****



**REPUBLIQUE DU NIGER**

Fraternité – Travail – Progrès

Ministère de l’Enseignement Supérieur de la Recherche et de l’Innovation

Direction Générale de l’Enseignement

**Direction de l’Enseignement Privé**

***Mémoire de fin de cycle présenté en vue de L’Obtention du Diplôme de Master Professionnel***

***Option : Génie Logiciel***

**Thème : Etude et Déploiement d’une Plateforme de gestion des sessions des conseils de la Faculté des Sciences et Techniques (FAST) de l’Université Abdou Moumouni (UAM)**

**Présenté par: Encadré par :**

**Mlle. HALIDOU ALI YACOUBOU M. BONCANO ABDOU**

**(Ingénieur en informatique – UAM TICE)**

**Année académique** : **2024-2025**

Avant-propos

Dans un monde en constante évolution, marqué par une digitalisation croissante des processus, le secteur de l’enseignement supérieur n’échappe pas à cette dynamique. Les institutions universitaires sont appelées à repenser leurs modes de gestion, en particulier ceux liés à la gouvernance académique. C’est dans ce contexte que s’inscrit le présent mémoire, fruit d’un travail réalisé dans le cadre du parcours de Master en Génie Logiciel.

Ce mémoire retrace une expérience à la croisée de l’ingénierie logicielle et de l’organisation universitaire, à travers l’étude et le déploiement d’une plateforme de gestion des sessions des conseils au sein de la Faculté des Sciences et Techniques (FAST) de l’Université Abdou Moumouni (UAM). Il s’agit d’une immersion concrète dans les réalités d’un établissement d’enseignement supérieur confronté à des défis liés à la dématérialisation des procédures, à la conservation des données sensibles, et à la fluidification des échanges décisionnels.

Le travail présenté ici reflète non seulement une application pratique des compétences acquises durant le cursus académique, mais également une volonté de contribuer de manière concrète à l’amélioration des pratiques de gestion à la FAST. Guidé par le souci de rigueur, d’innovation et d’adaptation aux besoins spécifiques du terrain, ce projet a été mené dans un esprit de collaboration, sous la supervision d’un encadrement qualifié et l’accompagnement d’acteurs engagés dans la transformation numérique de l’université.

Dédicaces

Je dédie ce mémoire :

À mes parents, pour leur amour inconditionnel, leur soutien indéfectible et les sacrifices consentis tout au long de mon parcours.

À ma famille, mes frères et sœurs, qui m’ont toujours encouragé et soutenu avec fierté.

À mes amis proches et camarades de promotion, pour leur bienveillance, leur patience et leurs encouragements constants.

À tous les enseignants de l’EPI-Niger, pour leur encadrement et la transmission du savoir avec rigueur et passion.

A mon frère BELLO ISSA Idrissa auprès de qui j’ai eu le microsoft office pour rédiger mon mémoire.

Que ce travail soit le témoignage de ma gratitude profonde envers toutes ces personnes qui ont marqué mon chemin.

Remerciements

La réalisation de ce mémoire n’aurait pas été possible sans l’appui, les conseils et le soutien de nombreuses personnes que je tiens à remercier sincèrement.

Je remercie tout d’abord Monsieur Boncano Abdou, mon encadrant, pour sa disponibilité, sa patience et ses orientations pertinentes tout au long de ce travail. Son accompagnement a été précieux à chaque étape de la conception et du développement de la plateforme.

Mes sincères remerciements vont également à l’ensemble du personnel de la Faculté des Sciences et Techniques (FAST), pour leur accueil chaleureux, leur collaboration et leur contribution active à la réussite de cette étude.

Je tiens à remercier les enseignants du département de Génie Logiciel, pour la qualité de l’enseignement reçu tout au long de ma formation.

Je n’oublie pas mes camarades de promotion et mes amis, avec qui j’ai partagé les moments forts de ce parcours académique.

Enfin, mes pensées les plus respectueuses vont à mafamille, pour son soutien moral, financier et affectif, sans lequel ce travail n’aurait pas pu aboutir.

Glossaire des abréviations

Liste des figures

Liste des tableaux

Sommaire

Introduction Générale

Dans un contexte mondial marqué par une transformation numérique accélérée, les établissements d’enseignement supérieur se trouvent face à la nécessité de repenser leurs modes de gouvernance et de gestion interne. L’amélioration des processus décisionnels, la traçabilité des informations et la fiabilité des données sont désormais des impératifs stratégiques pour garantir la qualité du pilotage académique et administratif.  
Au Niger, comme dans de nombreux pays, les facultés et institutions universitaires fonctionnent encore en grande partie selon des méthodes de gestion traditionnelles, caractérisées par la circulation physique de documents, la centralisation manuelle des informations et des difficultés de suivi des décisions prises en conseil. Ces pratiques, bien qu’historiquement éprouvées, montrent aujourd’hui leurs limites : lenteurs administratives, risques de pertes ou de duplications d’informations, difficultés à produire des statistiques fiables et absence d’outils de communication en temps réel avec les acteurs concernés.

C’est dans ce contexte qu’intervient le présent mémoire, qui s’inscrit dans le cadre de la formation de Master Professionnel en Génie Logiciel. Il porte sur l’étude et le déploiement d’une plateforme de gestion des sessions des conseils de la Faculté des Sciences et Techniques (FAST) de l’Université Abdou Moumouni (UAM). L’objectif principal est de proposer une solution technologique innovante permettant d’automatiser la planification, la convocation, la gestion et le suivi des sessions de conseil de faculté. Cette plateforme devra également assurer la conservation sécurisée des données, l’accessibilité des documents de travail, la gestion des comptes rendus et la production de statistiques fiables sur la participation et le fonctionnement du conseil.

La réalisation de ce projet répond à plusieurs enjeux majeurs :

* **Modernisation des processus** : réduire la charge administrative et améliorer l’efficacité de la gestion des conseils.
* **Dématérialisation et traçabilité** : garantir une meilleure conservation des informations et une traçabilité complète des décisions.
* **Interopérabilité et évolutivité** : concevoir une solution adaptable aux besoins futurs de la faculté et éventuellement à d’autres structures de l’UAM.
* **Renforcement de la gouvernance académique** : mettre à disposition des responsables des outils d’aide à la décision fondés sur des données fiables.

Pour atteindre ces objectifs, une méthodologie rigoureuse a été adoptée, comprenant l’analyse de l’existant, l’étude de l’état de l’art, la modélisation du système à l’aide de méthodes de conception appropriées, le choix d’outils et de technologies adaptés, puis le développement et le déploiement effectif de la plateforme.

Le présent mémoire est structuré en trois chapitres principaux :

* **Le premier chapitre** présente le cadre général de l’étude, à savoir l’environnement institutionnel, organisationnel et technique de la FAST et de l’UAM, ainsi qu’une analyse des limites des méthodes actuelles de gestion des sessions de conseil.
* **Le deuxième chapitre** est consacré à l’état de l’art. Il expose les concepts théoriques, les standards techniques et les outils logiciels utilisés dans le domaine de la conception et du déploiement de solutions de gestion informatisée.
* **Le troisième chapitre** détaille la conception, la réalisation et le déploiement de la plateforme, depuis la modélisation des données jusqu’à l’implémentation pratique et l’évaluation des résultats obtenus.

À travers ce travail, il s’agit donc d’apporter une contribution concrète à la modernisation des processus internes de la FAST, tout en démontrant les compétences techniques et méthodologiques acquises au cours du cursus de Master en Génie Logiciel.

Chapitre I. Présentation du cadre d’études

Introduction

La Faculté des Sciences et Techniques (FAST) de l’Université Abdou Moumouni (UAM) constitue un pôle majeur d’enseignement et de recherche scientifique au Niger. Comme toute institution académique, elle est dotée d’organes de gouvernance, dont le **Conseil de Faculté**, qui joue un rôle central dans la prise de décisions pédagogiques, administratives et stratégiques. Les sessions du conseil, organisées périodiquement, réunissent différents acteurs: enseignants, responsables administratifs et représentants des étudiants. Elles permettent de débattre et de statuer sur des questions essentielles, telles que l’adoption des programmes, la validation des résultats académiques, ou encore l’orientation des activités de recherche.

I.1. Présentation de l'UAM de Niamey

I.1.1. Présentation générale

L’Université Abdou Moumouni a été créée par la Loi n°71-31 du 6 septembre 1971 sous l’appellation de Centre d’Enseignement Supérieur (CES) de Niamey. Le CES était placé sous la tutelle du Président de la République ou du Ministre désigné par lui, en l’occurrence le Ministre de l’Éducation Nationale, de la Jeunesse et des Sports et était administré par un Conseil d’administration et dirigé par un Directeur. Il comprenait un Institut Pédagogique et une École Supérieure des Sciences.

* Par la Loi N°73-23 du 20 septembre 1973, le CES sera transformé en Université dite Université de Niamey administrée par un Conseil de l’Université et dirigé par un Recteur. L’arrêté n°044/MEN/JS/CAB du 3 octobre 1973 remplaçait l’Institut Pédagogique par un Institut Universitaire de Formation Pédagogique (IUFP).
* Le décret 74-108/PCMS/MEN/JS du 28 mai 1974, créa des écoles et instituts. Il s’agit de l’École des Lettres, l’École Supérieure d’Agronomie, l’École de Pédagogie, l’École des Sciences, l’Institut de Recherche sur l’Enseignement des Mathématiques et l’Institut de Recherche en Sciences Humaines. Le décret n°74-262/PCMS/MEN/JS du 1er octobre 1974 créait l’École des Sciences de la Santé.
* L’ordonnance n°84-3 du 12 janvier 1984 et son décret d’application n°84- 8/PCMS/MES/R du 12 janvier 1984 redonnèrent une nouvelle physionomie à l’Université de Niamey. Ces écoles sont devenues des facultés administrées par des Doyens et des Conseils de Faculté sous l’autorité du Recteur.

L’Université de Niamey deviendra en août 1992 Université Abdou Moumouni de Niamey en hommage à un de ses anciens Recteurs, le Professeur Abdou Moumouni Dioffo.

I.1.2. Missions

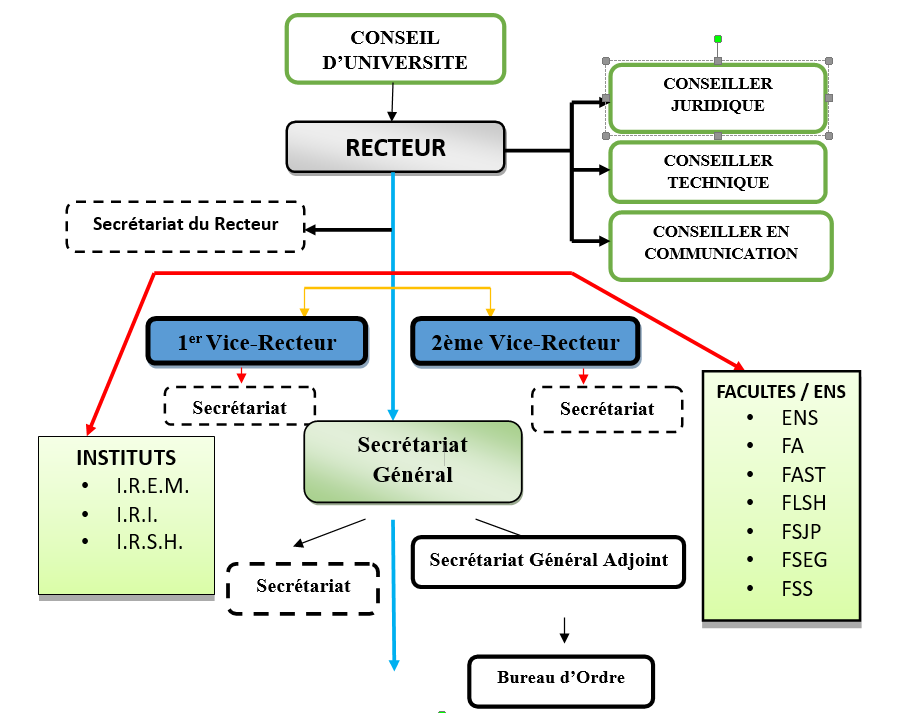
L’UAM a pour mission essentielle de promouvoir les formations initiales et continue, ainsi que la recherche scientifique fondamentale et appliquée. Il est attendu en terme de bilan, la diffusion des résultats de cette institution, notamment dans les domaines en rapport avec les besoins du pays afin de contribuer à la diffusion de la culture et de l’information scientifique et technique et de former une identité culturelle et une conscience nationale et africaine.

I.1.3. Objectifs

I.1.4. Structure organisationnelle

I.1.5. Organigramme

La figure suivante présente l’organigramme actuel de l’UAM :



**SERVICES CENTRAUX**

L'Agence Comptable - Le Service Central des Personnels - Le Service de l'Intérieur et des Affaires Générales - Le Service Central de la Solde - Le Service Central des Affaires Financières - Le Service Central de la Scolarité - Le Service des Équivalences des Diplômes - Le Service Central des Relations Extérieures - La Bibliothèque Universitaire Centrale - Le Service des Examens du Baccalauréat - Le Service Central des Annales –

Le Service Central de Développement - Le Service Central de l’Informatique et des Nouvelles Technologies de l’Information et de la Communication.

**SERVICES CENTRAUX**

L'Agence Comptable - Le Service Central des Personnels - Le Service de l'Intérieur et des Affaires Générales - Le Service Central de la Solde - Le Service Central des Affaires Financières - Le Service Central de la Scolarité - Le Service des Équivalences des Diplômes - Le Service Central des Relations Extérieures - La Bibliothèque Universitaire Centrale - Le Service des Examens du Baccalauréat - Le Service Central des Annales –

Le Service Central de Développement - Le Service Central de l’Informatique et des Nouvelles Technologies de l’Information et de la Communication.

**SERVICES CENTRAUX**

L'Agence Comptable - Le Service Central des Personnels - Le Service de l'Intérieur et des Affaires Générales - Le Service Central de la Solde - Le Service Central des Affaires Financières - Le Service Central de la Scolarité - Le Service des Équivalences des Diplômes - Le Service Central des Relations Extérieures - La Bibliothèque Universitaire Centrale - Le Service des Examens du Baccalauréat - Le Service Central des Annales –

Le Service Central de Développement - Le Service Central de l’Informatique et des Nouvelles Technologies de l’Information et de la Communication.

I.2. Présentation de la FAST

I.2.1. Présentation générale

I.2.2. Missions et attributions

I.2.3. Organigramme

I.3. Etude de l’existant

I.3.1. Recensement des équipements informatiques

I.3.2. Architecture du réseau

La précédente architecture comporte de nombreux équipements. Les équipements Switch du constructeur Cisco y sont intensivement utilisés. Cette architecture laisse entrevoir l’absence de matériels pare-feu physiques et d’une DMZ, ce qui fait expose les usagers au risque d’attaque informatique. De nombreuses autres caractéristiques de ce réseau motivent la nécessite de la conduite à son terme, du processus de résolution de notre problématique.

La figure suivante représente une vision globale de l’architecture du réseau de l’UAM.



I.3.3. Procédé de gestion actuelle du processus d’octroi des crédits particuliers

Le doyen de la Faculté des Sciences et Technique (FAST) organise régulièrement et si la situation l’exige, les sessions des conseils de cette faculté dans la salle habituelle réservée à cet effet. Pour cela, il désire mettre en place un outil automatisé de gestion de ces sessions des conseils pour une meilleure efficacité dans sa gestion, un suivi des décisions prises, une meilleure conservation des comptes-rendus et une production des statistiques fiables. Pour cela, la note de convocation d’une session indique la date, l’heure de début et est accompagnée des dossiers à étudier. Cette note et les documents qui l’accompagnent est transmise à tous les membres permanents de ce conseil en plus des invités s’il y en a. Le nouveau système envisagé doit envoyer automatiquement un message à cet effet à tous ces membres. Le jour de la réunion une liste de présence électronique est ouverte, un comptage des présents est réalisé, et si le quorum est atteint, alors les débats sont engagés autour des points inscrits à l’ordre du jour. A la fin de la réunion, un compte rendu est rédigé sous forme de fichier PDF et mis à la disposition de tous les participants à la session du conseil de fac dans l’application. Globalement, le processus à automatiser consiste à implémenter une solution pour remplacer la procédure manuelle utilisée actuellement permettant de :

* Gérer les membres du conseil de fac avec confirmation de participation ou non à une session
* Créer et planifier les sessions du conseil de la faculté
* Gérer les documents à étudier sous forme de fichier joint à la convocation
* Gérer les ordres de jour
* Gérer les notifications et rappels
* Gérer les comptes rendus
* Fournir l’historique des sessions passées du conseil de la faculté
* Suivre les présences aux sessions du conseil de la faculté
* Fournir des statistiques sur les sessions du conseil de la faculté : nombre de réunions avec un taux de participation

I.3.4. Critiques et limites de la gestion actuelle

L’organisation, la convocation et la gestion globale des sessions du conseil de faculté sont manuelles avec les inconvénients suivants :

* Lenteur, risque de perte des documents et des informations, difficultés de fournir des statistiques

I.4. Problématique traitée

I.4.1. Enoncé de la problématique

I.4.2. Axes de résolution du problème

I.5. Synthèse

Chapitre II. Etat de l’art

Introduction

Dans la réalisation d’une plateforme de gestion des sessions des conseils, il est essentiel de s’appuyer sur des concepts, des modèles et des outils éprouvés dans le domaine de l’ingénierie logicielle et des technologies de l’information. Avant de passer à la conception et au déploiement, une revue des connaissances disponibles permet de poser les bases théoriques et technologiques indispensables pour garantir la qualité, la robustesse et l’évolutivité de la solution à développer.

Ce chapitre a pour objectif de présenter les fondements techniques sur lesquels s’appuie le projet. Il s’agit notamment des normes et standards réseaux, des cycles de développement logiciel, des méthodes de conception, des Framework de développement, des systèmes de gestion de bases de données (SGBD), des langages de programmation et des environnements serveurs couramment utilisés dans la mise en place d’applications web professionnelles.

La présentation de cet état de l’art se décline en plusieurs sections :

* Les principes de normalisation et les prérequis pour la mise en place d’un réseau informatique.
* Les modèles et cycles de développement logiciel adaptés aux projets web.
* Les méthodes de conception telles que MERISE et UML.
* Les Framework les plus utilisés pour le développement d’applications web (React, Next Js, TypeScript, Angular, Laravel, Symfony, etc.).
* Les systèmes de gestion de bases de données et les critères de choix.
* Les langages de programmation adaptés au contexte.
* Les environnements serveurs et leur configuration.

L’objectif est de fournir une base solide et documentée pour les choix technologiques et méthodologiques qui seront détaillés et justifiés dans le chapitre suivant consacré à la conception et à la réalisation de la plateforme.

II.1. Normalisation et pré-requis de mise en place d’un réseau

Dans toute architecture informatique moderne, la mise en place d’un réseau fiable et performant repose sur des normes internationales et des principes techniques clairement définis. Ces normes garantissent l’interopérabilité entre les équipements, la sécurité des communications et la pérennité des infrastructures. Avant de concevoir ou de déployer une application de gestion telle que celle des sessions du conseil de la FAST, il est nécessaire de s’assurer que l’environnement réseau est bien dimensionné, normalisé et conforme aux standards en vigueur.

II.1.1. Le modèle ISO/OSI

Le modèle OSI (Open Systems Interconnection) est un modèle de référence en ce qui concerne les réseaux, il est proposé par l'ISO (International Standards Organisation), il décrit les concepts et les démarches à suivre pour interconnecter des systèmes. C’est un cadre conceptuel qui divise les fonctions de communication réseau en sept couches. L'envoi de données via un réseau est complexe, car les différentes technologies matérielles et logicielles doivent fonctionner de manière cohérente au-delà des frontières géographiques et politiques. Le modèle de données OSI fournit un langage universel pour la mise en réseau informatique, de sorte que diverses technologies peuvent communiquer à l'aide de protocoles ou de règles de communication standard. Chaque technologie d'une couche spécifique doit fournir certaines capacités et exécuter des fonctions spécifiques pour être utile dans la mise en réseau. Il est composé de 7 couches :

* La couche physique

La couche physique fait référence au support de communication physique et aux technologies permettant de transmettre les données sur ce support. La communication de données renvoie essentiellement à la transmission de signaux numériques et électroniques via divers canaux physiques tels que les câbles à fibres optiques, les câbles en cuivre et l'air. La couche physique inclut des normes pour les technologies et les métriques étroitement liées aux canaux, tels que le Bluetooth, la NFC et les vitesses de transmission de données.

* La couche de liaison de données

La couche de liaison de données fait référence aux technologies utilisées pour connecter deux machines sur un réseau où la couche physique existe déjà. Elle gère les trames de données, qui sont des signaux numériques encapsulés dans des paquets de données. Le contrôle du flux et le contrôle des erreurs des données sont souvent au cœur de la couche de liaison de données. L'Ethernet est un exemple de norme en la matière. La couche de liaison de données est souvent divisée en deux sous-couches : Media Access Control (MAC) et Logical Link Control (LLC).

* La couche réseau

La couche réseau renvoie à des concepts tels que le routage, le transfert et l'adressage sur un réseau dispersé ou plusieurs réseaux connectés de nœuds ou de machines. La couche réseau peut également gérer le contrôle du flux. Sur Internet, les protocoles Internet v4 (IPv4) et IPv6 constituent les principaux protocoles de couche réseau.

* La couche transport

L'objectif principal de la couche de transport est de s'assurer que les paquets de données arrivent dans le bon ordre, sans pertes ni erreurs, ou qu'ils peuvent être récupérés en toute harmonie si nécessaire. Le contrôle du flux, autant que le contrôle des erreurs, est souvent au cœur de la couche de transport. À cette couche, les protocoles couramment utilisés associent le protocole TCP (Transmission Control Protocol), un protocole basé sur une connexion quasiment sans perte, et le protocole UDP (User Datagram Protocol), un protocole sans perte de connexion. Le protocole TCP est couramment utilisé lorsque toutes les données doivent être intactes (par exemple le partage de fichiers), tandis que le protocole UDP est utilisé lorsque la conservation de tous les paquets est moins critique (par exemple le streaming vidéo).

* La couche de session

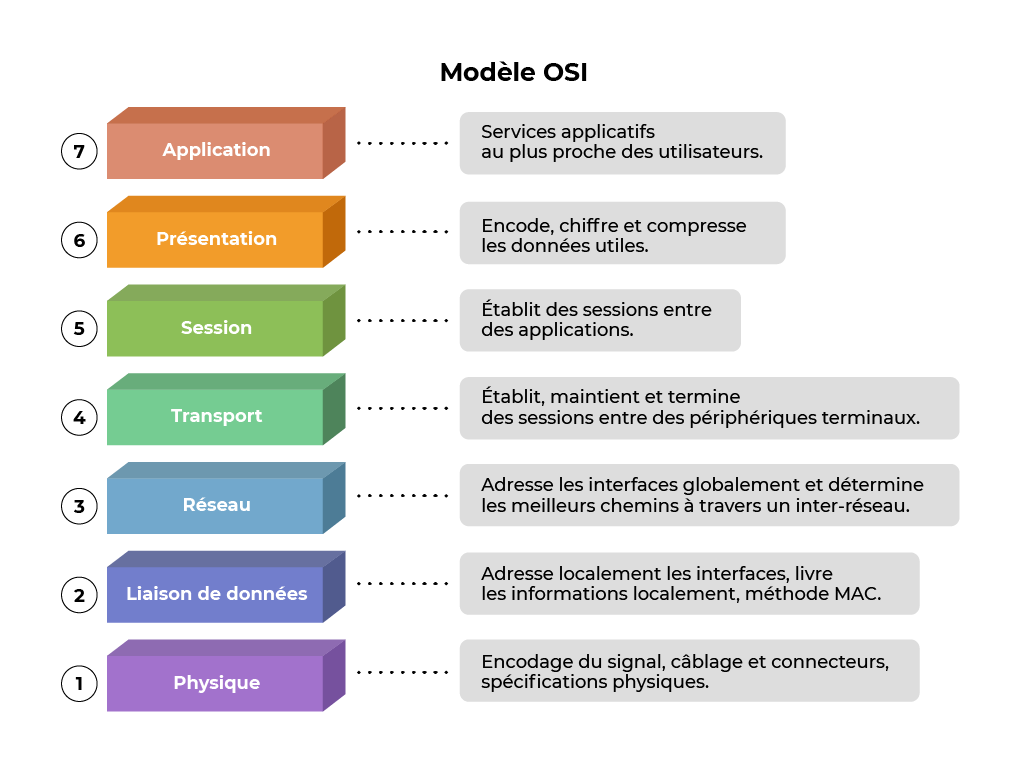
La couche de session est responsable de la coordination réseau entre deux applications distinctes au cours d'une session. Une session gère le début et la fin d'une connexion individuelle à une application et les conflits de synchronisation.

* La couche de présentation

La couche de présentation concerne principalement la syntaxe des données elles-mêmes que les applications peuvent envoyer et utiliser. Par exemple, HTML (Hypertext Markup Language), JSON (JavaScipt Object Notation) et CSV (valeurs séparées par des virgules) sont tous des langages de modélisation qui décrivent la structure des données au niveau de la couche de présentation.

* La couche d’application

La couche d'application renvoie au type spécifique d'application lui-même et à ses méthodes de communication normalisées. Par exemple, les navigateurs peuvent communiquer à l'aide du protocole HTTPS (HyperText Transfer Protocol Secure), et les clients HTTP et de messagerie via POP3 (Post Office Protocol version 3) et SMTP (Simple Mail Transfer Protocol).



II.1.2. Le modèle TCP/IP

On peut très bien naviguer sur des sites web et utiliser des applications sans rien savoir du fonctionnement d’Internet. Mais en coulisses, une grande partie de tout cela repose sur des ordinateurs, des serveurs et des modems qui communiquent entre eux. Le protocole TCP/IP est une norme de communication qui contribue au fonctionnement d’Internet. C’est un protocole de liaison de données utilisé sur Internet pour permettre aux ordinateurs et autres appareils d’envoyer et de recevoir des données. L’acronyme TCP/IP signifie Transmission Control Protocol/Internet Protocol. Il permet aux appareils connectés à Internet de communiquer entre eux via les réseaux.

C’est un modèle qui se compose de :

* Le protocole TCP : (Transmission Control Protocol) est une norme de communication qui permet aux programmes applicatifs et aux dispositifs informatiques d’échanger des messages sur un réseau. Il est conçu pour envoyer des paquets sur Internet et assurer la transmission réussie des données et des messages sur les réseaux.
* Le protocole IP : qui est la méthode utilisée pour envoyer des données d’un appareil à un autre via Internet. Chaque appareil dispose d’une adresse IP qui l’identifie de manière unique et lui permet de communiquer et d’échanger des données avec d’autres appareils connectés à Internet.  Aujourd’hui, elle est considérée comme la norme pour une communication rapide et sécurisée directement entre les appareils mobiles.

Le modèle TCP/IP comporte **quatre couches : accès réseau, Internet, transport et application**. Ensemble, ces couches composent une suite de protocoles. Le modèle TCP/IP fait transiter les données par ces couches dans un ordre bien défini lorsque l’utilisateur envoie une information, puis dans l’ordre inverse lorsque des données arrivent.

* Couche 1 : la couche d’accès à réseau

La couche d’accès réseau, aussi appelée couche de liaison de données, **gère l’infrastructure physique** qui permet aux ordinateurs de communiquer entre eux via Internet. Cette infrastructure comprend les câbles Ethernet, les réseaux sans fil, les cartes réseau, les pilotes de périphérique de votre ordinateur, etc. Elle **comprend aussi l’infrastructure technique**, par exemple le code qui convertit les données numériques en signaux transmissibles, qui rend possible la connexion réseau.

* Couche 2 : la couche internet

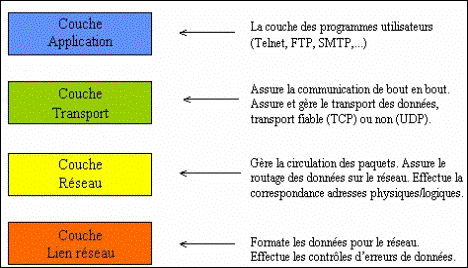
La couche Internet, aussi appelée couche réseau, **contrôle le flux ou le routage des paquets sur le réseau** afin que les données soient envoyées rapidement et à la bonne destination. Cette couche est également chargée de réassembler le paquet de données côté destination. Si le trafic Internet est important, la couche Internet peut avoir besoin d’un peu plus de temps pour envoyer un fichier, mais les risques qu’une erreur vienne endommager ce fichier sont réduits.

* Couche 3 : la couche transport

La couche transport **fournit une connexion des données fiable** entre deux appareils qui communiquent entre eux. C’est comme si vous envoyiez un colis avec une assurance. La couche de transport décompose les données en paquets, accuse réception des paquets qu’elle a reçus de l’expéditeur et s’assure que le destinataire accuse réception des paquets qu’il reçoit.

* Couche 4 : la couche application

La couche application est le groupe d’applications qui **permettent à l’utilisateur d’accéder au réseau**. Pour la plupart d’entre nous, il s’agit des e-mails, des applications de messagerie et des programmes de stockage dans le cloud. C’est ce que l’utilisateur final voit et utilise lorsqu’il envoie et reçoit des données.



II.1.3. Le matériel de mise en place du LAN Ethernet

La conception d’un **réseau local (LAN – Local Area Network)** performant constitue un prérequis indispensable pour le bon fonctionnement d’une application de gestion interne telle que celle des sessions de conseils. Un LAN fiable permet la circulation fluide des informations, la disponibilité constante des services et la sécurité des échanges. Sa mise en place nécessite un ensemble cohérent d’équipements matériels, chacun ayant un rôle bien déterminé dans l’architecture globale. On peut distinguer dont entre autres :

* Les câbles muraux

Les câbles sont les artères du réseau. Dans un environnement académique comme celui de la FAST, les câbles utilisés sont généralement de types pairs torsadés (catégories 5 e, 6 ou 6a), permettant des débits allant de 100 Mbps à 10 Gbps. Ils assurent la transmission physique des données entre les postes de travail, les switch et les équipements centraux.

* Les prises murales RJ45

Installées dans les bureaux ou salles de réunion, elles constituent le **point d’accès physique** entre les équipements terminaux (ordinateurs, imprimantes réseau) et l’infrastructure câblée. Chaque prise est reliée par un câble à l’armoire de brassage.

* Le panneau de brassage

C’est un module situé dans l’armoire de brassage qui permet de regrouper toutes les connexions des câbles provenant des prises murales. Il facilite l’organisation et la maintenance du réseau, en permettant des interventions rapides et ordonnées.

* L’armoire de brassage

Elle centralise les équipements réseau (switchs, routeurs, panneaux de brassage). Elle doit être installée dans un lieu sécurisé et ventilé afin de protéger les appareils contre les pannes ou les accès non autorisés.

* Le Switch Ethernet

Le switch est un élément clé du LAN. Contrairement au hub, il est **intelligent** et gère le trafic en fonction des adresses MAC des équipements connectés. Il assure ainsi une communication efficace entre les postes, réduit les collisions et augmente la bande passante disponible pour chaque utilisateur.

* Le routeur et le pare-feu

Le routeur permet d’interconnecter le réseau interne de la faculté avec d’autres réseaux (par exemple Internet ou le réseau de l’université). Le pare‑feu (souvent intégré au routeur moderne) assure la **sécurité** en filtrant les accès entrants et sortants, protégeant ainsi l’infrastructure contre les attaques extérieures

* Hub Ethernet

Les réseaux Hub sont un type de réseau informatique qui relie plusieurs périphériques ou nuds avant de partager des ressources, des informations et des services.

| **Composant** | **Image** | **Rôle et Description** |
| --- | --- | --- |
| Câbles muraux |  | Transmission physique des données à haut débit. |
| Hub Ethernet |  | Relie plusieurs périphériques ou nuds avant de partager des ressources, des informations et des services. |
| Switch Ethernet |  | Gestion intelligente du trafic entre équipements du réseau. |
| Prises murales |  | Point de connexion entre terminaux et câblage du réseau. |
| Armoire de brassage |  | Centralisation et protection des équipements réseau. |
| Panneau de brassage |  | Organisation centralisée des connexions réseau. |
| Routeur firewall Ethernet |  | Interconnexion du LAN avec d’autres réseaux et sécurisation des flux. |

Tableau : Liste de matériel du réseau Ethernet

II.1.4. Les protocoles Internet

Dans une infrastructure réseau, la communication entre les différents équipements repose sur des **protocoles normalisés**. Ces protocoles définissent les règles et les formats d’échange des données afin d’assurer l’interopérabilité entre matériels et logiciels, même lorsqu’ils proviennent de fournisseurs différents. Dans le cadre du déploiement d’une plateforme de gestion des sessions du conseil, la compréhension et l’utilisation adéquate de ces protocoles sont essentielles pour garantir la fiabilité, la sécurité et la performance du système. Les principaux protocoles utilisés dans un réseau moderne sont présentés ci‑après.

* FTP (File Transfer Protocol)

Le protocole FTP permet le transfert de fichiers entre un client et un serveur sur un réseau TCP/IP.  
Dans une application de gestion documentaire, comme celle envisagée pour la FAST, le FTP peut être utilisé pour échanger des fichiers volumineux (comptes rendus, rapports) entre le serveur central et les utilisateurs habilités.

Il offre deux modes de fonctionnement :

* Actif : le serveur initie la connexion de données vers le client.
* Passif : le client initie toutes les connexions, souvent préféré pour traverser les pare-feu.
* HTTP (HyperText Transfer Protocol)

L’HTTP est le protocole fondamental du World Wide Web. Il définit la manière dont un client (navigateur) interagit avec un serveur web pour demander et recevoir des ressources (pages HTML, scripts, images, documents PDF).

La plateforme de gestion des sessions reposera sur ce protocole pour permettre aux membres du conseil d’accéder aux informations et aux comptes rendus via une interface web.

* HTTPS (HyperText Transfer Protocol Secure)

Variante sécurisée de l’HTTP, le protocole HTTPS chiffre les communications grâce au SSL/TLS.  
Il est indispensable dans un contexte où des données sensibles sont échangées (listes des membres, délibérations, documents internes).

Son usage garantit :

* La confidentialité des échanges,
* L’authentification du serveur,
* L’intégrité des données transmises.
* SMTP (Simple Mail Transfer Protocol)

Le SMTP est le protocole standard pour l’envoi de courriers électroniques.  
Dans le cadre du système automatisé de gestion des sessions, il sera utilisé pour envoyer les convocations et notifications aux membres du conseil directement dans leur boîte mail.

* Telnet

Telnet est un protocole permettant une connexion distante sur une machine afin de la contrôler en mode texte. Bien que de moins en moins utilisé aujourd’hui pour des raisons de sécurité (il ne chiffre pas les données), il reste important sur le plan pédagogique pour comprendre les communications réseau de bas niveau.

* UDP (User Datagram Protocol)

UDP est un protocole de communication rapide et léger, utilisé lorsqu’une transmission rapide est privilégiée à la fiabilité (par exemple pour la voix sur IP ou la vidéo en streaming).  
Même s’il n’est pas central dans la plateforme de gestion, il est important dans l’écosystème réseau global pour d’autres services.

* TCP (Transmission Control Protocol)

TCP est un protocole orienté connexion qui garantit :

* L’ordre des paquets,
* La fiabilité de la transmission,
* Le contrôle d’erreur.

La plupart des services de la plateforme (HTTP, FTP, SMTP) s’appuient sur TCP pour assurer la stabilité des échanges de données.

* IPv4 et IPv6 (Internet Protocol)

L’IP est le protocole chargé de l’adressage et du routage des paquets à travers le réseau.

IPv4 utilise des adresses de 32 bits et reste le plus répandu.

IPv6, avec des adresses de 128 bits, est de plus en plus adopté pour pallier la saturation des adresses IPv4.

* ARP (Address Resolution Protocol)

L’ARP permet de traduire une adresse IP en adresse MAC afin de permettre la communication effective sur le réseau local.

* Ethernet

Bien plus qu’un protocole, Ethernet définit un ensemble de standards pour les réseaux locaux filaires. Il spécifie notamment :

* Les méthodes d’accès au support (CSMA/CD),
* Les formats de trames,
* Les débits supportés.

| **Protocole** | **Description** |
| --- | --- |
| FTP (File Transfer Protocol) [] | Transfert de fichiers entre client et serveur |
| HTTP (Hypertext Transfer Protocol) [] | Accès et échange de données via le web (HTTPS sécurisé) |
| SMTP (Simple Mail Transfer Protocol) [] | Envoi de courriers électroniques |
| Telnet [] | Connexion distante en mode texte |
| UDP (User Datagramme Protocol) [] | Transmission rapide sans garantie de livraison |
| TCP [] (Transmission Control Protocol) | Transmission fiable et ordonnée des données |
| IPv4/v6 (Internet Protocol) [] | Adressage et routage des paquets |
| ARP (Adresse Résolution Protocol) [] | Résolution d’adresses IP en adresses MAC |
| Ethernet [] | Standard de communication pour les réseaux locaux |

Tableau : Synthèse des protocoles réseaux

II.1.5. Les services Internet de base

Au‑delà des protocoles, un réseau moderne repose sur un ensemble de **services Internet** qui assurent l’exploitation concrète de l’infrastructure pour les besoins des utilisateurs et des applications. Ces services, généralement fournis par des serveurs dédiés, facilitent la communication, la résolution de noms, l’accès aux ressources et la gestion centralisée des informations.  
Dans le cadre de la mise en place d’une plateforme de gestion des sessions du conseil de la FAST, certains services Internet fondamentaux doivent être compris et intégrés pour garantir un fonctionnement optimal.

* Le service DNS (Domain Name System)

Le DNS est un service essentiel qui assure la traduction des noms de domaine (faciles à mémoriser par les humains) en adresses IP (utilisées par les machines).

Exemple : la résolution du nom *fast.uam.ne* vers une adresse IP du serveur hébergeant la plateforme.

* Le service WEB

Le service web permet l’hébergement et la mise à disposition de contenus accessibles via le protocole HTTP/HTTPS. Il se matérialise par un serveur web (par exemple Apache, Nginx) qui reçoit les requêtes des utilisateurs et renvoie les pages, scripts et documents demandés.

* Le service de messagerie électronique

Ce service englobe plusieurs protocoles et serveurs (SMTP pour l’envoi, IMAP/POP3 pour la réception). Il permet l’échange d’e‑mails entre les utilisateurs internes et externes.

* Le service d’annuaires

Un service d’annuaires (comme LDAP ou Active Directory) centralise la gestion des identités, comptes utilisateurs et droits d’accès. Il fournit un mécanisme de contrôle d’accès uniforme et simplifie l’authentification sur plusieurs services.

| **Protocole** | **Description** |
| --- | --- |
| DNS []  Domain Name System | simplifier l’accès à la plateforme en utilisant un nom de domaine au lieu d’une suite de chiffres. |
| WEB [] | fournir l’interface utilisateur de la plateforme de gestion des sessions de conseil. |
| Le service de messagerie [] | transmettre automatiquement les convocations, rappels et notifications aux membres du conseil. |
| Le service d’annuaires | sécuriser et gérer l’accès des membres autorisés à la plateforme. |

Tableau : Synthèse des services réseaux standards

II.2. Cycle de développement

La réalisation d’une application logicielle suit généralement un cycle de développement bien défini. Ce cycle décrit l’ensemble des étapes nécessaires pour passer de l’expression d’un besoin à la mise en production d’une solution opérationnelle. Dans le cadre du projet de plateforme de gestion des sessions du conseil de la FAST, le choix d’un cycle de développement approprié est crucial pour assurer la qualité du produit final, maîtriser les coûts et respecter les délais. Parmi les différents modèles existants, deux sont particulièrement adaptés aux projets d’ingénierie logicielle : le modèle en Y et le modèle 2TUP (Two-Track Unified Process).

II.2.1. Modèle en Y

Le **modèle en Y** est un modèle de cycle de vie qui met l’accent sur la **spécification parallèle** des traitements et des données. Il est très utilisé dans les approches de conception issues de la méthode MERISE.

* **Principe :**

Le modèle en Y décompose le processus de conception en deux axes principaux :

L’axe fonctionnel (les traitements à réaliser), L’axe structurel (les données à manipuler),  
qui convergent ensuite vers une phase d’implantation où les deux dimensions sont intégrées dans le système final.

* **Étapes principales :**

Analyse et modélisation des données (MCD, MLD, MPD), Analyse et modélisation des traitements (MCT, MOT), Convergence et implantation technique dans l’environnement choisi.

* **Avantage :**

Permet une bonne séparation des préoccupations, Facilite l’évolution du système grâce à une vision claire des données et des traitements, Favorise la qualité de la conception.

II.2.2. Modèle 2TUP

Le **2TUP (Two-Track Unified Process)** est une adaptation du processus unifié (UP), adaptée aux environnements où la flexibilité et l’itérativité sont requises.

* **Principe**

Le modèle 2TUP se fonde sur deux axes :

L’axe **produit** qui suit les étapes classiques du cycle de vie (conception, réalisation, tests),

L’axe **projet** qui gère les aspects organisationnels, la planification et les ajustements en cours de développement.

* **Caractéristiques**

Itératif et incrémental : le développement se fait par cycles successifs, chacun apportant une version améliorée du produit,

Orienté risque : les aspects les plus critiques du projet sont traités en priorité,

Centré sur l’utilisateur : chaque itération est validée par des retours du client ou des utilisateurs.

* **Avantages**

Une meilleure prise en compte des évolutions de besoins en cours de projet,

Une livraison progressive de fonctionnalités,

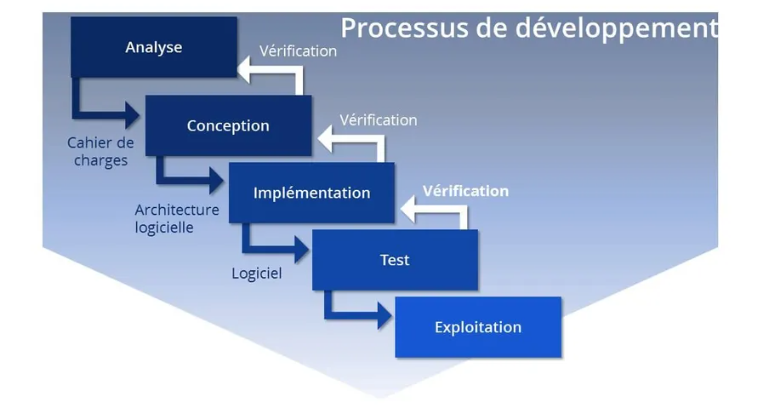
Une grande flexibilité et une réduction des risques liés à l’incertitude initiale.

II.2.3. Le modèle en cascade

Le modèle en cascade (en anglais : waterfall model) est un **modèle de gestion linéaire** qui divise les processus de développement en phases de projet successives. Contrairement aux modèles itératifs, chaque phase est effectuée une seule fois. Les sorties de chaque phase antérieure sont intégrées comme entrées de la phase suivante. Le modèle en cascade est principalement utilisé dans le développement de logiciels.

En pratique, **plusieurs versions du modèle en cascade** sont utilisées. Les modèles les plus courants divisent les processus de développement en cinq phases. Les phases 1, 2 et 3 définies par Royce sont parfois regroupées en une seule et même phase, qualifiée d’analyse des besoins.

* **Analyse :** planification, analyse et spécification des besoins
* **Conception :** conception et spécification du système
* **Implémentation :** programmation et tests des modules
* **Test :** intégration du système, tests du système et de l’intégration
* **Exploitation :** livraison, maintenance, amélioration



II.2.4. Choix d’un modèle

Dans le cadre de notre travail qui est le développement de la plateforme de gestion des sessions du conseil s’appuie sur un modèle de cycle de vie séquentiel proche du modèle en cascade, car les besoins fonctionnels ont été clairement identifiés en amont et validés par la FAST. Cette approche garantit une meilleure traçabilité et une validation formelle de chaque étape avant d’entamer la suivante.

II.3. Méthodes de conception

La phase de conception constitue une étape déterminante dans le cycle de développement d’une application. Elle permet de traduire les besoins fonctionnels exprimés lors de l’analyse en une architecture technique claire, compréhensible et exploitable pour le développement.  
Plusieurs méthodes de conception sont couramment utilisées dans l’ingénierie logicielle. Parmi elles, **MERISE** et **UML** se distinguent par leur maturité et leur large adoption dans les projets de gestion informatisée.

Le choix d’une méthode appropriée permet de structurer le projet, d’assurer la cohérence des données et des traitements, et de faciliter la communication entre les différents acteurs impliqués.

II.3.1. MERISE

La conception d’un système d’information n’est pas évidente car il faut réfléchir à l’ensemble de l’organisation que l’on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s’appuyer.

Merise est une méthodologie de modélisation à usage général dans le domaine du développement de systèmes d’information, du génie logiciel et de la gestion de projet. Introduit pour la première fois au début des années 1980, il était largement utilisé en France. Il a été développé et perfectionné à un point tel que la plupart des grandes organisations gouvernementales, commerciales et industrielles françaises l'ont adopté. Merise procède à un traitement séparé des données et des processus, où la vue des données est modélisée en trois étapes : de la conception à la physique en passant par la logique. De même, la vue axée sur les processus passe par les trois étapes conceptuelles, organisationnelles et opérationnelles. Ces étapes du processus de modélisation sont parallèles aux étapes du cycle de vie : planification stratégique, étude préliminaire, étude détaillée, développement, mise en œuvre et maintenance. C'est une méthode d'analyse basée sur le modèle entité-relation. En utilisant Merise, vous pouvez concevoir des tables avec des relations pour créer une base de données relationnelle.

Elle se repose sur 3 principes de base :

II.3.2. UML

UML est une méthode de modélisation standardisée largement utilisée dans le génie logiciel pour concevoir des systèmes orientés objet. Contrairement à MERISE qui se concentre principalement sur la modélisation des données, UML offre une approche plus globale en intégrant à la fois la structure statique et le comportement dynamique d'un système.

Développé dans les années 1990, UML permet de visualiser l'architecture d'un système à travers différents diagrammes, classés en deux catégories principales : les diagrammes structurels (comme le diagramme de classes) et les diagrammes comportementaux (comme le diagramme de séquence). Cette séparation permet une modélisation claire et précise des composants logiciels et de leurs interactions.

L'un des principaux avantages d'UML réside dans sa flexibilité. Il s'adapte aussi bien aux méthodologies traditionnelles (cycle en V) qu'aux approches Agile, où il peut être utilisé pour des modélisations légères. De plus, UML facilite la communication entre les différents acteurs d'un projet, des analystes aux développeurs, en passant par les clients, grâce à sa notation graphique standardisée.

Cependant, UML présente certaines limites. Sa prise en main peut être complexe pour les débutants, en particulier lorsqu'il s'agit de maîtriser l'ensemble des diagrammes disponibles. Par ailleurs, une utilisation trop rigide d'UML peut entraîner une surcharge de documentation, ce qui peut être contre-productif dans des projets nécessitant une grande réactivité.

En résumé, UML est un outil puissant pour la conception de systèmes logiciels complexes, complétant utilement des méthodes comme MERISE. Son choix dépendra des besoins spécifiques du projet, notamment lorsqu'une approche orientée objet est privilégiée.

II.3.3. Choix d’une méthode

Dans le cadre de notre thème qui est développement de la plateforme de gestion des sessions du conseil de la FAST, il est nécessaire de disposer :

D’une méthode rigoureuse pour la modélisationdesdonnées et la conception d’un schéma relationnel, d’outils de modélisation des interactionsutilisateurs et des processus.

**MERISE** est particulièrement adaptée au contexte, car :

* Elle met l’accent sur la structuration des données et des traitements,
* Elle est éprouvée dans le domaine de la gestion et largement enseignée dans les formations francophones,
* Elle permet d’obtenir des modèles directement exploitables pour le SGBD choisi.

II.4. Les Frameworks

Un **Framework** est un ensemble d’outils, de bibliothèques et de conventions qui facilitent et accélèrent le développement d’applications logicielles. Dans le domaine du développement web, le choix d’un Framework conditionne la productivité des développeurs, la qualité du code et l’évolutivité de l’application. Plusieurs Framework existent, chacun ayant ses avantages, son domaine d’application privilégié et sa communauté. Cette section présente quelques Framework représentatifs, puis justifie le choix de celui utilisé pour la plateforme de gestion des sessions du conseil.

II.4.1. Quick PHP

QuickPHP est un mini serveur web et un micro‑framework qui permet de tester des applications PHP localement.

* **Avantages**

Très léger et facile à configurer,

Idéal pour des tests rapides ou de petites applications PHP.

* **Limites**

Fonctionnalités limitées, pas conçu pour des projets complexes,

Peu de modularité et communauté restreinte.

* **Utilisation typique**

Petits projets locaux ou scripts simples nécessitant un environnement PHP minimal.

II.4.2. CodeIgniter

CodeIgniter est un framework PHP basé sur le modèle **MVC** (Modèle-Vue-Contrôleur).

* **Avantages**

Installation simple et rapide,

Courbe d’apprentissage courte,

Faible consommation de ressources serveur,

Bonne documentation.

* **Limites**

Moins riche en fonctionnalités avancées par rapport à d’autres frameworks modernes,

Communauté moins active qu’auparavant.

* **Utilisation typique**

Projets PHP de taille moyenne où la rapidité de développement est prioritaire.

II.4.3. Laravel

Laravel est aujourd’hui l’un des Framework PHP les plus populaires et les plus complets.

* **Avantages**

Basé sur MVC, architecture propre et modulaire,

Intègre un ORM (Eloquent) pour simplifier la gestion des données,

Authentification, migrations, queues et tests intégrés,

Grande communauté et documentation abondante.

* **Limites**

Peut-être plus lourd pour de très petits projets,

Courbe d’apprentissage un peu plus longue pour les débutants.

* **Utilisation typique**

Applications web complexes nécessitant une architecture robuste et des fonctionnalités avancées.

II.4.4. Synfony

Symfony est un framework PHP orienté entreprise, très modulaire et basé sur des composants réutilisables.

* **Avantages**

Respect strict des standards (PSR, SOLID),

Grande flexibilité et extensibilité,

Large écosystème de bundles,

Utilisé comme base pour d’autres Framework (dont Laravel à ses débuts).

* **Limites**

Installation et configuration plus complexes,

Courbe d’apprentissage plus longue.

* **Utilisation typique**

Projets à grande échelle, nécessitant un fort niveau de personnalisation et une architecture très structurée.

II.4.5. React Js

ReactJS est une bibliothèque JavaScript développée par Meta (Facebook), spécialisée dans la construction d’interfaces utilisateur dynamiques et réactives.

* **Avantages**

Permet de créer des Single Page Applications **(**SPA**)** rapides et fluides,

Utilise un VirtualDOM pour améliorer les performances d’affichage,

Composants réutilisables, facilitant la maintenance et l’évolution,

Grande communauté et écosystème riche.

* **Limites**

Ce n’est qu’une bibliothèque orientée “vue”, il faut ajouter d’autres outils pour gérer le routage, le rendu côté serveur ou la logique back end,

Nécessite une bonne maîtrise de JavaScript moderne.

* **Utilisation typique**

Interfaces web interactives nécessitant une mise à jour fréquente sans rechargement complet de la page.

II.4.6. Next Js

**Next.js** est un framework basé sur **ReactJS**, développé par Vercel, qui ajoute des fonctionnalités essentielles pour créer des applications web complètes.

* **Avantages**

**Rendu côté serveur (SSR)** : améliore les performances et le SEO,

**Rendu statique (SSG)** : permet de générer des pages prêtes à l’emploi pour plus de rapidité,

**Routing intégré** : simplifie la navigation entre les pages sans configuration supplémentaire,

**API Routes** : permet de créer des points d’accès backend dans le même projet,

Gestion optimisée des images et des performances.

* **Limites**

Moins connu que certains frameworks PHP dans certains milieux académiques, mais en forte adoption,

Demande une bonne maîtrise de l’écosystème JavaScript/TypeScript.

* **Utilisation typique**

Applications web modernes nécessitant une interface utilisateur dynamique, un rendu côté serveur et une bonne optimisation des performances.

II.4.7. Choix d’un Framework

Après l’étude des différentes solutions, le choix du Framework pour la plateforme de gestion des sessions du conseil de la FAST s’est orienté vers un environnement moderne et performant.

Les Framework PHP classiques (Laravel, Symfony) sont excellents pour des applications orientées serveur, mais la plateforme nécessite :

* Une **interface réactive et fluide** pour les utilisateurs,
* Un **rendu côté serveur** afin d’améliorer les performances et la compatibilité,
* Une architecture facilement extensible et intégrable avec MySQL.
* **Next.js** a donc été retenu car
* Il s’appuie sur **ReactJS** pour construire des interfaces utilisateur modernes,
* Il offre des fonctionnalités avancées comme le SSR et le SSG,
* Il s’intègre facilement à un backend MySQL via des API,
* Il dispose d’une large communauté et d’outils de déploiement modernes (Vercel, Docker, etc.).

II.5. Les SGBD ou Système de Gestion des Bases de Données

Un Système de Gestion de Base de Données (SGBD) est un logiciel permettant de créer, manipuler, sécuriser et maintenir des bases de données. Dans un projet comme la plateforme de gestion des sessions du conseil de la FAST, le SGBD constitue l’épine dorsale du système : il stocke et organise les informations relatives aux membres du conseil, aux sessions, aux ordres du jour et aux comptes rendus. Le choix d’un SGBD adapté garantit la cohérence des données, la sécurité des accès, et la performance des requêtes. Plusieurs SGBD relationnels et non relationnels sont disponibles. Nous présentons ci‑après quelques solutions couramment utilisées avant de justifier le choix pour notre projet.

II.5.1. MariaDB

MariaDB est un SGBD relationnel open-source, développé à partir du code source de MySQL suite à son rachat par Oracle.

* **Avantages**

Totalement compatible avec MySQL,

Améliorations constantes en termes de performance et de sécurité,

Licence libre (GPL) et forte communauté.

* **Limites**

Certaines fonctionnalités avancées diffèrent de celles de MySQL, ce qui peut poser des problèmes dans certains environnements spécifiques.

* **Utilisation typique**

Applications web nécessitant des bases de données relationnelles avec haute disponibilité et performances améliorées.

II.5.2. Postgre

PostgreSQL est un SGBD relationnel avancé, reconnu pour sa robustesse, sa conformité aux standards SQL et ses fonctionnalités étendues.

* **Avantages**

Gestion des types complexes, indexation avancée, triggers, vues matérialisées,

Haute fiabilité dans les environnements critiques,

Très bonne extensibilité et support du langage PL/pgSQL.

* **Limites**

Courbe d’apprentissage plus longue,

Plus complexe à configurer et à administrer pour des projets simples.

* **Utilisation typique**

Systèmes critiques, data warehouses, grandes applications métiers.

II.5.3. My SQL

MySQL est un SGBD relationnel libre et extrêmement répandu, très bien adapté aux applications web, en particulier dans les environnements Node.js.

* **Avantages**

Léger, rapide, et facile à prendre en main,

Large communauté, documentation abondante,

Parfaitement compatible avec des ORM modernes comme **Prisma**, **Sequelize** ou **TypeORM**, utilisés dans l’écosystème **Next.js**,

Fortement intégré aux hébergeurs web.

* **Limites**

Moins riche en fonctionnalités complexes que PostgreSQL,

Certaines limitations dans les transactions avancées (dans la version communautaire).

* **Utilisation typique**

Applications web dynamiques à fort trafic nécessitant un accès rapide et structuré aux données.

II.5.4. MongoDB

MongoDB est un SGBD **NoSQL**, orienté documents, qui stocke les données au format BSON (similaire au JSON).

* **Avantages**

Très flexible (schéma non strict),

Idéal pour les données non relationnelles ou hétérogènes,

Bonne scalabilité horizontale.

* **Limites**

Moins adapté aux applications nécessitant des relations complexes entre les données,

Moins de contrôle sur l’intégrité référentielle.

* **Utilisation typique**

Applications temps réel, gestion de contenus, big data.

II.5.5. Choix d’un SGBD

Compte tenu de notre thème qui est le développement de la **plateforme de gestion des sessions du conseil** de la FAST, les besoins identifiés sont :

* Une structure relationnelle bien définie (membres, sessions, ordres du jour, comptes rendus),
* Une bonne intégration avec le Framework Next.js, via un ORM compatible,
* Une performance satisfaisante pour des consultations fréquentes et des écritures fiables,
* Une solution bien documentée et maîtrisable dans un contexte académique.

**MySQL** a été retenu comme SGBD pour les raisons suivantes :

* Il offre une compatibilité parfaite avec les outils de l’environnement JavaScript/Node.js,
* Il est simple à déployer et à administrer,
* Il répond pleinement aux exigences d’une application de gestion structurée,
* Il est fiable, performant et largement utilisé dans le monde professionnel et académique.

II.6. Les langages de programmation

Le langage de programmation est un élément central dans la mise en œuvre d’une application. Il détermine la façon dont les fonctionnalités sont développées, les performances atteintes et la facilité de maintenance du code. Dans le domaine du développement web, plusieurs langages se distinguent par leur maturité et leur large adoption. Le choix du langage dépend du type d’application, des compétences de l’équipe et des outils disponibles. Cette section présente quelques langages couramment utilisés, puis justifie le choix du langage approprié pour la plateforme de gestion des sessions du conseil.

II.6.1. PHP

**PHP** (Hypertext Preprocessor) est un langage de script côté serveur largement utilisé dans les applications web traditionnelles.

* **Avantages**

Intégration native avec de nombreux serveurs web (Apache, Nginx),

Large communauté et abondante documentation,

Très utilisé dans les CMS (WordPress, Joomla) et frameworks tels que Laravel ou Symfony.

* **Limites**

Architecture parfois moins adaptée aux applications Single Page Applications (SPA) modernes,

Le rendu côté client reste limité sans combinaison avec JavaScript.

II.6.2. Python

**Python** est un langage polyvalent et très lisible, reconnu pour sa simplicité.

* **Avantages**

Large écosystème de bibliothèques et Framework web (Django, Flask),

Idéal pour des projets nécessitant du traitement de données ou de l’intelligence artificielle.

* **Limites**

Moins optimisé que Node.js pour l’exécution côté client,

Moins natif pour les interactions riches dans le navigateur.

II.6.3. Java

**Java** est un langage orienté objet robuste, utilisé dans les applications d’entreprise et les systèmes embarqués.

* **Avantages**

Grande portabilité grâce à la JVM (Java Virtual Machine),

Frameworks solides (Spring, JSF),

Bonne gestion de la sécurité et de la performance.

* **Limites**

Courbe d’apprentissage plus longue,

Plus lourd à mettre en place pour une application web de taille moyenne.

II.6.4. C#

**C#** est le langage principal de l’écosystème Microsoft (.NET et ASP.NET).

* **Avantages**

Fortement intégré à l’environnement Windows,

Outils et frameworks matures pour le développement web et desktop,

Bonne gestion des transactions et de la sécurité.

* **Limites**

Dépendance fréquente aux solutions Microsoft,

Moins courant dans les environnements open source modernes.

II.6.5. Choix d’un langage de programmation

Pour notre projet de **plateforme de gestion des sessions du conseil** de la FAST, les critères suivants ont guidé le choix du langage :

* **Compatibilité avec le Framework choisi** : l’application sera développée avec **Next.js**, qui repose sur l’environnement **Node.js**.
* **Interface réactive et moderne** : la solution doit proposer une expérience utilisateur fluide et dynamique.
* **Écosystème riche et moderne** : pour garantir l’évolutivité, l’intégration avec MySQL et la disponibilité de bibliothèques.
* **Choix retenu : JavaScript et TypeScript**
* **JavaScript (ES6+)** est le langage natif du web, supporté par tous les navigateurs, et s’exécute aussi côté serveur grâce à Node.js.
* **TypeScript**, surcouche de JavaScript, apporte un typage statique et une meilleure maintenabilité du code dans des projets complexes comme celui-ci.

Le projet sera implémenté principalement en **JavaScript** et **TypeScript** au sein du Framework **Next.js**. Ce choix permet d’assurer la compatibilité totale entre le front‑end et le back‑end, une grande réactivité de l’interface et une intégration fluide avec la base de données MySQL.

II.7. Les environnements serveurs

Un **environnement serveur** est l’ensemble des logiciels installés sur un serveur permettant d’exécuter une application et de fournir des services aux clients (navigateurs web, applications mobiles, etc.). Dans le cadre d’une plateforme de gestion, il est essentiel de choisir un environnement serveur stable, performant et compatible avec les technologies utilisées dans le projet. Traditionnellement, de nombreux environnements sont orientés vers les langages comme PHP. Cependant, avec l’essor des frameworks modernes tels que **Next.js**, de nouveaux environnements basés sur **Node.js** s’imposent pour répondre aux exigences actuelles en termes de performance et de flexibilité.

II.7.1. EasyPHP

**EasyPHP** est un environnement de développement pour Windows qui regroupe :

* **Apache** comme serveur web,
* **MySQL** comme base de données,
* **PHP** comme langage de script.
* **Avantages**
* Installation simple et rapide,
* Intègre des outils comme phpMyAdmin pour gérer les bases de données,
* Idéal pour un apprentissage ou des prototypes rapides en PHP.
* **Limites**
* Spécifique aux technologies PHP/MySQL,
* Peu adapté pour exécuter des applications en JavaScript côté serveur comme celles développées avec Next.js,
* Ne propose pas nativement de gestion des rendus côté serveur (SSR) en Node.js.

II.7.2. WampServer

**WampServer** est un autre environnement populaire sous Windows, regroupant également **Apache**, **MySQL** et **PHP** (d’où l’acronyme WAMP : Windows, Apache, MySQL, PHP).

* **Avantages**
* Interface intuitive pour démarrer ou arrêter les services,
* Permet d’héberger des applications PHP en local sans configuration complexe,
* Supporte plusieurs versions de PHP.
* **Limites**
* Environnement conçu spécifiquement pour le couple PHP/MySQL,
* Ne prend pas en charge nativement les Framework Node.js/Next.js,
* Moins flexible pour des applications modernes avec rendu côté serveur en JavaScript.

II.7.3. Lamp

**LAMP** est un acronyme pour **Linux, Apache, MySQL, PHP**. C’est un environnement serveur classique utilisé depuis de nombreuses années pour héberger des sites et applications web.

* **Avantages**
* Très stable et éprouvé dans le temps,
* Large communauté et documentation abondante,
* Très bien supporté par de nombreux hébergeurs.
* **Limites**
* Spécifique au langage PHP,
* Nécessite des ajustements supplémentaires pour exécuter des applications Node.js,
* Moins optimisé pour des applications SPA/SSR basées sur JavaScript moderne.

II.7.4. Choix d’un serveur

Dans cadre de notre travail, les critères suivants ont été retenus pour le choix de l’environnement serveur :

* Compatibilité avec **Next.js**, qui repose sur **Node.js**,
* Support du **rendu côté serveur (SSR)** et de la génération statique (SSG),
* Intégration facile avec **MySQL** pour la persistance des données,
* Simplicité de déploiement et bonne performance.
* **Environnement retenu : un serveur basé sur Node.js**.

Au lieu d’utiliser un environnement classique (EasyPHP, WampServer ou LAMP) centré sur PHP, le projet adoptera un **serveur Node.js** car :

* Il permet d’exécuter du **JavaScript côté serveur**,
* Il s’intègre parfaitement avec **Next.js** et ses fonctionnalités avancées,
* Il offre une gestion simple des dépendances grâce à **npm** ou **yarn**,
* Il est largement supporté sur des plateformes de déploiement modernes (Vercel, Heroku, serveurs Linux).

II.8. Synthèse

Dans ce chapitre, nous avons présenté les bases théoriques et technologiques nécessaires à la réalisation de la plateforme de gestion des sessions du conseil de la FAST.  
Nous avons d’abord rappelé les **normes et standards réseau** (modèles ISO/OSI et TCP/IP, matériel LAN, protocoles et services Internet), indispensables pour garantir la connectivité et la sécurité.

Ensuite, nous avons analysé différents **cycles de développement** (modèle en Y et 2TUP) et retenu le **modèle en Y**, plus adapté à la modélisation rigoureuse des données et traitements.  
Concernant les **méthodes de conception**, **MERISE** a été choisie comme méthode principale, avec un apport d’UML pour représenter les interactions.

Enfin, nous avons comparé plusieurs **Framework**, **langages** et **SGBD**. Le projet adopte **Next.js** (basé sur ReactJS) avec **JavaScript/TypeScript**, un **SGBD MySQL** et un **environnement serveur Node.js**, pour garantir performance, évolutivité et maintenabilité.

Chapitre III. Conception et réalisation de l’application de gestion des sessions du conseil de la FAST

Introduction

Ce chapitre présente la conception et la réalisation effective de la plateforme de gestion des sessions du conseil de la FAST. Il s'articule autour de plusieurs axes : l'amélioration de l'architecture réseau existante, la modélisation des données et des traitements avec la méthode MERISE, la mise en place de l'environnement technique, le développement des interfaces utilisateur, la sécurisation du système et enfin l'évaluation des coûts de mise en œuvre.

L'objectif est de démontrer comment les choix technologiques présentés dans l'état de l'art ont été concrètement appliqués pour répondre aux besoins identifiés lors de l'analyse de l'existant.

III.1. Architecture améliorée du réseau

III.2. Conception et modélisation avec MERISE

III.2.1. Dictionnaire des données

En MERISE (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise), le dictionnaire de données est un élément essentiel qui permet de documenter et de définir de manière précise et structurée toutes les données manipulées par le système d'information. Il s'agit d'une référence centrale pour décrire les entités, les attributs, les relations et les contraintes qui régissent le fonctionnement de la base de données.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom du champ | Type | Taille | Description | contraintes |
| Id\_membre | entier | - | Identifiant unique du membre | Clé primaire |
| nom | text | 100 | Nom du membre | obligatoire |
| prenom | text | 100 | Prénom du membre | obligatoire |
| email | text | 150 | Adresse email du membre | Unique, obligatoire |
| fonction | text | 100 | Fonction dans le conseil | facultatif |
| Mot\_de\_passe | Text(hash) | 255 | Mot de passe chiffré | obligatoire |
| Profile\_utilisateure | Enum | - | Role : Admin ou membre | Par defaut : membre |
| Id\_session | entier | - | Identifiant unique de la session | Clé primaire |
| Date\_session | date | - | Date de la session | Obligatoire |
| lieu | text | 100 | Lieu de tenue de la session | Obligatoire |
| president | text | 100 | Nom du président de séance | Facultatif |
| Id\_ordre | entier | - | Identifiant de l’ordre du jour | Clé primaire |
| Titre\_point | text | 200 | Intitulé du point à l’ordre du jour | Obligatoire |
| Description\_point | text | 500 | Description détaillée du point | Facultatif |
| Id\_convocation | entier | - | Identifiant de convocation | Clé primaire |
| Date\_envoi | date | - | Date d’envoi de la convocation | Automatique |
| statu | enum | - | Statu : envoyé/en attente/annulé | Par defaut : en attente |
| Id\_pv | entier | - | Identifiant du procès-verbal | Clé primaire |
| Contenu\_pv | text | - | Contenu du compte rendu | Obligatoire |
| Auteur\_pv | text | 100 | Rédacteur du procès-verbal | Facultatif |

III.2.2. Le MCD

Le MCD est une représentation abstraite et indépendante des détails techniques de la façon dont les données sont stockées ou manipulées. Il se concentre sur les concepts métier et les entités qui sont importantes pour le domaine d'application. Les principales composantes du MCD sont les entités, les relations et les attributs. Les entités représentent des objets du monde réel, les relations définissent les liens entre ces entités, et les attributs spécifient les caractéristiques de ces entités. Le MCD est souvent représenté sous forme de diagrammes entité-association (EA) et aide à définir la structure générale des données.

III.2.3. Le MLD

Le modèle logique de données (MLD) est l'une des étapes clés de la méthodologie de conception des systèmes d'information Merise. Il s'agit du deuxième modèle dans la séquence conceptuelle -> logique -> physique et il vise à décrire comment les données sont organisées, structurées et interconnectées dans le système d'information, indépendamment de toute considération technique spécifique.

Le MLD est une représentation intermédiaire entre le modèle conceptuel des données (MCD) et le modèle physique des données (MPD). Il traduit les entités, les attributs, les relations et les règles métier du MCD en structures plus concrètes, telles que des tables, des vues et des contraintes, tout en évitant les détails spécifiques au système de gestion de bases de données (SGBD).

III.2.4. Le MCT

Le modèle conceptuel de traitement permet de capturer la logique et la séquence des opérations métier au sein du système d'information. Il favorise une compréhension commune des processus et des règles de gestion, facilitant ainsi la conception d'une solution qui répond aux besoins métier. Le MCT sert également de base pour le développement ultérieur du modèle logique des traitements (MLT), qui se concentre davantage sur les aspects techniques et les détails d'exécution.

III.3. L’environnement de déploiement

III.3.1. Installation et configuration du Framework

III.3.2. Installation et configuration du serveur

III.3.3. Installation et configuration du SGBD

III.4. Réalisation et mise en exploitation de l’application

III.4.1. Interface d’ouverture de session

III.4.2. Interface de gestion des membres du conseil

III.4.3. Interface de gestion des sessions du conseil

III.4.4. Interface de gestion des ordres de jour

III.4.5. Interface de gestion des convocations

III.5. Sécurisation de l’environnement

III.5.1. Système de fichiers NTFS de Windows

III.5.2. Gestion de l’authentification par Windows

III.5.3. Gestion de l’authentification par l’application

III.5.4. Gestion des profils par l’application

III.6. Evaluations et Coûts de mise en œuvre

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Libellé** | **Caractéristiques** | **Quantité** | **Prix unitaire** | **Total** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | |  |  |  |
| TOTAL | | | |  |

Tableau : Evaluation des coûts de mise en œuvre

III.7. Synthèse

Conclusion Générale

Bibliographie

Table des matières