau collège a été introduit.

La création du lycée s’est faite en 2013 avec seulement deux classes de seconde.

Depuis les premières promotions, on a un taux moyen de réussite à 87% pour le CEPE, 80% pour le BEPC et 62% pour le baccalauréat.

Pour cette année scolaire 2020-2021, avant le confinement dû à la pandémie du Covid-19, l’établissement compte environ 1000 élèves tous niveaux confondus.

* + - 1. Dirigeants

Le directeur qui est aussi fondateur et propriétaire s’appelle RANDRIANARIVONY Fanomezantsoa, certifié par l’ONG ADM Mada et ADM Italie en formation d’entreprenariat en 1996.

* + - 1. Les locaux de l’entreprise

On occupe une surface d’environ 3000m2 dans le Fokontany Mahafaly Vatofotsy et d’environ 5000m2 à Miaramasoandro, avec 6 immeubles dont 4 pour les salles de classe, 1 logement du directeur, 1 pour les bureaux d’administration, et un terrain de sport.

Les locaux sont propriétaires à l’établissement.

* + - 1. L’emploi, la formation, le personnel

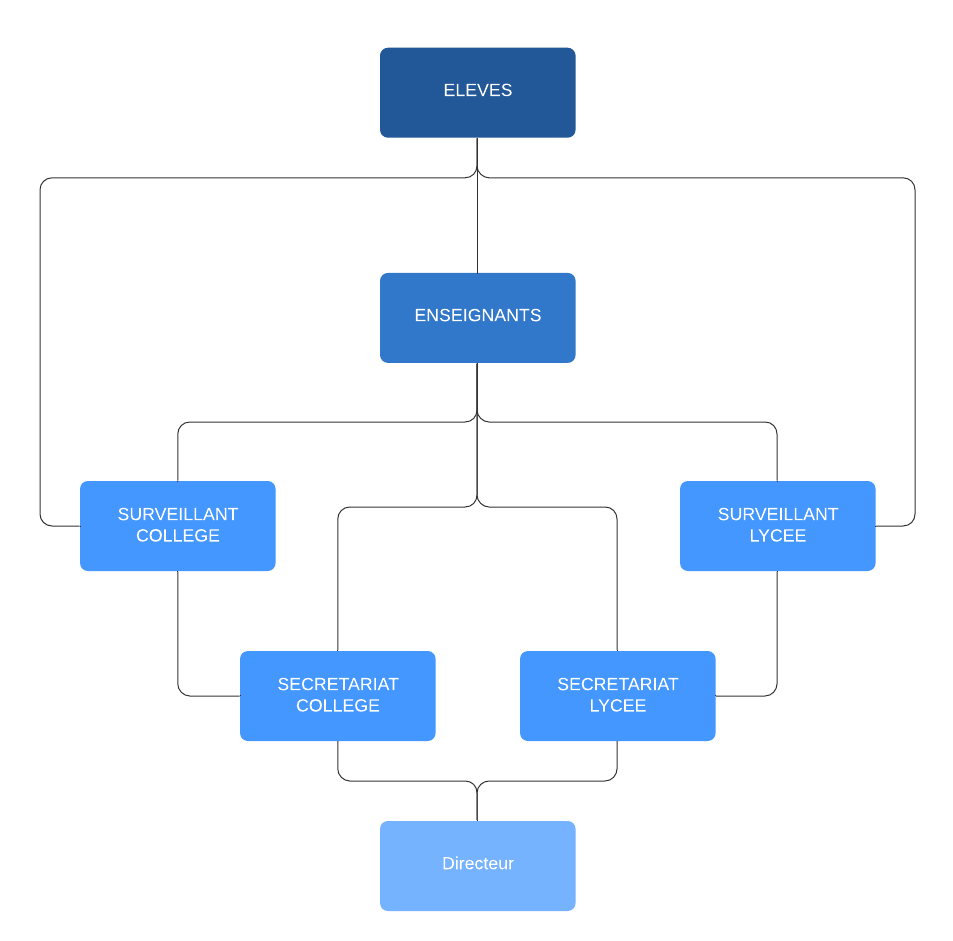
Le personnel compte 50 membres dont 5 personnels administratifs et 45 enseignants/institutrices.

* + - 1. Valeurs

Pour assurer le bon fonctionnement de cette institution, acquérir un avantage concurrentiel et gérer son prestige, elle doit avoir son propre ensemble de valeurs. Ces valeurs sont pertinentes pour Le lycée privée les orchidées :

* Le savoir : posséder des connaissances étendues et approfondies et une riche expérience.
* L’éthique : en agissant avec intégrité et principe, c'est-à-dire en adhérant aux normes professionnelles et en agissant de manière irréprochable ;
* L’excellence : accomplir la tâche au mieux de ses capacités, c'est-à-dire à un degré qui n'atteint pas l’idéal.
  + 1. Organigramme

Un organigramme est un diagramme qui décrit les liens et les interrelations opérationnels, organisationnels et hiérarchiques entre les éléments et les individus d'une organisation formelle (association, entreprise, réseau, etc.) ou d'un programme, et met en évidence sa structure organisationnelle. De ce fait, la figure suivante nous montre l’organigramme au sein du Lycée Privée les orchidées et nous permet d’apercevoir les différentes directions et postes en ce lieu.



Organigramme du lycée privée « Les Orchidées »

* 1. Présentation du projet
     1. Généralité

L'informatique est une discipline omniprésente qui a un impact significatif sur de nombreux aspects de la vie, tels que le social, l'économique, le juridique, etc.

Il y a une tendance générale à l'harmonisation des supports utilisés dans les entreprises qui proposent des services de distribution de courriers (tels que la fiche de suivi des courriers, le bordereau de dépôt, le relevé des clients, etc.) afin de pallier les problèmes fastidieux et répétitifs et de répondre aux besoins d'évaluation et de suivi des activités de l'organisation. Le souci d'optimiser les méthodes de gestion des clients en rationalisant leur dossier est également présent.

L'informatisation vas permettre de suivre de manière détaillée la gestion de chaque courrier, ce qui permettra aux responsables de prendre de bonnes décisions et d'offrir la possibilité de retrouver les traces ou les opérations accomplies, tout en générant des statistiques.

En ce qui concerne la distribution de courriers, l'informatique offre la possibilité de suivre les différents courriers vers les clients destinataires de manière simple et efficace via internet, sans être limité par des contraintes de temps ou de lieu.

En somme, l'informatique touche presque tous les secteurs de la vie grâce au progrès constant observé dans chacune de ses branches, notamment avec la présence d'un site web dans notre cas.

Avec tous les bienfaits de l’informatisation le responsable informatique a pris pour projet d'informatiser la gestion des courriers entrants, sortants et internes de la Direction Régionale du Budget dans l'optique d'améliorer la gestion de son service de courrier.

* + 1. Objectif et besoins de l’utilisateur
       1. Objectifs

Nous avons comme objectif de concevoir et de réaliser un système de gestion de courriers pour la Direction Régionale du Budget de Vakinankaratra. Ce système permettra d'optimiser la gestion des courriers entrants, sortants et internes en automatisant les différentes étapes du processus, de la réception à l'archivage en passant par la distribution et le suivi des courriers. Le système visera également à améliorer la traçabilité et la sécurité des échanges de courriers, en assurant notamment la confidentialité et l'intégrité des données. Enfin, notre objectif est de fournir un outil convivial et facile d'utilisation, qui permettra aux utilisateurs de gagner en efficacité et en productivité dans leur travail quotidien.

* + - 1. Enumération des besoins

Le tableau ci-dessous énumère les besoins selon le type de personne au sein du service :

|  |  |
| --- | --- |
| **Type d’utilisateur** | **Les besoins** |
| Administrateur | * Permettre la gestion des courriers. * Offrir la possibilité de visualiser les courriers créés en affichant l'identité de leur créateur. * Fournir une consultation des profils utilisateurs. * Permettre la gestion des utilisateurs ainsi que les tâches qui leur sont attribuées dans l'application. * Offrir une consultation du profil de l'utilisateur actif. * Permettre la gestion des directions. * Fournir un accès au tableau de bord pour consultation. |
| Secrétaire | * Permettre la création, la modification, la suppression et la recherche des courriers * Offrir la possibilité de consulter son profil * Permettre la consultation d'un courrier * Fournir un tableau de bord pour une vue d'ensemble de l'activité de gestion des courriers. * Recevoir et distribuer les courriers vers leurs destinataires au sein du service |
| Utilisateur Simple | * Consultation de son profil * Consultation des courriers * Suivie des traitements des courriers envoyé ou reçu |

Tableau démontrant les besoins selon le type d’utilisateur

* + 1. Moyens nécessaires à la réalisation du projet :
       1. Moyens humains

Les personnes qui ont contribué pour la bonne réalisation de ce projet sont :

* Un encadreur professionnel qui a suivi de près l’état d’avancement du projet à réaliser.
* Un stagiaire qui est à la fois le concepteur et le réalisateur du projet
  + - 1. Moyens matériels

En termes de matériel, nous avons utiliser un ordinateur portable dont les caractéristiques sont mentionnées dans le tableau ci-dessous :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Marque et modèle | Processeur | RAM | Stockage | Système d’exploitation |
| DELL LATITUDE E5430 | Intel Core i5 3320M 2.60GHz | 4Go ddr3 de fréquence 1333mhz | 256 Go en disque dur | Windows 10 Edition Professionnel Version 1909 |

Spécifications de l’ordinateur portable utilisé

* + - 1. Moyens logiciels

Pour bien parfaire ce projet, nous avons eu recours à l’utilisation de divers logiciels :

* VSCode : Pour le développement de l’application
* MySql Workbench : pour la modélisation de la base de données
* Google Chrome : pour les différentes recherche et test de l’application
  + 1. Résultats attendus

Le temps impartis ainsi terminer, nous devons fournir une application :

* Rapide et efficace
* Possédant toutes les fonctionnalités requises dans la gestion de tout type de courriers
* Fiable et sécurisé
* Facile à utiliser
* Permettant de suivre le traitement des courriers qui circule dans le service

1. Analyse et Conception du projet
   1. Méthodes et notations utilisées
      1. Introduction :

La conception est la partie la plus crucial dans la réalisation d’un projet informatique et c’est dans ce chapitre que nous allons détailler les différentes conceptions effectuées jusqu’à maintenant pour pouvoir répondre aux besoins des utilisateurs.

* + 1. Méthode de gestion de projet informatique adopté
       1. Méthode de développement de projet informatique
          1. Définition

Une approche de développement de projet informatique est une stratégie de gestion de projet qui aide et facilite les tâches de manière logique et ordonnée, du début à la fin du projet. De ce fait, la gestion de projet n'est rien sans organisation et procédure.

* + - * 1. Type de méthode de développement

Il existe de nombreuses méthodes, dont trois (03) sont particulièrement populaires, ces derniers sont les suivants :

* Modèle en cascade
* Méthode agile
* Méthode hybride

|  |  |
| --- | --- |
| Méthode | Fonctionnement |
| Modèle en cascade | Travail étape par étape |
| Méthode Agile | Amélioration continue |
| Méthode hybride | Combine méthode Agile et en cascade |

Type de méthode de gestion de projet

Dans notre cas, nous avons opté pour la méthode Agile en connaissant qu’elle a été conçue pour gestion de développement des applications en amélioration continue. Cette méthode aura expliqué la partie qui suit.

* + - 1. Choix de la méthode Agile
         1. Description de la méthode

Il est conçu pour être plus adaptable et flexible, avec les besoins du client au premier plan des priorités du projet.

Cette méthode a été initialement développée pour des projets de développement web et informatique. Parce qu'elle peut être utilisée pour un large éventail de projets dans de nombreux secteurs, la technique Agile devient de plus en plus populaire. Des boucles de rétroaction plus courtes résultent d'un travail itératif, ce qui facilite la modification des articles en fonction de la technologie actuelle et/ou des préférences des clients. De nombreux chefs de projets informatiques et développeurs l’utilisent pour parfaire leur travail.

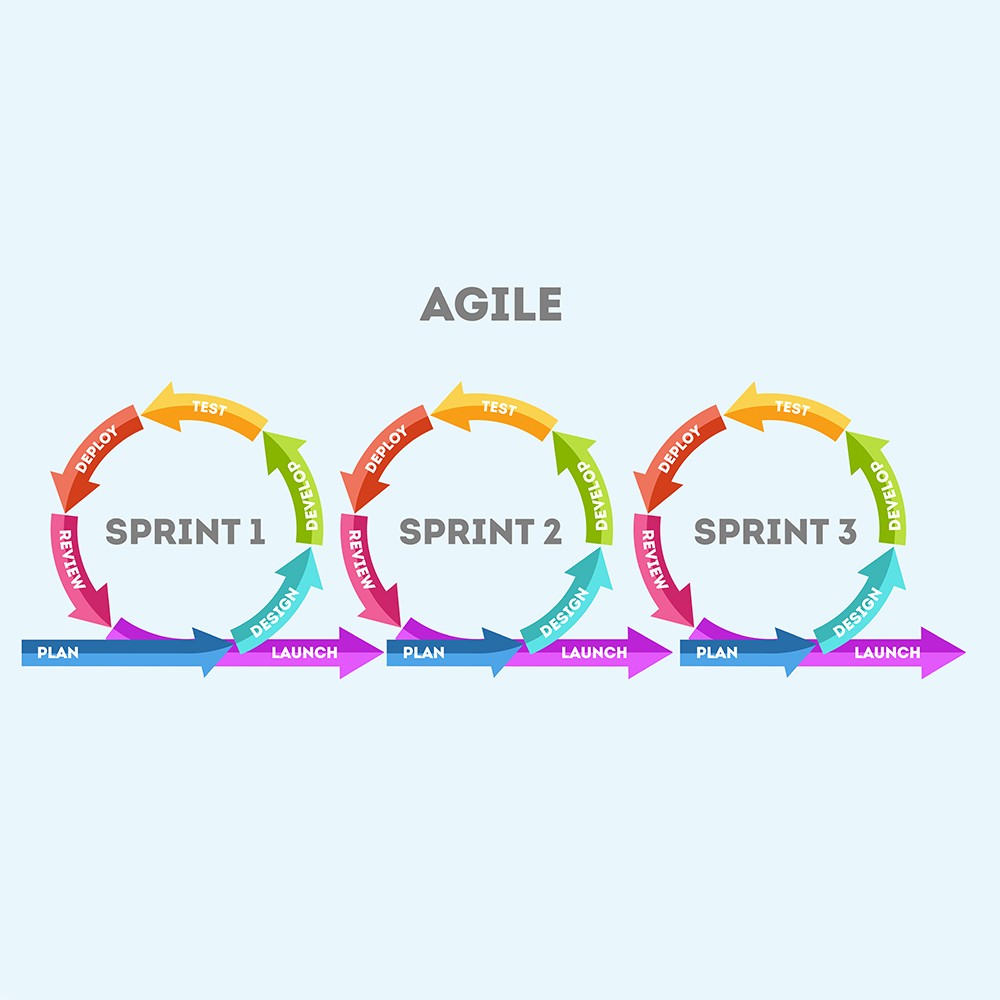
* + - * 1. Fonctionnement de la méthode

La technique Agile est fondée sur un principe de base : planifier votre projet dans ses moindres détails avant de le développer est inefficace.

Organiser chaque détail de votre projet est une perte de temps car rarement tout se passe comme prévu. Les dangers se développent fréquemment, vous obligeant à ajuster vos plans.

Fixer des objectifs à court terme est recommandé par la technique Agile. En conséquence, le projet est divisé en plusieurs sous-projets. On passe à l'objectif suivant une fois le premier atteint, et ainsi de suite jusqu'à ce que le dernier objectif soit atteint. Cette méthode est plus adaptable. Parce qu'il est difficile de tout planifier et de tout prévoir, il y a toujours des opportunités pour les imprévus et les ajustements.

Autre avantage, la méthode Agile repose sur une relation avantagée entre le client et l'équipe responsable du projet. La satisfaction du client étant une préoccupation majeure, le personnel doit être pleinement engagé et réactif aux changements d’exigences et aux événements imprévus. Il faut privilégiés les moments de rencontres avec le client. Car c'est lui qui vérifie le projet à chaque étape. Afin de répondre à leurs attentes, l'évolution de leurs besoins est prise en compte et des ajustements sont effectués en temps réel.

Rien n'est gravé dans le marbre en utilisant la méthodologie Agile. L'équipe projet doit pouvoir e remettre en question régulièrement et chercher à s'améliorer.­­­

Cycle de développement selon la méthode Agile

* 1. Analyse du projet
     + 1. Analyse des besoins

La spécification des besoins est la première étape du développement de toute application, au cours de laquelle nous déterminerons les différentes exigences. Il doit décrire clairement le programme qui sera développé. Il se compose d'un ensemble de documents et de modèles.

La définition des besoins doit être accessible à toutes les personnes impliquées dans le projet. Nous distinguons les besoins fonctionnels qui présentent les fonctionnalités attendues de notre application des besoins non fonctionnels qui quant à eux permettent d'éviter le développement d'une application insatisfaisante.

* + - * 1. Spécification des besoins fonctionnels

Les besoins fonctionnels, également appelés besoins métier, sont les tâches que le système doit effectuer ; elle ne peut devenir opérationnelle que si elles sont respectées. Ils spécifient une action que le système doit entreprendre en réponse à une demande (sorties pour un ensemble spécifique d'entrées).

L'application que nous allons développer devra regrouper les fonctionnalités qui permettront à l'utilisateur de :

* Gérer les élèves
* Gérer les classes
* Gérer les cours
* Gérer les enseignants
* Gérer le personnel
* Gérer l’emploi de temps
* Gérer le salaire des enseignants et du personnel administrative et d’entretien
  + - * 1. Spécification des besoins non fonctionnels

Les exigences non fonctionnelles sont celles qui ne sont pas directement liées au comportement du système, mais identifient plutôt les restrictions internes et externes du système. Voici les principales exigences non fonctionnelles de l'application :

Besoins de sécurité :

Les niveaux d'accès des utilisateurs au système sont définis par eux. Dans notre cas, le programme doit avoir un niveau de sécurité élevé, avec des comptes d'utilisateurs protégés par mot de passe.

Ergonomie

L'ergonomie de l'application doit favoriser la lisibilité et l'accessibilité des informations, ainsi que leur réutilisation.

L'interface utilisateur doit être simple à utiliser et à comprendre pour les non-professionnels de l'informatique.

* + 1. Analyse de l’existant

Une analyse a été menée avant de commencer la conception et le développement de l'application afin de justifier la réalisation du projet.

Et, sur la base de cette étude, nous pouvons déterminer que l’établissement scolaire utilise actuellement les programmes Microsoft Office pour le suivi des élèves, enseignant et la gestion financière, les données étant obtenues manuellement dans un premier temps. En effet, le programme Microsoft Word est utilisé pour enregistrer les fichiers, et le tableur Excel est utilisé pour suivre et traiter les informations sur les salaires et dépenses.

En plus de cela le réseau interne n'existe pas et la base des données centralisée non plus.

* + - 1. Organigramme de fonctionnement
         1. Cas de l’inscription

La figure suivante illustre le fonctionnement de la gestion de l’inscription et d’embauche d’enseignant ou de personnel avant la mise en place du nouveau système informatique :

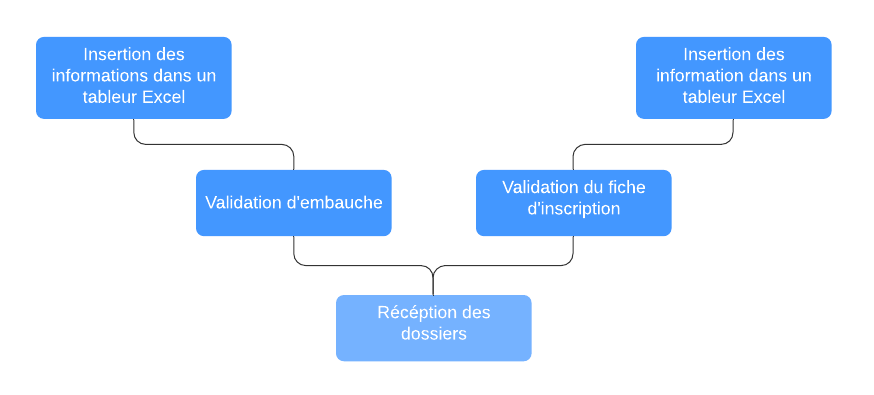
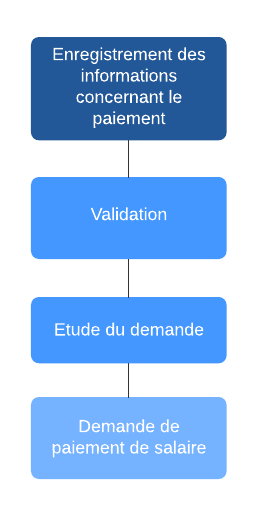


Illustration de l’organisation des inscription et embauche dans l’établissement

* + - * 1. Cas du paiement de salaire

La figure suivante illustre le fonctionnement du paiement de salaire des enseignants et du personnel qui est actuellement en utilisation :

Illustration de l’organisation des paiements de salaires au sein de l’établissement

* + - 1. Moyens matériels et logiciels
         1. Moyens matériels

L’établissement a en leur disposition :

* Un (01) ordinateur de bureau
* Un (01) ordinateur portable
* Deux (02) imprimantes

Ces matériels on les caractéristique suivantes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Libellé | Processeur | Disque dur | Ram | Marque |
| Desktop | Pentium 4 | 320 | 2go | DELL |
| Laptop | Core i5 | 500 | 4go | HP |

Caractéristiques des ordinateurs au sein de l’établissement

* + - * 1. Moyens logiciel

Le système d'exploitation Windows est utilisé dans les bureaux de l’établissement, et celui qui est utilisé par l’ordinateur de bureau est le Windows 7, tandis que celle utilisé par l’ordinateur portable est le Windows 10.

Le tableau 6.2 ci-dessous explique les détails du logiciel utilisé par l’établissement en question :

|  |  |
| --- | --- |
| Noms | Fonctions générale et spécifique |
| Microsoft Word | * Traitement de texte * Rédaction, modification, ou suppression des données. |
| Microsoft Excel | * Feuille de calcul, * Insertion, modification ou suppression de données dans un tableau. |

Outils utilisés par l’établissement

* + 1. Critique de l’existant

Certes, après un examen approfondi de la situation, une critique objective est proposée. Ainsi, on a pu constater que le traitement manuel et l'utilisation des produits Microsoft Office sont toujours utilisés dans l’administration de l’établissement. De ce fait, celle-ci porte sur les points forts et faibles du système existant.

* + - 1. Points forts
* Il vaut mieux ne pas informatiser la plupart des procédures, telles que les signatures et la vérification des informations
* Les documents utilisés dans le service et bureau inscriptions ont une bonne présentation.
  + - 1. Points faibles
* Défaut de sécurité des données : les outils Microsoft Office ne sont pas conçus pour protéger les données. L'utilisation de ces papiers n'étant pas limitée par l'authentification, il existe un risque de suppression et de modification.
* Défaut de directive : l’utilisateur peut faire de mauvaise manipulation dans les outils traditionnelle risquant d’altérer ou de supprimer les informations conduisant au mauvais fonctionnement de la gestion
* Redondances des informations car ils sont réenregistrés chaque année
* Non cohérence entre les informations anciennement enregistrées et nouvellement inscrite
  + 1. Solutions proposées

Deux remèdes sont présentés après un examen objectif du système existant, et ces derniers sont les suivants :

* La restructuration
* L'informatisation.
  + - 1. La solution de restructuration

Au vu des failles évoquées dans la critique du système précédent, la solution actuelle est de conserver le système existant mais de le réorganiser comme suit :

* Créer des disques de récupération semblable à l’état actuel des informations contenu dans les disques dur à chaque fin de journée,
* Limiter l’accès à certaine action avec la restriction d’opération faisable dans un fichier
  + - 1. Solution d’informatisation

Cela conduira à la conception d'un nouveau système d'information informatisé.

* + - 1. Comparaison de deux (02) solutions proposées

|  |  |  |
| --- | --- | --- |
| Solution | Avantages | Inconvénients |
| Restructuration | Pas de besoin de reformer le personnel  Maintien de la méthode actuelle de travail | Nécessité de beaucoup plus de stockage  Données non-sécurisé |
| Informatisation | La diffusion rapide et facile des infirmations  Fiabilité des résultats  Mise en jours des données facile  Sécurité des informations. | Besoin en formation du personnel |

Comparaison entre la solution de restructuration et celle de l’informatisation.

* + 1. Solution retenue

En conséquence des faits vu dans le **tableau 6.3** ci-dessus, la restructuration n’est pas la solution adéquate vu que celui-ci est plus coûteuse. Ainsi, nous pensons que la conception d'un nouveau système d'information informatisé serait une solution supérieure car elle inclurait des fonctionnalités plus appropriées aux besoins de l’établissement. De ce fait, nous avons conçu et mis en œuvre ce projet conformément au cahier des charges établi, ainsi qu'à quelques fonctionnalités additionnelles, afin que le travail au sein de l’établissement soit plus simple et non fastidieux, ce qui se traduit par des résultats supérieurs et la performance voulue par ce dernier. Nous avons également optimisé les facteurs de sécurité et de gestion des données tout en soulageant le personnel d’une énorme dépense de temps.

* 1. Conception du projet
     1. Définition de quelques concepts
        1. Etape conceptuelle

L'étape conceptuelle marque le début de la conception et de la modélisation du domaine de gestion. C'est aussi le niveau auquel des règles de gestion peuvent être précisées, permettant de voir directement les relations entre l'information et son utilisation en suivant les contraintes associées à l'esprit.

* + - 1. Conception

La conception est un scénario abstrait qui aide à la présentation d'entités conceptuelles, telles que des objets et des relations.

* + 1. Identification des acteurs

Les acteurs sont des entités non-système qui interagissent avec le système, comme un humain ou un robot. Parce qu'une même personne (ou robot, etc.) peut jouer plusieurs rôles dans un système, les acteurs doivent être décrits avant tout par leur rôle, qui décrit les besoins et les capacités de l'acteur. Un acteur effectue une action sur le système (accès en lecture/écriture).

L'activité du système cherche à répondre aux besoins de l'acteur. Un pictogramme humanoïde avec le nom de l'acteur est utilisé pour représenter les acteurs. Dans notre application, nous avons identifié deux (02) acteurs :

* Administrateur

Rôles : gérer les élèves, gérer les enseignants, gérer les emplois du temps

* Administrateur principal

Rôles : gérer l’écolages des élèves, les salaires des enseignants et les autres utilisateurs.

* + 1. Dictionnaire de données

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entité | Propriété | Description | Type | Taille |
| Student | ID | Identification de l’élève | N | 3 |
| im | Numéro d’immatriculation de l’élève | AN | 10 |
| firstname | Prénoms de l’élève | AN | 30 |
| lastname | Nom de l’elève | AN | 50 |
| birthday | Date de naissance de l’élève | D | - |
| birthplace | Lieu de naissance de l’élève | AN | 50 |
| sex | Sexe de l’élève | AN | 1 |
| address | Adresse de l’élève | AN | 50 |
| actNum | Numéro de l’acte de naisance de l’élève | AN | 20 |
| Image | Chemin de l’image de l’élève | AN | 255 |
| isRedoubling | Information si l’élève est un redoublant ou non | B | 1 |
| Parent | ID | Identifiant du parent | N | 3 |
|  | cin | Numéro de la carte d’identité nationale du parent | N | 3 |
|  | firstname | Prénoms du parent | AN | 30 |
|  | lastname | Nom du parent | AN | 30 |
|  | type | Le type de parent, si c’est un père ou une mère ou un tuteur | AN | 10 |
|  | Job | Le travail du parent | AN | 40 |
|  | Phone | Le numéro de téléphone du parent | N | 10 |
| Teacher | ID | Identifiant de l’enseignant | N | - |
| cin | Numéro de la carte d’identité nationale de l’enseignant | N | 12 |
| firstname | Prénoms de l’enseignant | AN | 30 |
| lastname | Nom de l’enseignant | AN | 30 |
| teachLic | Numéro d’autorisation d’enseigné | AN | 50 |
| teachLicDate | Date d’obtention de l’autorisation d’enseigner | D | - |
| im | Immatriculation de l’enseignant | AN | 10 |
| takingOffice | Date de la prise de poste de l’enseignant | D | - |
| birthday | Date de naissance de l’enseignant | D | - |
| image | Chemin vers l’image de l’enseignant | AN | 255 |
| accountID | Identifiant du compt de l’enseignant | N | - |
| Grade | desc | Description du diplôme | AN | 255 |
| receivedAt | Date d’obtention du diplôme | D | - |
| type | Type du diplôme | AN | 30 |
| teacherID | Identifiant de l’enseignant détenant le diplôme | N | - |
| Person | ID | Identifiant de la personne | N | - |
| firstname | Prénoms de la personne | AN | 30 |
| lastname | Nom de la personne | AN | 30 |
| cin | Numéro de la carte d’identité nationale de la personne | N | 12 |
| image | Chemin vers l’image de la personne | AN | 255 |
| Staff | personID | Identifiant le la personne qui est un agent | N | - |
| Admin | PersonID | Identifiant de la personne qui est un administrateur | N | - |
| accountID | Identifiant de du compte appartenant à l’administrateur | N | - |
| Account | ID | Identifiant du compte | N | - |
| username | Nom d’utilisateur du compte | AN | 30 |
| passwd | Mot de passe du compte | AN | 50 |
| level | Niveau d’accès du compte | N | 1 |
| Work | ID | Identifiant du travail | N | - |
|  | salaryAmount | Salaire du travail convenue | N | 10 |
|  | personID | Identifiant de la personne faisant le travail | N | - |
|  | functionID | Identifiant de la fonction à occupé | N | - |
|  | workPeriodPeriodID | Identifiant de la période où le travail a été occupé par une personne | N | - |
| Period | ID | Identifiant de la période | N | - |
| begining | Date du début du période | D | - |
| ending | Date terminant le période | D | - |
| schoolyearID | Identifiant de l’année scolaire à laquelle la période appartienne | N | - |
| WorkPeriod | workID | Identifiant du période à laquelle le période de travaille appartient | N | - |
| Function | ID | Identifiant de la fonction | N | - |
|  | label | Libellé de la fonction | AN | 30 |
| PresTeacher | teacherID | Identifiant de l’enseignant qui est un enseignant des classes préscolaire et primaire | N | - |
| HighTeacher | teacherID | Identifiant de l’enseignant qui est un enseignant des classes du collège et du lycée | N | - |
| Class | ID | Identifiant de la classe | N | - |
| label | Libellé de la classe | AN | 30 |
|  | levelID | Identifiant du niveau dans laquelle la classe se situe | N | - |
| presClass | classID | Identifiant de la classe qui est une classe de préscolaire ou secondaire | N | - |
| HighClass | classID | Identifiant de la classe qui est une classe du collège ou du lycée | N | - |
| Holiday | ID | Identifiant du vacances | N | - |
|  | label | Libellé du vacances | AN | 30 |
|  | offDay | Date concernée par la vacance | D | - |
| HolidayHour | ID | Identifiant de l’heure concerné par la vacance | N | - |
|  | hourNb | Le numéro de l’heure dans une journée qui commence par 7h-12h le matin et de 14h-18h l 'après midi | N | 2 |
|  | holidayID | Identifiant de la vacance à laquelle l’heure appartient | N | - |
| Level | ID | Identifiant du niveau | N | - |
| label | Libéllé du niveau | AN | 30 |
| tuition | Ecolage par mois pour le niveau | N | 10 |
| registrationFees | Droit d’inscription pour les nouveaux | N | 10 |
| oldRegistraionFees | Droit d’inscription pour les anciens | N | 10 |
| Activité | Activité à payer pour le niveau | N | 10 |
| fram | FRAM à payer pour le niveau | N | 10 |
| travellingExpenses | Le frais de transport à payer pour le niveau | N | 10 |
| cycleID | Identifiant du cycle dans lequel le niveau appartient | N | - |
| Cycle | ID | Identifiant du cycle | N | - |
|  | Label | Libellé du cycle | AN | 30 |
| Historic | ID | Identifiant de l’historique | N | - |
|  | dateBegin | Date du début de l’historique | D | - |
|  | dateEnd | Date auquelle l’historique se termine | D | - |
|  | content | Contenue de l’historique | AN | 255 |
|  | studentID | Identifiant de l’étudiant propriétaire de l’historique | N | - |
| OtherCoast | ID | Identifiant du frais divers à payer par l’élève | N | - |
|  | label | Libellé du frais | AN | 30 |
|  | amountPaid | Montant payé sur le frais | N | 10 |
|  | totalAmount | Montant total du frais | N | 10 |
|  | studentID | Identifiant de l’élève auquel le frais appartient | N | - |
| TuitionFees | ID | Identifiant de l’écolage | N - |  |
|  | totalAmount | Montant total de l’écolage | N | 10 |
|  | beginning | Date débutant le mois d’étude | D | - |
|  | ending | Date terminant le mois d’étude | D | - |
|  | studentID | Identifiant de l’élève auquel l’écolage appartient | N | - |
| Payment | ID | Identifiant d’un paiement | N | - |
|  | paidAt | Date du paiement | D | - |
| PaymentDetail | paymentID | Identifiant du paiement qui est un paiement d’écolage | N | - |
|  | tuitionsFeesID | Identifiant de l’écolage concerné par le paiement | N | - |
| PaymentSalary | paymentID | Identifiant du paiement qui est un paiement de salaire | N | - |
|  | salaryID | Identifiant du salaire concerné par le paiement | N | - |
| Salary | ID | Identifiant du salaire | N | - |
| month | Le numéro du mois du salaire | N | 2 |
| year | L’année du salaire | N | 4 |
| TeacherSalary | salaryID | Identifiant du salaire qui est un salaire d’enseignant | N | - |
| PersonSalary | salaryID | Identifiant du salaire qui est un salaire du personnel | N | - |
| Overtime | ID | Identifiant de l’heure supplémentaire | N | - |
| label | Libellé de l’heure supplémentaire | AN | 30 |
| Date | La date quand l’heure supplémentaire sera faite | D | - |
| courseID | Identifiant du cours concerné par l’heure supplémentaire | N | - |
| teacherID | Identifiant de l’enseignant qui fera l’heure supplémentaire | N | - |
| Note | ID | Identifiant du remarque | N | - |
| content | Contenu du remarque | AN | 255 |
| createdAt | Date de création du remarque | D | - |
| isNegative | Valeur pour savoir si une remarque est négative ou non | B | 1 |

Description des données utilisées

* + 1. Diagramme des cas d’utilisation
       1. Généralité

Les cas d'utilisation nous permettent de collecter, d'analyser et d'organiser les besoins, ainsi que d'identifier les principales fonctionnalités du système. En conséquence, c'est la première étape d’analyse UML d'un système

Dans cette section, nous présenterons des cas d'utilisation relatifs aux modules de :

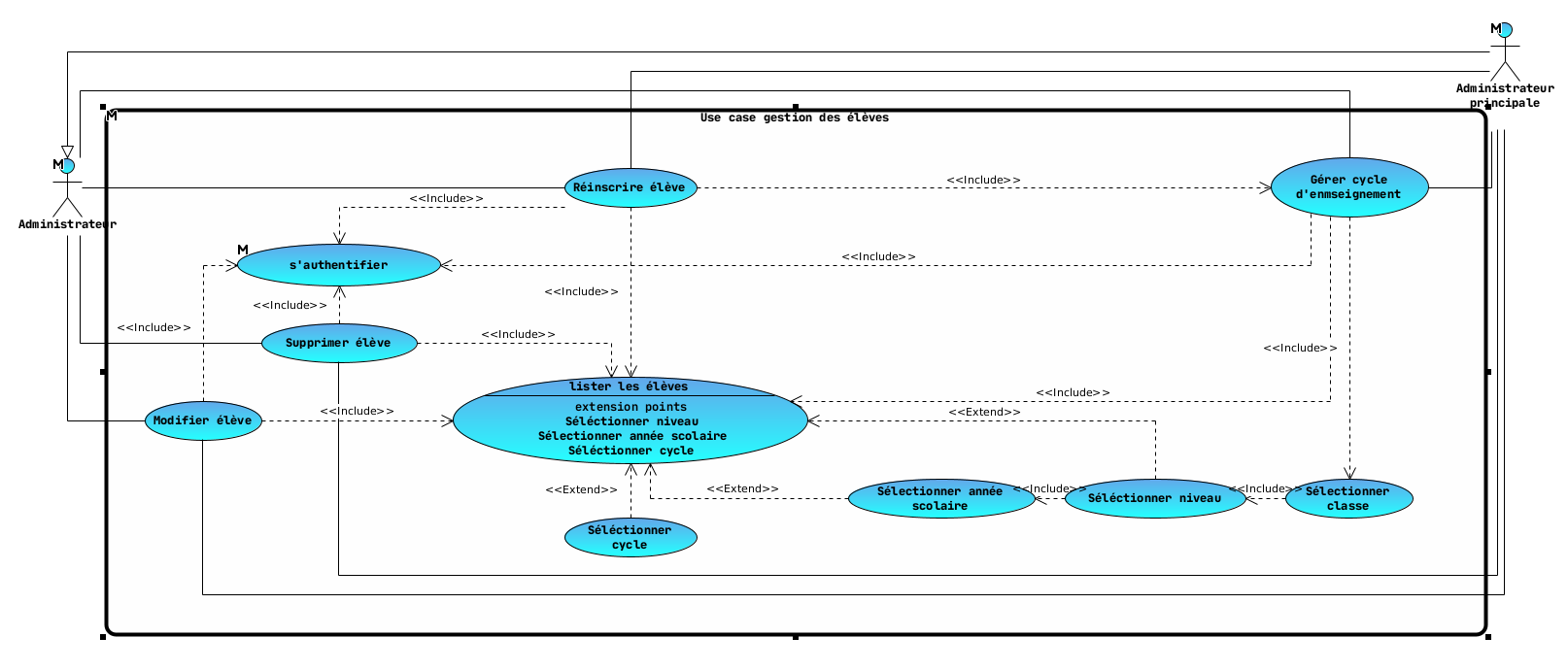
* Gestion des élèves
* Gestion cycle d’enseignement
* Gestion du personnel administratifs et pédagogique
  + - 1. Cas d’utilisation « Gestion des élèves »

Diagramme de cas d’utilisation « Gestion des élèves »

* + - * 1. Description du cas d’utilisation « Gestion des élèves »

|  |  |
| --- | --- |
| Cas d’utilisation | Lister les élèves |
| Acteur principale | Administrateur |
| Objectif | Voir la liste des élèves dans le système |
| Pré conditions | * S’authentifier * Voir la liste des élèves * Sélectionner l’année scolaire * Sélectionner le niveau |
| Post conditions | Affichage de la liste des élèves |
| Description | * L’administrateur se connecte et se dirige vers la page de la liste des élèves * Le système affiche la liste de tous les élèves |
| Exigences supplémentaires | * L’administrateur doit sélectionner l’année scolaire, le cycle et le niveau des élèves à affiché |

Description du cas d’utilisation « lister les élèves » »

|  |  |
| --- | --- |
| Cas d’utilisation | Inscrire élève |
| Acteur principale | Administrateur |
| Objectif | Inscrire un élève dans une classe pendant l’année scolaire en cours |
| Pré conditions | S’authentifier  Voir la liste des élèves  Sélectionner l’année scolaire  Sélectionner le niveau  Sélectionner la classe  Vérification des informations fournis |
| Post conditions | * Un nouvel élève est inscrit dans la classe pour l’année scolaire * L’élève inscrite passe au paiement des divers frais et écolage du premier mois. |
| Scénario nominale | * L’élève présente sa demande, les informations nécessaires et les frais d’inscription * Gestionnaire vérifie que toutes les informations sont correctes. * Gestionnaire rempli le formulaire et enregistre le nouvel élève, l’élève/étudiant est alors inscrit. |
| Scénario secondaire | Les informations fournis ne sont pas suffisantes donc l’inscription est reportée jusqu’à la complétion des documents  L’âge fourni n’est pas valide |
| Exigences supplémentaires | L’élève doit apporter un photo version ou numérique ou doit-être prise en photo |

Description du cas d’utilisation « inscrire élève »

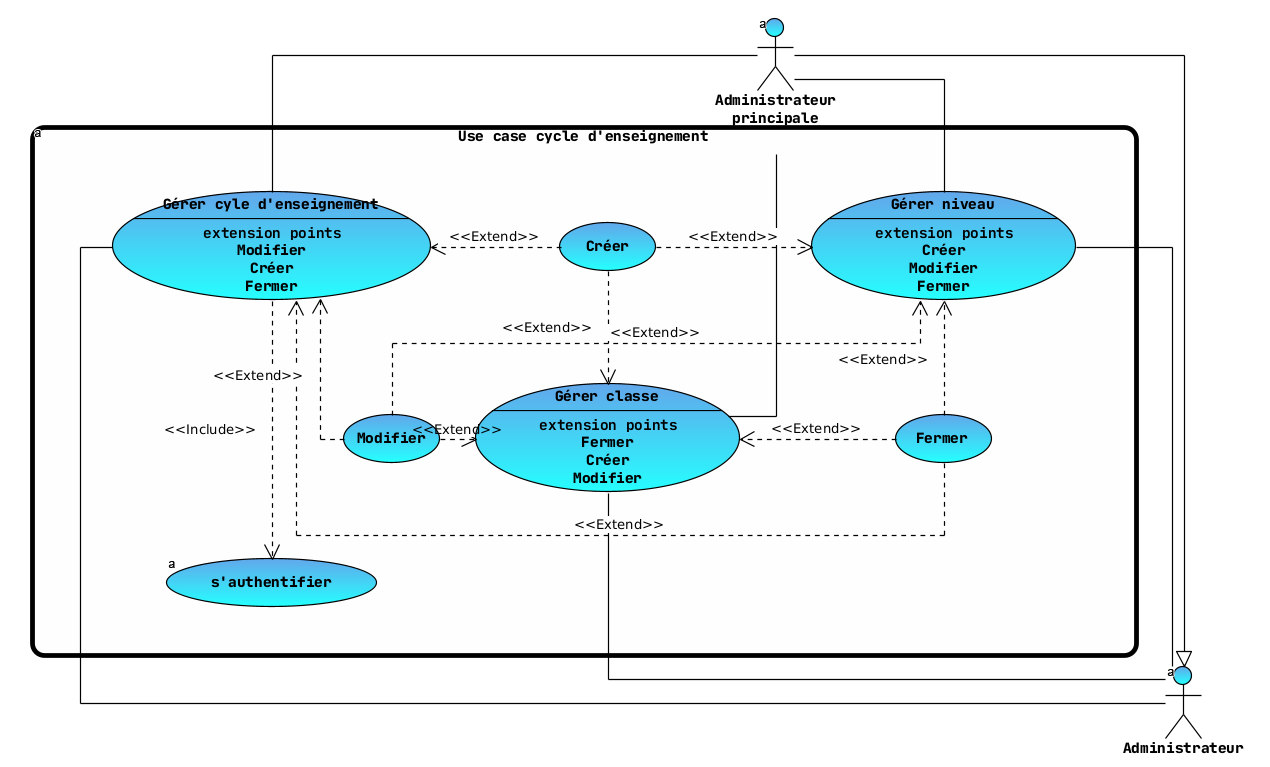
* + - 1. Cas d’utilisation « Gestion cycle d’enseignement »

Diagramme du cas d’utilisation « Gestion cycle d’enseignement »

* + - * 1. Description du cas d’utilisation « Gestion cycle d’enseignement »

|  |  |
| --- | --- |
| Cas d’utilisation | Gestion cycle d’enseignement |
| Acteur principale | Administrateur |
| Objectif | Ce cas d’utilisation permet de gérer un cycle d’enseignement concernant l’établissement :   * Création d’un nouveau cycle, d’un nouveau niveau ou d’une nouvelle classe, * Modification d’un cycle, d’un niveau ou d’une classe, * Fermeture d’un cycle, d’un niveau ou d’une classe |
| Pré conditions | * S’authentifier |
| Scénario nominale | * L’administrateur ajoute un nouveau cycle ou ajoute un nouveau niveau dans un cycle ou ajoute une nouvelle classe dans un niveau * L’administrateur modifie un cycle ou un niveau ou une classe * L’administrateur ferme un cycle ou un niveau ou une classe |
| Scénario secondaire | * L’administrateur essaie de supprimer un cycle qui contient plusieurs niveaux donc alerter du risque, * L’administrateur essaie de supprimer un niveau qui contient plusieurs classes donc alerter du risque, * L’administrateur essaie de supprimer une classe qui contient plusieurs élèves donc alerter du risque, |

Description « Gérer cycle d’enseignement »

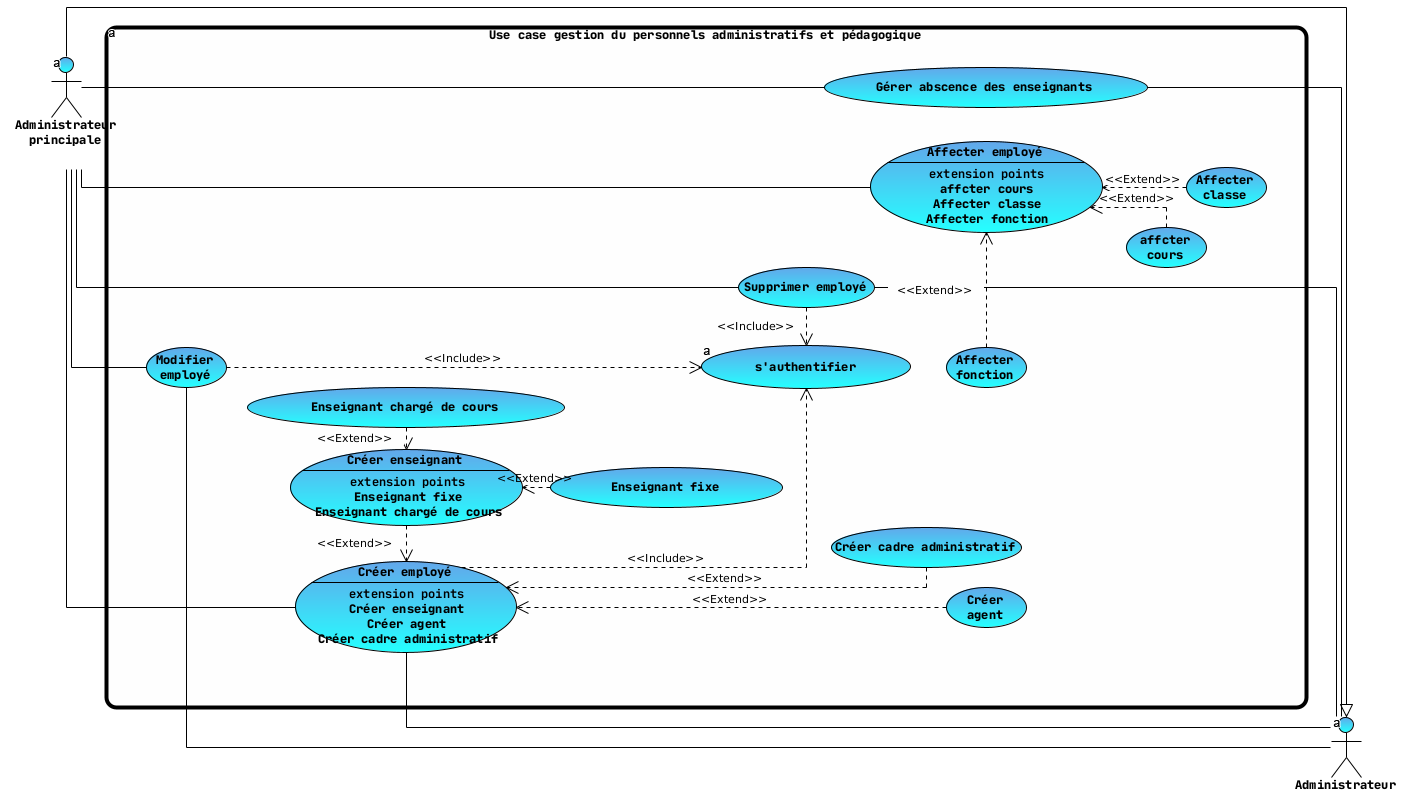
* + - 1. Cas d’utilisation « Gestion du personnels administratifs et pédagogique »

Diagramme du cas d’utilisation « gestion du personnels administratifs et pédagogique »

* + - * 1. Description cas d’utilisation « Gestion du personnels administratifs et pédagogique »

|  |  |
| --- | --- |
| Cas d’utilisation | Créer cadre administratifs |
| Acteur principale | Administrateur principale |
| Objectif | * Créer un membre de l’administration pouvant gérer la majorité des actions au sein de l’établissement |
| Pré conditions | * S’authentifier avec le compte possédant le niveau d’accès le plus élevé |
| Post conditions | * Un nouvel administrateur possédant un compte est ajouté dans le système * Les salaires du nouveau membre de l’administration pendant l’année scolaire sont générés son créer |
| Scénario nominale | * L’administrateur se connecte et ouvre la liste du personnel et clique sur le bouton « créer administrateur » * Un formulaire s’ouvre et l’administrateur principale le rempli et valide l’insertion * Après l’insertion de ses informations, l’administrateur continue avec la création de son compte pour pouvoir utiliser l’application |
|  | * Le gestionnaire consulte les informations d’un employé, les modifie en cas de données erronées. |

Description du cas d’utilisation « Créer cadre administratifs »

|  |  |
| --- | --- |
| Cas d’utilisation | Créer enseignants fixe |
| Acteur principale | Administrateur |
| Objectif | * Créer un enseignant des classes préscolaire et primaire |
| Pré conditions | * S’authentifier |
| Post conditions | * Un nouvel enseignant fixe est ajouté * Les salaires du nouvel enseignant pendant l’année scolaire sont générer |
| Scénario nominale | * L’administrateur se connecte et ouvre la liste des enseignant et clique sur le bouton ajouter enseignant * Un formulaire s’ouvre et l’administrateur principale le rempli, choisi le type « enseignant fixe » et valide le formulaire |
| Scénario secondaire | * Le gestionnaire consulte les informations d’un employé, les modifie en cas de données erronées. |

Description du cas d’utilisation « créer enseignant fixe »

|  |  |
| --- | --- |
| Cas d’utilisation | Créer enseignant chargé de cours |
| Acteur principale | Administrateur |
| Objectif | * Créer un nouvel enseignant qui se chargera des classes du collège ou du lycée |
| Pré conditions | * S’authentifier |
| Post conditions | * Un nouvel enseignant qui se chargera des cours est ajouté dans le système * Les salaires du nouvel enseignant pendant l’année scolaire sont générer tandis la somme exacte attendra son affection à un cours |
| Scénario nominale | * L’administrateur se connecte et ouvre la liste du personnel et clique sur le bouton « créer administrateur » * Un formulaire s’ouvre et l’administrateur principale le rempli et valide l’insertion |
| Scénario secondaire | * Le gestionnaire consulte les informations d’un employé, les modifie en cas de données erronées. |

Description du cas d’utilisation « Créer un enseignant chargé de cours »

* + 1. Diagrammes d’activités

L'utilisation de diagrammes d'activités nous permet de nous concentrer sur les traitements. De ce fait, ils sont particulièrement bien adaptés à la modélisation des flux de contrôle et des flux de données. De ce fait, ils permettent de représenter graphiquement le comportement d'une méthode ou la progression d'un cas d'utilisation.

Dans cette section, nous présenterons des diagrammes d'activités pour le processus d'inscription des étudiants ainsi que la gestion des absences au sein d'un établissement d'enseignement.

* + - 1. Diagramme d’activité d’inscription d’étudiants

En effet, après la demande d’inscription de l’étudiant, le gestionnaire vérifie si l’étudiant complète les conditions d’admission, et l’ajoute à la base de données et l’enregistre en lui affectant le niveau et la classe.

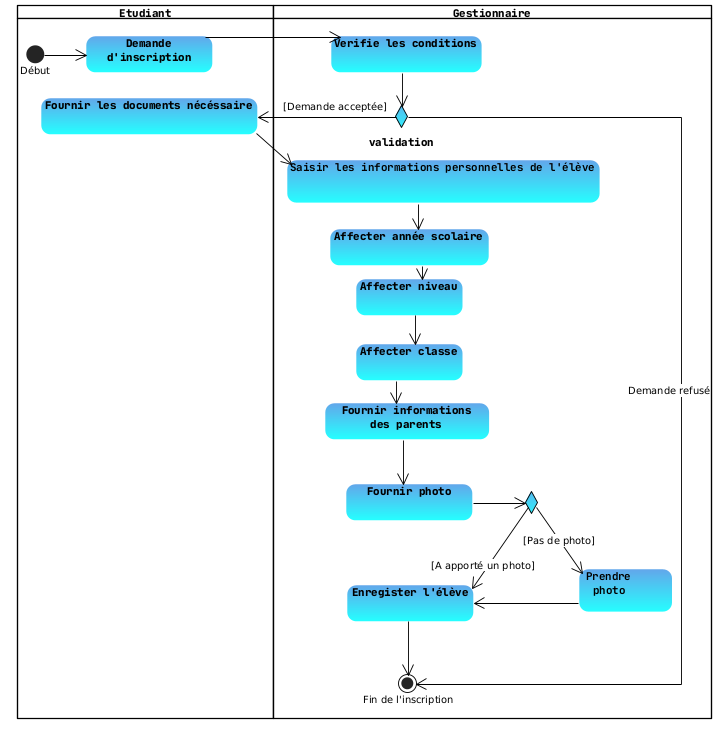


Diagramme d’activité du cas d’utilisation « Gestion élève »

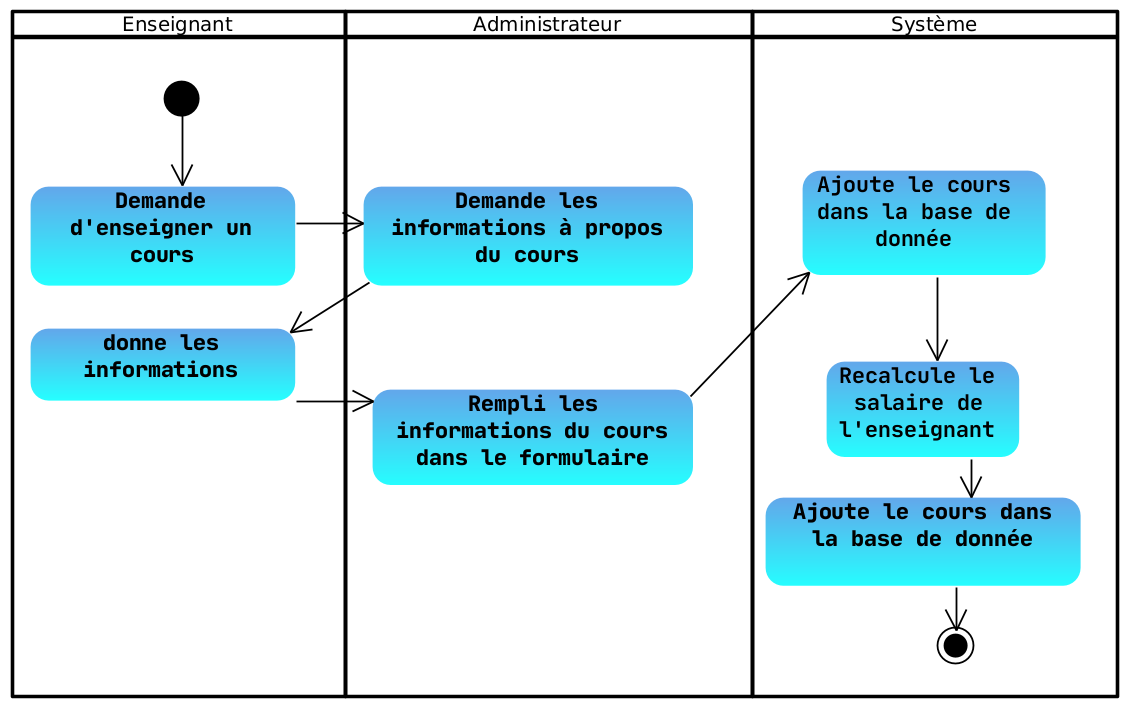
* + - 1. Diagramme d’activité d’ajout d’un cours 

Diagramme d’activité du cas d’utilisation « ajouter cours »

* + 1. Diagramme de séquence

Les diagrammes de séquence sont des diagrammes d'interaction qui vous permettent de modéliser des scénarios. Ils visent à mieux représenter les interactions temporelles entre les objets de notre projet.

En effet, une fois les exigences du système formalisées à l'aide de schémas d'utilisation, il est nécessaire de modéliser les interactions entre le système et les différents acteurs à l'aide d'une succession séquentielle d'actions.

La figure suivante nous montre les séquences d’action entre l’utilisateur et le système lors de la première utilisation :

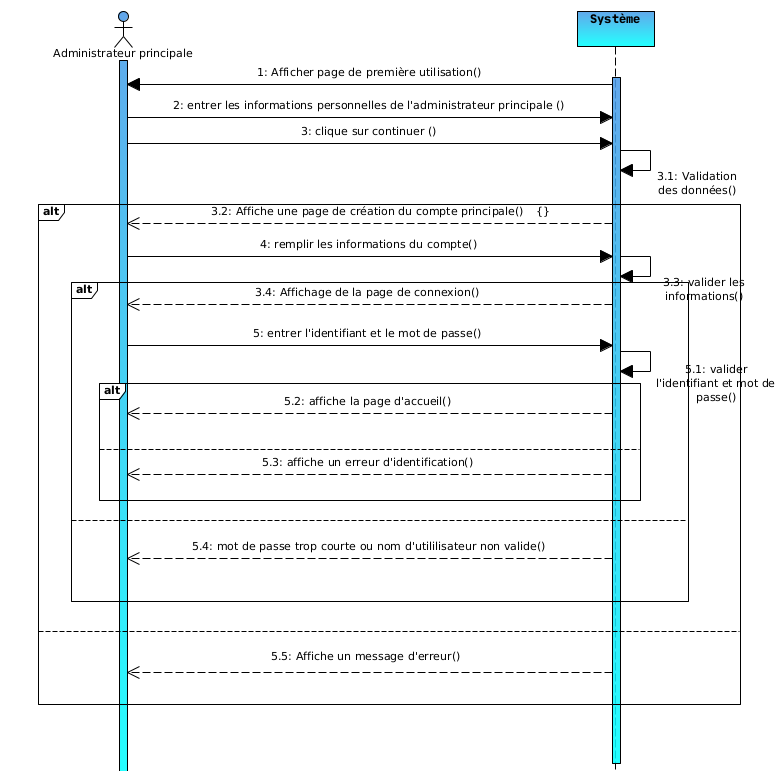


Diagramme de séquence du cas d’utilisation « Première connexion »

La figure 7.7 suivante nous expose les suites d’actions entre le gestionnaire et le système lors du paiement d’écolage d’un élève :

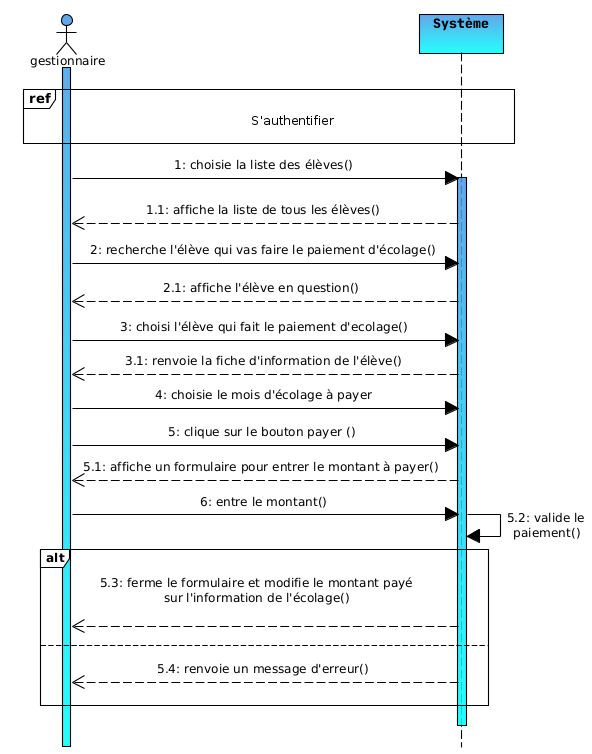


Diagramme de séquence du cas d’utilisation « Paiement écolage »

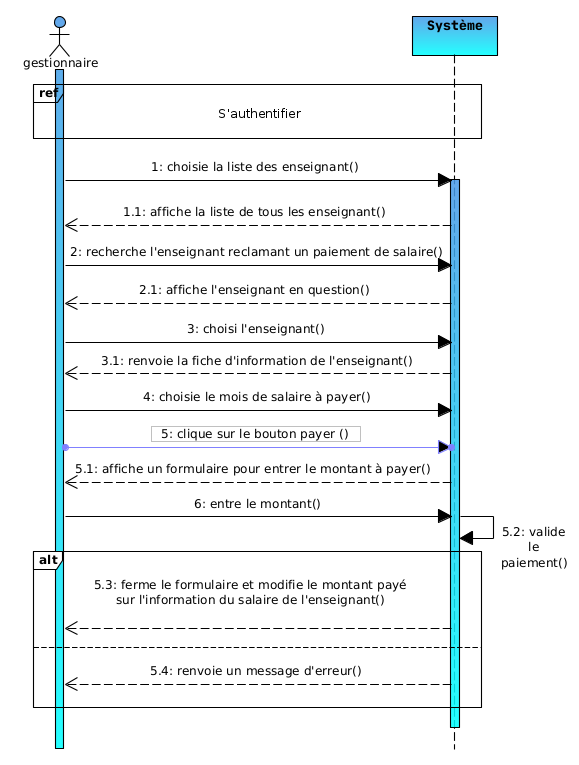
La figure suivante nous illustre les séquences d’actions entre le gestionnaire et le système lors du paiement de salaire d’un enseignant

Diagramme de séquence du cas d’utilisation « Paiement salaires enseignants ».

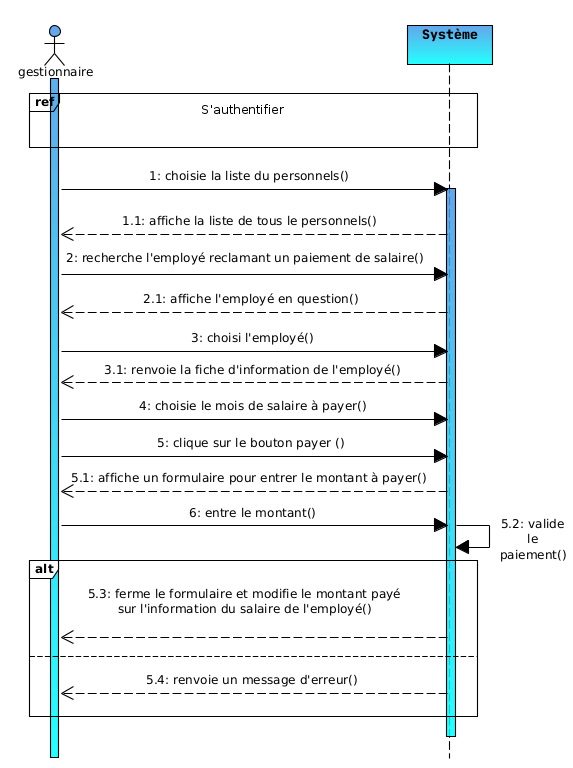
La séquence de travails entre le gestionnaire et le système lors du paiement du salaire d'u personnel est illustrée par la figure suivante :

Diagramme de séquence du cas d’utilisation « Paiement salaire personnel »

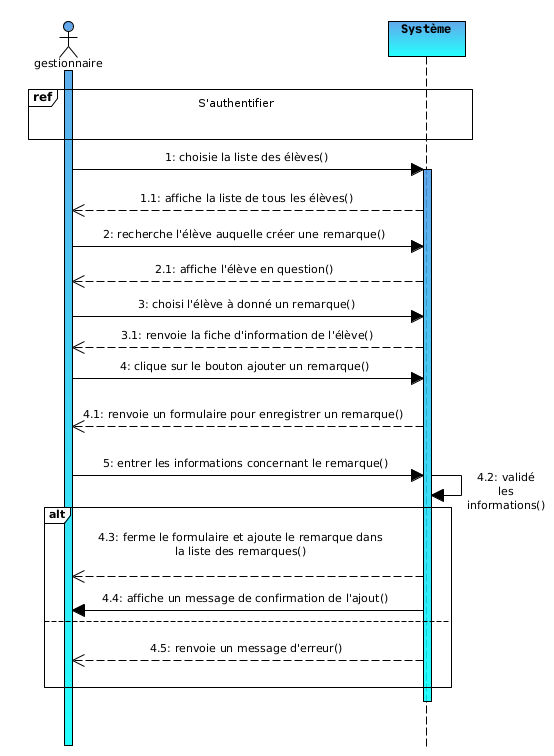
La séquence de fait entre le gestionnaire et le système lors de l’ajout d’une remarque sur un élève est illustre sur la figure suivante :

Diagramme de séquence du cas d’utilisation « ajouter une remarque à un élève »

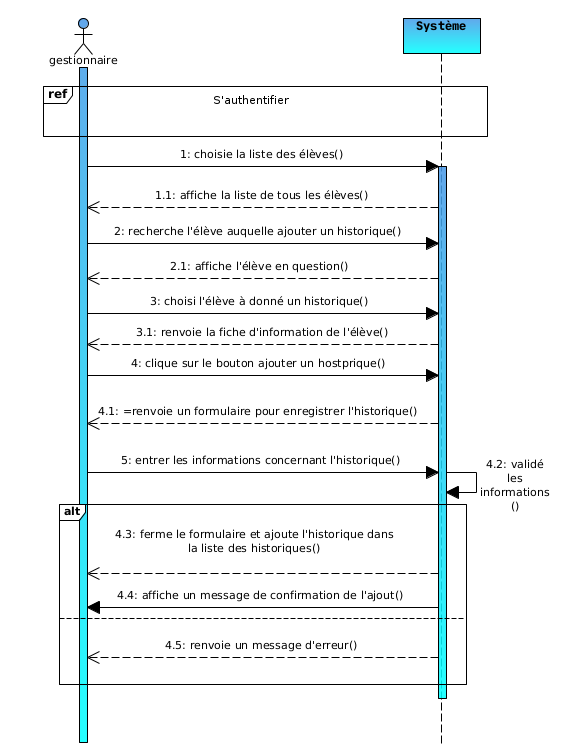
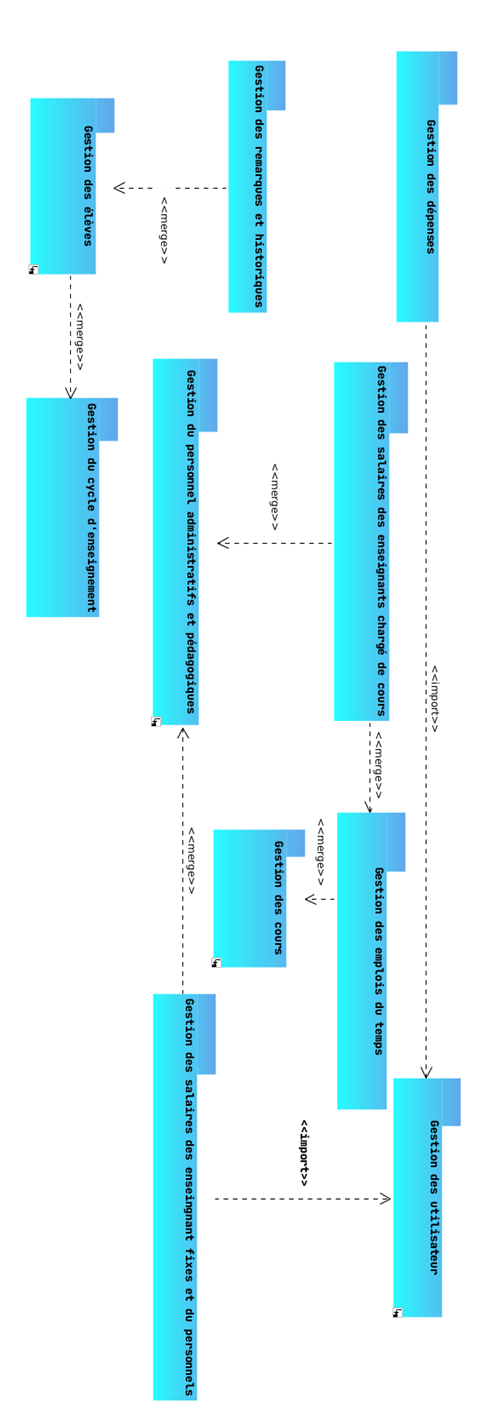
La figure suivante décrit la séquence d'événements entre le gestionnaire et le système lors de l'ajout d'une historique sur un étudiant :

Diagramme de séquence du cas d’utilisation « ajouter historique sur un élève »

* + 1. Diagramme de paquetage

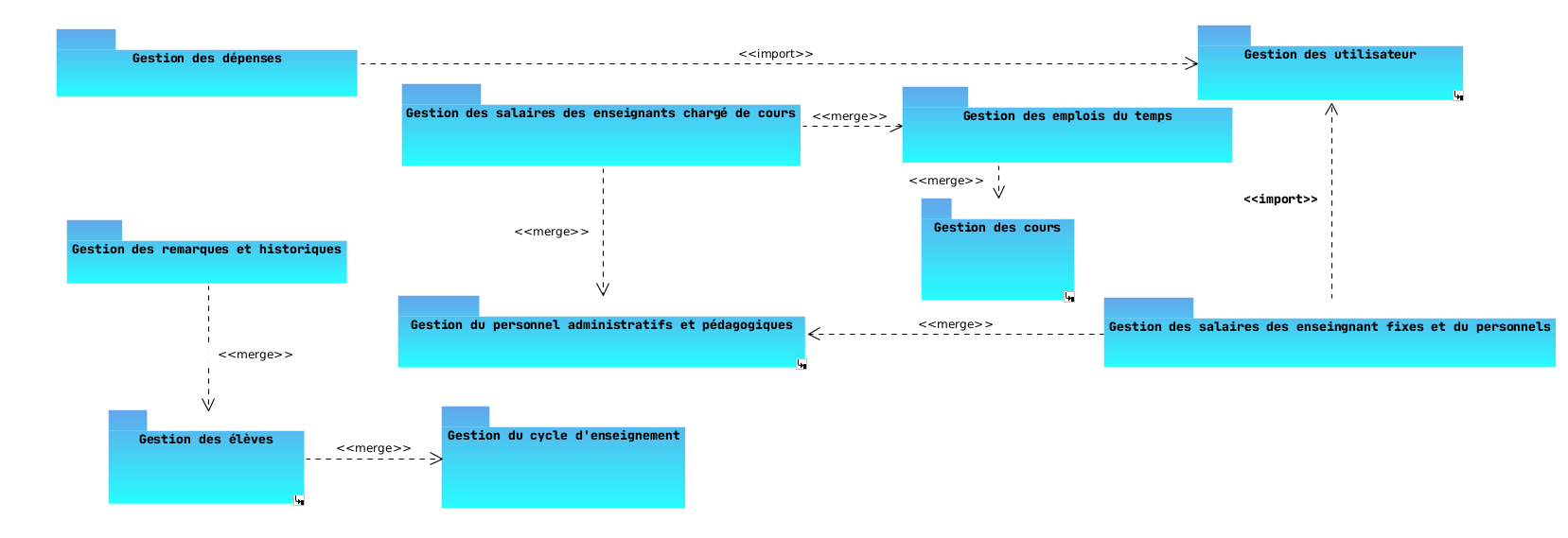
Le diagramme ci-dessous montre les différents packages notre système :

Diagramme de packages du système

* Gestion des salaires des enseignants fixe et du personnel :
* Assurer le paiement, l’affichage des salaires des enseignants fixes et du personnel
* Gardez chaque information importante sur chaque paiement (date de paiement, montant payer)
* Gestion du personnel administratifs et pédagogique
* Conserver des informations détaillées sur un enseignant, y compris des renseignements personnels, la classe dont il est responsable et son salaire par mois convenu durant une année scolaire ou un période spécifique,
* Conserver chaque information pertinente sur les personnelles de l’administration
* Gestion des salaires des enseignants chargés de cours :
* Conserver des informations détaillées sur un enseignant, y compris des renseignements personnels, matières enseignées (par année et période), salaire par heure de cours de tous les cours enseignés,
* Afficher le calcul du salaire par mois de l’enseignant en se référant à l’emploi du ou des cours qu’il compte tout tenant compte des heures supplémentaire et congés durant le période donné
* Gestion des remarques et historiques :
* Garder traces de tous les parcours scolaires antérieur d’un élève,
* Préserver chaque révélation à propos d’un élève, que ce soit une information au niveau scolaire, familiale ou comportementale.
* Gestion des élèves :
* Assurer toutes les opérations de gestion en vigueur sur une entité élève : première inscription, réinscription, niveau, cycle, classe, ⎫
* Garder toutes les informations pertinentes qui contiennent des données personnelles (date et lieu de naissance, adresse nom et prénom l’élève et de ses parents ou tuteurs, …), les informations concernant le parcours scolaire de l’élève (les établissements et classes dans lesquelles l'élève a étudié, …
* Gestion des cours :
* Assurer l’affectation d’un cours à une classe et le faire assurer par enseignant,
* Garder trace de tous les enseignants qui ont pris le cours avec la période de leur travail ainsi que le salaire convenu.
* Gestion des emplois du temps :
* Assurer d’affecter les cours d’une classe dans une année scolaire à des jours de la semaine. Tout ça, en gardant en historique tous les changements d’emploi à chaque période pour pouvoir fournir le calcul des salaires des enseignant mensuel.
  + 1. Diagramme de classe
       1. Généralités

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, et on peut sans doute dire qu’il est le seul obligatoire lors d'une telle modélisation. Contrairement au diagramme de cas d’utilisation qui montre le système du point de vue des acteurs, il montre la structure interne. Il permet de générer une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Mais aussi le diagramme de classe est une vue statique car il ne tient pas compte des facteurs temporel et dynamique dans le comportement du système.

En effet, le diagramme de classe nous donne la structure générale des classes et des relations entre eux dans un système. Chaque langage de programmation orienté objet possède son propre moyen d’implémenter le paradigme objet mais le diagramme de classe permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Composant d’un diagramme de classe

Le diagramme de classe est composé de trois (03) parties :

* Nom de la classe
* Attributs de la classe
* Méthode de la classe

Nom de la classe :

Le nom de la classe doit évoquer le concept décrit par la classe. Il commence par une majuscule.

Attributs de la classe :

Les attributs définissent des informations qu'une classe ou un objet doivent avoir. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisée. Le nom de l'attribut doit être unique dans la classe.

La visibilité déclare la possibilité pour un élément de modélisation de référencer un élément qui se trouve dans un espace différent de celui de l'élément qui établit la référence. Il existe quatre visibilités prédéfinies :

*Public* ou + :

* Tout élément qui peut voir le conteneur peut également voir l'élément indiqué.

Protected ou # :

* Seul un élément situé dans le conteneur ou un de ses descendants peut voir l'élément indiqué.

*Private* ou - :

* Seul un élément situé dans le conteneur peut voir l'élément.

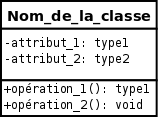
*Package* ou ∼ ou rien :

* Seul un élément déclaré dans le même paquetage peut voir l'élément.

Ici, le conteneur est la classe.

Méthode de la classe

Les opérations décrivent les éléments individuels d'un comportement que l'on peut invoquer. Ce sont des fonctions qui peuvent prendre des valeurs en entrée et modifier les attributs ou produire des résultats. Dans une classe, une opération (même nom et mêmes types de paramètres) doit être unique. Quand le nom d'une opération apparaît plusieurs fois avec des paramètres différents, on dit que l'opération est surchargée. En revanche, il est impossible que deux opérations ne se distinguent que par leur valeur retournée.

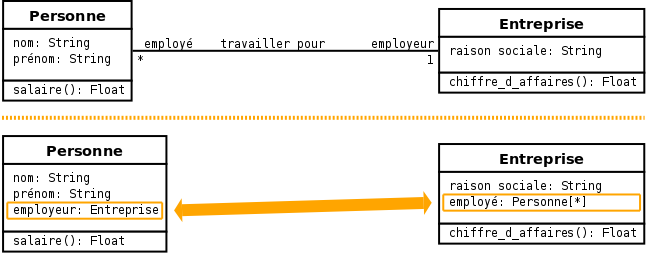
Représentation graphique

Exemple d’un diagramme de classe

Elle est représentée par un rectangle divisé en trois compartiments dont le premier indique le nom de la classe, le deuxième ses attributs et le troisième ses opérations (ou comportements).   
Relations entre classe

Une association est une relation entre deux classes (association binaire) ou plus (association n‑aire), qui décrit les connexions entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées. On peut constater les différents types de relation suivante :

* L’héritage
* L’agrégation
* La composition
* L’association bidirectionnelle
* L’association unidirectionnelle



Deux exemples de façons de modéliser une association.

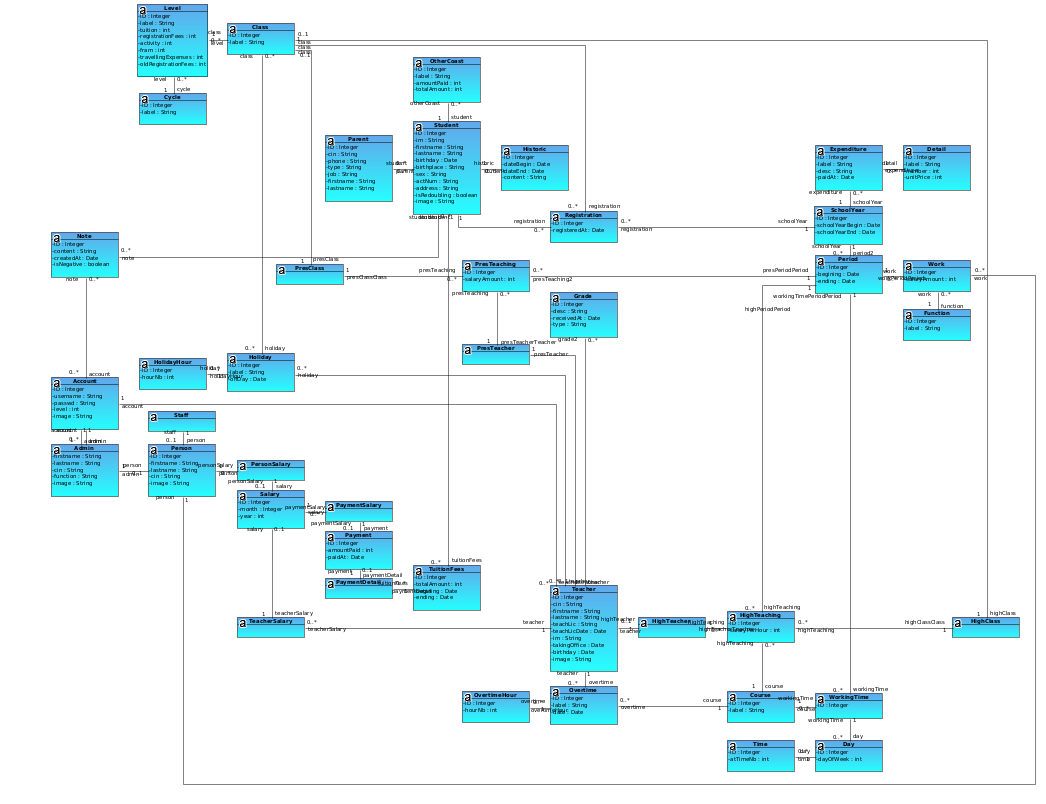
* + - 1. Diagramme de classe globale

Diagramme de classe globale.

* + 1. Conclusion

Nous avons présenté dans ce chapitre une vue conceptuelle de la solution à mettre en place selon les besoins de l’utilisateur et les améliorations proposées, en se reposant sur l’utilisation du langage de modélisation UML. Cette dernière étape nous permet d'assurer le bon fonctionnement de notre application et nous aide pendant la phase développement de notre application desktop.

Sans tarder, nous allons avancer vers le dernier chapitre consacré à la réalisation, dans lequel on va étaler l'environnement de travail de notre projet, ainsi que quelques interfaces représentants les fonctionnalités de notre application.

1. Réalisation du projet
   1. Spécification des outils de réalisation
      1. Introduction

Ce chapitre couvrira l'installation des différents outils nécessaires au développement de notre application, ainsi que le processus de développement lui-même, autant qu'un aperçu de notre application de suivi.

* + 1. Présentation des outils utilisés
       1. Le langage UML

UML (Unified Modeling Langage) est un langage de modélisation unifié qui permet de modéliser une application logicielle d'une façon standard dans le cadre de conception orientée en se libérant totalement du concept de langage de programmation.

En fournissant plusieurs moyens de description et de modélisation des acteurs et des usages du système, le comportement des objets, le flux des contrôles internes aux opérations, la mise en œuvre des composants et leurs relations, la structure matérielle et la répartition des objets et composants, UML permet de couvrir le cycle de vie d'un logiciel depuis la spécification des besoins jusqu'au codage.

* + - * 1. Les principaux diagrammes UML

Il existe neuf (09) diagrammes UML principaux qui dépendent hiérarchiquement et se complètent pour permettre la modélisation du projet tout au long de son cycle de vie. On distingue :

* Le diagramme de cas d'utilisation qui représente les fonctions du système du point de vue de l'utilisateur ;
* Le diagramme d'objets qui représente les objets et leurs relations ;
* Le diagramme de classes qui représente la structure statique en termes de classes et de relations ;
* Le diagramme de composants qui représente les composants physiques d'une application ;
* Le diagramme de déploiement qui décrit le déploiement des composants sur les dispositifs matériels ;
* Le diagramme de collaboration qui est une représentation spatiale des objets, des liens et des interactions ;
* Le diagramme de séquence qui représente temporellement les objets et leurs interactions ;
* Le diagramme d'états-transition qui décrit le comportement d'une classe d'objet en termes d’états ;
* Le diagramme d'activités montre les enchaînements des activités d'un cas d'utilisation ou d'une opération.
  + - * 1. Outils de modélisation

Il existe plusieurs types d’outils de modélisation. Mais parmi ceux qui sont utiliser pour UML, le tableau suivant nous montre leur comparaison.

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de l’outils | Type | Prix | Langage de programmation pris en charge |
| * LucidChart | * Logiciel de diagramme Web | * Abonnement gratuit et abonnement Premium | * JavaScript, HTML5, SQL |
| * MagicDraw | * Logiciel propriétaire avec licence utilisateur unique, licence flottante et mobile | * Paiement unique relativement élevé | * Java, C++, C#, CIL, XML, CORBA, WSDL, EJB, DDL, IDL |
| * Visual Paradigm | * Existe en version web et desktop | * Paiement de licence mais existe une version community gratuit | * Java, C++ et tous les langages SQL |

Comparaison des outils de modélisation pour UML

On considère Visual paradigme comme le plus adéquat pour la modélisation qu’on va effectuer. En effet, cet outil, en plus des avantages cités dans le tableau précédent :

* Inclut une large gamme de diagramme : 13 types de diagramme ;
* Est facile à manipuler ;
* Intègre une fonctionnalité d’utilisation d’Hibernate ;
* Permet de générer directement des codes.
  + - * 1. Le langage et l’IDE

Langage

Pour faire une application de bureau, il existe plusieurs langages et le tableau ci-dessous montre leur comparaison :

|  |  |  |
| --- | --- | --- |
| Nom du langage | Avantages | Inconvénient |
| C++ | * Utilisation de Qt pour faire une interface * Un exécutable facile à déployer | * Langage proche du système donc difficile à utiliser |
| Python | * Langage avancé, existence de nombreuse librairie * Utilisation de PyQt pour faire une interface | * Application difficile à déployer |
| Java | * Existence de plusieurs outils pour faire une interface * Code réutilisable pour faire une application Android * Une large communauté | * Nécessité de toujours travailler dans une classe |

Comparaison des différents langages

Java est un langage de programmation populaire et puissant qui sert également de plate-forme. Il s'agit d'un langage de programmation de haut niveau et sécurisé qui est utilisé sur diverses plates-formes à travers le monde, notamment les applications Web, Android, Big Data, les services bancaires, informatiques et financiers. Et dans notre cas, nous avons choisi Java pour la réalisation de notre application vu qu’une des perspectives de notre projet et de fournir une application mobile à l’enseignant de l’établissent.

L’IDE

Pour implémenter le langage de programmation Java, nous avons besoin de certains environnements dans lesquels nous pouvons développer l’application. Alors dans le tableau subséquent nous allons comparer les différent IDE pour le langage de programmation Java :

|  |  |  |
| --- | --- | --- |
| Nom | Avantages | Inconvénients |
| NetBeans | * NetBeans permet aux développeurs de déployer le code à partir de son propre environnement. * Les utilisateurs peuvent formater et définir des règles pour toutes les langues | * Le traitement est parfois lent |
| Eclipse | * Possède une bonne fonction d'intégration pour créer des outils comme ANT et Maven * Les utilisateurs peuvent développer différentes applications sur la même plate-forme, telles que des applications Web et autonomes, des services Web | * Sans directives et documentation appropriées, la configuration initiale peut être difficile. |
| IdeaIntelliJ | * Intellij est apte à détecter les blocs de code répétitifs et à afficher les erreurs avant la compilation * IntelliJ Idea est livré avec une grande variété d'outils intégrés tels que GIT, le contrôle de version, le décompilateur, la couverture, la base de données SQL * **Il possède une méthode de complétion intelligent** | * La courbe d'apprentissage est raide et la documentation des outils pourrait être améliorée. |

Comparaison des différent IDE de Java

* + - * 1. Choix de la technologie

Il existe deux (02) différentes technologies pour faire une interface avec java, ainsi nous allons faire leur comparaison dans le tableau suivant :

|  |  |  |
| --- | --- | --- |
| Nom de la technologie | Avantages | Inconvénients |
| Java Swing | * Possède de nombreuse composant | * Une interface utilisateur basique * Une technologie obsolète * Manque support pour l’architecture MVC |
| JavaFX | * Possibilité de faire une interface riche, moderne et intuitive * Possibilité de beaucoup d’amélioration dans le futur * Basé sur l’architecture MVC | * Possède peu de composant |

Comparaison des technologies pour l’interface graphique en Java

Suivant le constat contenu dans le tableau ci-dessus nous avons opter pour JavaFX pour la création d’interface graphique de notre application en vue d’une interface utilisateur plus moderne et plus facile à utiliser.

* + - * 1. La technologie orientée objet

La programmation orientée objet (POO) est un paradigme de programmation informatique qui consiste en la définition et l'interaction de briques logicielles appelés objets ; un objet représente un concept, une idée ou toute entité physique comme une voiture, une personne ou même une page de livre Il a une structure en son sein.et le comportement, et il comprend comment interagir avec ses pairs. Par conséquent, il est nécessaire de représenter ces objets ainsi que leurs interactions.

L'interaction entre les objets via leurs relations permet la conception et la réalisation des fonctionnalités attendues, permettant une meilleure solution au(x) problème(s). En conséquence, l'étape de modélisation est critique et nécessaire pour la POO. C'est elle qui permet la virtualisation des éléments de la réalité.

Dans notre cas, confirmer que java est un langage orienté objet, ce concept sera alors utilisé tout au long de la réalisation de notre application.

* + - * 1. Système de Gestion des Bases de Données

Généralité sur les SGBD

Un SDBG est un logiciel système conçu pour stocker et partager des informations dans une base de données en garantissant la qualité, la durabilité et la confidentialité des informations tout en masquant la complexité des opérations. Le SDBG a un large éventail de capacités. En conséquence, nous pouvons insérer, rechercher, modifier et imprimer des données. Il permet la création de rapports basés sur des données enregistrées et comprend des processus pour assurer l’homogénéité des données, éviter les pertes dues à des pannes et assurer la confidentialité.

Choix de la SGBD

Afin de faire le bon choix pour le SGBD, nous allons voir dans le tableau 8.5 suivante :

|  |  |  |  |
| --- | --- | --- | --- |
|  | MySQL | PostgreSQL | Oracle |
| Performance | Moyen | Bien | Bien |
| Licence | Gratuit | Gratuit | Payant |
| Langage supporté | C, C#, C++, Java, Ruby, et Objective C | C, C#, C++, D, Java, Ruby, et Objective C | C, C#, C++, Java, Ruby, et Objective C |

Comparaison entre quelques SGBD relationnel.

Avec notre projet nous avons choisi d 'utiliser postgreSQL à raison de sa performance car, notre application exécute de nombreuses requêtes durant son utilisation.

* 1. Mise en œuvre et implémentation
     1. Architecture logicielle
        1. Architecture MVC (Model-View-Controller)

L'architecture MVC est une méthode d'organisation de l'interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes : le modèle, la vue et le contrôleur, chacune ayant un objectif précis dans l'interface.

L'organisation globale d'une interface graphique est souvent difficile. Bien que la façon dont MVC organise une interface ne soit pas une solution miracle, elle fournit souvent un point de départ qui peut ensuite être adapté. Il fournit également un cadre pour l'organisation d'une application.

Les rôles des trois entités dans l'architecture MVC sont les suivants :

* Modèle : données (accès et mise à jour)
* Vue : interface utilisateur (entrées et sorties)
* Contrôleur : gestion des événements et synchronisation
  + - * 1. Rôle du modèle

Le composant "Modèle" regroupe et collecte toutes les données. Il contient, après tout, les classes dont les instances doivent être visualisées et traitées. Le modèle spécifie quelles données doivent être incluses dans l'application. Si l'état de ces données change, le modèle avertira généralement la vue (afin que l'affichage puisse changer au besoin) et occasionnellement le contrôleur (si une logique différente est requise pour contrôler le changement de vue). Il encapsule l'accès aux données.

* + - * 1. Rôle de la vue

La vue sert d'interface utilisateur. Sa première tâche consiste à afficher les données du modèle qu'il a récupérées. Sa deuxième tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'entrée, boutons, etc.) Ses différents événements sont transmis au contrôleur.

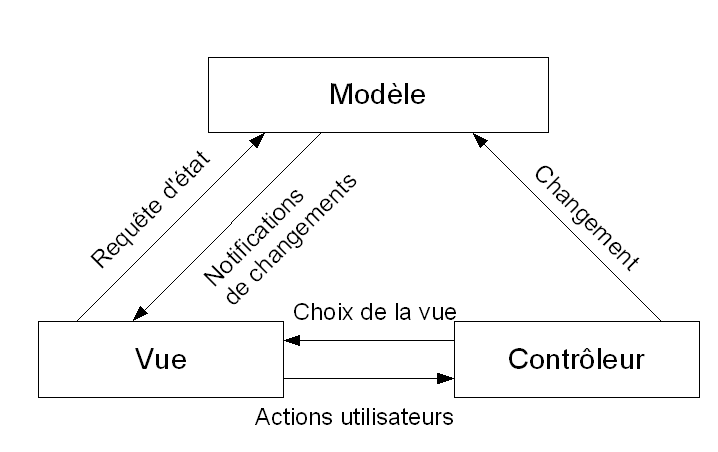
La vue peut également fournir plusieurs vues, partielles ou complètes. Par exemple, la donnée unique de l'application de conversion de base de données est un entier. Ce même entier est affiché de diverses manières (sous forme de texte dans diverses bases de données, bit par bit avec des cases à cocher et avec des curseurs). La vue peut également offrir à l'utilisateur la possibilité de modifier la vue.

* + - * 1. Rôle du contrôleur

Le contrôleur synchronise le modèle avec la vue. Tous les événements utilisateurs sont reçus et les actions à effectuer sont initiées. Si une modification de données est nécessaire pour une action de données, le contrôleur demande que les données du modèle soient modifiées, puis informe que les données ont été modifiées pour mettre à jour la vue. Certains événements utilisateur concernent la vue, pas les données. Le contrôleur demande un changement dans ce cas.

* + - * 1. Fonctionnement

Le fonctionnement de ces différentes parties sont illustrées par la figure 9.1 suivante :

Architecture MVC.

* + 1. Architecture matérielle

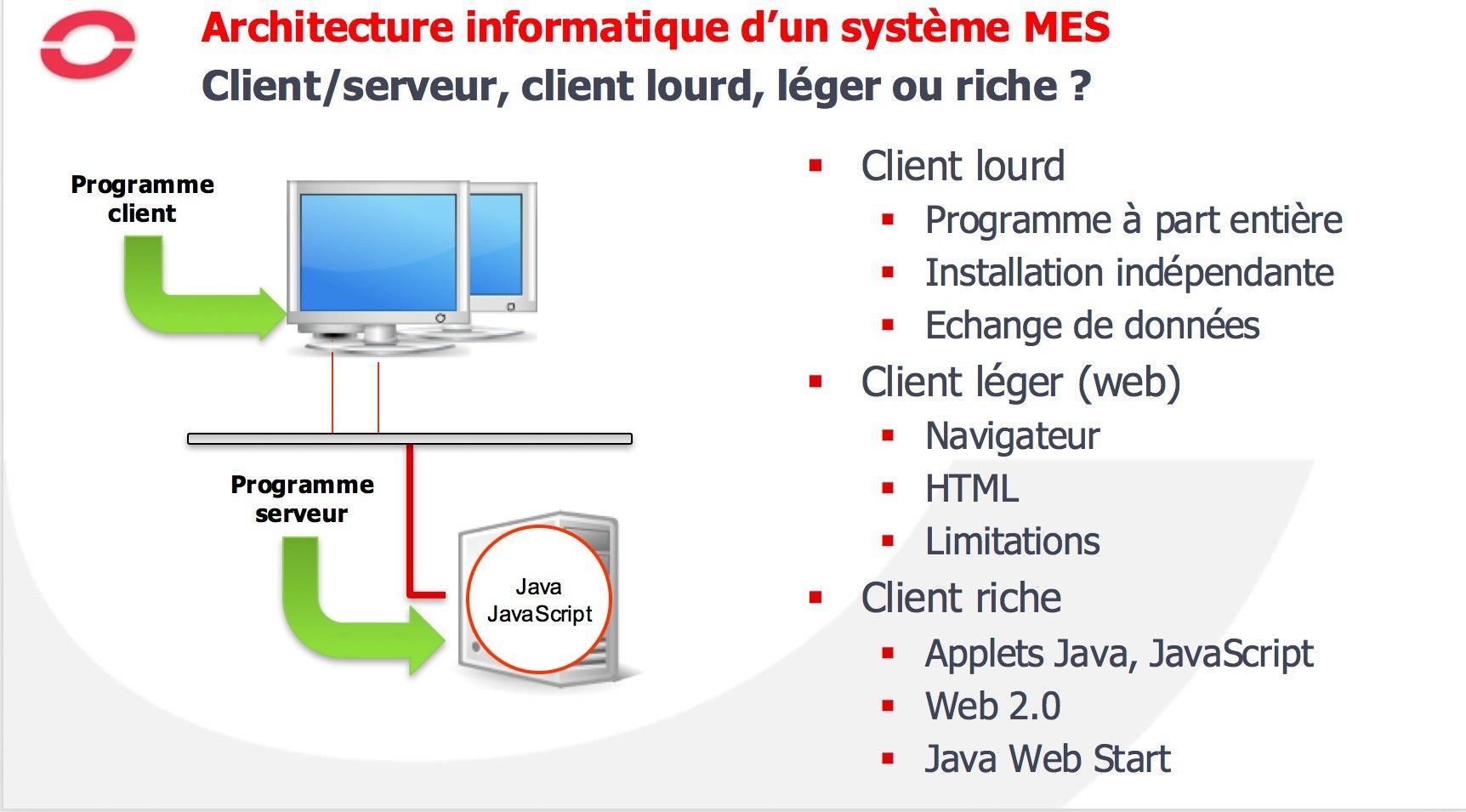
Les architectures matérielles sont désormais beaucoup plus similaires à quelques exceptions près. Presque tous les architectes s'appuient sur des ordinateurs de type PC, parfois complétés par d'autres appareils tels que des terminaux, des tablettes et des smartphones. Généralement, plusieurs ordinateurs sont mis en réseau et un ou plusieurs ordinateurs fonctionnent comme serveur, et les autres ordinateurs servent à des fonctions clientes. Il existe cependant trois (03) types d’architecture matériel, et ces derniers sont répertorié dans la liste suivante :

* Client lourd
* Client léger
* Client riche

Mais dans notre cas, on utilisera l’architecture client lourd.

* + - 1. Description

Cette architecture offre plusieurs avantages : en termes d'apparence et de graphisme, elle offre à l'utilisateur la même interface riche qu'une application de bureau classique. Comme une grande partie du traitement est effectuée localement, l'interface "répond" rapidement quelle que soit la vitesse du réseau et certaines opérations peuvent être effectuées alors que le réseau n'est pas opérationnel.

figure d’un client lourd

* + 1. Les différents modules de l’application
       1. Le paiement de l’écolage

Le module de gestion des paiements des écolages dans notre application de suivi des élèves et enseignants permet de prendre en compte les paiements en retard, les facilités de paiement et les ajournements de paiement. Il nous permet de suivre de manière efficace les paiements des étudiants en fournissant des options de paiement flexibles pour les aider à s'acquitter de leurs frais de scolarité, tout en assurant un suivi rigoureux des paiements en retard. Le module facilite également la communication entre l'école et les parents ou tuteurs en fournissant des rappels de paiement et des informations sur les ajournements et les facilités de paiement, une partie du code qui s’occupe du paiement de l’écolage est la suivante :



Code source pour le paiement de l’écolage

* + - 1. Le paiement des salaires des enseignants

Le module de gestion des paiements des salaires des enseignants permet de calculer les salaires des enseignants en prenant en compte plusieurs paramètres. Tout d'abord, les heures de cours effectuées par les enseignants sont comptabilisées et rémunérées en conséquence. Ensuite, si un enseignant remplace un autre professeur lors d'un cours, les heures effectuées sont également prises en compte dans le calcul de son salaire. De même, les heures de surveillance d'examens sont également comptabilisées.

De plus, les enseignants ont la possibilité de demander des avances sur salaire, ce qui est également pris en compte dans le module de gestion. Les avances sont déduites du salaire final.

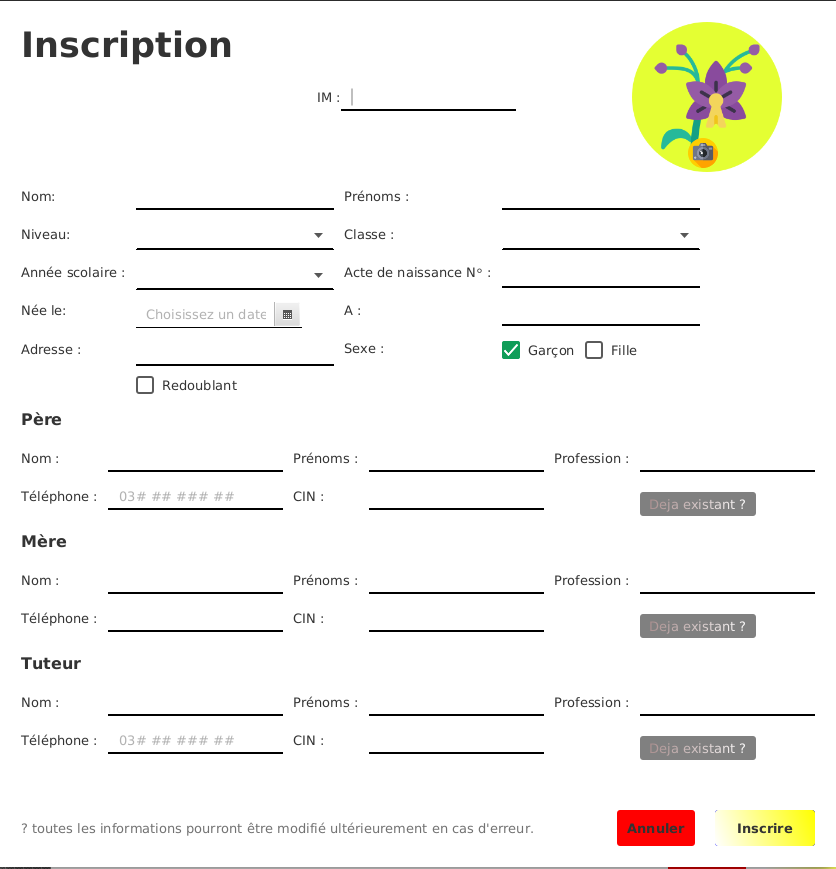
En résumé, ce module de gestion permet de prendre en compte tous les éléments qui peuvent impacter le salaire d'un enseignant, y compris les demandes d'avance sur salaire, les heures de cours effectuées, les heures de remplacement et les heures de surveillance d'examens. Cela permet de calculer de manière précise le salaire final de chaque enseignant en fonction de leur activité.

L’image suivante nous montre une partie de code de ce module :

Code source pour le paiement du salaire d’un enseignant

* 1. Présentation de l’application développée

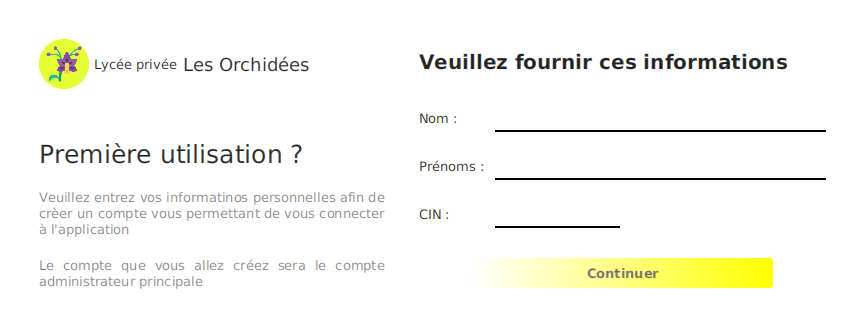
Les figures suivantes illustres l’une des fonctionnalités importantes de l’application :



Formulaire d’inscription des élèves



Formulaire pour gérer du temps des différentes classes



Formulaire lors de la première connexion

* + 1. Conclusion

Ce chapitre avait pour objectif de décrire l'environnement technique et les différentes étapes de l'application.

En fait, l'architecture technique a été décrite en premier.

Ensuite, nous avons identifié les différentes phases de l'étape de développement et finalement présenté quelques interfaces décrivant certaines fonctionnalités de l'application.

CONCLUSION GÉNÉRALE

L'étude, l'analyse, la conception et l'exécution de l’application de gestions financière et de suivi des élèves et des enseignants ont fait l'objet de notre projet de fin d'études au sein du lycée privée « Les Orchidées ». En effet, cette étape pratique de notre formation à l'analyse, la conception et la programmation nous a conduit à satisfaire les besoins réels en fonction des objectifs et des contraintes constatés à partir des problèmes aperçus.

Lors de ce stage, l’une des principales contraintes était le fait d’intégrer un système de gestion dans un environnement où il n’y a aucun réseau de communication entre les ordinateurs. Mais après réflexion, on a pu arriver à le résoudre en créant une architecture de type « client lourd » au sein des bureaux de l’établissement.

On peut dire que la réalisation de cette application répond aux besoins des administrateur et résout le problème de la gestion de l’inscription, d’embauche et du paiement des salaires du personnel administratifs et pédagogique au sein de l’établissement. L’application génère automatiquement les salaires et écolages sans devoir les ajouter chaque mois pendant l’année scolaire. Et même si l’année scolaire s’allonge ou se raccourci, l’application ici présent sera capable de recalculer tous les salaires tels que les écolages des élèves. Cela garantit le bon fonctionnement de l’établissement, n'entraîne aucune perte financière au niveau administratif. Ainsi que l’adoption d'une politique d'utilisation du système d'information sécurisé est assurée pas cette application.

Ce stage nous a permis d'établir un lien entre nos connaissance universitaires et celle de la profession. En effet, il nous a permis de développer nos compétences techniques, d'approfondir et de mettre en pratique nos connaissances théoriques. Cela a renforcé nos remontées mécaniques d'analyse et de synthèse et nous a permis, notamment d’approfondir nos connaissances du langage UML et de la mise en place des architectures de type « client lourd ».

De plus, les améliorations futures à apporter au niveau des applications doivent être envisagées par d'autres. Et puisque notre application tire ses contenues dans une base de données déjà prête pour de nombreuse autre fonctionnalité à venir, améliorer l’application ou même intégrer de

Nouvelle technologie sont des actions des plus faisable sans perdre de temps ni prendre des difficultés.

BIBLIOGRAPHIE

Nous tous chez monday.com, 2021. Guide de la gestion de projet informatique [en ligne].

Mondayfrancais. Disponible sur : <https://monday.com/blog/fr/the-newbies-guide-to-it-project-management/> (consulté le 15/04/2021)

Larreur G.,2021. Qu’est-ce-qu’un cahier des charges informatique [en ligne]. Ozytis. Disponible sur : <https://ozytis.fr/cahier-des-charges-informatique/ (consulté le 15/04/2021)

Client lourd et client léger », https://www.supinfo.com/articles/single/3424-client-lourdclient-leger, du 05 Mars 2021

« Architecture d’un système d’information », http://www.guillaumeriviere.name/ estia/si/pub/SI\_COURS-01\_2012\_introduction.pdf, (consulté le 5/04/2021).

« UML », http://openclassrooms.com /courses/26832-apprenez-a-programmer-enjava/21530-modeliser-ses-objets-grace-a-uml, (consulté le 17/04/2021).

RÉSUMÉ

Ce projet décrit comment une application desktop pour gestion financière et de suivi des élèves et enseignants est conçu et mis en œuvre. Il montre la gestion d'un projet informatique au sein d'une entreprise, montrant les différentes étapes qui ont été franchies et les différents outils utilisés. Nous avons utilisé le langage de modélisation UML pour modéliser une étape importante dans le déroulement d'un projet informatique. Pour ce projet, nous nous sommes penchés sur la technologie javaFx, comme son nom l’indique il est basé sur le langage JAVA et que nous avons utilisé IdeaIntellij pour l’IDE et postgreSQL pour la SGBD. Afin de bien structurer notre application et faciliter l'adoption de l'architecture MVC

Mots-clés : MVC, SGBD, UML, javaFX, IdeaIntellij ;

ABSTRACT

This project describes how a student and teachers desktop application has been designed and implemented for financial management and monitoring. It shows how an IT project is managed in a company and shows the different phases and tools used. In order to model an important progress step in the IT project, we have used the UML modeling language. We looked at JavaFx technology for this project because its name suggests that it is based on JAVA and IdeaIntellij is used for the DBMS and posgreSQL for the IDE. To structure our application properly and facilitate the implementation of the MVC architecture.

Keywords : MVC, DBMS, javaFX, IdeaIntellij