UNIVERSITÉ DU BURUNDI
GRENIER DU SAVOIR DU BURUNDI
Dépôt institutionnel officiel
https://repository.ub.edu.bi

Mise en place d'un système automatise et sécurisé d'évaluation en ligne des étudiants

NKENGURUTSE, Pacifique

2023-03
UB
https://repository.ub.edu.bi/handle/123456789/482

Document téléchargé depuis le **Grenier du Savoir du Burundi** — https://repository.ub.edu.bi

République du Burundi

Ministère de l'Education Nationale et de la Recherche Scientifique



Master en Génie Informatique

Année Académique : 2021-2022

Université du Burundi

Faculté des Sciences de l'Ingénieur

MISE EN PLACE D'UN SYSTEME AUTOMATISE ET SECURISE D'EVALUATION EN LIGNE DES ETUDIANTS

MEMOIRE

Présenté Par

NKENGURUTSE Pacifique

à la

Faculté des Sciences de l'Ingénieur (FSI)

En vue de l'obtention du grade de

MASTER

en

Génie Informatique

Soutenu le 01/03/2023, devant le jury composé de :

Pr. NDIKUMAGENGE Jérémie	: Président
Dr. NKUNZIMANA Hilaire	: Vice-Président
Dr. SAHINGUVU William	: Secrétaire
Dr. NDAYISABA Longin	: Directeur
Dr. MUKESHIMANA Michele	: Membre

IDENTIFICATION DES MEMBRES DU JURY

Prof.	NDIKUMAGENGE Jérémie	: Président du Jury
Dr.	NKUNZIMANA Hilaire	: Vice-Président du Jury
Dr.	SAHINGUVU William	: Secrétaire du Jury
Dr.	NDAYISABA Longin	: Directeur de Thèse
Dr.	MUKESHIMANA Michele	: Membre du Jury

DEDICACES

A Dieu tout puissant ;

A mon regret Père qui nous a quittés sitôt ;

A ma très chère maman ;

A mes frères et sœurs ;

A mes neveux ;

A mes Cousins et Cousines ;

A tous mes camarades de classe ;

A tous mes amis et connaissances.

REMERCIEMENTS

J'adresse ma plus profonde gratitude à Dieu le Tout Puissant, le très Miséricordieux pour m'avoir accordé la vie et le courage de réaliser ce travail jusqu'au bout.

Nos plus vifs remerciements s'adressent à ma chère maman qui n'a pas cessé de me donner du courage et de fournir tous ces efforts durant toute la période de mes études. Tu es un exemple de courage et détermination, une vraie exemplarité et inspiration de ma vie !

Mes sincères remerciements s'adressent à Dr Longin NDAYISABA, enseignant à l'Université du Burundi dans la Faculté des Sciences de l'Ingénieur pour m'avoir guidé dans la réalisation de ce travail malgré ses multiples et lourdes fonctions. Ses conseils judicieux et sa rigueur scientifique m'ont été d'une très grande utilité. Je tiens à remercier également toute personne qui, de près ou de loin, a participé à la réalisation de ce travail, à mes éducateurs de l'école primaire jusqu'à l'Université, plus particulièrement à tous les enseignants du département de Technologie de l'Information et de la Communication.

Nos remerciements vont aussi aux membres du jury qui ont accepté d'examiner ce travail et de nous accompagner durant le moment de la défense de ce mémoire.

Je remercie vivement l'Université du Burundi pour m'avoir fourni le matériel nécessaire durant toute la période de mon cursus. Je ne peux pas aussi oublier tous mes amis et camarades de classe.

Enfin, que toutes personnes qui m'ont assisté tant matériellement que moralement au cours de mes études, trouvent dans ce travail le couronnement de leur générosité.

RESUME

Le système éducatif dans le monde entier est un système qui gère le processus d'enseignement et de distribution des cours dans des salles de classes qu'en ligne. Après la distribution de chaque cours, chaque enseignant doit évaluer manuellement les étudiants pour prouver leurs connaissances et compréhensions sur un sujet particulier. Un étudiant est sanctionné par une note qu'il obtiendra dans une évaluation effectuée dans des murs de classe. Aujourd'hui, avec la gratuite de la scolarité, le nombre élevé des étudiants constitue un problème majeur lors de l'évaluation. De plus, chaque année, l'Etat doit donner à l'université du Burundi les frais logistiques pour les papiers sur lesquels on imprime les questionnaires des évaluations. Le nombre élevé des étudiants, les frais logistiques pour l'Etat, les cas de tricheries qui s'observent ainsi que les frais de déplacement vers les salles des évaluations constituent aussi un problème à résoudre. Le présent projet de fin d'études de master vise à mettre fin à tous ces problèmes énoncés ci-haut en mettant en place un système automatisé et sécurisé d'évaluation en ligne des étudiants afin de pallier tous ces problèmes.

Dans la deuxième partie, nous avons entamé la modélisation du système en utilisant le langage de modélisation UML, et la troisième partie nous avons développé la théorie des files d'attentes comme modèle mathématique permettant d'appliquer à notre système les outils, les techniques et les théories mathématiques, puis généralement, en sens inverse, la traduction des résultats mathématiques obtenus en prédictions ou opérations dans le monde réel. La sécurité des données transitant sur le réseau est préoccupante raison pour laquelle nous avons décidé d'utiliser le chiffrement de Vigenère pour éviter tout accès non autorisé aux données.

Enfin, nous avons implémenté le système en utilisant de multiples technologies comme le Framework PHP Codeigniter, la librairie JQuery Data table pour dynamiser les recherches, MYSQL comme système de gestion des bases de données, Sublime Text comme éditeur de texte. Une fois le système est mis en place, il permettra aux enseignants d'évaluer et corriger facilement les classes possédant de nombreux étudiants, aux étudiants de faire les évaluations en ligne afin d'économiser les frais des papiers et de déplacement lors des évaluations, de diminuer voire éradiquer les cas de tricheries, réduire le taux de transmission des maladies transmissibles ainsi que les frais logistique des papiers pour l'Etat.

Mots clés : Système éducatif, questionnaires des évaluations, évaluation, système automatisé, sécurisé, évaluation en ligne, modélisation, UML, Codeigniter.

ABSTRACT

The education system worldwide is a system that manages the process of teaching and distributing courses both in classrooms and online. After each course has been delivered, each teacher must manually assess the students to prove their knowledge and understanding of a particular subject. A student is awarded a grade in a classroom-based assessment. Today, with free tuition, the large number of students is a major problem when it comes to assessment. What's more, every year the government has to provide the University of Burundi with the logistical costs of the paper on which the evaluation questionnaires are printed. The high number of students, the logistical costs for the state, the cases of cheating that occur and the cost of travel to the evaluation rooms also constitute a problem to be solved. The aim of this final-year Master's project is to put an end to all these problems by setting up an automated and secure online student evaluation system.

With this in mind, in the first part we studied the existing system. In the second part, we began modeling the system using the UML modeling language, and in the third part we developed queuing theory as a mathematical model for applying mathematical tools, techniques and theories to our system, and then generally, in reverse, translating the mathematical results obtained into real-world predictions or operations. The security of data passing over the network is a major concern, so we decided to use Vigenère encryption to prevent unauthorized access to the data.

Finally, we implemented the system using multiple technologies such as the Codeigniter PHP framework, the JQuery Data table library for dynamic searches, MYSQL as a database management system, and Sublime Text as a text editor. Once the system is up and running, it will enable teachers to easily assess and correct classes with large numbers of students, students to take assessments online to save paperwork and travel costs during assessments, reduce or even eradicate cases of cheating, reduce the rate of transmission of transmissible diseases as well as the logistical costs of paperwork for the state.

Keywords: Educational system, assessment questionnaires, assessment, automated system, secure, online assessment, modeling, UML, Codeigniter.

TABLE DES MATIERES

IDENTIFICATION DES MEMBRES DU JURY i

DEDICACES ii

REMERCIEMENTS iii

RESUME iv

ABSTRACT v

TABLE DES MATIERES vi

LISTES DES FIGURES ix

LISTE DES TABLEAUX x

LISTE DES SIGLES ET ABREVIATIONS xi

AVANT-PROPOS xiii

CHAPITRE I : INTRODUCTION GENERALE 1

I.1 Justification du contexte 1

I.1. 1 Généralités 1

I.2. Objectifs du projet 2

I .2.1 Objectif global 2

I.2.2 Objectifs spécifiques 2

I.3. Problématiques 3

I.4. Solutions proposées 3

I.5. Résultats attendus 4

I.6 Apports scientifiques et technologiques 4

I.7 Délimitation du sujet 4

I.8 Méthodologie de recherche utilisée 5

I.9 Outils de modélisation et de développement utilisés 5

I.9.1 Outils de modélisation 5

I.9.2 Outils de développement utilisés 5

CHAP II : ANALYSE DU SYSTEME D'INFORMATION 6

II.1 Introduction 6

II.2 Méthode d'analyse et de conception 6

II.3 Etapes du cycle de vie du logiciel 7

II.3. 1 Analyse et définition des besoins 7
II.3.2 Analyse et conception 7
II.3.3 Mise en œuvre 7
II.3.4 Validation 8
II.3. 5 Evolution et maintenance 8
II.4 Modèle du cycle de vie du développement logiciel 8
CHAP III : MODELISATION DU SYSTEME AVEC LE LANGAGE UML 10
III.1 Introduction 10
III.2 Outils de modélisation UML 10
III.3 Modélisation du système d'information 10
III.3 .1 Axe statique 11
III.3 .2 Axe dynamique 13
CHAP IV : MODELISATION MATHEMATIQUE DU SYSTEME PAR LES FILES D'ATTENTE 23
IV.1 Introduction 23
IV.2 Définition 24
IV.2.1 Notation de Kendall 24
IV.3 Caractéristiques d'une file d'attente 24
IV.4 Réseaux de files d'attente 26
IV.4.1 Classification des réseaux de file d'attente 26
IV.4.2 Processus d'arrivée de Poisson 27
IV.5 Principaux modèles des files d'attente 31
IV.6 Processus de naissance et de mort 34
CHAP V : IMPLEMENTATION DU SYSTEME DE GESTION DU PROCESSUS D'EVALUATION EN LIGNE DES ETUDIANTS 43
V.1 Introduction 43
V.2 Architecture client/serveur 43
V.3 Outils et langages de développement utilisés 44
V.4 Technologie de développement utilisée dans notre projet 45

CHAP VI : PROTECTION DES DONNEES CONTRE LES ACCES NON AUTORISES 47

VI.1 Introduction 47

VI.2 Protection des données par le chiffrement de Vigenère 48

VI.2 .1 Types de chiffrement des données 48

VI.2 .2 Chiffrement asymétrique 49

VI.2 .3. Chiffrement symétrique 49

VI.3 Chiffrement de Vigenère comme outils de sécurité 50

VI.3 .1 Cryptanalyse du chiffrement de Vigenère 53

VI.3 .2 Application du chiffrement de Vigenère dans notre projet 54

VI .4 Estimation du coût de mise en œuvre du logiciel 55

VI.5 Méthode COnstructive COst MOdel 55

VI.5.1 Méthode COCOMO II 57

VI.5.2 Méthode WEBMO 57

VI .6 Calcul du coût de mise en place du logiciel 58

VI .7 Calendrier de réalisation du projet 59

VI. 8 Présentation de quelques fenêtres du système développé 61

VI.8.1 Authentification 61

VI.8.2 Interfaces d'accueil des différents utilisateurs 62

CONCLUSION GENERALE ET RECOMMENDATIONS 66

Conclusion générale 66

RECOMMANDATIONS 67

BIBLIOGRAPHIE 68

Ouvrages généraux 68

WEBOGRAPHIE 68

LISTES DES FIGURES

Figure 1: Modelé de développement d'un logiciel 9
Figure 2: Algorithme de fonctionnement général du système. 11
Figure 3:Diagramme de classe. 12
Figure 4:Diagramme de déploiement. 13
Figure 5:Diagramme de cas d'utilisation. 15
Figure 6:Diagramme d'activité pour le cas d'utilisation « Elaborer un questionnaire ». 19
Figure 7:Diagramme de séquence du cas d'utilisation « Se connecter » 20
Figure 8: Diagramme de séquence du cas d'utilisation « Ajouter ». 21
Figure 9:Diagramme de séquence du cas d'utilisation « Rechercher ». 21
Figure 10: Réseau mono-classe fermé et réseau mono-classe ouvert. 27
Figure 11:Diagramme de transition entre états. 35
Figure 12: Modèle MVC 46
Figure 13:Processus de cryptage et de décryptage de l'algorithme asymétrique. 49
Figure 14:Processus de cryptage et de décryptage de l'algorithme symétrique. 50
Figure 15:Application de chiffrement avec chiffrement de Vigenère. 54
Figure 16:Déchiffrement avec chiffrement de Vigenère. 55
Figure 17: Diagramme de Gantt des activités du projet. 61
Figure 18:Interface d'authentification. 61
Figure 19:Interface de l'enseignant 62
Figure 20: Elaboration des questions ouvertes. 63
Figure 21:Interface d'accueil de l'étudiant. 63
Figure 22:Interface des réponses pour les questions ouvertes 64
Figure 23:Interface pour la correction et l'attribution des points. 64

LISTE DES TABLEAUX

Tableau 1:Description brève du cas d'utilisation « s'authentifier » 14
Tableau 2:Description détaillée du cas d'utilisation « S'authentifier » 16
Tableau 3:Description détaillée du cas d'utilisation « Faire une évaluation » 17
Tableau 4:Description détaillé du cas d'utilisation « Elaborer un questionnaire » 18
Tableau 5:Diagramme d'activité du cas d'utilisation « S'authentifier » 19
Tableau 6:Table de Vigenère 51
Tableau 7:Paramètres du modèle COCOMO simple 56
Tableau 8:Valeurs Paramètres du modèle COCOMO simple 56
Tableau 9:Paramètres WebMo 58
Tableau 10:Chronogramme des activités 59

LISTE DES SIGLES ET ABREVIATIONS

AES : Advanced Encryptions System

AJAX : Asynchronous JavaScript And XML

B2B : Business To Business

BAC : Baccalauréat

BDD : Base De Données

COCOMO : COnstructive COst Model

CSS : Cascading Style Sheets

DES : Data Encryption System

FIFO : First In First Out

HM : Homme-Mois

HTML : HyperText Markup Language

KLSL : Kilo Lignes Sources Livrées

LIFO : Last In Last Out

MDN : Multiple Domain Network

Msc-Ir : Maitre en Science de l'Ingénieur

MVC : Modèle Vue Contrôleur

MySQL : My Structured Query Language

NTIC : Nouvelles Technologies de l'Information et de la Communication

P2P : Peer to Peer

PDM : Processus de Décision Markovien

PHP : HyperText Preprocessor

PKI : Public Key Infrastructure

PS : Processing Sharing

RSA : Rivest Shamir Adleman

SI : Système d'Information

SLOC : Source Line Of Code

SQL : Structured Query Language

TDEV : Temps de Développement

TIC : Technologie de l'Information et de la Communication

UGS : UserGroup Web Service Protocol

UML : Unified Modeling Language

USSD : Unstructured Supplementary Service Data

WIP : Work In Process

XAMPP : X Apache MariaDB Perl PHP

XLSX : eXcel Spreadsheet

AVANT-PROPOS

Comme le stipule le sujet de ce travail « Mise en place d'un système automatisé et sécurisé d'évaluation en ligne des étudiants. Ce travail a été réalisé dans le but de mettre fin aux problèmes liés à l'évaluation des classes possédant un nombre élevé des étudiants. Nous avons subdivisé ce travail en six chapitres.

Le premier chapitre est consacré à la présentation du projet de recherche en mettant accent sur l'étude de l'existant.

Dans le deuxième chapitre, nous avons vu l'analyse du système d'information .Le troisième chapitre consiste à modéliser le système à l'aide des diagrammes UML.

Dans le quatrième chapitre, nous avons développé la théorie des files d'attentes comme modèle mathématique permettant d'appliquer à notre système les outils, les techniques et les théories mathématiques, puis généralement, en sens inverse, la traduction des résultats mathématiques obtenus en prédictions ou opérations dans le monde réel.

Le cinquième chapitre consiste à implémenter le système en utilisant de multiples technologies comme le Framework PHP Codeigniter, la librairie JQuery Data table pour dynamiser les recherches, MYSQL comme système de gestion de base de données, Sublime Text comme éditeur de texte.

Le sixième chapitre consiste à sécuriser les données à l'aide du chiffrement de Vigenère pour éviter tout accès non autorisé aux données.

En fin, une conclusion générale consiste à récapituler les principaux résultats de ce travail et mettre en pratique tous ces théories et techniques de développement utilisés pour développer le sujet de mémoire.

CHAPITRE I : INTRODUCTION GENERALE

I.1 Justification du contexte

I.1. 1 Généralités

Par nature l'être humain est doté d'une capacité et une volonté d'apprendre continuellement comme le dit Charles Perrault dans son ouvrage Querelle des anciens et des modernes. Le processus d'apprentissage chez l'être humain se fait distinctement des autres animaux car il est qualifié plus intelligent que les autres. Par son intelligence, l'homme n'a cessé d'améliorer le système d'apprentissage jusqu'à la formation des écoles et universités. A l'heure actuelle, les étudiants ou apprenants sont civilisés par les enseignants qualifiés travaillant comme employés des écoles ou des universités. A l'université, après la formation, chaque enseignant doit évaluer manuellement les étudiants et chaque étudiant est sanctionné par une note qu'il obtiendra. Au cours du processus d'évaluation, il faut du temps pour faire tous les travaux sur papiers et obtenir la validité des résultats. C'est un processus couteux, qui consomme beaucoup des ressources et qui prend du temps. Il y a aussi des cas où certains documents sont égarés et manquent de sécurité. L'évaluation par les enseignants est un processus devenant pénible suite au nombre élevés des étudiants ce qui aura comme impact retard de la publication et affichage des résultats. De plus, les frais de logistiques pour les papiers, frais de papiers aussi pour les étudiants, perte des questionnaires des évaluations, les cas de tricheries qui s'observent sont des problèmes liés à la gestion manuelle du système d'évaluation actuel. En revanche l'informatique en tant que science d'automatisation des tâches semble inévitable dans ce siècle afin de chercher à exterminer les embarras liés à la lenteur des services rendus dans plusieurs voire toutes institutions. Nous avons décidé de mettre en place un système automatisé d'évaluation en ligne des étudiants qui sera une solution excellente pour extirper tous ces problèmes mentionnés ci-haut. La transformation du système manuel en système automatisé facilitera la correction, la disponibilité des résultats en un laps de temps, le gain du temps pour la publication des résultats, la conservation des documents dans un endroit sûr et sécurisé. L'évaluation en ligne par les enseignants avec un ordinateur ou support Android améliorera le système actuel. Les documents seront cryptés et conservés à distance dans un endroit sécurisé accessible à tout moment pendant une longue période ce qui constitue une solution incontournable pour les cas de perte des documents. Au surplus, les travaux pour les enseignants diminueront aussi en raison du nouveau système que nous avons proposé.

I.2. Objectifs du projet

I .2.1 Objectif global

L'objectif global est de concevoir un système automatisé et sécurisé d'évaluation en ligne des étudiants qui facilitera aux enseignants en général et particulièrement aux étudiants et à l'Etat dans la résolution des problèmes liés aux nombres élevés des étudiants, aux frais de papiers et logistiques, longue période d'attente pour publier les résultats après les évaluations et frais de déplacement vers les salles des évaluations, aux pertes des papiers qui s'observent dans certains cas.

I.2.2 Objectifs spécifiques

Stocker en toute sécurité les papiers des questions d'examen et il est directement visible par les étudiants qui ont participé à l'examen et évaluer les classes possédant un nombre élevé des étudiants. Les objectifs spécifiques de notre système sont :

1) Concevoir et développer une application accessible à tous et compatible avec tous les terminaux (Smartphones et Ordinateurs) ;
2) Atténuer la fatigue aux enseignants qui évaluent les classes possédant un nombre élevé des étudiants ;
3) Stocker en toute sécurité les papiers des questions d'examen et il est directement visible par les étudiants qui ont participé à l'examen afin de pallier aux tricheries ;
4) Economiser le coût des papiers, car l'examen se déroulera en ligne ;
5) Réduire le coût de la logistique pour livrer les papiers de questionnaire au centre d'évaluations ;
6) Economiser le temps de l'institution qui a organisé l'évaluation et de l'étudiant qui a participé à l'évaluation ;
7) Participer à l'examen n'importe où en utilisant n'importe quel dispositif électronique dans lequel la facilité d'accès d'internet est disponible ;
8) Gérer les étudiants, les enseignants et les horaires des examens ;
9) Réduire le temps de publication du résultat de l'évaluation parce que le résultat sera généré en un seul clic ;
10) Stocker les copies des évaluations des années passées dans un endroit sûr et sécurisé et rapidement accessible pour tous (étudiants et enseignants) avec la connexion internet.

I.3. Problématiques

Etant donné qu'actuellement le processus d'évaluation se fait manuellement, au cours de ce processus, s'il s'agit des classes comprenant un nombre élevé d'étudiants il faut beaucoup du temps pour faire tous les travaux sur papiers et corriger toutes les copies d'évaluations afin d'obtenir la validité des résultats, les cas de tricherie qui s'observent, il y a aussi des cas où certains documents sont égarés et manquent de sécurité. De plus, les examens sur papiers nécessitent beaucoup de papiers pour imprimer les feuilles de questions et de réponses. Il y a également beaucoup de gaspillage dû aux erreurs d'impression, sans parler aussi de la logistique liée aux feuilles. Toutes ces lacunes mentionnées ci-haut nous ont amenés à se poser une question si l'analyse, conception et développement d'un système automatisé et sécurisé d'évaluation en ligne des étudiants ne serait-il pas un remède pour ces problèmes ?

I.4. Solutions proposées

Notre solution primordiale vise à analyser, concevoir et développer un système automatisé et sécurisé d'évaluation en ligne des étudiants. Après la mise en place de ce système nous avons proposé comme solution concevoir une application web permettant l'évaluation des étudiants à distance ,correction automatique des évaluations et publication des résultats en un laps de temps ,concevoir une base de données stockant les évaluations faites ainsi que les informations y relatives ,génération automatique des rapports ,chiffrement des données stockées dans la base de données avec un cryptosystème fort ,accessibilité facile à l'information et consultation des résultats à chaque instant si l'étudiant ou l'enseignant veut consulter les résultats.

I.5. Résultats attendus

Après la mise en place de notre système, de nombreux problèmes seront résolus et voici ci-dessous les résultats :

1. Possibilité d'évaluer un nombre illimité des étudiants ;
2. Apaisement de la fatigue pour les enseignants qui évaluent des classes possédant un effectif élevé des étudiants ;
3. Réduction du temps d'attente entre la passation des évaluations et la publication des résultats ;
4. Stockage des données dans un endroit sécurisé ;
5. Un gain remarquable en termes de ressources humaines et de temps est observé ;
6. Publication instantanée des résultats ;
7. Réduction des cas de tricheries ;
8. Diminution des cas de transmission des maladies surtout pendant les périodes de pandémie.

I.6 Apports scientifiques et technologiques

Ce travail apporte une application de la théorie des files d'attente comme outils mathématique permettant de modéliser le système et apporte aussi des principes sur lesquels tournent l'application, le développement d'un système de chiffrement de Vigenère pour garantir la sécurité des données transitant sur le réseau ,la mise en place d'un système automatisé et sécurisé de gestion du processus d'évaluation en ligne des étudiants constitue un apport technologique et un sujet de recherche pour l'Université du Burundi et aussi pour les futurs chercheurs et étudiants de l'université désirant apporter des améliorations à notre système.

I.7 Délimitation du sujet

Notre travail a été délimité dans l'espace, dans le temps et dans le domaine. Dans l'espace, notre système sera utilisé dans le domaine éducatif spécialement par l'Université du Burundi ainsi que d'autres universités désirant intégrées ce système , Dans le temps, il a été réalisé pendant une période de six mois dans le cadre d'un projet de fin d'études du cycle de Master en Sciences de l'Ingénieur dans le département des TICs envie de l'obtention du diplôme de Msc-Ir, Dans le domaine, il se limite sur la gestion du processus d'évaluation en ligne des étudiants afin que les évaluations se fassent en ligne.

I.8 Méthodologie de recherche utilisée

Compte tenu de la nature du sujet, la lecture des livres et des mémoires, des articles scientifiques et la recherche sur internet sont aussi des techniques beaucoup utilisées pour aboutir à la mise en place de notre système. Nous avons utilisé aussi les modèles du cycle de développement du logiciel pour développer et mettre en place ce système.

I.9 Outils de modélisation et de développement utilisés

I.9.1 Outils de modélisation

Pour finaliser notre travail nous avons utilisé la théorie des files d'attentes comme outils mathématique utilisé pour modéliser le système d'évaluation en ligne. C'est dans cette perspective que cet outil a permis de montre en concret comment le système fonctionne tout en permettant de montre comment un client c'est-à-dire un étudiant pour notre cas demande du service auprès du serveur c'est-à-dire un enseignant lui répond en lui donnant les ressources qui lui sont réservés. Le modèle mathématique permet de montre et calculer les paramètres nécessaires pour qu'ils n'y auraient pas effet d'étranglement de la demande de service auprès du serveur.

I.9.2 Outils de développement utilisés

Pour développer notre système nous avons utilisés les outils de développement comme le Framework Codeigniter 3.1.12 pour le développement de l'application, UML pour la conception du système d'information, le Framework front end BOOTSTRAP 4, librairie DOMPDF pour générer des rapports en PDF, librairie JavaScript DATATABLE pour dynamiser les tableaux, MySQL comme système de gestion de bases de données et Apache comme serveur web pour tester l'application. Dans ce premier chapitre nous avons développé la raison d'être du projet (le contexte et la justification du projet). L'objectif du projet, la problématique, les solutions proposées, les résultats attendus, Apports scientifiques et technologiques, méthodologie utilisée, outils de modélisation et développement du système ont été évoqués à l'issue de ce chapitre.

CHAP II : ANALYSE DU SYSTEME D'INFORMATION

II.1 Introduction

Nous aborderons dans cette partie les notions et concepts utiles nous permettant d'évoquer l'analyse informatique dans les termes appropriés afin de dégager les éléments constitutifs du système d'information. Processus est un ensemble d'opérations, logiquement liées, aboutissant à certains résultats. En conception de systèmes d'information, selon la méthode Merise, le processus se situe au niveau du modèle conceptuel de traitement. [1]. Information encore appelée donnée est un élément de connaissance susceptible d'être codé pour être conservé, traité ou communiqué. Un système est un ensemble de composants en interaction destiné à accomplir une tâche donnée. C'est le cas par exemple des systèmes de production, systèmes de transport, systèmes informatiques, etc... [2]. Le système d'information c'est l'ensemble des ressources de l'entreprise qui permettent la gestion de l'information. Le SI est généralement associé aux technologies, aux processus qui les accompagnent, et aux hommes qui les supportent [3]. Le SI est composé tout d'abord des individus, ce sont toutes personnes qui interagissent avec le système. Elles sont concernées soit en utilisant des informations pour réaliser leurs tâches, soit en participant aux événements liés à l'enregistrement, traitement ou échange des informations. Ils sont aussi les concepteurs et les gestionnaires des systèmes d'information. Ensuite, il y a les matériels, c'est à partir d'eux que les précédents parviennent à réaliser tout ce dont ils ont besoin. Il y a également les logiciels et applications, ce sont des programmes nécessaires au fonctionnement du système d'information informatisé. Il y a enfin des données qui constituent l'information en soi. Parler du système d'information, n'est pas parler d'un système informatique, il n'est qu'un sou ensemble.

II.2 Méthode d'analyse et de conception

En ingénierie, une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client [4]. Pour ce faire, on part d'un énoncé informel (le besoin tel qu'il est exprimé par le client, complété par des recherches d'informations auprès des experts du domaine fonctionnel, comme les futurs utilisateurs d'un logiciel), ainsi que de l'analyse de l'existant éventuel (c'est-à-dire la manière dont les processus à traiter par le système se déroulent actuellement chez le client). La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation [5]. Elle permet aussi de lister les résultats

attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité etc.

II.3 Etapes du cycle de vie du logiciel

Le développement logiciel désigne l'ensemble du processus consistant à bâtir tout type d'applications informatiques fiables et performantes et va de l'étude du besoin du client, en passant par la conception, la mise en œuvre jusqu'à la maintenance de l'application. Ces diverses étapes sont possibles grâce à un langage informatique spécifique ou plutôt grâce à plusieurs langages et à des développeurs maitrisant ces langages [6]. Pour désigner les différentes étapes du développement d'un logiciel, de sa conception à sa fin de vie, on parle de « cycle de vie d'un logiciel ». Le développement de logiciel suit obligatoirement un certain nombre d'étapes.

II.3. 1 Analyse et définition des besoins

L'étape d'analyse et définition des besoins consiste à déterminer les attentes des futurs utilisateurs, par exemple avec un cahier des charges. Il faut décrire à la fois le système et l'environnement dans lequel le système sera exécuté. Cette étape comprend également une étude de faisabilité de la part des experts.

II.3.2 Analyse et conception

L'étape d'analyse et conception consiste à analyser, spécifier et effectuer les choix de conception du système. Cette étape comporte plusieurs sous-étapes comme spécification du système qui est une description des fonctionnalités. Il s'agit de décrire ce que le système doit faire, sans préciser comment ces fonctionnalités seront implémentées, conception de l'architecture qui consiste à décrire la structure générale du système et conception détaillée qui consiste en une description des composants, des algorithmes et des structures de données.

II.3.3 Mise en œuvre

L'étape de mise en œuvre consiste à programmer le logiciel, en suivant les choix effectués lors de l'analyse et la conception. Le développement logiciel désigne le processus consistant à bâtir des applications informatiques, qu'elles soient élaborées par l'entreprise pour son propre compte ou par un éditeur qui les commercialise. Ce processus s'appuie sur des langages particuliers conçus pour donner à la machine des instructions à exécuter.

II.3.4 Validation

La validation consiste à s'assurer que le programme est de qualité. Il existe plusieurs techniques de validation : Analyse statique (typage, conventions de programmation, détection d'erreurs pouvant survenir à l'exécution), preuve formelle (couteuse, peu utilisée), Revue de code (efficace) et les tests qui constituent la principale méthode de validation. On distingue les tests unitaires, les tests d'intégration, les tests système et les tests d'acceptation.

II.3. 5 Evolution et maintenance

L'étape d'évolution et de maintenance consiste à effectuer des modifications du logiciel après sa livraison. On distingue plusieurs types de maintenance : Maintenance corrective (correction de défauts), Maintenance évolutive (ajout de nouvelles fonctionnalités) et Maintenance adaptative (portage sur une nouvelle plate-forme).

II.4 Modèle du cycle de vie du développement logiciel

Les cycles de vie du développement logiciel sont des « plans de travail » qui permettent de planifier le développement. Plus le logiciel à développer est complexe (taille, algorithmes) et critique, plus il est important de bien contrôler le processus de développement et plus les documents qui accompagnent le logiciel doivent être précis et détaillés. Dans le modèle en cascade les différentes étapes du cycle de vie du logiciel sont effectuées de façon séquentielle. Les interactions ont lieu uniquement entre étapes successives : on s'autorise des retours en arrière uniquement sur l'étape précédente. Par exemple, un test ne doit pas remettre en cause la conception architecturale. Le modèle en V du cycle de vie du logiciel précise la conception des tests : Les tests système sont préparés à partir de la spécification, les tests d'intégration sont préparés à partir de la conception architecturale et les tests unitaires sont préparés à partir de la conception détaillée des composants. Le modèle incrémental est un modèle itératif, qui consiste à sélectionner successivement plusieurs incréments. Un incrément du logiciel est un sous-ensemble du logiciel complet, qui consiste en un petit nombre de fonctionnalités. Le modèle du cycle de vie en spiral est un modèle itératif, où la planification de la version se fait selon une analyse de risques. L'idée est de s'attaquer aux risques les plus importants assez tôt, afin que ceux-ci diminuent rapidement. Le modèle par prototypage est un modèle dans lequel un ou plusieurs prototypes sont soumis au client pour évaluation ou révision de chaque étape. Il permet également d'examiner et d'explorer certains aspects du système pour évaluer et choisir les meilleures stratégies [7]. Dans ce chapitre, nous avons introduit la notion de système d'information. Nous avons développé en premier lieu les éléments constitutifs du système

d'information, par après nous avons inséré la notion d'étapes du cycle de vie d'un logiciel qui nous a montré les étapes à suivre lors du développement d'un système d'information et enfin nous avons vu les modèles du cycle de vie d'un logiciel qui sont des éléments indispensables dans le développement des systèmes d'information. Tous ces éléments vus nous ont donné l'envie de faire la conception du système en utilisant le langage de modélisation UML. Le chapitre suivant va faire la conception et la modélisation du système à l'aide des diagrammes que comporte le langage UML.

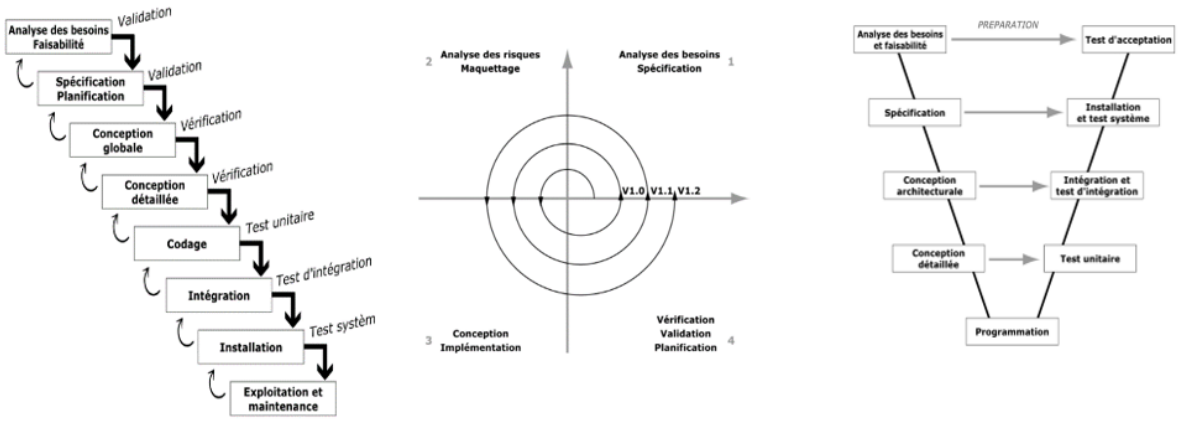


Figure 1: Modelé de développement d'un logiciel

CHAP III : MODELISATION DU SYSTEME AVEC LE LANGAGE UML

III.1 Introduction

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML est un langage, plus précisément une notation graphique, de modélisation à objets dont ces concepteurs avaient initialement les buts de représenter des systèmes entiers (pas uniquement logiciels) par des concepts objets, lier explicitement des concepts et le code qui les implantent, pouvoir modéliser des systèmes à différents niveaux (pour permettre d'appréhender des systèmes complexes) et créer un langage de modélisation utilisable à la fois par les humains et les machines.

III.2 Outils de modélisation UML

Dans cette partie, nous allons faire une brève présentation du langage UML, nous allons essayer de définir d'une manière brève les diagrammes UML que nous avons utilisé pour faire une représentation visuelle du système du processus d'évaluation en ligne. De manière générale, UML comporte 13 types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ces diagrammes se répartissent en deux grands groupes qui sont les diagrammes structurels ou diagrammes statiques et les diagrammes comportementaux ou diagrammes dynamiques. La différence entre ces deux catégories de diagramme UML est qu'une vue statique permet de représenter la structure du modèle sans tenir compte de l'évolution au cours du temps. Une vue dynamique représente au contraire les changements qui interviennent au cours du temps.

III.3 Modélisation du système d'information

Dans ce point, nous mettons en pratique la théorie UML vue ci-dessus, en modélisant notre nouveau système d'information. En vue de construire le modèle du système d'information du processus d'évaluation en ligne, il nous faut représenter les deux axes de modélisation du standard UML à savoir l'axe statique ou structurel et l'axe dynamique ou comportemental. Nous ne pouvons pas modéliser tous les 13 diagrammes UML mais nous avons choisi certains de ces diagrammes jugés plus importants. Nous avons utilisé 5 diagrammes dont 2 décrivent la représentation statique du système (diagramme de classe et diagramme de déploiement) et 3

autres pour la représentation dynamiques (diagramme de cas d'utilisation, diagramme de séquence et diagramme d'activités). Avant de faire une modélisation UML avec certains diagrammes, nous avons décidé de vous montre schématiquement la structure fonctionnelle de notre système :

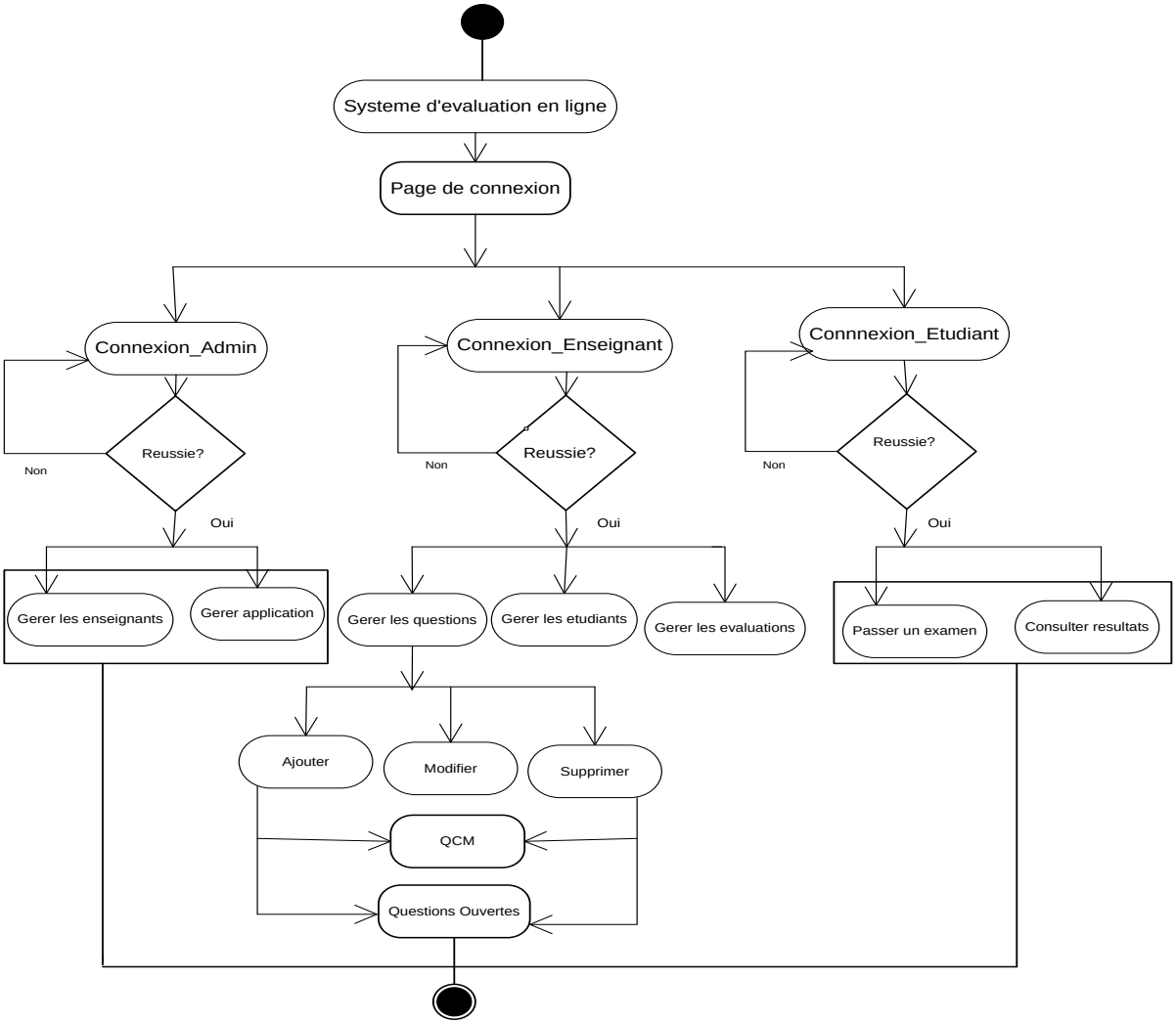


Figure 2: Algorithme de fonctionnement général du système.

III.3 .1 Axe statique

Les diagrammes statiques qui nous ont intéressés beaucoup pour pouvoir implémenter le code, il s'agit de diagramme de classes et diagramme de déploiement. Le diagramme de classe est généralement considéré comme le plus important dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet, ou au mieux les modules du langage de développement [8]. Le schéma

ci-dessous nous donne une vue globale de notre application. On a les classes principales qui vont nous servir à réaliser l'application :

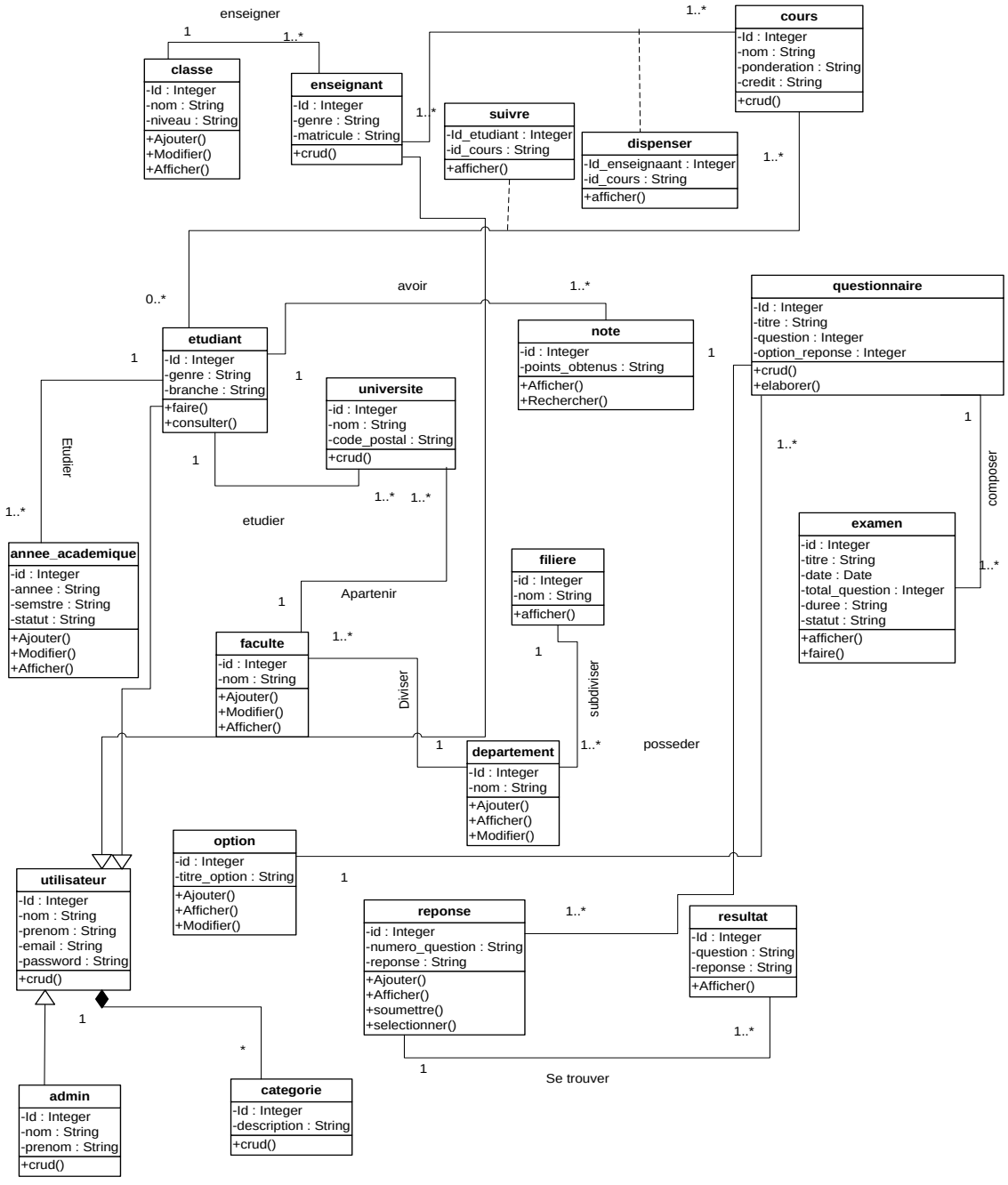


Figure 3:Diagramme de classe.

Diagramme de déploiement

Un diagramme de déploiement est un type de diagramme UML qui montre l'architecture d'exécution d'un système, y compris les nœuds tels que les environnements d'exécution

matériels ou logiciels, et l'intergiciel qui les relie [9]. Les diagrammes de déploiement sont généralement utilisés pour visualiser le matériel physique et les logiciels d'un système. En l'utilisant, vous pouvez comprendre comment le système sera physiquement déployé sur le matériel. Les diagrammes de déploiement aident à modéliser la topologie matérielle d'un système par rapport à d'autres types de diagrammes UML qui décrivent principalement les composants logiques d'un système.

Voici ci-dessous le diagramme de déploiement du système :

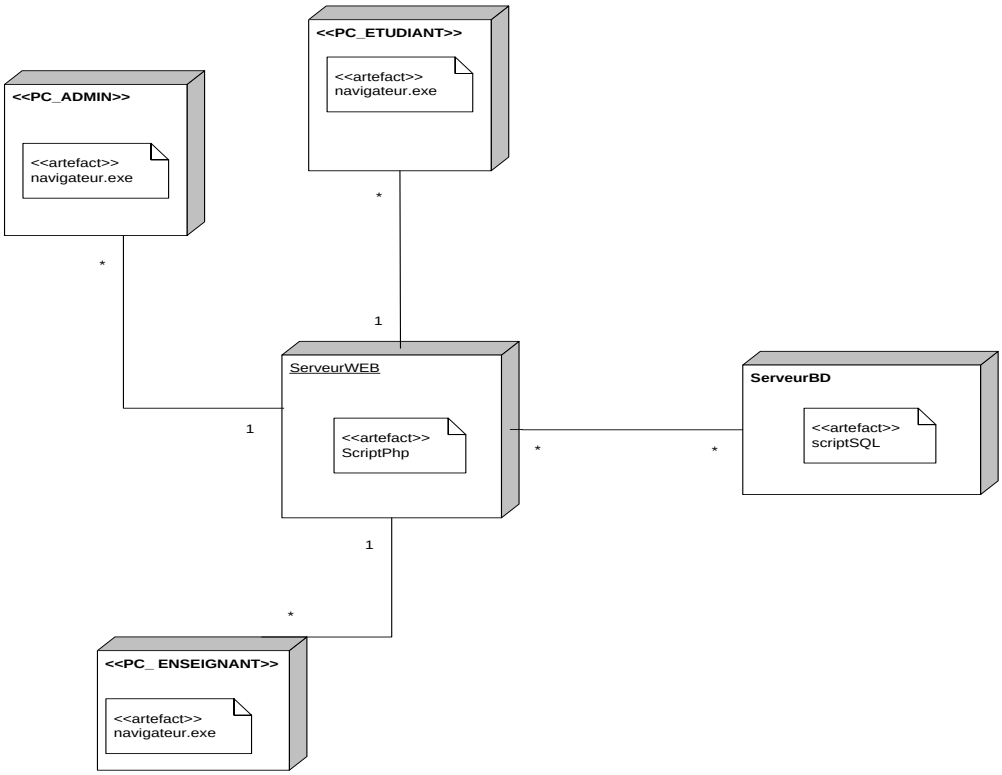


Figure 4:Diagramme de déploiement.

III.3 .2 Axe dynamique

Diagramme de cas d'utilisation

Le schéma ci-dessous nous donne une vue du diagramme de cas d'utilisation pour authentification des utilisateurs.

L'identification des acteurs et des cas d'utilisation de notre système.

Le tableau ci-dessous illustre l'identification des cas d'utilisation de notre système :

Tableau 1:Description brève du cas d'utilisation « s'authentifier »

ACTEURS	**CAS D'UTILISATION**
Tous utilisateurs	Authentification
Administrateur	Gérer les utilisateurs
Enseignant	Gérer les étudiants Ajouter de nouvelles universités Ajouter de nouvelles facultés Créer de nouvelles classes Elaborer un questionnaire Télécharger les fichiers
Etudiant	Participer à l'évaluation Consulter les résultats Télécharger un fichier

Description brève du cas d'utilisation « s'authentifier »

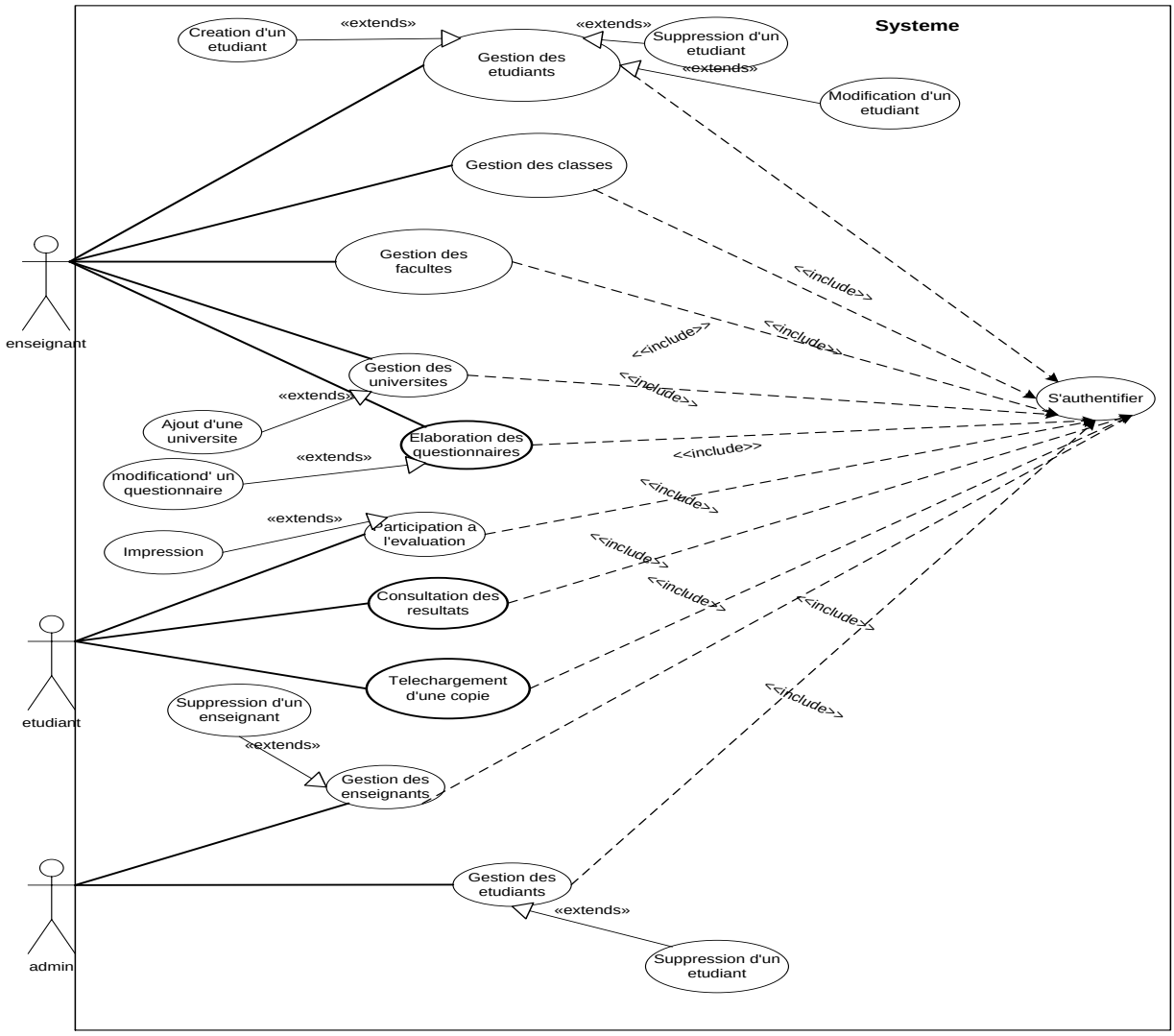


Figure 5:Diagramme de cas d'utilisation.

Description détaillée de certains cas d'utilisation

Un tel travail de recherche doit comporter plusieurs cas d'utilisation afin de répondre convenablement aux besoins des utilisateurs. Nous ne pouvons pas faire la description sommaire ni la description de tous les cas d'utilisation raison pour laquelle nous avons décidé de décrire brièvement certains cas d'utilisation jugés primordiaux et utiles.

Description du cas d'utilisation « s'authentifier »

Chaque utilisateur du système accède à son espace de travail après avoir s'authentifier avec succès. Si ses coordonnes d'authentification sont incorrects il ne peut pas avoir accès à son espace membre, c'est pourquoi au départ l'utilisateur s'authentifie en saisissant dans l'espace réservé son nom d'utilisateur et son mot de passe et après il sera redirigé automatiquement dans son espace si ses coordonnées sont correctes, le cas échéant reste sur la même page d'authentification et les droits et privilèges lui seront attribués selon sa fonction.

Tableau 2:Description détaillée du cas d'utilisation « S'authentifier »

Acteurs	Chaque utilisateur du système
Précondition	Tous utilisateurs enregistrés dans la base de données doivent connaître ses identifiants (Email/username et password) pour pouvoir se connecter.
Post-condition	L'utilisateur connecté. Il peut exécuter toutes les fonctions qui lui sont attribuées dans son espace selon son profil ou se déconnecter du système.
Scénarios normaux	Le système affiche un formulaire d'authentification où l'utilisateur saisit ses coordonnées de connexion, le système vérifie et valide les informations saisies. Le système redirige ensuite l'utilisateur vers sa page d'accueil selon son profil.
Scénarios anormaux	Un messag`e d'erreur s'affiche si les identifiants de connexion sont erronés. L'utilisateur peut essayer de se reconnecter encore.

Description du cas d'utilisation « Faire une évaluation »

Un étudiant enregistré dans le système a une possibilité de faire une évaluation qui lui est attribué et voir les résultats après la passation de l'évaluation.

Tableau 3:Description détaillée du cas d'utilisation « Faire une évaluation »

Acteurs	Etudiant
Précondition	Chaque étudiant enregistré dans la base de données doit connaître ses identifiants (Email/username et password) pour pouvoir se connecter.
Post-condition	L'étudiant peut faire l'évaluation et voir la note obtenue
Scénarios normaux	L'étudiant consulte toutes les informations qui lui concerne et qui ont été mises dans son profil mais sans aucune possibilité de les modifier. La note obtenue par l'étudiant est gardée inchangeable l'étudiant n'a pas le droit de la modifier.
Scénarios anormaux	Impossible de participer à l'évaluation.

Description du cas d'utilisation « Elaborer un questionnaire »

La phase d'élaboration d'un questionnaire permet à un enseignant enregistré et authentifié correctement de pouvoir préparer un questionnaire qui sera la future évaluation des étudiants. Dans son profil il a le droit de créer plusieurs questionnaires et faire inscrire les étudiants.

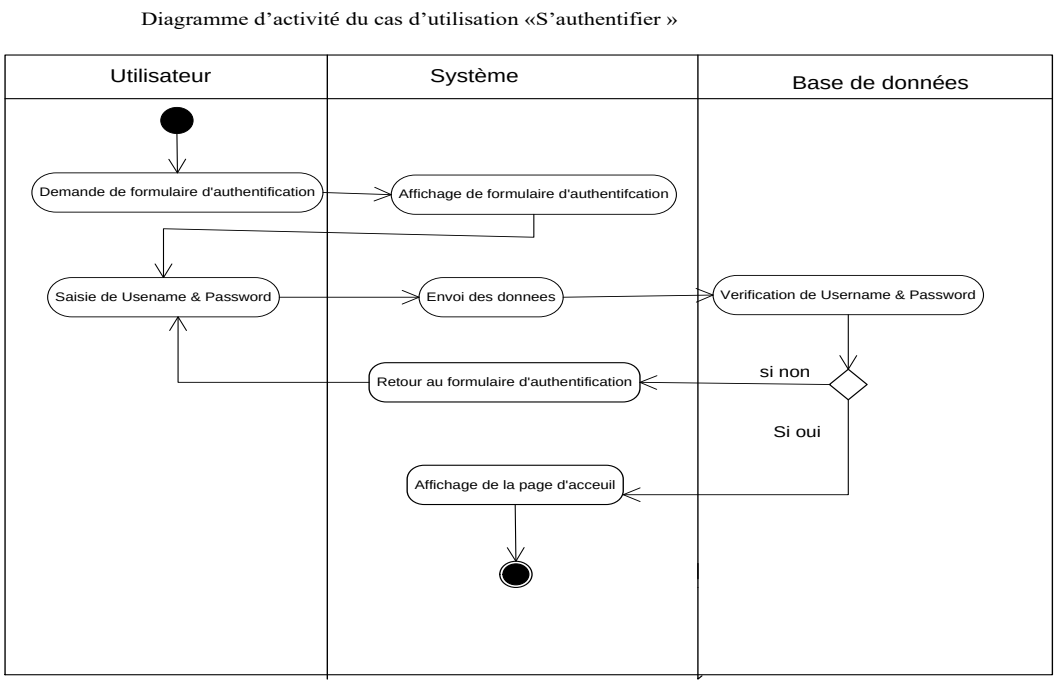
Tableau 4:Description détaillé du cas d'utilisation « Elaborer un questionnaire »

Acteurs	Enseignant
Précondition	Chaque enseignant enregistré dans la base de données doit connaître ses identifiants (Email/username et password) pour pouvoir se connecter.
Post-condition	L'enseignant peut élaborer un questionnaire d'évaluation et livrer l'évaluation aux étudiants qui ont le droit de participer.
Scénarios normaux	L'enseignant consulte toutes les informations qui lui concernent et qui ont été mises dans son profil mais à la possibilité de les modifier mais pas en totalité. La note obtenue par l'étudiant est gardée non falsifiable l'enseignant n'a pas le droit de la changer.
Scénarios anormaux	Impossible d'élaborer un questionnaire.

2. Diagramme d'activités

Dans le langage UML, un diagramme d'activité fournit une vue du comportement d'un système en décrivant la séquence d'actions d'un processus. Les diagrammes d'activité sont similaires aux organigrammes de traitement de l'information, car ils montrent les flux entre les actions dans une activité. Les diagrammes d'activité peuvent, cependant, aussi montrer les flux parallèles simultanés et les flux de remplacement [10]. Dans les diagrammes d'activité, vous utilisez des nœuds d'activité et des bords d'activité pour modéliser le flux de commande et de données entre les actions. Les diagrammes d'activité sont utiles avant de démarrer un projet, vous pouvez créer des diagrammes d'activité pour modéliser les principaux flux de travaux pendant la phase d'exigences, vous pouvez créer des diagrammes d'activité pour illustrer le flux d'événements décrit dans les cas d'utilisation et pendant les phases d'analyse et de conception, vous pouvez utiliser des diagrammes d'activité pour faciliter la définition du comportement des opérations.

Tableau 5:Diagramme d'activité du cas d'utilisation « S'authentifier »



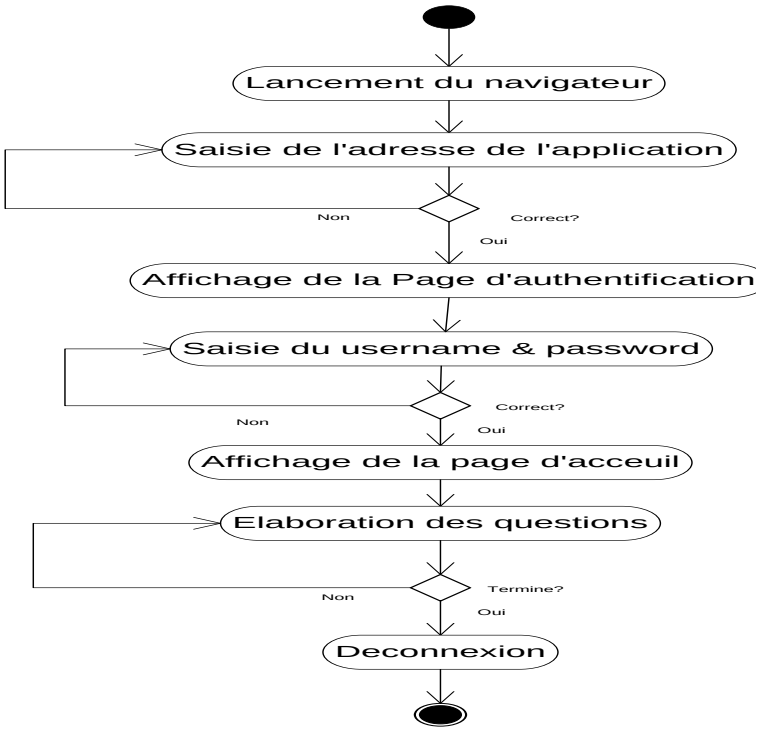


Figure 6:Diagramme d'activité pour le cas d'utilisation « Elaborer un questionnaire ».

3. Diagramme de séquence

Un diagramme de séquence montre les interactions entre les objets, arrangés en séquence dans le temps. En particulier il montre les objets participants dans l'interaction par leur ligne de vie et les messages qu'ils s'échangent ordonnancés dans le temps. Il ne montre pas les associations entre les objets. L'ordre d'envoi des messages est déterminé par sa position sur l'axe vertical (l'axe du temps) du diagramme. Le temps s'écoule de haut en bas. Les diagrammes de séquences permettent entre autres de faire une description graphique d'un cas d'utilisation [11]. Sur un diagramme de séquence, il est possible de représenter de manière explicite les différentes périodes d'activité d'un objet au moyen d'une bande rectangulaire superposée à la ligne de vie de l'objet. Ci-dessous voici un schéma d'un diagramme de séquence du cas d'utilisation « Se connecter » :

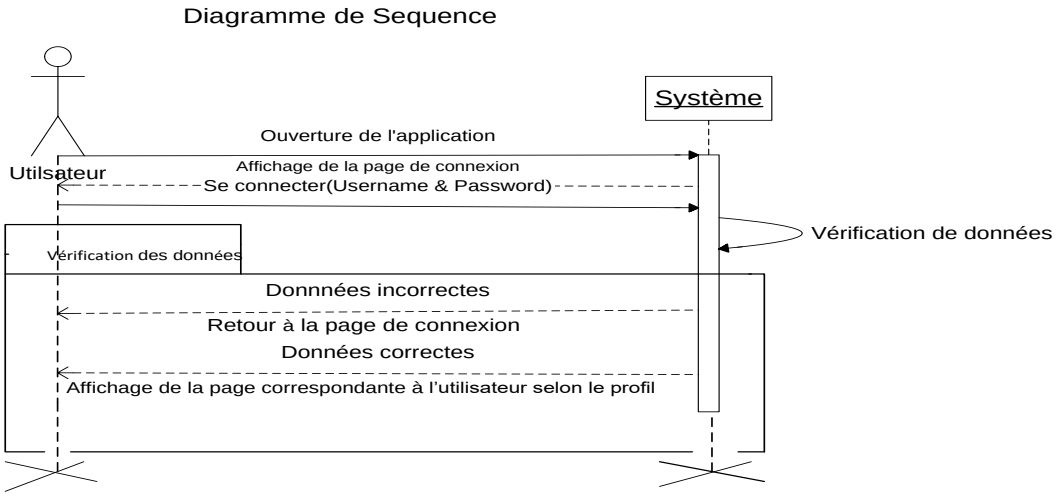


Figure 7:Diagramme de séquence du cas d'utilisation « Se connecter »

Diagramme de séquence du cas d'utilisation « Ajouter »

L'enseignant après avoir réussi à se connecter, demande au système le formulaire d'ajout en cliquant sur le bouton ajouter, le système lui renvoi le formulaire, il remplit tous les champs pour compléter le formulaire et soumettre les données en cliquant sur le bouton de validation. Le système envoi alors les données dans la base de données et seront sauvegardées en permanence. L'enseignant peut éditer les données une fois a mal complété le formulaire d'ajout.

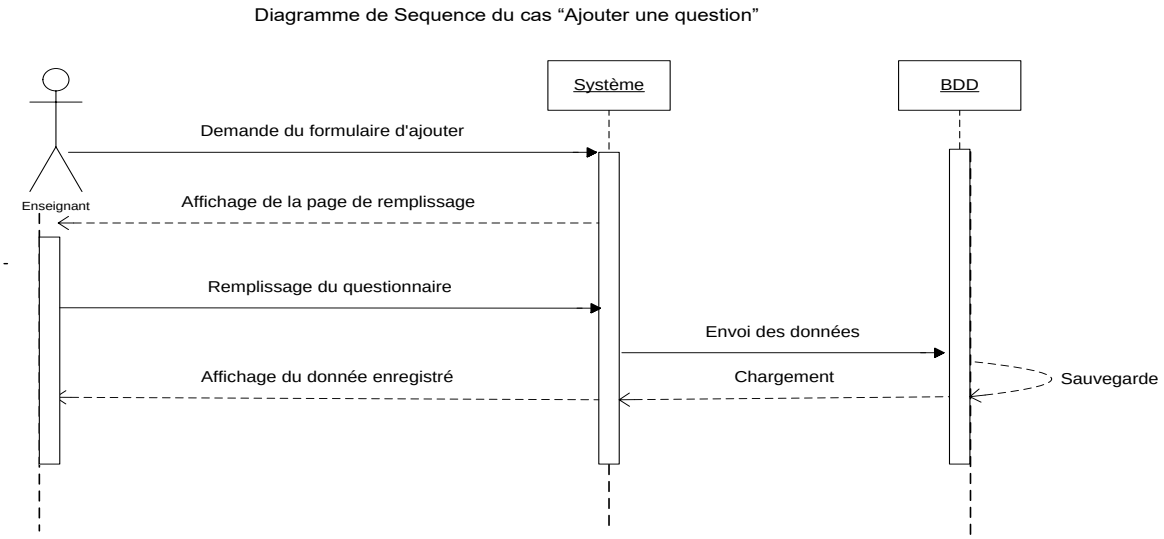


Figure 8: Diagramme de séquence du cas d'utilisation « Ajouter ».

Diagramme de séquence du cas d'utilisation « Rechercher »

Une fois l'utilisateur a passé la page de connexion, il demande le formulaire de rechercher, il saisit dans la zone de recherche les données à rechercher, le système envoie la requête à la base de données qui exécute la requête et charge les données à base des mots clés que l'utilisateur a utilisées pour effectuer une requête. Enfin, le système affiche les données à l'utilisateur selon sa demande. Voici le schéma du diagramme de séquence du cas d'utilisation « Rechercher » :

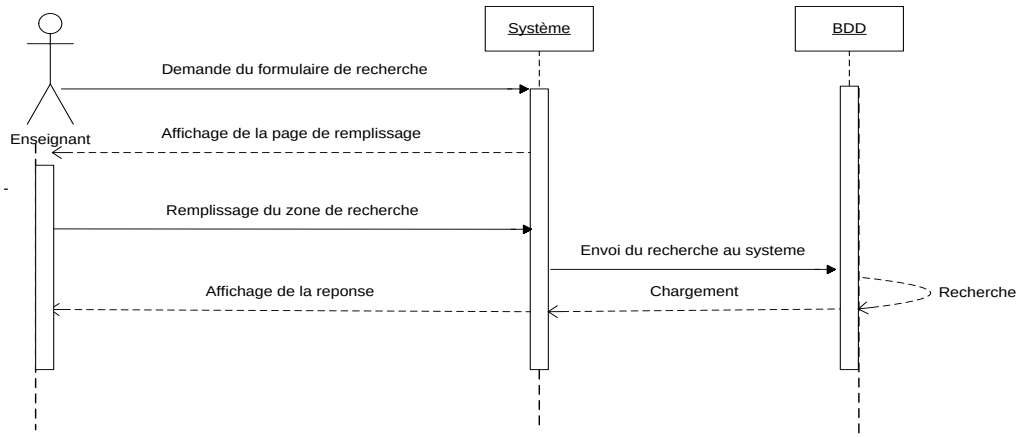


Figure 9:Diagramme de séquence du cas d'utilisation « Rechercher ».

Le troisième chapitre nous a placés dans la conception des systèmes d'information, modélisation des systèmes d'information à l'aide du langage UML. Dans la première partie nous avons essayé de voir les notions théoriques liées à la conception des systèmes d'information afin de s'en attaquer à la deuxième partie permettant de mettre en exergue la théorie que nous avons dans cette partie. En définitif nous avons exploré diagonalement les théories liées avec la modélisation et par après nous avons modélisé le système à l'aide de certains diagrammes UML choisies comme modèle afin de montre graphiquement le fonctionnement du système dans le but de rendre le système flexible pour tous les internautes. Nous avons décidé de modéliser le système avec cinq diagrammes : Le diagramme de cas d'utilisation, de séquence, d'activité, diagramme de déploiement et le diagramme de Classe qui sont considérés comme des diagrammes pivots pour montre globalement le fonctionnement du système développé.

CHAP IV : MODELISATION MATHEMATIQUE DU SYSTEME PAR LES FILES D'ATTENTE

IV.1 Introduction

Chaque système d'information est basé sur un modèle mathématique qui explique concrètement son principe de fonctionnement. C'est le modèle mathématique qui nous aide à comprendre la structure fonctionnelle du système. Les modèles mathématiques connus comme le réseau de pétri, Chaine de Markov, les réseaux de neurones, la régression linéaire, file d'attente etc. peuvent être utilisé pour représenter mathématiquement un système d'information selon le domaine. La modélisation d'un système est « l'opération par laquelle on établit un modèle d'un phénomène, afin d'en proposer une représentation, interprétable, reproductible et simulable. ». Les réseaux des files d'attente constituent un formalisme de modélisation, largement utilisé pour l'évaluation des performances des systèmes à événements discrets tels que les systèmes informatiques, les réseaux de communication et les systèmes de production. Ce modèle permet de représenter la notion de partage des ressources, où une ressource est partagée entre plusieurs clients. Notre système utilise une architecture client-serveur, le serveur le mieux adapté à notre système est Apache. Le client se connecte au serveur par des requêtes SQL pour demander des données lui correspondant selon son profil. On doit optimiser le serveur pour qu'il puisse répondre à toutes les requêtes issues des clients pour que tout le monde soit servi. Le serveur Apache utilise deux modes de gestion des requêtes des clients : le mode itératif et le mode concurrent. Le mode itératif qui n'est pas fiable aux applications web, le processus serveur traite les requêtes les unes après les autres. Quant au mode concurrent qui est très fiable, le serveur exécute plusieurs requêtes simultanément grâce au schéma veilleur exécutants [12]. Dans ce chapitre, nous allons présenter le formalisme des réseaux de files d'attente. Nous commençons dans une première section par introduire la notation de Kendall et la formule de Little. Dans une deuxième section, nous donnons les principaux résultats des files d'attente sur lesquels se basent notre travail. Une troisième section sera dédiée aux réseaux de files d'attente, à savoir leurs types et leurs méthodes de résolution analytique. La dernière section présentera la simulation des réseaux de files d'attente, qui constitue un dernier recours lorsque le réseau à analyser ne peut pas être résolu analytiquement. Comme notre système est une architecture client-serveur, nous allons aussi explorer et étudier la théorie des files d'attentes comme outils mathématiques de gestion des requêtes auprès du serveur. La théorie de la file d'attente vise à étudier scientifiquement les attentes des clients (étudiants pour notre cas) lorsqu'ils reçoivent un service non immédiat. Il est fréquemment utilisé pour résoudre des problèmes de conception et de

détermination du nombre de serveurs nécessaires. La théorie des files d'attente permet d'étudier de manière scientifique l'attente que doivent attendre les clients lorsqu'ils demandent un service.

IV.2 Définition

Un système d'attente simple ou station consiste en une file d'attente, appelée aussi buffer ou tampon, et d'une station de service constituée d'une ou plusieurs ressources appelées serveurs. Des entités dites clients ou travaux (jobs), générées par un processus d'arrivée externe, rejoignent la file pour recevoir un service offert par l'un des serveurs. Une politique de service (généralement FIFO, premier servi de la file) est adoptée pour servir les clients. A la fin du service, le client quitte le système. La théorie des files d'attente a été développée pour fournir des modèles mathématiques pour prédire le comportement des systèmes admettant un phénomène d'attente. Un tel système est caractérisé par le nombre de serveurs, la discipline de service, La capacité du buffer, le processus d'arrivée, Le processus de service. Ces caractéristiques sont définies à travers la notation suivante, dite de Kendall [13].

IV.2.1 Notation de Kendall

Dans la théorie des files d'attente, la notation de Kendall est un standard utilisé pour décrire un modèle de file d'attente. Une file d'attente s'écrit sous la forme T/X/K/C/m/Z avec T qui décrit la distribution d'inter-arrivée des clients, X décrit la distribution de service, K décrit le nombre de serveurs, C décrit La capacité de la station, m décrit le nombre maximum de clients susceptibles d'arriver dans la file d'attente, Z décrit la discipline de service (FIFO, LIFO, PS). T et X peuvent être donnés par plusieurs types de distributions. Les plus répandues sont M qui est la loi sans mémoire, E_k qui est la loi constante, H_k *qui est la l*oi hyperexponentielle-k,C_k qui est la loi de Cox-k ,PH_k qui est la loi de type « phase » à k étages ,G qui est la loi générale et GI qui sont les lois générales indépendantes. Les valeurs par défaut des trois derniers paramètres de la notation de Kendall sont C=+∞, m=+∞ et Z=FIFO. Dans la suite, nous utilisons la notation T/X/k. Nous expliquons maintenant chacune des caractéristiques d'une file d'attente.

IV.3 Caractéristiques d'une file d'attente

L'arrivée des clients, décrite par le symbole A, est définie à l'aide d'un processus stochastique de comptage N_t avec $t \geq 0$. Soit A_n la date d'arrivée du n ème client dans le système :

$$A_0 \equiv 0 \; et \; A_n \equiv inf\,\{t | N_t = n\}$$

Soit T_n le temps séparant l'arrivée du $(n-1)i^{ème}$ client et celle du n ème client. La loi de T_n est dite distribution des inter-arrivées :

$$T_n = A_n - A_{n-1}$$

Un processus de comptage N_t avec $t \geq 0$ est un processus de renouvellement si et seulement si les variables aléatoires $(T_n)_{n=1, 2, \ldots}$ Sont des variables indépendantes et identiquement distribuées. La loi de T décrivant les durées d'inter-arrivées suffit alors pour caractériser le processus de renouvellement. Notons que, lorsque les inter-arrivées sont de loi exponentielle, le processus d'arrivé est un processus de Poisson. Ce dernier est le processus le plus couramment employé pour caractériser les processus d'arrivée. Le temps de service est défini par le temps séparant le début de la fin du service. On note X_n le temps de service du $n^{ème}$ client. La distribution du temps de service la plus couramment utilisée est la distribution exponentielle, qui est caractérisée par la propriété sans mémoire. Une station peut contenir une ou une infinité de serveurs. Dans le cas multiserveur, la détermination de la distribution de service de chacun des serveurs est recommandée voire indispensable. La plupart du temps, les serveurs sont considérés identiques et indépendants les uns des autres. Pour modéliser un phénomène de retard pur ou un système dont le nombre de serveur est toujours supérieur au nombre de client, on utilise une station avec un nombre infini de serveur. L'attente est alors réduite à zéro et temps de séjour dans la station est égal au temps de service. La capacité d'une station notée C est le nombre de clients maximal qui peut se trouver simultanément dans la file et les serveurs. Cette capacité peut être finie ou infinie. Par conséquent, si un client arrive et qu'il y a déjà c clients dans le système, le client peut être accepté ou rejeté suivant la politique de débordement de la station. L'objectif de la discipline de service est de déterminer l'ordre de service des clients dans la file d'attente et leurs passages dans les serveurs, afin d'être servis. Les disciplines les plus courantes sont FIFO (First In First Out) ou les clients sont servis dans leur ordre d'arrivée, LIFO (Last In First Out) ou le dernier client arrivé sera placé en tête de file pour être servi en premier, PS qui est une discipline définie pour modéliser des systèmes informatiques. Tous les clients sont servis à tour de rôle. Chaque client effectue un quantum de temps très petit dans le serveur et revient dans la file d'attente jusqu'à terminer complètement son service et Random qui définit un ordre aléatoire d'accès aux serveurs, etc. Dans cette première partie nous avons introduit la notation et les caractéristiques d'une file d'attente simple qui nous permettra de modéliser les systèmes. Toutefois, la majorité des systèmes sont assez complexes pour qu'ils soient modélisés par une seule file d'attente simple, d'où la notion de réseau de files d'attente.

IV.4 Réseaux de files d'attente

Un réseau de files d'attente est un ensemble de file d'attentes interconnectées. La définition de ce réseau requiert, d'une part, la définition de toutes les files qui le constituent, et d'autre part, la définition du routage entre ses files. En effet, après avoir terminé son service dans un serveur, un client entre dans une autre station ou quitte définitivement le réseau. Plusieurs types de routage existent : Routage probabiliste où un client sortant d'une station i a une probabilité p_{ij} à passer à la station j ou une probabilité pi_0 pour quitter le réseau. L'ensemble des probabilités de routage de toutes les stations sont regroupées dans une matrice de routage, routage dynamique est un routage qui se fait suivant l'état du système au moment du routage. Par exemple, un client sortant d'une station i peut choisir la station comportant moins de clients parmi ces destinations possibles Routage cyclique où un client sortant d'une station choisira à tour de rôle une de ces destinations possibles. Un réseau de files d'attente peut être classé selon les types ou le nombre de classes de clients qui le parcourent ou suivant sa topologie.

IV.4.1 Classification des réseaux de file d'attente

Un réseau de file d'attente est un ensemble de files d'attente interconnectées, dans lesquelles circulent une ou plusieurs classes de clients. Chaque classe se caractérise par un schéma de routage, par des comportements différents au niveau de chaque station de service et de l'ordonnancement dans le buffer d'attente. On peut distinguer différentes classes de clients suivants : Des processus d'arrivées différents, des comportements des clients qui sont différents à chaque station et des routages différents dans le réseau. Un réseau parcouru par une seule classe de clients est appelé réseau mono-classe. Contrairement au réseau multi-classe dans lequel circulent plusieurs classes. Chaque classe de clients peut être soit ouverte soit fermée. On appelle réseau mono-classe ouvert un réseau avec une seule classe de clients où les clients arrivent de l'extérieur du système, séjournent pour recevoir un ou plusieurs services, puis quittent définitivement le système. Par conséquent, le nombre de clients est infini. La figure 10 montre un exemple de réseau mono-classe ouvert. Un réseau est dit mono-classe fermé lorsque tous les clients appartiennent à la même classe et leur nombre en entrée du système est constant. Il n'y a ni de départ ni d'arrivée de clients.

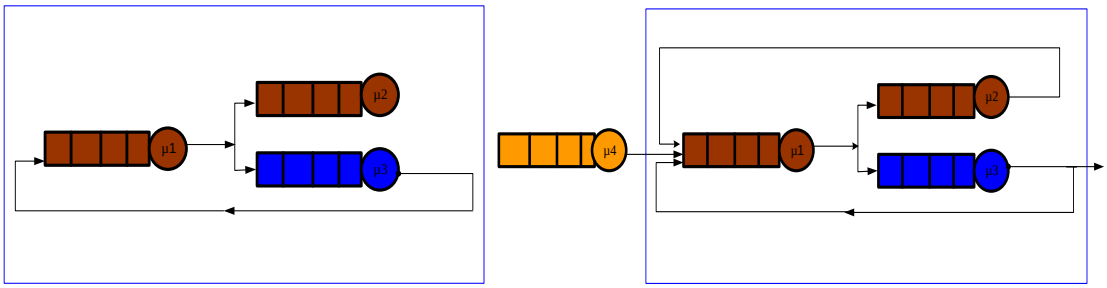


Figure 10: Réseau mono-classe fermé et réseau mono-classe ouvert.

On appelle réseau multi-classe mixte un réseau contenant au minimum une classe de clients ouverte et une classe fermée. La population de clients externes est infinie, alors que la population de clients internes est finie. La figure 16 montre un exemple de réseau multi-classe mixte.

IV.4.2 Processus d'arrivée de Poisson

Soit $N_{(t)} = max_i : t_i \leq t$ le nombre aléatoire d'arrivées ou avant l'instant $t \geq 0$, où t_i est l'instant d'arrivée du $ième$ client. Le processus stochastique $N_{(t)}$ est appelé un processus de Poisson si les clients arrivent un à la fois, Le nombre d'arrivée dans l'intervalle$(t, t + s]$, c'est-à-dire la variable aléatoire, $N_{(t+s)} - N_{(t)}$ est indépendant de $N_{(u)}, 0 \leq u \leq t$,La distribution de $N_{(t+s)} - N_{(t)}$ est indépendante de t pour tout $t > 0$, s > 0. Si $N_{(t)}$ est un processus de Poisson alors le nombre d'arrivée dans chaque intervalle de taille s est une variable aléatoire de Poisson avec paramètre λ_s avec $\lambda > 0$, c'est -à-dire Probabilité $\{N_{(t+s)} - N_{(t)} = k\} = exp^{-\lambda s}$, k = 0, 1, 2, . . ., t, s ≥ 0. Si $N_{(t)}$ est un processus de Poisson avec un taux λ alors les variables $A_i = t_i - t_{i-1}$ sont distribués selon la loi exponentielle de paramètre λ

Si X suit une loi de poisson de paramètre λ, soit $X \sim P_{(\lambda)}$

Alors, on a $P_{(X = k)} = e^{-\lambda} \frac{\lambda^k}{k!}$ et que

$$E_{(x)} = \sum_{k=1}^{+\infty} p(x)$$

$$E_{(x)} = \sum_{k=1}^{+\infty} k\, e^{-\lambda} \frac{\lambda^k}{k!}$$

$$E_{(x)} = e^{-\lambda} \sum_{k=1}^{+\infty} \frac{\lambda^k}{(k-1)!}$$

$$E_{(x)} = \lambda\, e^{-\lambda} \sum_{k=1}^{+\infty} \frac{\lambda^{k-1}}{(k-1)!}$$

$$E_{(x)} = \lambda\, e^{-\lambda}\, e^{\lambda} = \lambda \quad (1)$$

La loi de Little indique que pour réduire les temps de traversées (améliorer les délais), il faut réduire les en-cours ou augmenter le débit. Or il s'avère qu'un processus de production rencontre toujours un goulet (théorie des contraintes) limitant de fait le débit de production : la seule action efficace à conduire lorsque l'on souhaite réduire les temps de cycle reste donc de réduire les en-cours (en particulier les stocks). La loi de Little permet également de comprendre la relation entre taille de lot et temps de traversée d'une ligne de production. WIP représente alors la taille du lot et $\frac{1}{T}$ représente le temps de cycle du processus de production : plus les lots sont importants, plus les délais de traitement de ces lots sont importants et moins l'entreprise est agile pour répondre aux demandes du marché. La loi de Little s'applique à tous types de processus, quel que soit sa variabilité.

$$WIP = T * LT$$

Avec WIP : Stock d'encours

T : Débit par unité de temps

LT : Temps de Cycle moyen passé dans le système

La loi de Little énonce que le nombre moyen de clients dans un système de files d'attente est égal à leur fréquence moyenne d'arrivée λ multipliée par leur temps moyen W dans le système :

$$L = \lambda W$$

La formule est valide pour un système de files d'attente observé entre $[0, T]$ qui est vide aux temps 0 et T et avec $0 < T < \infty$. Exemple d'application : Les étudiants qui attendent le début de l'évaluation, La file d'attente des camions arrivant dans une station-service, etc.

1. Loi des temps d'attente et temps de séjour d'un client

Il est possible de déterminer la loi de probabilité du temps d'attente $W\infty$ (waiting time) d'un client générique en régime stationnaire grâce à la formule des probabilités totales :

$$P\ (W_{\infty \leq} t) = \sum_0^{\infty} P(W_{\infty} \leq |Q_{\infty} = n)\, P(Q_{\infty} = n).$$

Or la variable aléatoire conditionnelle (Q_{∞}= n) représente l'attente pendant laquelle n personnes consécutives doivent être servies selon un service exponentiel ε(μ), c'est donc la somme de n

variables aléatoires indépendantes de loi ε (μ). Si n = 0, le temps d'attente est nul. Si n > 1, cette variable aléatoire conditionnelle suit une loi d'Erlang E (n ; μ).

$$\text{D'où} \quad P_{(W\infty \leq t)} = \pi_0 + \sum_{n=1}^{\infty} \pi_n \int_0^t \frac{\mu n s n-1}{(n-1)!} e^{-\mu s}\, d_s$$

$$= (1-\rho)\,[1 + \rho_\mu \int_0^t \sum_{n=0}^{\infty} \frac{(\rho \mu s) n}{n!}\, e - \mu_s\, ds]$$

$$= (1-\rho)\,[1 + \lambda \int_0^t e - \mu(1-\rho)\, ds]$$

Soit $$P_{(W\infty \leq t)} = 1 - \frac{\lambda}{\mu} e^{-(\mu - \lambda)\, t} \qquad (2)$$

C'est une loi exponentielle $\varepsilon_{(\mu - \lambda)}$ avec un poids en $0 : P_{(W\infty = 0)} = 1 - \rho$. Notons l'égalité $P_{(W\infty = 0)} = P_{(Q_\infty = 0)}$ qui traduit le fait que la probabilité de ne pas attendre coïncide avec celle de trouver une file vide. L'attente moyenne d'un client peut se calculer selon le même procédé :

$$E_{(W\infty)} = \sum_{n=0}^{\infty} E(W_\infty \mid Q_\infty = n)\, P(Q_\infty = n)$$

Le raisonnement précédent montre que $E(W_\infty \mid Q_\infty = n) = \frac{n}{\mu}$ et donc

$$E_{(W\infty)} = \sum_{n=0}^{\infty} \frac{n}{\mu} P(Q_\infty = n) = \frac{1}{\mu} E(Q_\infty);$$

Soit $$E_{(W\infty)} = \frac{\lambda}{\mu(\mu - \lambda)} \qquad (3)$$

Enfin, l'expression $W_\infty + S_\infty$ représente le temps de séjour total (attente + service) du client générique. Sa loi s'obtient comme précédemment, c'est à présent une véritable loi exponentielle $\varepsilon_{(\mu - \lambda)}$:

$$P(W_\infty + S_\infty \leq t) = 1 - e^{-(\mu - \lambda)\, t}$$

Le temps de séjour moyen d'un client dans le système vaut donc

$$E(W_\infty + S_\infty) = \frac{1}{\mu - \lambda} \qquad (4)$$

2. Loi du temps d'activité du serveur

Une période d'activité est une période durant laquelle il sert les clients de manière continuelle. Elle démarre à partir de l'arrivée d'un client entrant dans le vide et cesse dès la fin du service du prochain client laissant derrière lui la file vide, puis une nouvelle période d'activité redémarre avec l'arrivée d'un autre client. Une période de vacances est une période (aléatoire) durant laquelle il n'y a aucune personne à servir. Un cycle d'activité est le laps de temps séparant deux personnes consécutives arrivant dans un système vide. Un tel cycle est donc la somme d'une période d'activité et d'une période de vacances du serveur. On peut démontrer que la loi du temps d'activité B_∞ (busy time) en régime stationnaire a pour densité :

$$fB\infty\,(t) \;=\; \sqrt{\frac{\mu}{\lambda}}\frac{1}{t}e^{-(\lambda+\mu)}I_1\,(2t\sqrt{\lambda\mu}) \qquad (5)$$

où $I1(z) \;=\; \sum_{n=0}^{+\infty}\frac{1}{(n+1)!n!}(\frac{z}{2})\,2_{n+1}$ est une fonction de Bessel.

Le temps d'activité moyen du serveur se calcule selon $\int_0^\infty t_{(t)}\mathrm{fB}\infty\mathrm{dt}$, il est donné par

$$E_{(\mathrm{B}\infty)} = \frac{1}{(\mu-\lambda)}$$

Ce résultat peut s'obtenir de manière empirique en multipliant le temps de service moyen d'un client par le nombre de clients présents dans la file lorsque cette dernière est non vide :

$$E_{(\mathrm{B}\infty)} = \; E\;_{(\mathrm{Q}\infty\,|\,\mathrm{Q}\infty\,\geq\,1)} \times\; E(\mathrm{S}_\infty).$$

On a

$$E\;(Q_\infty\,|\,Q_\infty \geq\; 1) \;=\; \frac{E(Q_\infty)}{P(Q_\infty \geq 1)} = \frac{\mu}{\mu-\lambda} \quad \text{et} \quad E(\mathrm{S}_\infty) = \frac{1}{\mu} \qquad (6)$$

Nous remarquons que le temps d'activité moyen du serveur est identique au temps de séjour moyen d'un client dans le système.

D'autre part, la propriété d'absence de mémoire de la loi exponentielle montre que le temps de vacances I_∞ du serveur (idle time), qui est la durée d'attente pour le serveur d'une nouvelle arrivée lorsque la file est vide, suit la loi $\varepsilon_{(\lambda)}$:

$$f\;I_\infty\,(t) \;=\; \lambda\,e^{\lambda t}$$

IV.5 Principaux modèles des files d'attente

Il existe plusieurs modèles des files d'attente. Voici ci-dessous la liste de quelques exemples :

Modèle M/M/1

La file $M/M/1$ est la file la plus simple et la plus utilisée pour modéliser les systèmes informatiques. L'utilisation de cette file est motivée par l'ensemble de ses résultats permettant de déterminer les paramètres de performances moyens. Elle est définie par le processus stochastique suivant :

1) Le processus d'arrivée des clients est distribué selon un processus de Poisson de paramètre λ;
2) Le processus de temps de service est indépendant du processus d'arrivée et suit la loi exponentielle de paramètre μ ;
3) Un seul serveur. On peut calculer les probabilités du régime transitoire et celles du régime permanent, ainsi que tous les paramètres de performance en utilisant les chaînes de Markov [14]. Soit $P_{(0)}$ la probabilité pour que le système ne contient aucun client en régime permanent et on note $\rho = \frac{\lambda}{\mu}$. Les paramètres de performance moyenne de cette file en régime stationnaire sont :
4) Le débit moyen D : $D = [1 - P_{(0)}]\,\mu = \rho\mu = \lambda$
5) Le taux d'utilisation U : $U = [1 - P_{(0)}] = \rho$
6) Le nombre moyen des clients N :

$$N = \frac{\rho}{(1-\rho)} \tag{7}$$

7) Le temps moyen de séjour R :

$$R = \frac{N}{D} = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu} + \frac{1}{\mu(1-\rho)} \tag{8}$$

Modèle M/M/k

Pour le cas markovien avec plusieurs serveurs, les paramètres de performance moyens en régime stationnaire sont :

a) Le débit moyen D : $D = \lambda$
b) Le nombre moyen des clients N : $N = RD = R\lambda$
c) Le temps moyen de séjour R :

$$R = P_{(0)}\frac{\rho^k}{\mu(k-\rho)^2(k-1)!} + \frac{1}{\mu} \tag{9}$$

Avec μ est le taux de service d'un serveur et tous les serveurs sont identiques.

Modèle M/M/1/C

Une file $M/M/1/C$ est identique à une file $M/M/1$, à l'exception de la capacité de la file considérée dans ce cas comme finie. Les paramètres de performances spécifiques à cette file sont comme-suit :

Le débit moyen : $D = \frac{1-\rho^C}{1-\rho^{C-1}}\lambda$ (10)

Le taux d'utilisation : $U\ [1\ -\ P_{(0)}]\ = \frac{1-\rho^C}{1-\rho^{C-1}}$ (11)

Le nombre moyen des clients : $N = \sum_{n=0}^{C} np(n)$

$$N = \frac{\rho}{1-\rho}\ \frac{1-(C+1)\rho^C+C\rho^{C-1}}{1-\rho^{C-1}} \quad (12)$$

Le temps moyen de séjour des clients non rejetés :

$$R = \frac{N}{D} = \frac{1-(C+1)\rho^C + C\rho^{C+1}}{(\mu-\lambda)(1-\rho^C)} \quad (13)$$

Modèle M/G/1

La file $M/G/1$ est une file à capacité illimitée ayant un seul serveur. Son processus d'arrivée suit la loi poissonienne de taux λ tandis que le temps de service est distribué selon une variable aléatoire générale G. Dans ce cas particulier, on peut déterminer tous les paramètres de performances moyens, même si son service ne vérifie pas la propriété sans mémoire. Ces paramètres sont déterminés grâce à la méthode de la chaîne de Markov induite. Cette méthode consiste à ramener l'étude à une chaîne de Markov à temps discret en considérant des instants d'observation particuliers (instants de début de service ou instants de fin de service). Les paramètres de performance moyens se présentent alors comme suit :

Le débit Moyen : $D\ =\ [1\ -\ P_{(0)}]\ \mu\ =\ \rho\mu\ =\ \lambda$; (14)

Le taux d'utilisation : $U\ =\ [1\ -\ P_{(0)}]\ =\ \rho$; (15)

Le nombre moyen des clients N :

$$N\ =\ \rho\ +\ \frac{\rho^2(1+CV^2)}{2(1-\rho)} \quad (16)$$

Avec CV^2 est le cœfficient de variation du temps de service

Le temps moyen de séjour R :

$$R = \frac{N}{D} = \frac{1}{\mu} + \frac{\lambda(1+cv^2)}{2\mu^2(1-\rho)} \quad (17)$$

Avec cv^2 est le carré du coefficient de variation de la loi de temps de service.

Modèle G/G/1

La file $G/G/1$ généralise les modèles de files d'attente à un seul serveur (les inter-arrivées et les services sont arbitraires). Cette généralisation rend impossible la détermination des résultats exacts des paramètres de performance pour ce modèle. En effet, un modèle contenant deux lois généraux (non Markoviennes) ne peut ni n'être résolu par la détermination des probabilités transitoires, ni par l'intermédiaire d'une chaîne de Markov induite [15]. Les principaux résultats de cette file sont des bornes inférieures et supérieures qui encadrent le temps moyen d'attente, noté W, dans la file [15] :

$$\frac{\lambda\sigma^2 X - X(2-\rho)}{2(1-\rho)} \leq W \leq \frac{\lambda(\sigma^2 T + \sigma^2 X)}{2(1-\rho)} \qquad (18)$$

Avec

X : le temps de service moyen

σ_X : L'écart type de la variable aléatoire qui décrit le temps de service

σ_T : L'écart type de la variable aléatoire qui décrit les inter-arrivées

Cette borne peut être décrite d'une façon plus simple si la condition suivante est vérifiée :

$$E[T - t | T > t] \leq \frac{1}{\lambda} \qquad \forall\, t \geq 0$$

Cette condition est satisfaite pour la plupart des distributions, à l'exception de la distribution hyper-exponentielle. Elle nous permet cependant de réécrire l'inégalité sous la forme :

$$W_{Sup} - \frac{1+\rho}{2\lambda} \leq W \leq W_{sup} \qquad (19)$$

Avec $W_{sup} = \frac{\lambda(\sigma^2 T + \sigma^2 X)}{2(1-\rho)}$

En appliquant la formule de Little, on obtient deux bornes pour le nombre moyen de clients N_{buffer} qui attendent le service :

$$\lambda W_{Sup} - \frac{1-\rho}{2} \leq N_{buffer} \leq W_{sup} \qquad (20)$$

Cette inégalité est très importante. En effet, les deux bornes encadrent de très près le nombre de clients : La différence entre la borne supérieure et la borne inférieure est de $\frac{1-\rho}{2}$ avec $0 \leq \rho \leq 1$ pour une file stable. En d'autres termes, la différence est comprise entre 0 et 1, ce qui représente une bonne approximation du nombre de client qui se trouve dans le buffer N_{buffer}.

Modèle G/G/k

La file $G/G/k$ généralise le cas d'une file à plusieurs serveurs (inter-arrivées et services arbitraires). De même que la file G/G/1, il est impossible de déterminer d'une façon exacte les paramètres de performance moyenne de cette file. Par ailleurs, des bornes pour les temps moyens d'attente dans la file W_q existent.

$$W_{Sup} - \frac{(k-1)X^2}{2kX} \leq W_{G/G/K} \leq \lambda \frac{\sigma^2 T + \frac{\sigma^2 X}{k} + \frac{(k-1)X^2}{K^2}}{2(1-\frac{\rho}{k})} \quad (21)$$

Avec S_{up} est la borne supérieure de W pour la file $G/G/1$ (voir la section précédente)

L'inégalité (19) montre la similitude qui existe entre les bornes déterminées dans le cas de la $G/G/1$ et de la $G/G/k$. En effet, le résultat de la G/G/k peut être considéré comme étant l'application de l'inégalité (16) pour un serveur k fois plus rapide avec un temps de service moyen de $\frac{X}{K}$ et une variance de $\frac{\sigma^2 X}{K^2}$. Le deuxième résultat important pour cette file est l'approximation pour le trafic dense (heavy trafic). Quand $\frac{\rho}{k}$ tend vers 1, le temps d'attente au régime stationnaire tend vers une variable aléatoire de distribution exponentielle avec une moyenne de :

$$W_G/G/k \approx \lambda \frac{\sigma^2 TX + \frac{\sigma^2 X}{k}}{2(1-\frac{\rho}{k})} \quad (22)$$

Les résultats déterminés pour cette file représentent des résultats globaux et valables pour toutes les autres files ($M/M/1, M/G/1, M/G/k \ldots$).

IV.6 Processus de naissance et de mort

Une file d'attente peut être considéré comme un processus de naissance et de mort du fait que le temps qui s'écoule entre deux arrivées et le temps de service sont distribués exponentiellement.

Naissance = Arrivée du client qui demande du service ;

Mort = Départ d'un client dans le système après son service.

Dans le processus de naissance et de mort :

Les temps s'écoulant entre deux naissances consécutives et entre deux morts consécutifs sont tous les deux distribués exponentiellement. A partir de l'état n, le temps de transition est de type : $n \rightarrow n+1$ (pour une seule naissance) et/ou $n \rightarrow n-1$ (pour une seule mort).

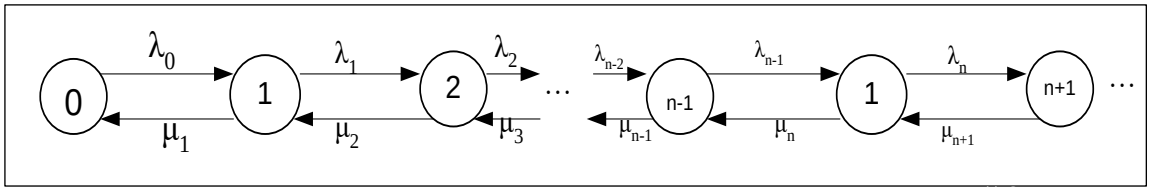


Figure 11:Diagramme de transition entre états.

Où λ_n = Taux moyen de naissance lorsque n individus sont dans le système ;

μ_n = Taux moyen de mort lorsque n individus sont dans le système ;

Le processus de naissance et de mort peut être considéré comme une chaîne de Markov en temps continu où les densités de transitions sont spécifiées à l'aide des λ_n et μ_n.

Soient les variables :

1) $\pi_j q_j$= Taux auquel le processus passe de j puisque π_j = probabilité (à l'équilibre) que le processus soit dans l'état j ;
2) q_j = taux de transition de sortir de l'état j étant donné que le processus est dans l'état j ;
3) $\pi_i q_j$ = taux de passage de l'état i à l'état j ;
4) q_{ij} = taux de transition de l'état i à l'état j étant donné que le processus est dans l'état i.

$$\sum_{\substack{i\neq 0 \\ i\neq j}}^{M} (\pi_i q_{ij})$$

Cette formule ci-dessus exprime le taux de passer à l'état j quel que soit l'état i dans lequel se trouve le processus. Nous ne constatons donc que le taux de départ de j = taux d'arrivée à j.

Nous allons donc utiliser les équations de balance suivantes : Equations d'équilibres

$$\pi_i q_j = \sum_{\substack{i\neq 0 \\ i\neq j}}^{M} (\pi_i q_{ij}) \qquad j \in \{0,\ldots, M\}$$

$$\Pi_j q_j = \sum_{i\neq 0}^{M} \pi_j = 1$$

Intensité de transition

$$q_j = \sum_{\substack{i\neq 0 \\ i\neq j}}^{M} q_{ij}$$

En remplaçant les valeurs des q_j dans les équations d'équilibre, on aura

$$\Pi q_j q_j = \sum_{\substack{i\neq 0 \\ i\neq j}}^{M} \pi_i q_{ij} \iff \Pi j \sum_{\substack{i\neq 0 \\ i\neq j}}^{M} q_{ji} = \sum_{i\neq j}^{M} \pi_i q_{ij} \qquad j \in \{0,\ldots, M\}$$

Les équations d'équilibre deviennent

$$\Pi j \sum_{i\neq j}^{M} q_{ji} = \sum_{i\neq j}^{M} \pi_i q_{ij} \qquad j \in \{0,\ldots, M\}$$

$$\Pi_j \sum_{J=0}^{M} \pi_j = 1$$

On a : taux de départ de j = taux d'arrivée à j.

On peut maintenant appliquer les équations d'équilibre pour déterminer les probabilités P_n à l'équilibre.

Les équations de balance deviennent :

$$\Pi_j \sum_{i\neq j}^{M} q_{ij} = \sum_{i\neq j} \pi_i q_{ij} \qquad j = 0, 1, 2, \ldots$$

$$\sum_j \pi_j = 1$$

Pour $n = 0$

$$P_0 \lambda_0 = p_1 \mu_1$$

Pour $n = 1, 2, \ldots$

$$p_n (\lambda_n + \mu_1) = p_n - 1 \lambda_n - 1 + p_n + 1 \mu_n + 1$$

Pour les états n on a :

A l'état 0 : $p_1 = \frac{\lambda_0}{\mu_0} p_0$

A l'état 1 : $p_2 = \frac{\lambda_1}{\mu_1} p_1 + \frac{1}{\mu_2}(\mu_1 p_1 - \lambda_0 p_0) = \frac{\lambda_1}{\mu_2} p_1 \frac{-\lambda_1}{\mu_2} \frac{\lambda_0}{\mu_1} p_0$

A l'état 2 : $p_3 = = \frac{\lambda_2}{\mu_3} p_2 + \frac{1}{\mu_3}(\mu_2 p_2 - \lambda_1 p_1) = \frac{\lambda_2}{\mu_3} p_2 = \frac{\lambda_2}{\mu_2} \frac{\lambda_1}{\mu_1} \frac{\lambda_0}{\mu_1} p_0$

A l'état n : $p_3 + 1 = \frac{\lambda_n}{\mu_{n+1}} p_n + \frac{1}{\mu_{n+1}} (\mu_n p_n - \lambda_{n-1} p_{n+1})$

$= \frac{\lambda_n}{\mu_{n+1}} p_n \frac{-\lambda_1}{\mu_2} \dots \frac{\lambda_2}{\mu_3} \frac{\lambda_1}{\mu_2} \frac{\lambda_0}{\mu_1} p_0$

Pour simplifier

$$p_n = \frac{\prod_{i=0}^{n-1} \lambda_0}{\prod_{i=0}^{n-1} \mu_i} \qquad n = 1,2, \dots \tag{23}$$

Pour déterminer p_0, on utilise

$$1 = \sum_j p_j = p_0 + \sum_{n\geq 1} \frac{\prod_{i=0}^{n-1} \lambda_i}{\prod_{i=1}^{n} \mu_i} p_0 = p_0 \left[1 + \sum_{n\geq 1} \frac{\prod_{i=0}^{n-1} \lambda_i}{\prod_{i=1}^{n} \mu_i}\right] \Leftrightarrow \quad p_0 = \frac{1}{1 + \sum_{n\geq 1} \frac{\prod_{i=0}^{n-1} \lambda_i}{\prod_{i=1}^{n} \mu_i}}$$

$\pi_i \sum_{i\neq j} q_{ji} = \sum_{i\neq j} \pi_i q_{ji}$ j=0, 1,2 …

$\sum_j \pi_j = 1$

Ce qui donne une file d'attente infinie à serveur unique ($C = 1$) : M/M/1 qui est un modèle applicable à notre système. Prenons un exemple de file d'attente où les arrivées et les départs se produisent comme dans 1.

$\lambda_n \equiv \lambda \ \forall n \ et \ \mu_n \equiv \mu \quad \forall n,$ independamment du nombre de clients dans le système

Alors

$$p_0 = \frac{1}{1 + \sum_{n\geq 1} \frac{\prod_{i=0}^{n-1} \lambda_i}{\prod_{i=1}^{n} \mu_i}} = \frac{1}{\sum_{n\neq 0}^{\infty} (\frac{\lambda}{\mu})^n} \tag{24}$$

Si on pose que le taux d'arrivée est plus petit que le taux de service $\lambda < \mu$, nous avons

$\frac{\lambda}{\mu} < 1,$ et la progression géométrique $\sum_{n\neq 0}^{\infty} (\frac{\lambda}{\mu})^n = \frac{1}{(\frac{1-\lambda}{\mu})}$

Lorsqu'il s'agit d'un serveur unique nous avons les caractéristiques comme :

M : Arrivées poissonniennes, taux λ ;

M : Service exponentiel, taux μ ;

1 : serveur.

$\rho = \frac{\lambda}{\mu}$ il s'ensuit que

$$p_n = (1 - \rho)\rho^n \ avec\ n = 0,1.2, \ldots$$

Le calcul des caractéristiques de la file d'attente a un seul serveur M/M/1 : Application pour notre système :

Déterminer le nombre moyen des étudiants dans la file (L) :

$$L = \sum_{n=0}^{\infty} np_n = \sum_{n=0}^{\infty} n(1 - \rho)p_n = (1 - \rho)\,\rho \sum_{n=0}^{\infty} np^n$$

$$= (1 - \rho)\,\rho \sum_{n=0}^{\infty} \frac{d}{d\rho}(p)^n$$

$$\Leftarrow L = (1 - \rho)\ \rho \frac{d}{d\rho} \sum_{n=0}^{\infty} \frac{d}{d\rho}(p)^n$$

$$= (1 - \rho)\,\rho \frac{d}{d\rho}\left(\frac{1}{1-p}\right) = (1 - \rho)\,\rho \frac{1}{(1-p)^2}$$

$$= \frac{1}{1-p} = \frac{\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}} = \frac{\lambda}{\mu - \lambda}$$

$$L = \frac{\lambda}{\mu - \lambda} \qquad (25)$$

Pour calculer les différentes caractéristiques on se base sur les paramètres suivants :

1. Nombre de serveurs : m ;
2. Capacité de la file : k clients (y compris ceux qui reçoivent du service) ;
3. Distribution des services : Durée moyenne ;
4. Distribution des inters arrivés : durée moyenne et nombre total de clients N.

Déterminer le nombre moyen des étudiants dans la file d'attente (Lq)

$$Lq = \sum_{n=1}^{\infty} (1 - \rho)p_n = \sum_{n=1}^{\infty} np_n - \sum_{n=1}^{\infty} p_n$$

$$= L - (1 - P_0) = \frac{\lambda}{\mu - \lambda} - \frac{\lambda}{\mu}$$

$$Lq = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

Le temps moyen qu'un étudiant passe dans le système

$$W = \frac{L}{\lambda} = \frac{\lambda}{\mu - \lambda}\frac{1}{\mu} = \frac{1}{\mu - \lambda}$$

Temps moyen qu'un étudiant passe dans la file d'attente

$$Wq = \frac{L_q}{\lambda} = \frac{\lambda^2}{\mu(\mu - \lambda)}\frac{1}{\lambda} = \frac{\lambda}{\mu(\mu - \lambda)}$$

Application à notre modèle : Nous prévoyons recevoir 5000 requêtes SQL par minute en moyenne avec un temps de service d'une seconde pour 400 requêtes simultanément (Server Limit). Nous allons calculer les paramètres de performances suivantes :

Le taux d'utilisation du système (ρ)

$\lambda = 5000\ requêtes/minutes$;

Temps de services = 1seconde/400 requêtes = $\frac{1}{\mu}$;

$$\mu = \frac{1}{1}\frac{400\text{requetes}}{\text{seconde}} * 60\ \text{secondes} = 24000\ \text{requêtes / minute}$$

$\lambda < \mu$ donc le système est stable.

Rappelons que le système est stable si et seulement si $\lambda \leq \mu$

$$\rho = \frac{\lambda}{\mu} = \frac{5000}{24000} = 0.208 \text{ soit } 20.8\%$$

$\rho < 1$ ⇔ Le système est stable

Le nombre moyen de requêtes SQL dans la file

$$Lq = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{(5000)^2}{24000(24000-5000)} = \frac{25000000}{456\,000\,000} =$$

$$21{,}92982456140350877193 \text{ soit } 21.9 \text{ requêtes}$$

Temps moyen qu'une requête SQL passe dans le système

$$Wq = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{5000}{24000(24000 - 5000)}$$

$$= \frac{5000}{456\,000\,000} * 400 * 60\ secondes =$$

$$0{,}26315789473684210526315789473684\ secondes$$

Temps inactif du serveur

Temps inactif du serveur $= 1 - \rho = 1 - 0.208 = 0{,}792\ \%\ soit\ 79.2\ \%$

Notre sujet de travail que nous avons développé est une architecture client-serveur. Pour assurer le bon fonctionnement du système il faut que le serveur réponde parfaitement aux requêtes que les clients lui adressent. Nous allons essayer de déterminer mathématiquement le nombre de requêtes que le serveur peut accueillir mais vous pouvez trouver en bas le schéma montrant l'architecture client-serveur où le client adresse une requête au serveur et le serveur lui répond en fonction de sa demande. Dans notre travail nous avons travaillé avec un seul serveur (machine de l'enseignant) avec plusieurs machines clientes (machines des étudiants), ce qui se modélise avec la notation $M/M/1/\infty/FIFO$. La politique de service est FIFO. Nous avons un serveur de base de données qui reçoit en moyenne 100 requêtes par seconde, arrivant selon un processus de Poisson. Le temps de traitement d'une requête suit la loi exponentielle de paramètre μ. Quand le serveur est occupé, les requêtes sont stockées sur un disque pour être traitées ultérieurement selon le principe "premier arrivé, premier servi". Il existe, sur le marché, 4 types différents de serveurs pouvant traiter respectivement 100, 150, 200 ou 300 requêtes par seconde quand ils fonctionnent sans interruption. Nous prévoyons installer un serveur pouvant traiter 100 requêtes par seconde.

Soit λ : Le taux d'arrivée des étudiants à un moment donné ;

μ : Le taux de services (nombre de clients servis à un moment donné) ;

S_i : La Durée moyen de service ;

ρ : La charge de service.

Pour notre cas il s'agit d'une file d'attente M/M/1, nous pouvons donc utiliser les lois de Little. Il y a distribution stationnaire si $\frac{\lambda}{\mu} < 1$, si on prend un serveur qui traite 100requetes/seconde alors nous avons $\frac{\lambda}{\mu} = 1$ la chaine n'est pas ergodique.

Le taux d'arrivée des étudiants à un moment donné : $\lambda = \frac{1}{100} = E(T) = \frac{1}{(\mu - \lambda)}$ $avec$

$$\mu = 200 requetes/secondes$$

$$\text{On a : } (200 - \lambda) = 100$$

$$-\lambda = 100 - 200$$

$$-\lambda = -100$$

$$\text{Donc } \lambda = 100\ étudiants/seconde$$

$$\lambda \leq \mu$$

Remarque : Pour qu'une file d'attente soit stable il faut que $\lambda \leq \mu$

On souhaite qu'un client qui émet une requête ait la réponse au bout de 1/100 seconde en moyenne. Alors on convient que lorsqu'il y a déjà 8 requêtes stockées dans la file d'attente, les nouvelles requêtes arrivantes seraient trop pénalisées au niveau du temps de réponse. Donc on décide que dans ce cas, elles seront redirigées instantanément sur un serveur auxiliaire. Pour simplifier, on supposera que cette nouvelle disposition a une influence négligeable sur l'ancienne distribution stationnaire (calculée sans serveur auxiliaire). Calculer la probabilité qu'une requête qui arrive sur le serveur principal soit redirigée sur le serveur auxiliaire. Les requêtes sont redirigées s'il y a déjà 8 requêtes dans le système. Donc il y a redirection lorsque le système atteint 9 requêtes dans le système. Il suffit de calculer π_9 pour connaitre la probabilité que la requête soit redirigée. D'après la loi de Little $\pi_n = (1 - \rho)\,\rho^n$. La valeur qui nous donne la réponse exacte est $\pi_9 = (1 - \rho)\,\rho^9$.

On calcule systématiquement les valeurs de $\pi_n \; \forall \; n = 1, \ldots, 9$

$$\Pi_1 = (1 - \rho)\,\rho^1 \; or \; \rho = \frac{\lambda}{\mu} \Rightarrow \rho = \frac{100}{200} = 0.5;$$

$$\Pi_1 = (1 - 0.5)0.5 = 0.25\ requête/seconde.$$

$$\Pi_2 = (1 - \rho)\,\rho^2 = (1 - 0.5)0.25 = 0{,}125\ requête/seconde.$$

$$\Pi_3 = (1 - \rho)\,\rho^3 = (1 - 0.5)(0.5)^3 = 0.5 * 0{,}125$$

$$= 0{,}0625 requête/seconde$$

$$\Pi_4 = (1 - \rho)\,\rho^4 = = 0.5 * (0.5)^4 = 0{,}03125 requête/seconde$$

$$\Pi_5 = (1 - \rho)\,\rho^5 = = 0.5 * (0.5)^5 = 0{,}015625 requête/seconde$$

$$\Pi_6 = (1 - \rho)\,\rho^6 = = 0.5 * (0.5)^6 = 0{,}0078125 requête/seconde$$

$$\Pi_7 = (1 - \rho)\,\rho^7 = = 0.5 * 0{,}00390625 = 0{,}00390625 requête/seconde$$

$$\Pi_8 = (1 - \rho)\,\rho^8 = = 0.5 * (0.5)^8 = 0{,}001953125 requête/seconde$$

$\Pi_9 = (1 - \rho)\,\rho^9 = 0.5 * 0{,}000976562 = 0{,}0009765625$requête/seconde soit 1/1024 requête/seconde

Comme le système reçoit 100 requêtes/seconde, le serveur auxiliaire reçoit $100 * 0{,}0009765625 = 0{,}09765625\ requêtes/seconde$.

En conclusion de ce chapitre, notre système est semblable à un système de files d'attente tournant sous le modèle client-serveur. La file d'attente est constituée par les machines étudiants qui se connectent sur notre système et les serveurs sont les machines professeurs sur lesquels les étudiants se connectent pour s'auto évaluer. Les étudiants arrivent suivant un processus de poisson de taux d'arrivée moyen λ et le temps de service suivant une loi exponentielle de paramètre μ.Nous avons essayé de définir la notion de file d'attente en premier lieu par après nous avons introduit la notion de Kendal pour expliciter les paramètres généraux d'une file d'attente. Nous avons aussi développé les différents types de file d'attente et leurs caractéristiques principaux. Ensuite nous avons développé et modélisé la notion de réseau de file d'attente et dernièrement essayé de faire une simulation en pratiquant les notions développées ci-haut pour vérifier la fiabilité et conditions de performance et de stabilité du système. Le point final était la mise en pratique et la démonstration de ce que nous avons développé en théorie.

CHAP V : IMPLEMENTATION DU SYSTEME DE GESTION DU PROCESSUS D'EVALUATION EN LIGNE DES ETUDIANTS

V.1 Introduction

Dans cette partie, nous allons réaliser l'architecture logicielle élaborée pendant la phase de modélisation. Le nouveau système est transposé dans un Framework PHP CodeIgniter et les différents composants du système sont développés à l'aide d'outils et technologies liés à ce Framework PHP. Le résultat de cette modélisation est un produit logiciel qui sera utilisé dans notre système d'évaluation en ligne des étudiants. Comme notre système d'information est une application web, nous présenterons en premier lieu l'architecture client/serveur qu'utilise notre système. Quant à la deuxième partie, nous spécifierons les outils, langages et les technologies utilisés.

V.2 Architecture client/serveur

Une architecture client-serveur représente l'environnement dans lequel des applications de machines clientes communiquent avec des applications de machines de type serveurs [16]. L'exemple classique est le navigateur Web d'un client qui demande (on parle de "requête") le contenu d'une page Web à un serveur Web qui lui renvoie le résultat (on parle de "réponse"). Si toutes les ressources nécessaires sont présentes sur un seul serveur, on parle d'architecture à deux niveaux ou 2 tiers (1 client + 1 serveur), si certaines ressources sont présentes sur un deuxième serveur (par exemple des bases de données), on parle d'architecture à trois niveaux ou 3 tiers (1 client interroge le premier serveur qui lui-même interroge le deuxième serveur), au-delà de 3 acteurs, on parle d'architecture à n tiers. Un client : Les caractéristiques d'un client sont les suivantes : Il est d'abord actif (ou maître), il envoie des requêtes au serveur, il attend et reçoit les réponses du serveur ; Un serveur : Un serveur est initialement passif, il attend, il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients. Dès qu'une requête lui parvient, il la traite et envoie une réponse ; Le dialogue : Le client et le serveur doivent bien sûr utiliser le même protocole de communication. Un serveur est généralement capable de servir plusieurs clients simultanément. Il existe aussi un autre type d'architecture réseau qui est le pair à pair (P2P), dans lequel chaque ordinateur ou logiciel est à la fois client et serveur. Cette architecture permet à plusieurs ordinateurs de communiquer via un réseau, de partager simplement des fichiers le plus souvent, mais également des flux multimédias ou encore un service (comme la téléphonie avec Skype par exemple), … sur internet.

V.3 Outils et langages de développement utilisés

Plusieurs langages de programmation les plus populaires évoluent en fonction des spécifications qui ont été établies au préalable. La spécification pour un langage de programmation est la source de référence pour sa syntaxe et son utilisation. Elle contient des informations détaillées sur tous les aspects du langage et définit un cadre pour son implémentation. La technologie de développement d'un nouveau système est le Framework CodeIgniter. Le Framework CodeIgniter, bien que jouissant d'une importante popularité dans le monde du développement Web, Dans le souci de concevoir une application web, nous avons choisi le Framework CodeIgniter comme technologie d'implémentation de notre application. CodeIgniter est un Framework de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur http, mais pouvant fonctionner comme n'importe quel langage interprété de toute façon. Sublime Text est un environnement de développement intègre la plupart des fonctionnalités de base d'un éditeur de texte, dont la coloration syntaxique personnalisable, l'auto-complétion, un système de plugins… L'éditeur propose cependant des fonctions plus avancées, dont Minimap : prévisualisation de tout le fichier dans une barre latérale, Sélection et édition dans plusieurs sections de code en parallèle, Marque-page au sein même des fichiers, Sauvegarde automatique, Recherche et remplacement par expressions régulières, Personnalisation des raccourcis clavier. Le logiciel propose également d'importer des packages (pour ajouter des langages ou bien des fonctionnalités de Sublime Text)**.** MySQL/Maria DB se compose de l'un des systèmes de gestion de base de données relationnelle les plus populaires au monde. En combinaison avec le serveur Web Apache et le langage script PHP, MySQL sert à l'enregistrement de données pour des services Web. Les versions actuelles de XAMPP favorisaient Maria DB à l'insu de MySQL comme gestionnaire de base de données, marquant un détachement avec ce dernier. XAMPP est un ensemble de logiciels libres. Le nom est un acronyme venant des initiales de tous les composants de cette suite. Ce dernier réunit donc le serveur Web **A**pache, la base de données relationnelle et système d'exploitation MySQL ou **M**aria DB ainsi que les langages scripts **P**erl et **P**HP. L'initiale X représente un logiciel pouvant tourner sur quatre systèmes d'exploitation, à savoir Linux, Unix, Windows et Mac OS X [17]. Apache est le serveur Web open source. Apache est utilisé mondialement et permet de délivrer des contenus Web. L'application de serveur est mise à disposition en open source par l'Apache Software Fondation. Boostrap est une combinaison des langages HTML, CSS et JavaScript qui fournit aux développeurs des outils pour créer un site facilement. Ce Framework est pensé pour développer des sites avec un design responsif, qui s'adapte à tout type d'écran, et en priorité pour

les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de Framework un "Front-End Framework". PHP est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP. PHP est un langage impératif orienté objet. Il s'agit d'un langage de script interprété côté serveur. HTML désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. Il permet, entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu. CSS correspond à un langage informatique permettant de mettre en forme des pages web (HTML ou XML). Ce langage est donc composé des fameuses « feuilles de style en cascade » également appelées fichiers CSS (.css) et contient des éléments de codage. JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. A chaque fois qu'une page web fait plus que simplement afficher du contenu statique, afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéos défilants, ou autre, JavaScript a de bonnes chances d'être impliqué. C'est la troisième couche des technologies standards du web, les deux premières (HTML et CSS) étant couvertes bien plus en détail dans d'autres tutoriels sur MDN [18]. Microsoft Visio (anciennement Microsoft Office Visio) est une application de création de diagrammes et de graphiques vectoriels et fait partie de la famille Microsoft Office. Le produit a été introduit pour la première fois en 1992, fabriqué par la Shareware Corporation, renommée ensuite Visio Corporation. Il a été acquis par Microsoft en 2000. Une version allégée de Visio est désormais incluse dans toutes les unités de service commerciales de Microsoft 365 et est connue sous le nom de Visio in Microsoft 365. Il existe deux autres UGS par abonnement. Le plan Visio 1 comprend l'application Web Visio, tandis que le plan Visio 2 donne accès à l'application Web et à l'application de bureau [19].

V.4 Technologie de développement utilisée dans notre projet

Pour développer notre système nous nous sommes servis de l'architecture logicielle modèle-vue -contrôleur du Framework « CodeIgniter ». Le patron MVC est une manière d'organiser le code. L'idée est de séparer le code qui sert à accéder aux données de celui servant à la gestion de l'affichage et de celui destiné à gérer les requêtes des utilisateurs ainsi que les interactions entre l'affichage et les données. Pour cela, nous utiliserons respectivement des modèles, des vues et des contrôleurs. CodeIgniter est un Framework PHP simple, léger, souple et performant conçu sur le modèle de développement MVC. Le modèle représente les données. Il s'agit bien souvent

d'une classe dont les méthodes permettent d'effectuer les actions de création, de lecture, de mise à jour et de suppression de vos données. C'est dans la vue que nous allons définir ce qui doit être affiché. En principe, pour une application web, c'est le seul endroit où nous pourrions retrouver du code HTML. Le contrôleur va recevoir les requêtes de l'utilisateur, charger les modèles nécessaires ainsi que les vues adéquates et retourner le résultat. Avec une application web, les requêtes se font au travers des URL.

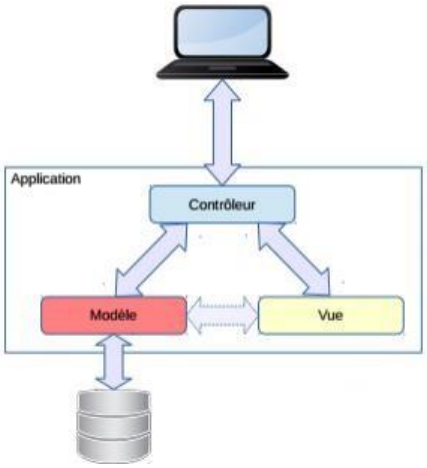


Figure 12: Modèle MVC

Compte tenu de la nature de notre système**,** dans ce chapitre nous avons présenté l'architecture client-serveur qui est une architecture utilisée par notre système. La présentation des outils, langages de développement et technologies utilisés pour pouvoir implémenter le système a paru inéluctable afin de montre aux internautes comment nous avons pu développer ce système.

CHAP VI : PROTECTION DES DONNEES CONTRE LES ACCES NON AUTORISES

VI.1 Introduction

Aujourd'hui, la protection des données est indispensable pour garantir la confidentialité entre les communicants. La sécurité des données est la pratique qui vise à protéger les informations contre les accès non autorisés, l'altération et le vol tout au long de leur cycle de vie. Il s'agit d'un concept qui englobe tous les aspects de la sécurité de l'information avec les algorithmes de chiffrement, de la sécurité physique du matériel au moyen des cadenas et des dispositifs de stockage aux contrôles administratifs et d'accès, en passant par la sécurité logique des applications logicielles. Avant l'usage généralisé d'équipements informatiques, la sécurité de l'information était assurée par des moyens physiques (cadenas). L'évolution de l'informatique n'a cessé d'apporter d'autres mécanismes de sécurité plus flexible, l'introduction de l'ordinateur et des NTIC pour la protection automatique des données stockées est devenu une solution répondant admirablement à tous ces problèmes. Ce besoin est accentué pour un système accessible via un téléphone public ou un réseau de données. Ainsi, donnons à cette collection d'outils conçus pour protéger et contrecarrer les pirates le nom de sécurité informatique. La sécurité satisfait au moins aux cinq principaux objectifs : La confidentialité, l'intégrité, la disponibilité, la non-répudiation et l'authentification. La confidentialité est le maintien du secret des informations. Dans le cadre d'un système d'information, cela peut être vu comme une protection des données contre une divulgation non autorisée. Deux actions complémentaires permettant d'assurer la confidentialité des données sont la limitation de leurs accès par un mécanisme de contrôle d'accès et la transformation des données par des procédures de chiffrement. Afin de satisfaire à tous ces critères, nous avons procédé à la gestion de l'accès aux données avec le principe de l'authentification ainsi que la garantie de la confidentialité par le chiffrement de Vigenère qui est un système cryptographique théoriquement et pratiquement sûr. L'intégrité permet de certifier que les données, les traitements ou les services n'ont pas été modifiés, altérés ou détruits tant de façon intentionnelle qu'accidentelle. L'altération est principalement occasionnée par le média de transmission mais peut être provenir du système d'information. Il faut également veiller à garantir la protection des données d'une écoute active sur le réseau. La disponibilité est de pair avec son accessibilité. Une ressource doit être accessible, avec un temps de réponse acceptable. La disponibilité des services, systèmes et données est obtenue par un dimensionnement approprié et une gestion opérationnelle des ressources et des services. Ce paramètre est mesuré par une montée en charge du système afin de s'assurer de la totale

disponibilité du service. La non-répudiation permet une transaction ne peut être niée par aucun des correspondants. La non-répudiation de l'origine et de la réception des données prouve que les données ont bien été reçues. Cela se fait par le biais de certificats numériques grâce à une clé privée. L'authentification limite l'accès aux personnes autorisées. Il faut s'assurer de l'identité d'un utilisateur avant l'échange de données. Dans notre système l'authentification est assurée du fait que chacun dans le système verra les informations qui lui sont autorisées après avoir s'authentifier. Ainsi les informations sensibles comme les mots de passe sont chiffrées à l'aide d'un cryptosystème fiable et chaque acteur devra s'authentifier pour avoir accès dans le système.

VI.2 Protection des données par le chiffrement de Vigenère

Le chiffrement des données est un moyen de convertir les données du texte en clair (non chiffré) en texte crypté (chiffré). Les utilisateurs peuvent accéder aux données chiffrées avec une clé de chiffrement et aux données déchiffrées à l'aide d'une clé de déchiffrement [20]. D'énormes quantités d'informations sensibles sont gérées et stockées en ligne dans le nuage ou sur des serveurs connectés. Le chiffrement utilise la cybersécurité pour se défendre contre les attaques par force brute et les cyberattaques, y compris les logiciels malveillants et les rançongiciels. Le chiffrement des données fonctionne en sécurisant les données numériques transmises dans le nuage et les systèmes informatiques. Il existe deux types de données numériques, les données transmises ou données en cours et les données numériques stockées ou données au repos. Les algorithmes de chiffrement modernes ont remplacé la norme de chiffrement de données devenue obsolète pour protéger les données. Ces algorithmes protègent les informations et alimentent les initiatives en matière de sécurité, notamment l'intégrité, l'authentification et l'irréfutabilité. Les algorithmes authentifient d'abord un message pour en vérifier l'origine. Ensuite, ils en analysent l'intégrité pour vérifier que le contenu n'a pas été modifié. Enfin, l'initiative d'irréfutabilité empêche les expéditeurs de nier une activité légitime.

VI.2 .1 Types de chiffrement des données

Il existe diverses méthodes de chiffrement, chacune d'entre elles ayant été développée en fonction de besoins différents en matière de sécurité. Les deux principaux types de chiffrement des données sont le chiffrement asymétrique et le chiffrement symétrique.

VI.2 .2 Chiffrement asymétrique

Le chiffrement asymétrique, également connu sous le nom de cryptographie à clé publique, chiffre et déchiffre les données à l'aide de deux clés cryptographiques asymétriques distinctes. Ces deux clés sont appelées « clé publique » et « clé privée ».

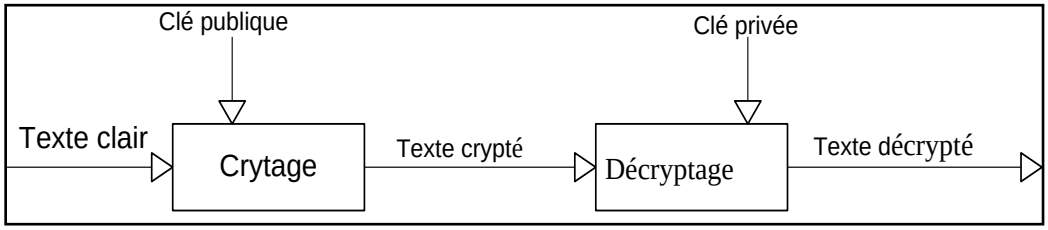


Figure 13:Processus de cryptage et de décryptage de l'algorithme asymétrique.

Deux exemples courants de chiffrement asymétrique sont RSA, du nom des informaticiens Ron Rivest, Adi Shamir et Leonard Adleman, qui est un algorithme couramment utilisé pour chiffrer les données à l'aide d'une clé publique et pour les déchiffrer avec une clé privée afin de les transmettre en toute sécurité et PKI qui est un moyen de gérer les clés de chiffrement par l'émission et la gestion de certificats numériques.

VI.2 .3. Chiffrement symétrique

Le chiffrement symétrique est un type de chiffrement dans lequel une seule clé symétrique secrète est utilisée pour chiffrer le texte en clair et déchiffrer le texte crypté. Voici quelques exemples de chiffrement symétrique courants :DES est un algorithme de chiffrement par blocs de bas niveau qui convertit le texte brut en blocs de 64 bits et les convertit en texte chiffré à l'aide de clés de 48 bits ,Triple DES exécute le chiffrement DES à trois reprises en chiffrant, en déchiffrant, puis en chiffrant à nouveau les données , Norme de chiffrement avancée AES est souvent désigné comme la référence absolue en termes de cryptage des données et est utilisé dans le monde entier, notamment par le gouvernement américain et Twofish qui est considéré comme l'un des algorithmes de chiffrement les plus rapides et son utilisation est gratuite.

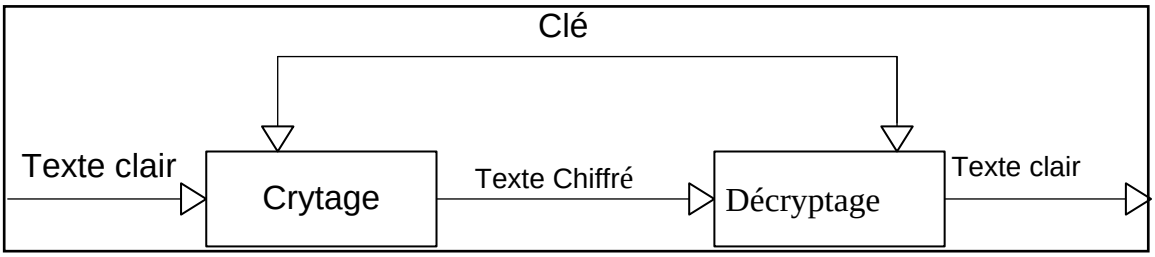


Figure 14:Processus de cryptage et de décryptage de l'algorithme symétrique.

Parmi toutes les cryptosystèmes, nous avons décidé d'utiliser le chiffrement de Vigenère qui est une méthode de chiffrement symétrique utilisant une substitution poly alphabétique pour chiffrer et déchiffrer le message secret afin de protéger les données contre les personnes males intentionnées.

VI.3 Chiffrement de Vigenère comme outils de sécurité

Le chiffre de Vigenère est un système de chiffrement par substitution poly alphabétique dans lequel une même lettre du message clair peut, suivant sa position dans celui-ci, être remplacée par des lettres différentes, contrairement à un système de chiffrement mono alphabétique comme le chiffre_de César (qu'il utilise cependant comme composant). Cette méthode résiste ainsi à l'analyse de fréquences, ce qui est un avantage décisif sur les chiffrements mono alphabétiques (21). L'idée de Vigenère est d'utiliser un chiffre de César, mais où le décalage utilisé change de lettres en lettres. Pour cela, on utilise une table composée de 26 alphabets, écrits dans l'ordre, mais décalés de ligne en ligne d'un caractère. On écrit encore en haut un alphabet complet, pour la clé, et à gauche, verticalement, un dernier alphabet, pour le texte à coder :

Tableau 6:Table de Vigenère

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Procéduralement, pour coder un message, on choisit une clé qui sera un mot de longueur arbitraire. On écrit ensuite cette clé sous le message à coder, en la répétant aussi souvent que nécessaire pour que sous chaque lettre du message à coder, on trouve une lettre de la clé. Pour coder, on regarde dans le tableau l'intersection de la ligne de la lettre à coder avec la colonne de la lettre de la clé. Pour déchiffrer, il suffit de prendre la colonne correspondante à la première lettre de la clé et de chercher dans cette colonne la lettre correspondant à la première lettre de la phrase codée, elle sera sur la ligne de la première lettre de la phrase normale (lettre qui se trouve dans la colonne la plus à gauche). Mathématiquement, pour coder un message, on identifie les lettres de l'alphabet aux nombres de 0 à 25 (A=0, B=1...). Les opérations de chiffrement et de déchiffrement sont, pour chaque lettre, celles du chiffre de César. En désignant la $i^{ème}$ lettre du texte clair par T[i], la $i^{ème}$ du chiffre par C[i], et la $i^{ème}$ lettre de la clé, répétée suffisamment de fois, par Clés[i], elle se formalise par C[i] = (T[i] + Clés [i]) modulo 26 ,T[i] = (C[i] - Clés[i]) modulo 26. Où x modulo 26 désigne le reste de la division entière de x par 26. Pour le chiffrement il suffit d'effectuer l'addition des deux lettres puis de soustraire 26 si le résultat dépasse 26. Pour le déchiffrement il suffit d'effectuer la soustraction et d'additionner 26 si le résultat est négatif. Le déchiffrement est aussi une opération identique à celle du chiffrement pour la clé obtenue par clé[i] = 26 - Clé[i]. Un disque à chiffrer, qui utilise une représentation circulaire de l'alphabet

(après Z on a A), permet de réaliser directement cette opération. Exemple d'application : Chiffrement de la phrase suivante avec la clé suivante : Texte clair : JE SUIS EN VAVANCES EN ESTONIE, Clé : BALAIS, Première étape : JE SUIS EN VAVANCES EN ESTONIE BA LAIS BA LAISBALA IS BALAISB. L'étape suivante est de chercher la lettre d'intersection dans la table de Vigenère entre la lettre du texte en clair et la lettre de la clé qui lui correspond en dessous, on répète le même processus jusqu'à parcourir toute la phrase en clair. Ce qui donne avec notre exemple au croisement de la colonne J (première lettre du mot je) et de la rangée B (première lettre du mot-clé balais) donne la lettre K. Pour chiffrer le E (seconde lettre du mot je) on croise la colonne E avec la rangée A (seconde lettre de balais) ce qui donne un E ; pour chiffrer le S du mot suis (troisième lettre de la phrase à chiffrer) on croise la colonne S avec la rangée L (troisième lettre de balais) ce qui correspond à la lettre D et ainsi de suite : la phrase à chiffrer donnera la suite suivante de lettres : KEDUQKFNGAKSOCPSMFFSEOVAF. En partant du principe mathématique que T[i] = (C[i] - Clés[i]) modulo 26 pour décrypter le texte crypté nous avons :

1. $T_{(1)}$ = K-B => 10-1 = 9 modulo 26 donne 9 correspond à la lettre J ;
2. $T_{(2)}$ = E-A => 4-0 = 4 modulo 26 donne 4 correspond à la lettre E ;
3. $T_{(3)}$ = D – L => 3 -11 = -8, comme $T_{(3)}$ est négatif, on va sommer $T_{(3)}$ avec 26 ce qui donne -8 + 26 = 18 modulo 26 donne 18 correspond à la lettre S ;
4. $T_{(4)}$ = U – A => 20 – 0 = 20 modulo 26 donne 20 correspond à la lettre U ;
5. $T_{(5)}$ = Q – I => 16 – 8 = 8 modulo 26 donne 8 correspond à la lettre I ;
6. $T_{(6)}$ = K – S=>10 – 18 = -8 + 26 =18modulo26 donne 18 donne la lettre S ;
7. $T_{(7)}$ = F – B => 5 – 1 =4 modulo 26 donne 4 correspond à la lettre E ;
8. $T_{(8)}$ = N – A => 13 – 0 =13 modulo 26 donne 13 correspond à la lettre N ;
9. $T_{(9)}$ = G – L => 6 – 11 = -5 + 26=21modulo26 donne 21 correspond à la lettre V ;
10. $T_{(10)}$ = A – A => 0 – 0 = 0 modulo 26 donne 0 correspond à la lettre A ;
11. $T_{(11)}$ = K – I => 10 – 8 =2 modulo 26 donne 2 correspond à la lettre C ;
12. $T_{(12)}$ = S – S => 18 – 18 = 0 modulo 26 donne 0 correspond à la lettre A ;
13. $T_{(13)}$ = O – B => 14 – 1=13 modulo 26 donne 13 correspond à la lettre N ;
14. $T_{(14)}$ = C – A => 2 – 0 = 2 modulo 26 donne 2 correspond à la lettre C ;

15. $T_{(15)}$ = P – L => 15 – 11 = 4 modulo 26 donne 4 correspond à la lettre E ;
16. $T_{(16)}$ = S – A => 18 – 0 = 18 modulo 26 donne 18 correspond à la lettre S ;
17. $T_{(17)}$ = M – I => 12 – 8 = 4 modulo 26 donne 4 correspond à la lettre E ;
18. $T_{(18)}$ = F – S => 5 – 18 = -13 + 26 = 13 modulo 26 donne 13 correspond à la N ;
19. $T_{(19)}$ = F – B => 5 – 1 = 4 modulo 26 donne 4 correspond à la lettre E ;
20. $T_{(20)}$ = S – A => 18 – 0 = 18 modulo 26 donne 18 correspond à la lettre S ;
21. $T_{(21)}$ = E – L => 4 – 11 = -7 +26 = 19 modulo 26 donne correspond à la lettre T ;
22. $T_{(22)}$ = O- A => 14 – 0 = 14 modulo 26 donne 14 correspond à la lettre O ;
23. $T_{(23)}$ = V – I => 21 – 8 =13 modulo 26 donne 13 correspond à la lettre N ;
24. $T_{(24)}$ = A – S => 0 – 18 = -18 +26 = 8modulo26 donne 8 correspond à la lettre I ;
25. $T_{(25)}$ = F – B => 5 – 1 =4 modulo 26 donne 4 correspond à la lettre E.

Le texte décrypté donne : JESUISENVACANCESENESTONIE qui correspond au texte clair.

Avec A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I=8, J = 9, K = 10, L = 11, M = 12, N = 13, O = 14, P = 15, Q = 16, R = 17, S = 18, T = 19, U = 20, V = 21, W = 22, X = 23, Y= 24, Z = 25.

NB : Le chiffre de Vigenère n'accepte pas les caractères spéciaux ni les lettres minuscules, nous avons travaillé avec les 26 lettre de l'alphabet.

VI.3 .1 Cryptanalyse du chiffrement de Vigenère

Le cryptosystème de Vigenère a longtemps été considéré incassable pendant plusieurs siècles. Cependant, sa cryptanalyse est très aisée à l'aide des ordinateurs. En supposant la longueur n de la clef connue, un message peut être décrypté rapidement de la façon suivante : Décomposer le texte chiffré en n textes. Le premier texte est la suite des lettres en positions 0, n, 2n, 3n,… Le deuxième est la suite des lettres en positions 1, n+1,2n+1,3n+1,…, pour chacun des n textes, compter le nombre d'occurrences de chaque lettre, dans chacun des n textes, la lettre la plus courante correspond presque surement à la lettre E. Les autres lettres suivent par décalage et recomposer les n textes ainsi décryptés pour composer le message d'origine.

VI.3 .2 Application du chiffrement de Vigenère dans notre projet

Nous avons développé une application qui repose sur le modèle client-serveur raison pour laquelle la sécurisation des données transitant sur le réseau est primordiale. Rappelons le principe de fonctionnement de ce modèle, le client envoie des requêtes au serveur pour lui demander la ressource qui lui est réservée et le serveur traite sa demande et envoie la réponse correspondant à sa demande au client. Tout échange se fait sur le réseau d'où la notion de chiffrement de données est requise pour déjouer toute tentative d'écoute des échanges. Nous allons faire un exemple de chiffrement et déchiffrement par une clé, ceux qui ont une connexion internet vous pouvez vérifier avec le site dcode.fr pour s'assurer que le texte chiffré est correct : Texte clair : Les étudiants de l'université du Burundi, Clé de chiffrement : master. Le programme lui-même va détecter toutes les espaces vides, les lettres minuscules et les caractères spéciaux afin de les supprimer. Voici sur la capture d'écran le résultat après avoir cliqué sur le bouton crypter :



Figure 15:Application de chiffrement avec chiffrement de Vigenère.

Si on veut décrypter le texte XEKXXLPISGXJPEDNRZHEJLMKQDMUYIGNVB que nous avons obtenu après le cryptage en utilisant toujours la même clé qui est master nous obtenons le texte original mais en majuscule et aussi avec les espaces supprimés.

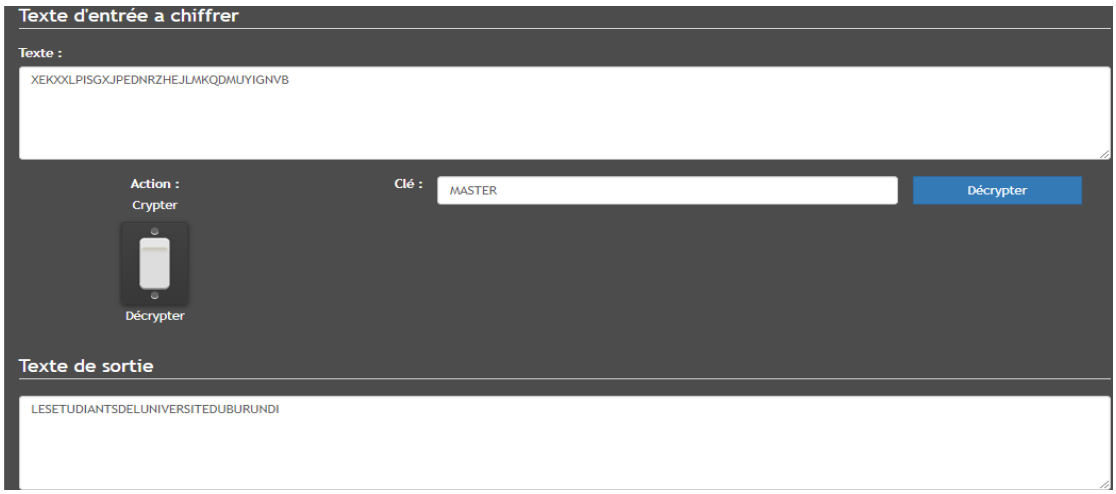


Figure 16:Déchiffrement avec chiffrement de Vigenère.

VI .4 Estimation du coût de mise en œuvre du logiciel

L'estimation permet de connaître le coût d'une "vue de l'esprit (vision) " qui deviendra réalité au bout d'un temps fini. L'estimation concerne l'effort de développement (coût), la durée du projet (temps) et autre (équipement, voyage, formation), et ajouter (la logique des calculs, les hypothèses). Les estimations sont basées sur des modèles mathématiques reposant sur divers paramètres. Dans notre travail nous avons utilisé la méthode COCOMO qui est une méthode utilisée pour estimer le coût d'un logiciel à partir des paramètres d'entrée.

VI.5 Méthode COnstructive COst MOdel

Le modèle d'estimation COCOMO est un modèle introduit par Dr Barry Boehm permettant de définir une estimation de l'effort à fournir dans un développement logiciel et la durée que ce dernier prendra en fonction des ressources allouées. Cette méthode évalue l'effort en nombre de mois-hommes (M/h) qui représente l'équivalent du travail d'une personne pendant un mois, généralement 20 jours. Elle permet aussi de faire catégoriser un projet selon le nombre des lignes-codes du logiciel. Un projet est qualifié de simple ou projet organique si le nombre de lignes est inférieur à 50 000, moyen ou semi détaché si le nombre de lignes est compris entre 50 000 et 300 000 et complexe ou imbriqué si le nombre de lignes est supérieur à 300 000[22]. Barry Boehm a proposé les formules pour calculer la charge, le délai ainsi que la taille de l'équipe pour la réalisation du projet.

$$\text{Charge} = a \times (KLSL)^{b}$$

La taille des projets se mesure en fonction de leur charge. Si Charge < 6 HM : Très petit projet, si 12 HM< Charge < 30 HM : Projet moyen, si 30 HM < Charge < 100 HM : Grand projet et si Charge > 100 HM : Très grand projet.

$$Délai = c \times (Charge)^{d}$$

$$Taille\ moyenne\ d'équipe = \frac{Charge}{Délai}$$

Avec KLSL est le nombre de milliers de lignes du code source et a, b, c et d des paramètres qui dépendent de la classification du projet.

Tableau 7:Paramètres du modèle COCOMO simple

Type de projet	Paramètres Charge (M/h)	Paramètres Délai(M)
Organique	a = 2,4	c =2,5
	b = 1,05	d = 0,38
Semi-détaché	a = 3	c = 2, 5
	b = 1,12	d = 0,35
Imbriqué	a = 3,6	c = 2,5
	b = 1,2	d = 0,32

Les paramètres a, b, c et d donnés dans le tableau ci-dessus nous permettent de calculer l'effort et le temps de développement selon le type de projet. Les paramètres a et b varient selon le modèle COCOMO utilisé.

Tableau 8:Valeurs Paramètres du modèle COCOMO simple

Type de projet	Charge (Effort)	Durée (Temps de Développement)
Organique	HM = 2.4 ∗(KLSL) $^{1.05}$	TDEV = 2.5 ∗(HM) $^{0.38}$

Semi-détaché	$HM = 3 * (KLSL)^{1.12}$	$TDEV = 2.5 * (HM)^{0.35}$
Imbriqué	$HM = 3.6 * (KLSL)^{1.2}$	$TDEV = 2.5 * (HM)^{0.32}$

Avant d'estimer l'effort ou la durée d'un projet, il est primordial d'avoir une estimation de la taille la plus précise possible car la taille est la variable principale dans l'estimation de l'effort.

VI.5.1 Méthode COCOMO II

COCOMO-II est la version révisée du modèle original COCOMO et est développé à l'Université de Californie du Sud. Il s'agit du modèle qui permet d'estimer le coût, l'effort et le calendrier lors de la planification d'une nouvelle activité de développement de logiciels. Notre projet est une application web utilisant l'architecture client-serveur. Nous avons fait une comparaison entre les méthodes d'estimation des coûts des logiciels mais la méthode WebMo semble plus parfaite pour répondre à ce problème.

VI.5.2 Méthode WEBMO

Pour le développement rapide des applications Web et leur estimation des coûts, nous avons pensé à une version COCOMO adaptée au Web et conçue par Reifer sous le nom de WebMo. Cette nouvelle méthode diffère de COCOMO II par l'introduction d'une nouvelle métrique de mesure de la taille appelée Objets Web (Web Objects) et par une récalibration des paramètres de COCOMO II [23]. Cependant, selon le groupe international [ISBSG2004], les projets Web offrent un champ nouveau d'application des modèles d'estimation du génie logiciel tels que le modèle COCOMO II ou le modèle des points de fonction, en particulier pour les applications Client-Serveur dans le contexte du Web. Cette méthode est le résultat d'une récalibration des paramètres de COCOMO II au moyen de 64 projets Web, ce qui se traduit par les modifications suivantes : Le nombre de multiplicateurs d'effort (EM) est passé de 17 à 9. Les 9 multiplicateurs correspondent aux 7 multiplicateurs de la version Early design (version abrégée) de COCOMO II à laquelle Reifer a ajouté 2 variables qui figuraient comme facteurs d'échelle dans COCOMO II , la valeur des 9 multiplicateurs d'effort retenus dans WebMo a été modifiée par rapport à leur valeur dans COCOMO II , les niveaux d'exigence dans WebMo ont été réduits à 5 (de très bas à très haut) alors que dans COCOMO II il y a 7 niveaux d'exigence , les 5 facteurs d'échelle (SF) de COCOMO II disparaissent dans WebMo ,pour valider WebMo avec COCOMO II, Reifer a

conçu la table de conversion qui convertit les objets Web en lignes de code (SLOC).L'équation de base de WebMo pour l'effort est alors la suivante :

$$\text{Effort} = A * (\text{Taille})^{P_1} * \prod_{i=1}^{9} EM_i$$

La constante A et l'exposant d'échelle P_l varient selon le domaine d'application, comme suit :

Tableau 9:Paramètres WebMo

Domaine d'application	**A**	**P1**
Portails Web	2.1	1.00
Utilitaires d'information Web	2.1	1.00
Applications commerciales et financières	2.7	1.05
Applications B2B	2.0	1.00
Commerce électronique	2.3	1.03

Nous allons calculer le coût du projet en se référant d'abord au type de projet et l'équation de base de WebMo nous fournira le coût estimatoire de notre travail. Le paramètre le plus prépondérant à déterminer est la taille du projet qui est égal au nombre de ligne que comporte notre système d'information.

VI .6 Calcul du coût de mise en place du logiciel

Le calcul du coût d'un logiciel varie selon le domaine d'application. C'est le domaine d'application qui nous permet de choisir la méthode de calcul à utiliser. Notre projet est enregistré dans la catégorie des applications web, l'utilisation des paramètres du portail web s'avère inévitable pour estimer le coût. Le premier paramètre à déterminer est la taille du projet.

$$Taille = 15246$$

$$A = 2.1$$

$$P_1 = 1.00$$

$$\prod_{i=1}^{9} EM_i = 1$$

$$Charge = 15246 * 2.1 = 32\ M/h$$

$$TDEV = 2.5 * (32)0.38 = 9{,}330329915 \cong 9\ Mois$$

$$Taille\ moyenne\ d'équipe = \frac{Charge}{TDEV} = \frac{32}{9} = 3.5 \cong 3\ personnes$$

Ce projet exigerait 9 mois s'il devait être exécuté par une seule personne mais comme la durée de réalisation du travail de fin d'étude de master est estimée à 6 mois nous avons réalisé ce travail en 6 mois comme on va le montrer dans le point de planification du chronogramme des activités à l'aide du diagramme de Gantt. Nous allons essayer de valoriser ce travail en se basant sur les paramètres que nous venons de démontrer en tenant compte de la réalité de l'environnement. Si nous posons un salaire forfaitaire par mois de 700000FBU pour un développeur senior, on aura comme coût financier : $700000 * 9 = 6.300.000FBU$. Le coût financier de développement de ce travail est évalué à 6.300.000 FBu

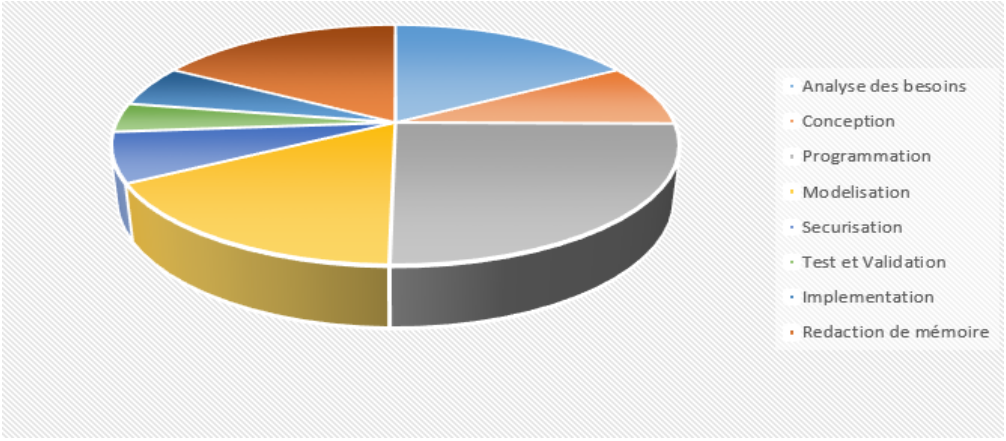
VI .7 Calendrier de réalisation du projet

En gestion de projet, le calendrier est la liste des activités et des livrables d'un projet. Le calendrier comprend aussi généralement les dates de début et de fin prévues, la durée et les ressources affectées à chaque activité. Un calendrier de projet efficace est essentiel pour bien gérer le temps [24]. L'élaboration du calendrier est une activité nécessaire pour montre le chronogramme des activités. Notre projet est un travail de fin d'étude de master qui doit se dérouler dans une période de 6 mois conformément au règlement académique de l'université du Burundi. Il existe plusieurs techniques utilisées pour élaborer le calendrier des activités d'un projet mais le diagramme de Gantt est le plus courant dans ce domaine, nous allons essayer de détailler les activités en utilisant ce digramme.

Tableau 10:Chronogramme des activités

Tâche	Date de début	Date de fin	Délai maximal
1. Analyse des besoins	16/05/2022	15/06/2022	1mois
2. Conception	15/06/2022	30/06/2022	15jours
3. Programmation	30/06/2022	15/08/2022	1mois 15jours
4.Modélisation	15/08/2022	15/09/2022	1mois
5. Sécurisation	15/09/2022	27/09/2022	12jours
6. Test et Validation	27/09/2022	03/10/2022	7jours
6. Implémentation	03/10/2022	13/10/2022	10jours
7. Rédaction du mémoire	13/10/2022	13/11/2022	1mois
Total des mois	16/05/2022	15/11/2022	**6 mois**

Tableau de bord du chronogramme des activités



Ce graphique montre la répartition des activités de développement du système. C'est la phase de développement qui comporte beaucoup de jours si on fait interpréter les données visualisées sur ce graphique. La planification du calendrier du projet indique la manière dont les tâches dépendent les unes des autres. La fin d'une tâche est marquée par le début d'une autre tâche, C'est le diagramme de Gantt qui va nous aider à modéliser toutes ces activités dressées dans le tableau ci-dessus.

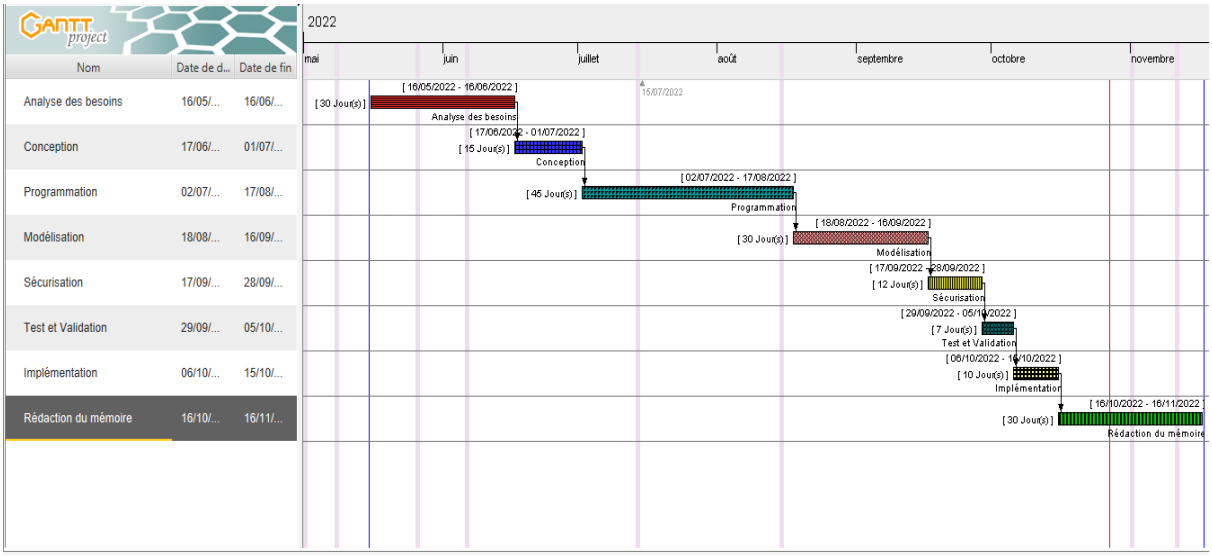


Figure 17: Diagramme de Gantt des activités du projet.

VI. 8 Présentation de quelques fenêtres du système développé

Notre application possède plusieurs interfaces mais nous présenterons quelques-unes jugées importantes.

VI.8.1 Authentification

L'authentification est un processus par lequel un système informatique vérifie et valide l'identité d'un utilisateur. Tout utilisateur du système doit d'abord s'authentifier pour accéder à son espace membre. Après authentification, une page d'accueil s'affiche et différentes fenêtres se présentent selon le profil de l'utilisateur.

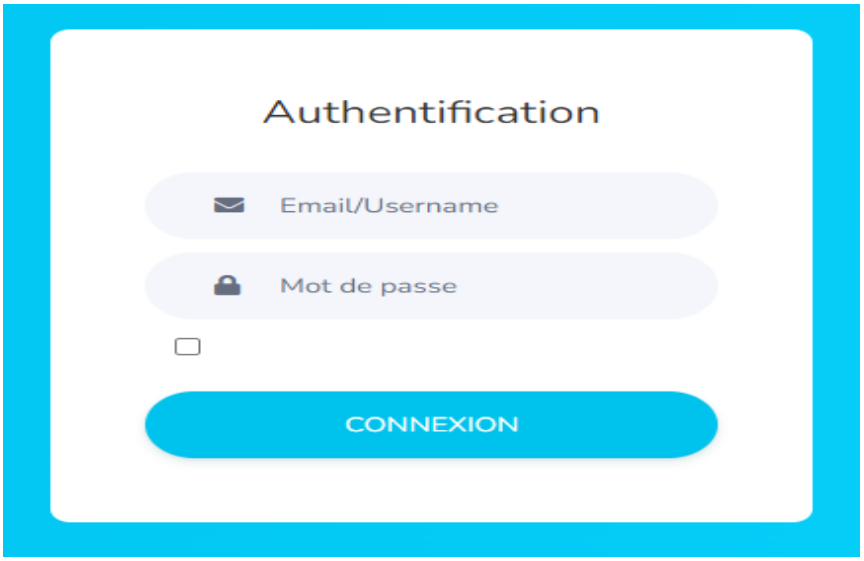


Figure 18:Interface d'authentification.

Arrivée sur cette page, chaque utilisateur doit s'authentifier avec nom d'utilisateur ou Email et le mot de passe. Nous avons deux profils : Enseignant et Etudiant, chaque utilisateur sera redirigé à sa page d'accueil.

VI.8.2 Interfaces d'accueil des différents utilisateurs

Comme nous l'avons vu dans la phase de conception, notre système possède deux acteurs et chacun a son interface après l'authentification. L'enseignant après avoir s'authentifier sera redirigé sur la page d'élaboration d'une évaluation de l'étudiant alors que l'étudiant sera redirigé sur la page de faire l'évaluation. Ces différentes interfaces présentent respectivement l'interface d'accueil de l'enseignant et celle des étudiants.

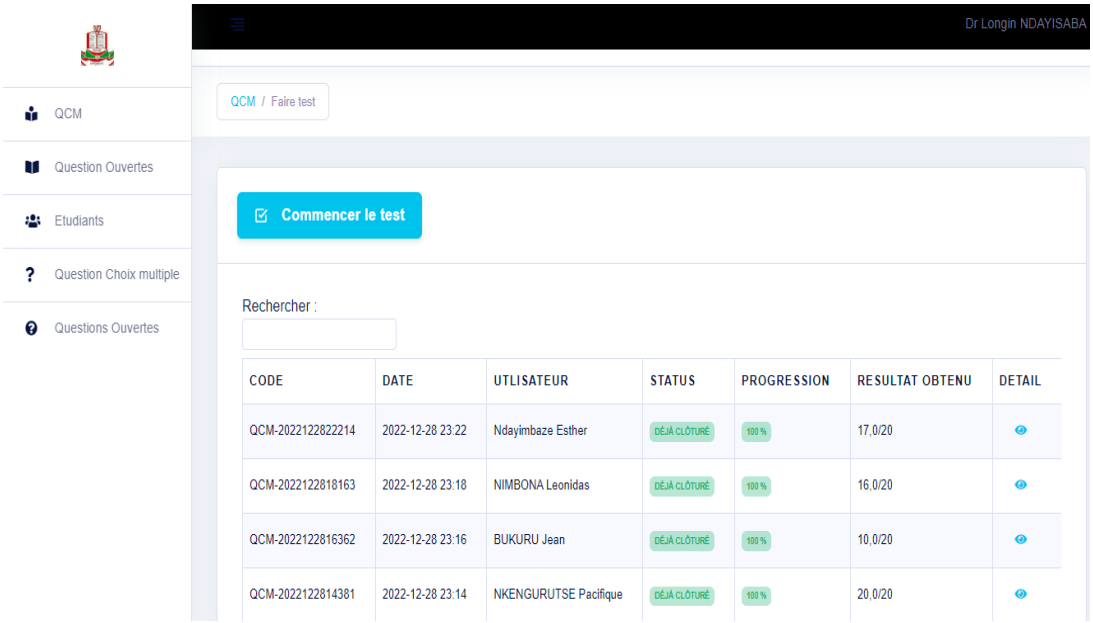


Figure 19:Interface de l'enseignant.

Si l'enseignant veut élaborer les questions ouvertes il passe par cette interface. Il saisit la question dans l'espace Description Question puis la réponse et le temps mis.

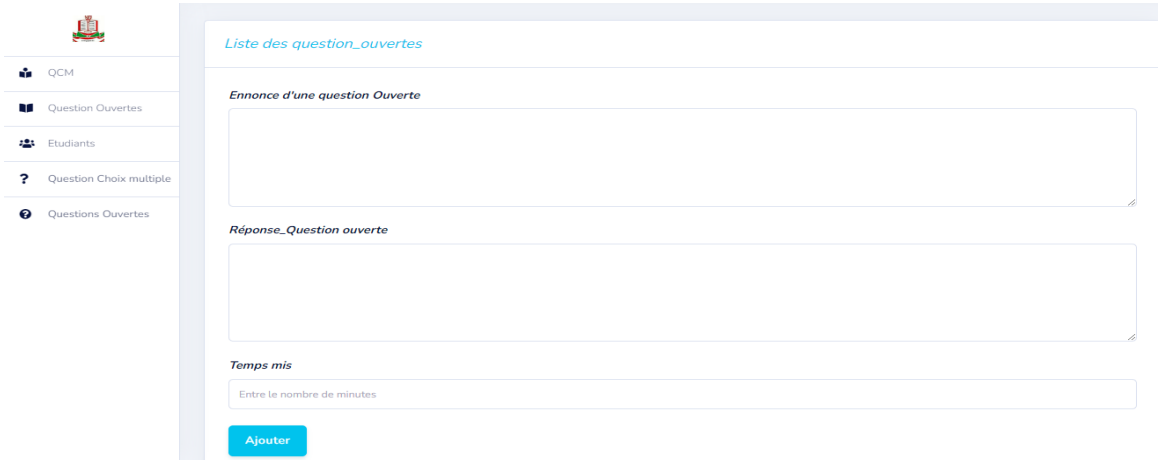


Figure 20: Elaboration des questions ouvertes.

Après avoir saisir énoncée de la question, la réponse et le temps mis, l'enseignant clique sur le bouton Ajouter et sera redirigé sur l'interface pour ajouter les mots clés et la note pour chaque mot clé. Quand l'enseignant a terminé d'élaborer les questions, l'étudiant peut faire le test, voici l'interface d'accueil quand il s'est authentifié correctement :

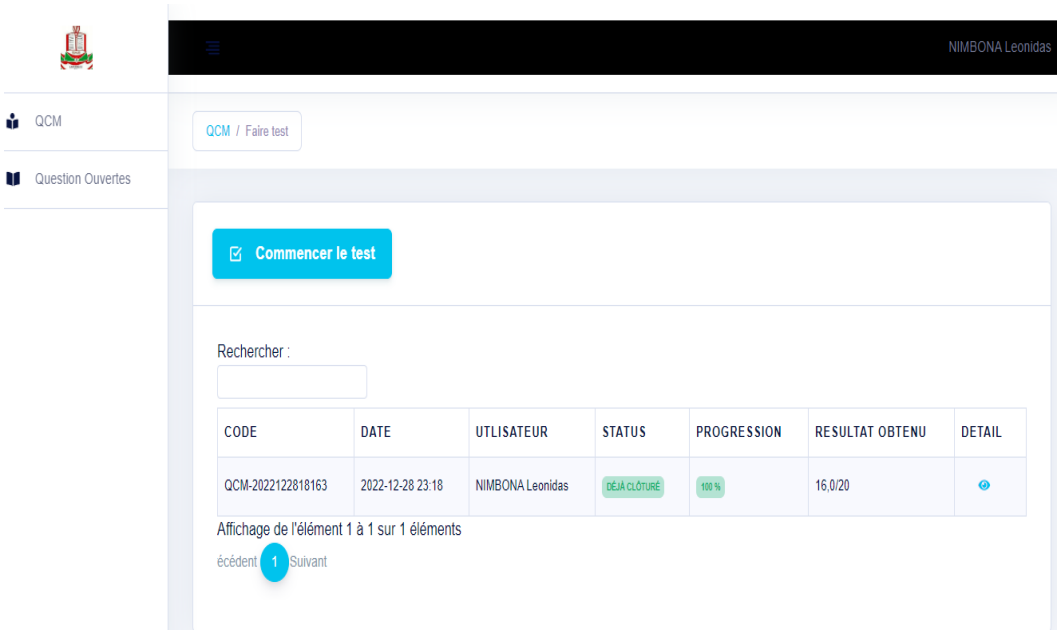


Figure 21:Interface d'accueil de l'étudiant.

L'étudiant peut faire maintenant le test en cliquant sur le bouton commencer le test. Si le temps programmé pour chaque question est dépassé le système passe automatiquement sur la question suivante. L'étudiant fait un choix parmi les quatre puis clique sur le bouton suivant, si le temps est dépassé il passe directement à la question suivante.

Si l'étudiant fait les questions ouvertes, voici l'interface

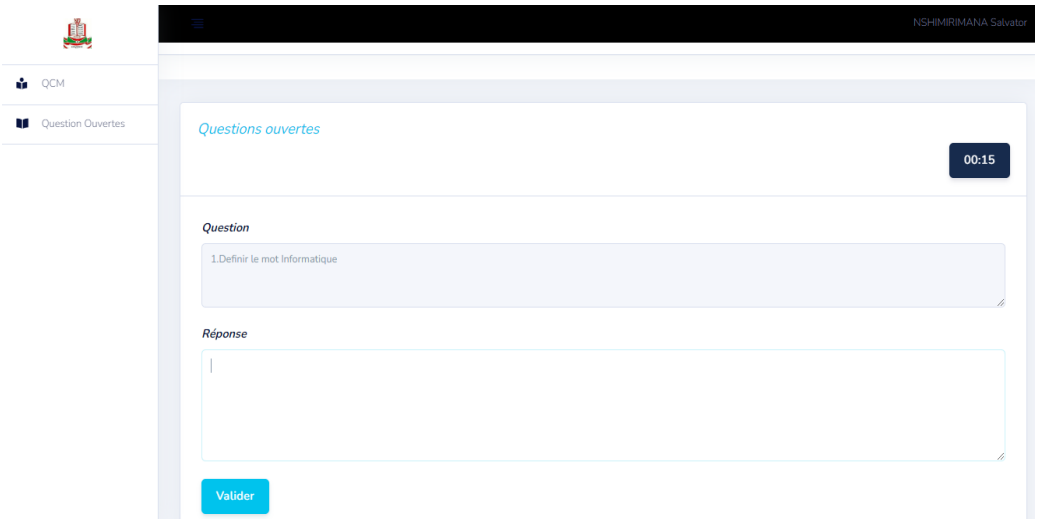


Figure 22:Interface des réponses pour les questions ouvertes

L'étudiant saisit la réponse et le système corrige automatiquement la réponse qu'il a saisie, le système fouille s'il trouve la bonne réponse il accorde automatiquement la note à l'étudiant sinon c'est zéro point. Pour chaque type de question si le temps est dépassé le système avance sur la question suivante puis à la fin le système attribue automatiquement les points à l'étudiant en fonction de bonnes réponses.

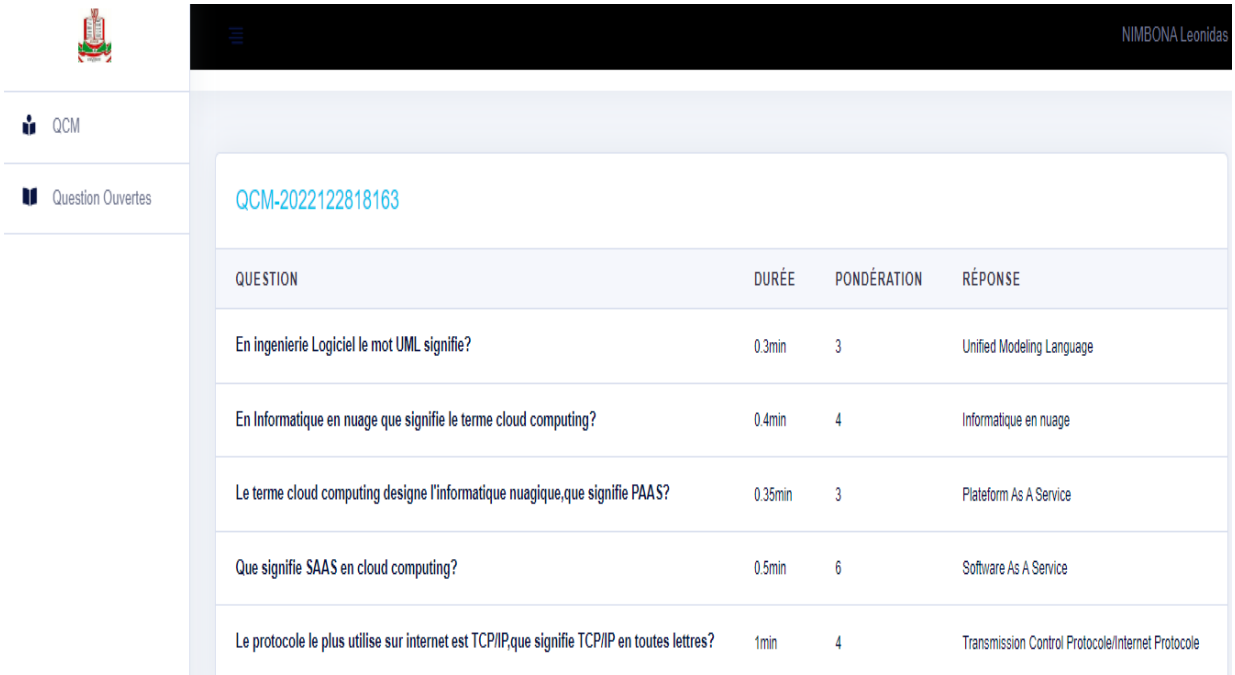


Figure 23:Interface pour la correction et l'attribution des points.

Dans ce chapitre nous avons présenté l'architecture utilisée par notre système, les outils, technologies et langages de développement que nous avons utilisé pour aboutir à l'implémentation de notre système, la sécurisation des données par le chiffrement de Vigenère, l'estimation du cout du projet avec la méthode WebMo, élaboration du calendrier des activités pour réaliser le projet à l'aide du diagramme de Gantt ainsi que quelques interfaces du système. Ce travail permettra de mettre en œuvre un système automatisé et sécurisé de gestion du processus d'évaluation en ligne des étudiants. A la fin de ce travail nous avons apporté une solution aux difficultés rencontrés dans le domaine de l'éducation surtout l'accessibilité de l'éducation au maximum des étudiants en permettant une évaluation à distance. Au regard à notre projet, nous nous sommes posé les questions suivantes : Comment contribuer au progrès de la nation en développant un système permettant d'examiner un nombre illimité des étudiants sans faire de déplacement ? Que faire pour rendre le système flexible étant centré sur le processus d'évaluation en ligne ? Ce travail a l'ambition intéressante d'améliorer mes connaissances et mes compétences professionnelles dans le domaine d'automatisation des tâches en combinaison avec d'autres technologies étudiées en classe.

CONCLUSION GENERALE ET RECOMMENDATIONS

Conclusion générale

Au terme du présent travail nous nous sommes fixés comme objectif d'analyser, concevoir et développer une application web automatisant le processus d'évaluation en ligne des étudiants. Dans la première partie nous avons développé la raison d'être du sujet de notre travail, objectifs du sujet, problématique, solutions proposées et apport scientifique. Dans la deuxième partie nous avons vu les notions d'analyse et conception du système d'information et modèle du cycle de vie d'un logiciel. La troisième partie est dédiée à la conception et modélisation du système avec le langage UML, nous avons modélisé le système avec certains diagrammes UML jugés primordiaux. Quant à la quatrième partie nous nous sommes prolongés dans la modélisation mathématique en utilisant la théorie des files d'attentes afin d'énumérer les paramètres de fiabilités pour assurer le bon fonctionnement du système. Dans la cinquième partie nous avons énuméré les technologies et outils utilisés pour mettre en place ce système. Enfin, tout échange transitant sur le réseau a besoin d'un niveau fiable de sécurité ce qui nous a poussé à développer et implémenter la nation de sécurité dans la sixième partie avec le chiffrement de Vigenère. L'analyse, conception et développement d'un système automatisé et sécurisé d'évaluation en ligne des étudiants est un travail visant à donner solutions aux problèmes dues au système manuel comme difficulté d'évaluer des classes détenant un nombre élevé des étudiants ce qui aura comme impact négatif retard de la publication des résultants, les cas de tricherie qui s'observent, il y a aussi des cas où certains documents sont égarés et manquent de sécurité. La mise en place de ce travail a permis l'évaluation d'un nombre illimité des étudiants en un laps de temps, réduction de la fatigue pour les enseignants qui évaluent des classes saturées, réduction du temps d'attente entre la passation des évaluations et la publication des résultats, stockage des données dans un endroit sécurisé accessible n'importe où à l'aide d'un dispositif électronique (smartphone et ordinateurs)**.** Nous sommes persuadés que ce travail est une solution triviale pour tous problèmes liés au système d'évaluation manuel c'est pourquoi nous allons recommander à l'université du Burundi d'adopter ce système afin qu'il soit la lumière des autres universités dans l'utilisation des systèmes automatiques.

RECOMMANDATIONS

En définitive, nous suggérons à l'Université du Burundi, à l'Etat Burundais, aux futurs étudiants chercheurs et à nos lecteurs les recommandations suivantes : A l'Université du Burundi : Adopter le nouveau système mis en place pour faciliter le processus d'évaluation en ligne, Compiler les projets de recherche réalisés par les différents étudiants s'inscrivant dans le même domaine afin de construire des gros systèmes répondant aux besoins des enseignants et étudiants. A l'Etat Burundais : Financer ces travaux de recherche menés par les étudiants de master afin de consolider les recherches dans notre pays, Encourager les enseignants qui suivent les étudiants pendant le processus de rédaction de mémoire car ils effectuent un travail fatiguant sans aucune récompense. Aux étudiants chercheurs : Apporter des améliorations au présent système comme l'ajout d'autres modules d'évaluations comme l'évaluation par voie orale, Intégrer le module de surveillance à distance face à face à l'aide de l'ordinateur et le module d'évaluation en ligne à l'aide des codes USSD afin de faciliter les étudiants ne possédant pas les smartphones ou ordinateur, Etendre le système pour qu'il soit utilisé concomitamment par l'Université du Burundi et autres universités privées. A nos lecteurs : Nous rappelons que ce travail n'est pas un modèle unique et parfait. C'est pourquoi, nous restons ouverts à toutes critiques et nous sommes prêts à recevoir toutes les suggestions, critiques et remarques tendant à améliorer davantage ce travail, chacun a la responsabilité de faire un critique et non pas un jugement dans le but d'améliorer ce travail.

BIBLIOGRAPHIE

Ouvrages généraux

[1] Olivier Gaudoin, *Fiabilité des Systèmes et des Logiciels*, Paris, 2001 ;

[2] Julie Vachon, *Introduction au génie logiciel*, Hiver 2003 ;

[3] I. Jacobson, G. Booch et J. Rumbaugh, *Le processus unifié de développement logiciel*, Paris, 2000 ;

[4] B. Baynat. *Théorie des files d'attente, des chaînes de Markov aux réseaux à forme produit*. Hermes Science Publications, Paris, 2000 ;

[5] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing networks and Markov chains. Modelling and Performance Evaluation with Computer Science Applications*. JOHN WILEY and SONS, Canada, 2006;

[6] Leonard Kleinrock. *Queueing Systems. Wiley-Interscience*, New York, 1975;

WEBOGRAPHIE

[1] http://revuesim.org/sim. Consulté le 25/07/2022 à 07h12 ;

[2] https://ingenierie-creations.fr/les-fonctions-du-systeme-dinformation/ .Consulté le 27/072022 à 09h17min ;

[3]https://www.semanticscholar.org/paper/M%C3%A9thodes-d'analyse-et-de-conception-orient%C3%A9es-objet-Bessai/d2090a54b63f995925d1a1848b3dec5a4c658e74. Consulté le 29/07/2022 à 20h 25min ;

[4]https://www.softfluent.fr/blog/developpement-logiciel-quand-quoi-comment/#:~:text=Le%20d%C3%A9veloppement%20logiciel%20d%C3%A9signe%20l,la%20maintenance%20de%20l'application .Consulté le 30/07/2022 à 06h10min ;

[5] https://creately.com/blog/fr/uncategorized-fr/tutoriel-sur-le-diagramme-de-deploiement/ .Consulté le 31/07/2022 à 19h15min ;

[6] https://www.ibm.com/docs/fr/rsar/9.5?topic=diagrams-activity consulte le 01/08/2022 à 07h 7min. Consulté le 01/08/2022 à 09h15min ;

[7] https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours05.html .Consulté le 01/08/2022 à 11h15min ;

[8] https://www.apachefriends.org/download.html.Consulté le 01/08/2022 à 14h25min ;

[9] https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript .Consulté le 02/08/2022 à 15h55min ;

[10] Flowchart Maker and Diagramming Software | Microsoft Visio .Consulté le 02/08/2022 à 21h15min

[11] https://www.ibm.com/fr-fr/topics/encryption. Consulté le 03/08/2022 à 19h15min ;

[12] An Interactive Cryptanalysis Algorithm for the Vigenere Cipher | SpringerLink.Consulté le 05/08/2022 à 11h45min ;

[13] https://archipel.uqam.ca/3264/1/M9720.pdf. Consulté le 07/10/2022 à 10h35min ;

[14] https://www.geeksforgeeks.org/software-engineering-cocomo-ii-model/ .Consulté le 15/10/2022 à 09h25min ;

[15] https://www.wrike.com/fr/project-management-guide/faq/quest-ce-que-le-calendrier-en-gestion-de-projet/. Consulté le 25/10/2022 à 09h25min ;