Une image contenant texte, décapsuleur, outil

Description générée automatiquement

**UNIVERSITE IBN TOFAIL**

FACULTE DES SCIENCES – KENITRA

### Département d’informatique

# Projet de fin d’études

Pour l’obtention de

Licence en informatique

# APPLICATION WEB de

# gestion d’une salle

# de sport

Présenté par

Bouktitiya hamza

Elghayouane mohamed

SOUTENUE LE : 25/06/2022

Devant la commission d’examen composée de :

M. lkram ELAZAMI Professeur à la Faculté des Sciences de Kénitra, Président de jury

M Khalid HOUSNI Professeur à la Faculté des Sciences de Kénitra, Encadrant

**Remerciement**

##### Nous tenons à remercier nos professeurs de nous avoir incités à travailler en mettant à notre disposition leurs expériences et leurs compétences. Nous tenons à remercier dans un premier temps, toute l’équipe pédagogique de département d’informatique et les intervenants professionnels responsables de la filière Science Mathématique et Informatique au sein de la faculté des sciences IBN TOFAIL, pour tout le savoir-faire qu’ils nous ont transmis

##### Avant d’entamer ce rapport, nous profitons de l’occasion pour remercier tout d’abord notre professeur Monsieur Khalid Housni qui n’a pas cessé de nous encourager pendant la durée du projet, ainsi pour sa générosité en matière de formation et d’encadrement. Nous le remercions également pour l’aide et les conseils concernant les missions évoquées dans ce rapport, qu’il nous a apporté lors des différents suivis, et la confiance qu’il nous a témoigné.

Merci pour tous

**Dédicace**

**A ceux qui ont contribué à l’élaboration de ce travail**

**Et ceux à qui je dois tant**

**Ma plus grande gratitude et tout mon amour à mes parents, qui ont su me faire confiance, me soutenir et m’encourager au cours de ma vie.**

**A mes sœurs et frères**

**A TOUS QUI M'AIMENT**

**A touts mes amis en qui j’ai toujours trouvé le soutien et le réconfort.**

# Table des matières

TABLE DES MATIERES 1

[TABLES DES FIGURES 3](#_TOC_250034)

INTRODUCTION GENERALE 5

CADRE GENERAL 5

TRAVAIL DEMANDE 5

[PLAN DU RAPPORT 5](#_TOC_250033)

CHAPITRE I : ETUDE, SPECIFICATION ET ANALYSE DES BESOINS 7

* 1. [INTRODUCTION 8](#_TOC_250032)
  2. [ETUDE DE L’EXISTANT 8](#_TOC_250031)
  3. [SPECIFICATION DES BESOINS 9](#_TOC_250030)
  4. choix de cycle de vie 9
     1. [Introduction 9](#_TOC_250029)
     2. [Spécification des besoins 10](#_TOC_250028)

[1.3.2.1 Les acteurs 10](#_TOC_250027)

* + 1. [Les besoins fonctionnels 11](#_TOC_250026)
    2. [Les diagrammes de cas d’utilisation 12](#_TOC_250025)
       1. [Diagramme de Cas d’utilisation : L’administrateur 13](#_TOC_250024)
       2. [Diagramme de Cas d’utilisation : Directeur de département 15](#_TOC_250023)
       3. [Diagramme de Cas d’utilisation : Cas Enseignant 16](#_TOC_250022)
       4. [Diagramme de Cas d’utilisation : Etudiant 17](#_TOC_250021)

1.5 [ANALYSE DE LA VUE DYNAMIQUE 18](#_TOC_250020)

* + 1. [Diagramme de séquence : « Authentification » 18](#_TOC_250019)
    2. [Diagramme de Séquence : « Ajout Département » 18](#_TOC_250018)
    3. [Diagramme de séquence « Modifier un département » 19](#_TOC_250017)
    4. [Diagramme de séquence : « Supprimer un département » 19](#_TOC_250016)
    5. [Diagramme de séquence « Affectation en tant que Vacant » 20](#_TOC_250015)
    6. [Diagramme de séquence : « Affectation Charge enseignant» 21](#_TOC_250014)
    7. [Diagramme de séquence : « Consultation Emploi » 21](#_TOC_250013)
  1. Conclusion 9

CHAPITRE II: CONCEPTION 22

* 1. [INTRODUCTION 23](#_TOC_250012)
  2. [DIAGRAMME DE CLASSE 24](#_TOC_250011)
  3. Conclusion 26

CHAPITRE III: REALISATION 29

* 1. [INTRODUCTION 30](#_TOC_250010)
  2. [ENVIRONNEMENT DE TRAVAIL 30](#_TOC_250009)
     1. [Environnement matériel 30](#_TOC_250008)
     2. [Environnement logiciel 30](#_TOC_250007)
        1. [Choix des technologies de développement 30](#_TOC_250006)
        2. [Les Outils de développement 32](#_TOC_250005)
  3. ETAPES DE REALISATION 33
     1. [L’architecture de l’application Web 33](#_TOC_250004)
  4. [LES INTERFACES DE L’APPLICATION 34](#_TOC_250003)
  5. [Conclusion 34](#_TOC_250003)

[CONCLUSION 40](#_TOC_250002)

[BIBLIOGRAPHIE 41](#_TOC_250001)

[ANNEXE A 42](#_TOC_250000)

# Tables des figures

Figure 1 : Cycle de vie en V 11

Figure 1. 1 : Diagramme de Cas d’utilisation Général 12

Figure 1. 2 : Diagramme de Cas d’utilisation Pour l’Authentification 13

Figure 1. 3 : Diagramme de Cas d’utilisation Pour l’Administrateur 14

Figure 1. 4 : Diagramme de Cas d’utilisation Pour un client 16

Figure 1. 5 : Diagramme de séquence « Authentification » 18

Figure 1. 6 : Diagramme de séquence « gestion produits » 19

Figure 1.7 : Diagramme de séquence « suppression » 19

Figure 1. 8 : Diagramme de séquence : « modification » 20

Figure 2. 1 : Le modèle Conceptuel de Données 23

Figure 2. 2 : Diagramme de classe 25

Figure 3. 1 : Authentification 33

Figure 3. 2 : Page inscrire 34

Figure 3. 3 : Page d’administrateur 34

Figure 3. 4 : List des inscription 35

Figure 3. 5 : page d’ajoute un membre 35

Figure 3. 6 : liste des membres 35

Figure 3. 7 : modifier membre 36

Figure 3. 8 : List des entraineurs 36

Figure 3. 9 : page d’ajoute un entraineur 37

Figure 3. 10 : page modifier un entraineur 38

Figure 3. 11 : page contrats obsolètes 39

Figure 3. 12 : page liste de commandes 39

Figure 3. 13 : page modifier contrats obsolètes 41

Figure 3. 14 : page d’ajoute produit 41

Figure 3. 14 : page liste des produits 41

Figure 3.16 :totale………………………………………………………………………………….42

Figure 3. 19 : page des informations sur le club 43

Figure 3.20 : page d’exposition des produits………………………………………………………43

Figure 3.21 : les produits…………………………………………………………………………...44

Figure 3.22 : contact………………………………………………………………………………..44

Figure 3.23 : les calendriers………………………………………………………………………...45

Figure 3.24 : les entraineur…………………………………………………………………………46

Liste des abréviations

|  |  |
| --- | --- |
| **PHP** | **H**ypertext **P**re**P**rocess |
| **JS** | **J**ava**S**cript |
| **SQL** | **S**tructured **Q**uery **L**anguage |
| **HTML** | **H**yper**T**ext **M**arkup **L**anguage |
| **CSS** | **C**ascading **S**tyle **S**heets |
| **UML** | **U**nified **M**odelng **L**anguage |
| **XML** | **E**xtensible **M**arkup **L**anguage |
| **MVC** | **M**odel **V**iew **C**ontroller |
| **MLD** | **M**odèle **L**ogique de **D**onnées |
| **POO** | **O**bject – **O**riented **P**rograming |

# Introduction Générale

## Cadre général

Avant l'invention de l'ordinateur, on enregistrait toutes les informations Manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières... etc.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de données des entreprises, ce qui est la partie essentielle dans leur développement

Aujourd’hui. Une salle de sport de musculation qui est un lieu où sont rassemblés des équipements permettant la Pratique d'exercices d'activité physique, qui n'échappe pas à cette règle(informatisation), de Plus les responsables des salles de musculation sont plus en plus conscients de l'importance des Applications de gestion. Les contraintes ne se limitent pas à ce que nous venons de citer, elles sont plus nombreuses et nous les découvrirons dans la suite de ce rapport.

Autrefois le stockage des données d'une entreprise est fait manuellement ce qui rend ses tâches plus Complexe et difficiles à gérer. Grâce à l'informatisation ou la digitalisation, ce problème ne sera plus existé, les gens peuvent effectuer leurs travails d'une manière plus rapide et efficace. Il est possible pour une entreprise d'engager moins de Personnel et d'économiser des dépenses ainsi qu'elles peuvent stocker des données plus facilement ce qui leur permet d'économiser le coût du stockage externe des fichiers.

## Travail demandé

Après pratiquée la musculation on a remarqué l’absence de bonne gestion de la salle et ses inscriptions. C’est pourquoi on a réalisé cette application web qui constitue de deux interfaces. la première interface pour la gestion de gym et la deuxième interface pour les clients.

## Plan du rapport

La première partie de ce rapport présente le contexte de notre projet, le travail à réaliser et la solution que nous proposons pour réaliser ce travail

La deuxième partie « Etude, Spécification et analyse des besoins » se focalise sur l’étude de l’existant, la présentation du projet à réaliser, ensuite, l’identification et l’analyse des besoins fonctionnels et non fonctionnels.

La troisième partie expose les différentes interfaces homme-machine développées ainsi que certains détails de l’implémentation. Les premières interfaces présentées sont parmi celles qui comportent des opérations d’ajout, de modifications et de suppression, les deuxièmes sont celles qui présentent une description des états possibles et des emplois prêts à être imprimés.

Nous clôturons ce rapport par une conclusion et les perspectives.

# Chapitre I : Etude, Spécification et analyse des besoins

## Introduction

Actuellement elle existe de nombreux logicielles conçus pour gérer les salles de sport de musculation malgré leur diversité, ces logiciels se basent sur les mêmes principes que ce soit au niveau des données, ou au niveau des traitements.

L'emploi du temps modifiable est l 'un des exemples les plus fréquents des problèmes d'optimisation dans un site de salle de sport. Il peut se manifester sous des formes selon le choix d’administrateur de club. Réaliser un emploi du temps pose un problème qu'on doit résoudre.

-absence des méthodes de vérification.

-un grand risque de faire des erreurs lors de de saisie des informations.

-difficulté de géré les inscriptions.

Tous ces problèmes nous amène à développer une application web Qui permet d’offrir aux responsables d’une salle de sport une bonne gestion d’information.

## Critique de l’existant

La critique de l’existant est une phase primordiale qui se fait après l’étude de l’existant. Cette étape a pour objectif la découverte et la précision des erreurs produites par l’utilisation manuelle a fin ’apporter les solutions convenables.

Le système actuel pose plusieurs problèmes

* Une grande partie est manuelle.
* Perte du temps

Jusqu’au aujourd’hui,nos salle de sport n’a aucune site web pour la publication des produits de salle et les machines afin d’attirer plus des membres de plus, l’administrateur utilise toujours les blocs-notes et les application de bureau.C’est dans ce cadre que s’applique notre projet.Gerer les membres et les produits et suivre les inscription dans la partie client au s’affiche les produits et les entreneur etc. Via la partie administrateur .

## Choix de cycle de vie

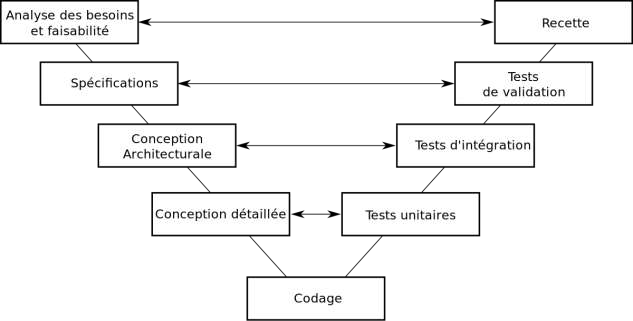
Pour choisir le processus de développement qui va mener au bon déroulement des étapes du projet et pour éviter l’échec d’un projet, il faut effectuer une étude profonde sur les processus de développement afin de choisir le processus adéquat

Les processus de développement classique nécessitent une idée précise du projet, avec un planning bien détaillé et où tous les risques possibles ont été anticipé. C’est exactement notre cas, alors le choix était clair « modèle en V. »

Le principe de ce modèle est qu’avec toute décomposition doit être décrite la recomposition et que toute description d’un composant est accompagnée de tests qui permettront de s’assurer qu’il correspond à sa description.

Le principal avantage du cycle en V est qu’il évite de revenir en arrière incessamment pour redéfinir les spécifications initiales, Son intérêt :

* + - Validations intermédiaires
    - Bon suivi du projet : avancement éclairé et limitation des risques en cascade d’erreurs
    - Favorise la décomposition fonctionnelle de l’activité
    - Génération de documents et outils supports
    - Modèle très utilisé et éprouvé

Dans ce modèle verticalement nous trouvons les étapes du développement et horizontalement la vérification.

**Figure 1:Cycle de vie en V**

## Spécification des Besoins

### Introduction

Une méthode de conception est une démarche générale reflétant une philosophie de présentation et de suivi du système. Elle propose des outils spécifiques permettant un suivi efficace de l’information relative au système. Et notre choix se porte sur le langage UML (Unified Modeling Language) qui facilite l’interactivité avec la base de données à l’aide des diagrammes de cas d’utilisation et des diagrammes de classes

L’UML (Unified Modeling Language) est un langage de modélisation orientée objet, elle est développée dans le but de définir la notion standard pour la modélisation des applications construites à l’aide des objets. Elle est utilisée pour spécifier un logiciel ou pour le concevoir, le modèle décrit les classes et les cas d’utilisation vus de l’utilisateur final du logiciel.

Le modèle produit par une conception orientée objet est en général une extension du modèle issu de la spécification, il l’enrichit de classe dites techniques qui n’intéressent pas l’utilisateur final du logiciel mais seulement ses concepteurs.

### Spécification des besoins

### 1.4.2.1 Les acteurs

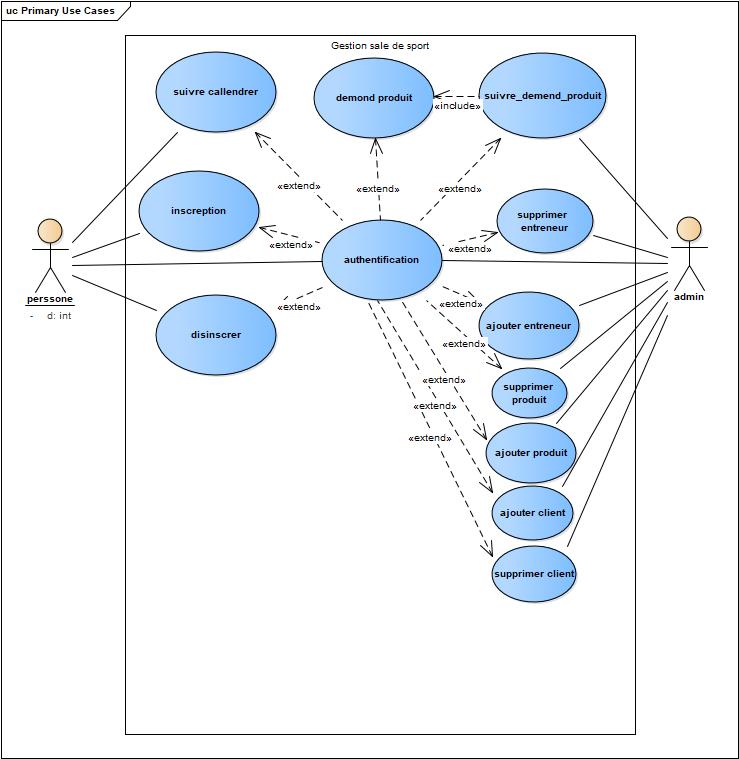
Un acteur est une entité externe qui interagit avec le système (opérateur, centre distant, autre système...). En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. Les acteurs peuvent être classés (hiérarchie).

* **L’administrateur** : c’est la personne primordiale dans le système . Il permet de manipuler toutes les tâches proposées et possibles pour gérer les engagée et les produits du club ; tels que Ajout, Modifier, Supprimer et consulter.
  + **Client:** toute personne qui suit une formation peut :
* Suivre des régimes
* Suivre les calandrer
* Demander du produit
* Voire les machines disponible
* Consulter les pages publique

### Les besoins fonctionnels

* La première étape consiste l’inscription de client dons le site et après dons le gym toutes les informations nécessaires tels que le nom, le prénom…
* Une personne est définie par son nom, prénom, ...
* Chaque client assure une ou plusieurs produits.
* Les clients peuvent suivre calendrier.
* Le client peut être faire l’inscription dons le site et aussi dans la gym.
* L´admin peut être faire ajouter entraineur supprimer entraineur modifier calandrer et aussi peut être faire ajouter et supprimer client.

La figure suivante illustre le diagramme des cas d’utilisation générale de notre application.



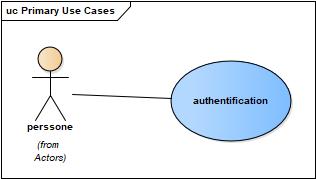
**Figure 1. 1 : Diagramme de Cas d’utilisation Général**

On entend dire par gérer les quartes actions principales sur un objet qui sont : ajouter, supprimer, modifier, consulter. Dans la conception textuelle et l’analyse des différents cas d’utilisation, on s’intéresse à l’action d’ajout.

### Les diagrammes de cas d’utilisation

Le diagramme de cas d’utilisation décrit la succession des opérations réalisées par un acteur. C’est le diagramme principal du modèle UML, celui qui assure la relation entre l’utilisateur et les objets que le système met en œuvre.

La description de l’interaction est réalisée suivant le point de vue de l’utilisateur, et les cas d’utilisation permettent de recueillir et de décrire les besoins des acteurs aux systèmes, il permet aussi de faciliter la structuration des besoins des utilisateurs et d’exprimer les limites et les objectifs du système.

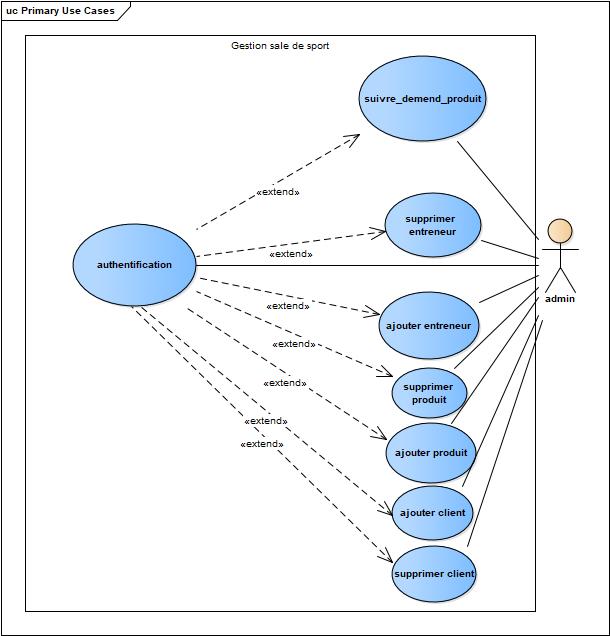


**Figure 1. 2: Diagramme de Cas d’utilisation Pour se connecter**

**Raffinement du cas d’utilisation « S’authentifier admin »**

|  |  |
| --- | --- |
| Cas d’utilisation | S’authentifier |
| Acteurs | Administrateur |
| Pré condition | Être un personnel de NCE |
| Post condition | Utilisateur du système authentifié |
| Scénario de base | * L’utilisateur saisit son login et son mot de passe. * L’utilisateur clique sur le bouton « se connecter ». * Le système vérifie la combinaison login et mot de passe.   Le système affiche la page d’accueil selon le profil de l’utilisateur. |
| Exception | - Si le login ou le mot de passe erroné, le système affiche un  message : « Login ou mot de passe incorrect ». |

### Diagramme de Cas d’utilisation : L’administrateur

****

**Figure 1. 3: Diagramme de Cas d’utilisation Pour l’Administrateur**

### Scénario du cas d’utilisation « Ajouter client »

|  |  |
| --- | --- |
| Cas d’utilisation | Ajouter client |
| Acteurs | Administrateur |
| Pré condition | Administrateur authentifié |
| Post condition | Client ajouté |
| Scénario de base | * L’administrateur saisit les informations sur le client (Intitulé, adresse, . . . ). * L’administrateur clique sur le bouton « Ajouter ». * Le système vérifie si un champ obligatoire est manquant. * Le système enregistre les données.   Le système affiche un message de succès. |
| Exception | - Si un champ manque le système affiche un message d’erreur. |

**Scénario du cas d’utilisation «Modifier client» :**

|  |  |
| --- | --- |
| Cas d’utilisation | Modifier client |
| Acteurs | Administrateur |
| Pré condition | Client affiché |
| Post condition | Client modifié |
| Scénario de base | * L’administrateur saisit les modifications. * L’administrateur clique sur le bouton « Enregistrer ».   Le système enregistre les modifications. |
| Exception | - Si un champ manque le système affiche un message d’erreur. |

**Scénario du cas d’utilisation «Supprimer client» :**

|  |  |
| --- | --- |
| Cas d’utilisation | Supprimer client |
| Acteurs | Administrateur |
| Pré condition | Client affiché |
| Post condition | Client supprimé |
| Scénario de base | * L’administrateur clique sur le bouton « Supprimer ». * Le système vérifie l’absence d’interventions référant à l’article. * Le système affiche un message de confirmation. * L’administrateur clique sur le bouton « Confirmer ».   Le système supprime le client. |
| Exception | - L’administrateur peut annuler la suppression. |

**Raffinement du sous cas d’utilisation «Modifier article»**

|  |  |
| --- | --- |
| Cas d’utilisation | Modifier article |
| Acteurs | Administrateur |
| Pré condition | Article affiché |
| Post condition | Article modifié |
| Scénario de base | * L’administrateur saisit les modifications. * L’administrateur clique sur le bouton « Enregistrer ». * Le système enregistre les modifications. |
| Exception | - Si un champ manque le système affiche un message d’erreur. |

### Diagramme de Cas d’utilisation : client

### 

**Figure 1. 4 :Diagramme de Cas d’utilisation Pour client**

**Raffinement du cas d’utilisation « S’authentifier client »**

|  |  |
| --- | --- |
| Cas d’utilisation | S’authentifier |
| Acteurs | **Client** |
| Pré condition | Être un personnel de NCE |
| Post condition | Utilisateur du système authentifié |
| Scénario de base | * Le client saisit son login et son mot de passe. * Le client clique sur le bouton « se connecter ». * Le système vérifie la combinaison login et mot de passe.   Le système affiche la page d’accueil selon le profil de l’utilisateur. |
| Exception | - Si le login ou le mot de passe erroné, le système affiche un  Message : « Login ou mot de passe incorrect ». |

**Raffinement du cas d’utilisation « demande produit »**

|  |  |
| --- | --- |
| Cas d’utilisation | Demande produit |
| Acteurs | Client |
| Pré condition | Client authentifier |
| Post condition | Produit demander |
| Scénario de base | -le client choisi un produit et clique sur ajout produit  -le système ajout le produit au panier  -le client faire lâchât  -le système vérifier le processus d’achat  -le système affiche le processus et effectuer avec succès |
| Exception |  |

R**affinement du cas d’utilisation « inscription »**

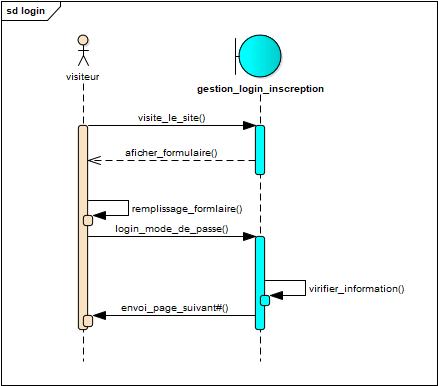
|  |  |
| --- | --- |
| Cas d’utilisation | Demande produit |
| Acteurs | Client |
| Pré condition |
| Post condition | Client inscrit |
| Scénario de base | * Le client saisit les informations personnelles. * Le client clique sur le bouton « inscrit ». * Le système vérifie si un champ obligatoire est manquant. * Le système enregistre les données.   -Le système affiche un message de succès.  -Le système affiche la page d’accueil selon le profil de client. |
| Exception | - Si un champ manque le système affiche un message d’erreur. |

## Analyse de la Vue dynamique

Le diagramme de séquence UML est un diagramme qui permet de représenter les interactions entre les objets, suite à un événement externe en précisant la chronologie des échanges de messages.

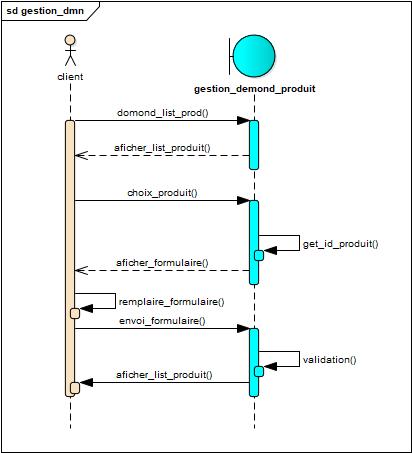
Pour analyser les comportements des cas d’utilisations, nous utilisons les diagrammes de séquences.

### Diagramme de séquence : « Authentification »



**Figure 1. 5: Diagramme de séquence « Authentification »**

### Diagramme de Séquence : Admin « gestion produit » :

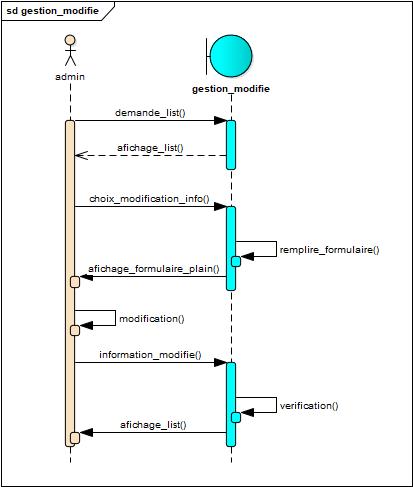


**Figure 1. 6 : Diagramme de séquence** : Admin « gestion produit »

### Diagramme de séquence : « Supprimer»

**Figure 1. 7: Diagramme de séquence : « Supprimer»**

### Diagramme de séquence «modifier »



**Figure 1. 8: Diagramme de séquence « Modifier»**

## Conclusion

Dans ce chapitre on a présenté une analyse de l’existant, les exigences fonctionnelles et non fonctionnelles et l’analyse du l’application, ainsi que la méthode de gestion adoptée durant la réalisation de notre projet.

Dans le chapitre suivant nous entamons la conception.

**Chapitre II: Conception**

## Introduction

La conception des bases de données est la tâche la plus ardue du processus de développement du système d’information

Recourir à une méthode de conception afin de faciliter la communication et la coopération entre les différents acteurs d’une application.

La conception d’une telle base de données consiste à suivre quatre étapes :

* + - Analyse de la situation existante et des besoins :

Cette première étape repose sur l’analyse de l’existant et des besoins, elle est très délicate et fondamentale dans le processus de conception

* + - Création d'une série de modèles conceptuels (canonique et vues externes) qui permettent de représenter tous les aspects importants du problème
    - Traduction des modèles conceptuels en modèle logique et optimisation (normalisation) de ce modèle logique
    - Implémentation d'une base de données dans un SGBD, à partir du modèle logique

Domaine

Problème posé

Représenter

Modèle

Conceptuel

BD

Solution proposée

Implémenter

Modèle

Logique

Traduire

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Figure 2. 1 : Le modèle Conceptuel de Données

Le modèle Conceptuel de Données (MCD) ou modèle Entité/Association est un modèle chargé de représenter, sous forme graphique, les informations manipulées par le système. Le MCD permet de décrire les données gérées sans tenir compte des choix d’organisation ou techniques.

Le MCD a pour objectif d’identifier, décrire et modéliser les entités et leurs associations à l’aide d’une présentation graphique

Certaines contraintes ne sont pas représentables par le seul formalisme de base (entité, association, propriétés, cardinalités) mais correspond à une règle que doit satisfaire le modèle pour être fidèle et cohérent avec l’activité à représenter.

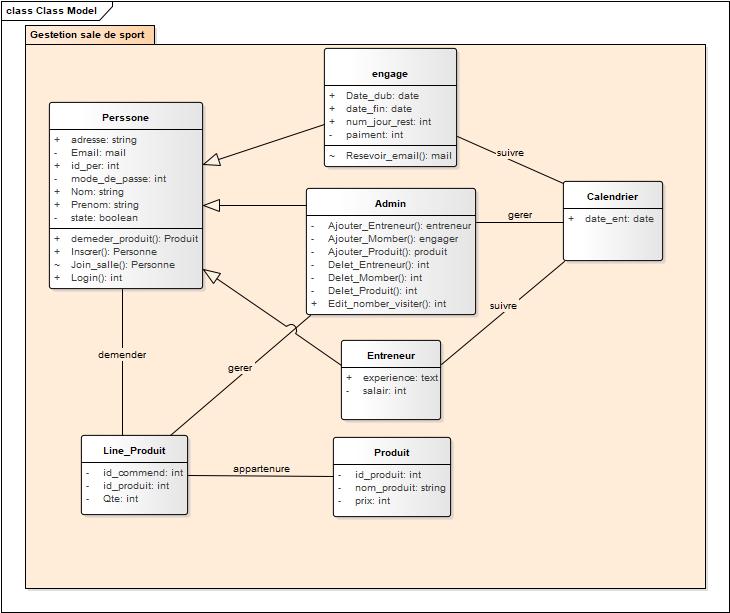
Par exemple : Pour une telle séance d’emploi du temps, un enseignant ne fait un cours que dans une seule salle.

## Diagramme de classe

Le diagramme de classe représente la description statique du système à développer en intégrant dans chaque classe la partie dédiée aux données et celle consacrée au traitement.

C’est un diagramme pivot de l’ensemble de modélisation d’un système, cette représentation est concentrée sur le concept de classe et d’associations, les traitements sont matérialisés par des opérations.

Une classe est une description abstraite d’un ensemble d’objet ayant des propriétés similaires, un comportement commun et des relations communes avec d’autres objets.



**Figure 2. 2 : Diagramme de classe**

## Conclution

Au cours de ce chapitre, nous avons conçu les différents composants de notre système.

Maintenant, notre application est prête à être codées. Le prochain chapitre concerne la mise en place de nos applications.

**Chapitre III: Réalisation**

## . Introduction

Ce dernier chapitre présente la partie de la réalisation et la mise en œuvre des différents composants décrits au niveau du chapitre précédent. Dans un premier temps, on présente l’environnement matériel et logiciel. Ensuite, on décrit le travail réalisé en détaillant quelques captures d’écrans des fonctionnalités réalisées.

## Environnement de travail

### Environnement matériel

Pour développer cette application nous avons utilise les matériels suivants :

* Un ordinateur portable php (système d’exploitation Windows 10, processeur Intel i5-8365U, 8 Go de RAM. La taille du disque dur est de 256Go SSD).
* Un ordinateur portable php (système d’exploitation Windows 10, processeur Intel i5-1035G1, 8 Go de RAM. La taille du disque dur est de 500Go SSD).

### Environnement logiciel

****

**Sublime Text** est un [éditeur de texte](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte) générique codé en [C++](https://fr.wikipedia.org/wiki/C%2B%2B) et [Python](https://fr.wikipedia.org/wiki/Python_(langage)), disponible sur [Windows](https://fr.wikipedia.org/wiki/Windows), [Mac](https://fr.wikipedia.org/wiki/Mac_OS) et [Linux](https://fr.wikipedia.org/wiki/Linux). Le logiciel a été conçu tout d'abord comme une extension pour [Vim](https://fr.wikipedia.org/wiki/Vim), riche en fonctionnalités[1](https://fr.wikipedia.org/wiki/Sublime_Text#cite_note-1).

Depuis la version 2.0, sortie le [26](https://fr.wikipedia.org/wiki/26_juin) [juin](https://fr.wikipedia.org/wiki/Juin_2012) [2012](https://fr.wikipedia.org/wiki/2012_en_informatique)[2](https://fr.wikipedia.org/wiki/Sublime_Text#cite_note-2), l'éditeur prend en charge 44 [langages de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares.

******

**XAMPP** est un ensemble de [*logiciels*](https://fr.wikipedia.org/wiki/Logiciel) permettant de mettre en place un [*serveur Web*](https://fr.wikipedia.org/wiki/Serveur_Web) local, un [*serveur FTP*](https://fr.wikipedia.org/wiki/Serveur_FTP) et un [*serveur de messagerie électronique*](https://fr.wikipedia.org/wiki/Serveur_de_messagerie_%C3%A9lectronique). Il s'agit d'une distribution de [*logiciels libres*](https://fr.wikipedia.org/wiki/Logiciel_libre) (**X** (*cross*) [*Apache*](https://fr.wikipedia.org/wiki/Apache_HTTP_Server) [*MariaDB*](https://fr.wikipedia.org/wiki/MariaDB) [*Perl*](https://fr.wikipedia.org/wiki/Perl_(langage)) [*PHP*](https://fr.wikipedia.org/wiki/PHP)) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les [*systèmes d'exploitation*](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) les plus répandus.



**Enterprise Architect** est un logiciel de [*modélisation*](https://fr.wikipedia.org/wiki/Mod%C3%A9lisation_informatique) et de [*conception*](https://fr.wikipedia.org/wiki/Conception_de_logiciel) [*UML*](https://fr.wikipedia.org/wiki/Unified_Modeling_Language), édité par la société australienne [*Sparx Systems*](https://fr.wikipedia.org/w/index.php?title=Sparx_Systems&action=edit&redlink=1). Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus.

### Longages utilisés

Une image contenant texte, clipart, signe

Description générée automatiquementLe **HTML** est un langage qui a pour rôle de gérer et organiser le contenu d'une page web. C'est un langage de description de données, et non un langage de programmation. J'ai utilisé le HTML 5 qui est la dernière version du HTML qui est actuellement toujours

en développement. Cette version apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires etc. [1]

Une image contenant texte, clipart, signe

Description générée automatiquement

Le rôle du **CSS** est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte…). Ce langage est le complément du langage HTML pour obtenir une page web avec du style. Le navigateur parcourt le

document HTML. Lorsqu'il rencontre une balise, il demande à la CSS de quelle manière il doit l'afficher.[2]



**JavaScript** désigne un langage de développement informatique, et plus précisément un langage de [script](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203599-script-definition/) orienté objet. On le retrouve principalement dans les pages Internet. Il permet, entre autres, d'introduire sur une page web ou HTML des petites animations ou des effets.

Créé en 1995 par Brendan Eich, en même temps que la technologie Java, le langage JavaScript se distingue des langages [serveurs](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203337-serveur-informatique-definition-traduction/) par le fait que l'exécution des tâches est opérée par le navigateur lui-même, sur l'ordinateur de l'utilisateur, et non sur le serveur web. Il s'active donc généralement sur le poste client plutôt que côté serveur.[3]



Le **PHP** est un langage informatique utilisé sur l'internet. Le terme PHP est un acronyme récursif de "*PHP: Hypertext Preprocessor*".

Ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que MySQL.

Exécuté du côté serveur (l'endroit où est hébergé le site) il n'y a pas besoin aux visiteurs d'avoir des logiciels ou plugins particulier. Néanmoins, les webmasters qui souhaitent développer un site en PHP doivent s'assurer que l'hébergeur prend en compte ce langage. Lorsqu'une page PHP est exécuté par le serveur, alors celui-ci renvois généralement au client (aux visiteurs du site) une page web qui peut contenir du [HTML](http://glossaire.infowebmaster.fr/html/), XHTML, [CSS](http://glossaire.infowebmaster.fr/css/), [JavaScript](http://glossaire.infowebmaster.fr/javascript/) ...[4]

## Etapes de réalisation

### L’architecture de l’application Web

### Le model MVC :

L’architecture MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d’une application. Son principal intérêt est la séparation des données (modèle), de l’affichage (vue) et des actions (contrôleur), ce qui assure la clarté de l’architecture et simplifie la tâche du développeur responsable de la maintenance et de l’amélioration du projet.

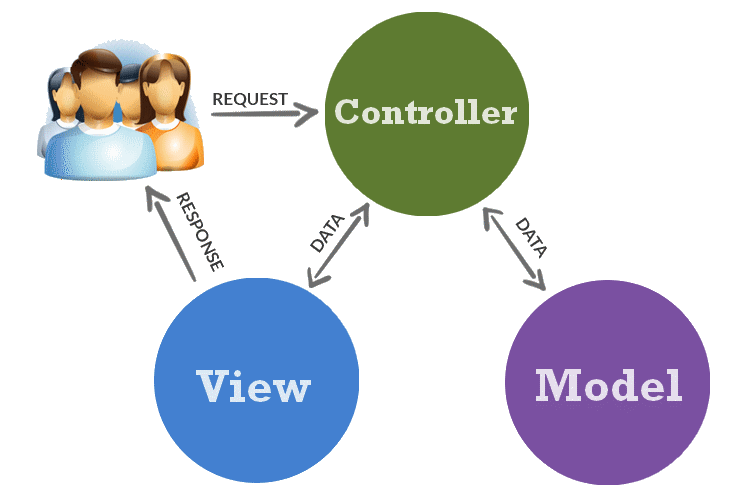
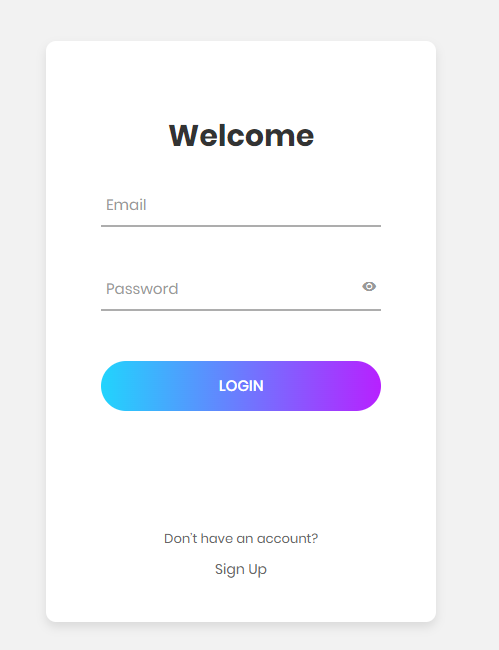
Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure :

Figure 4 : L'architecture MVC

* **Le Modèle :** Le modèle représente le cœur de l’application : traitements des données, interactions avec la base de données. Il décrit les données manipulées par l’application. Il regroupe la gestion de ces données et, il est responsable de leur intégrité. La base de données sera l’un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s’occupent pas de la présentation, Le modèle ne contient aucun lien direct vers la vue.
* **La Vue :** C’est avec quoi l’utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle, sa seconde tâche est de recevoir toute action de l’utilisateur (clic de souris, sélection d’un bouton radio, coche d’une case, entrée de texte, de mouvements, de voix, etc..). Ces différents événements sont envoyés au contrôleur. La vue n’effectue pas de traitement, elle se contente d’afficher les résultats des traitements effectués par le modèle et d’interagir avec l’utilisateur.
* **Le Contrôleur :** Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l’utilisateur et déclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ce dernier notifie la vue que les données ont changée pour qu’elle se mette à jour. D’après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est « observable ». Certains événements de l’utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n’effectue aucun traitement, ne modifie aucune donnée, il analyse la requête du client et se contente d’appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

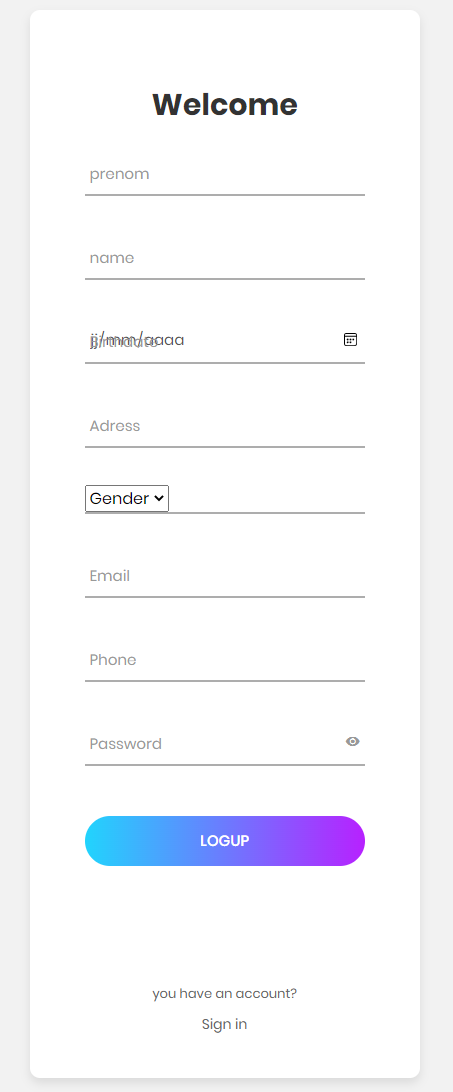
## Les Interfaces de l’application

### Les interfaces publiques



**Figure 3. 1 : Authentification**

La figure 3.1 : présente l’interface Authentification de l’application. Cette interface est décomposée deux champs pour la saisie du nom d’utilisateur (Login) et du mot de passe pour pouvoir accéder à l’application.



**Figure 3. 2 : page d’inscrire**

La figure 3.2 : Cette page permet aux internautes d’inscrire dans la site pour la visite automatique. .

### Les interfaces d’administrateur

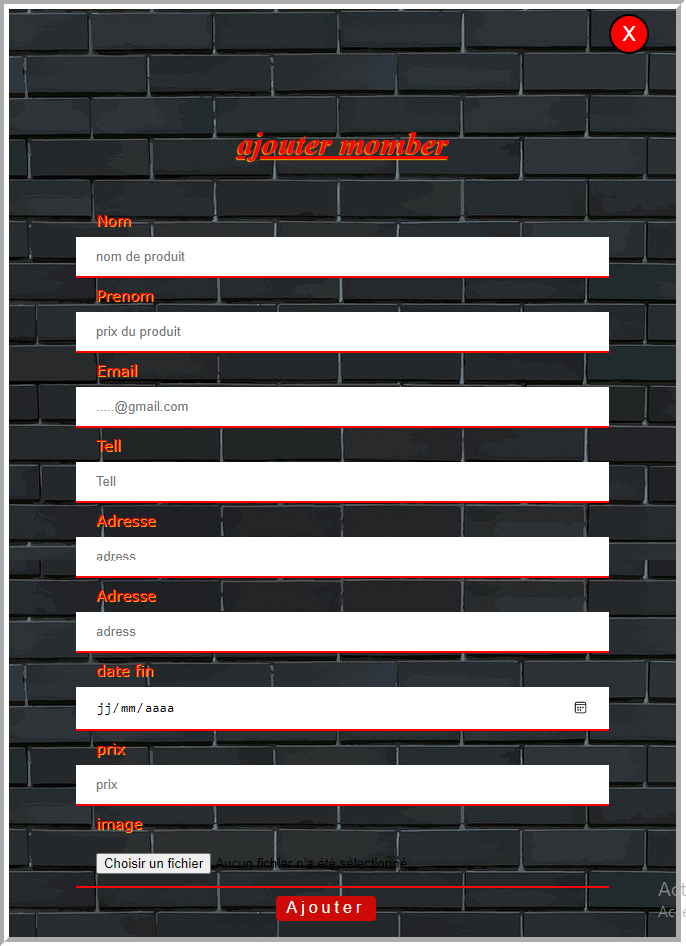
****

**Figure 3. 3 : page d’administrateur**

La figure 3.3 : Cette interface est accessible à l’administrateur seulement.

**Figure 3. 4: liste des inscription**

La figure 3.4 : Cette page permet d’afficher les informations des personnes qui sont inscrits dans la salle

** Figure 3. 5 : page d’ajoute un membre**

La figure 3.5 : Cette page permet d’ajouter un membre de salle du sport avec la validation des informations saisit par l’administrateur

**Une image contenant table

Description générée automatiquementFigure 3.6: page de liste des membres**

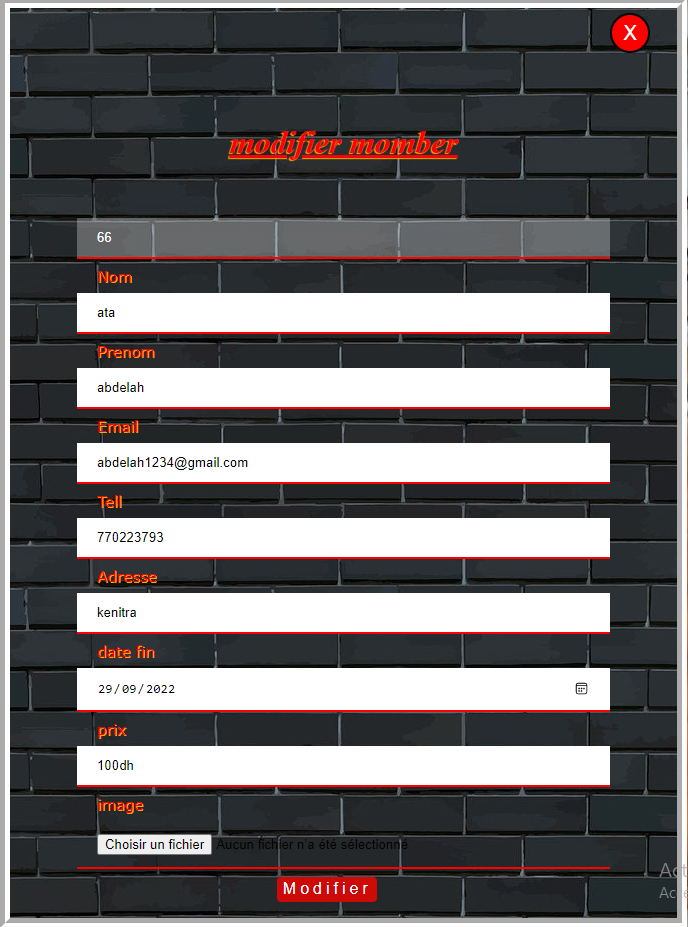
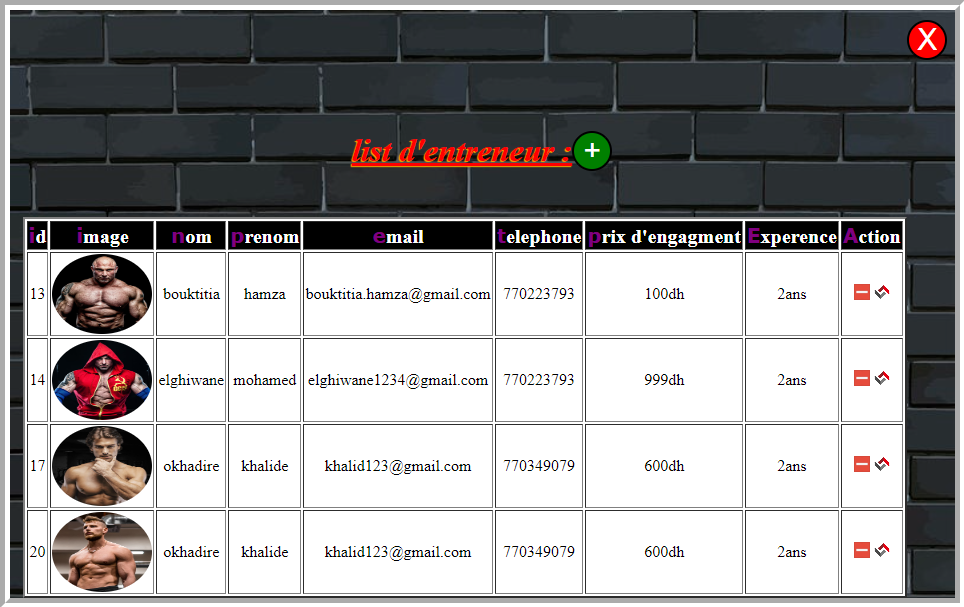
****

Figure 3. 7 : **modifier membre**

La figure 3.7 : Cette page permet modifier les informations des membres du club



### Figure 3. 8 : liste des entraineurs

La figure 3.8 : Cette page permet d’afficher les informations d'un entraîneur.

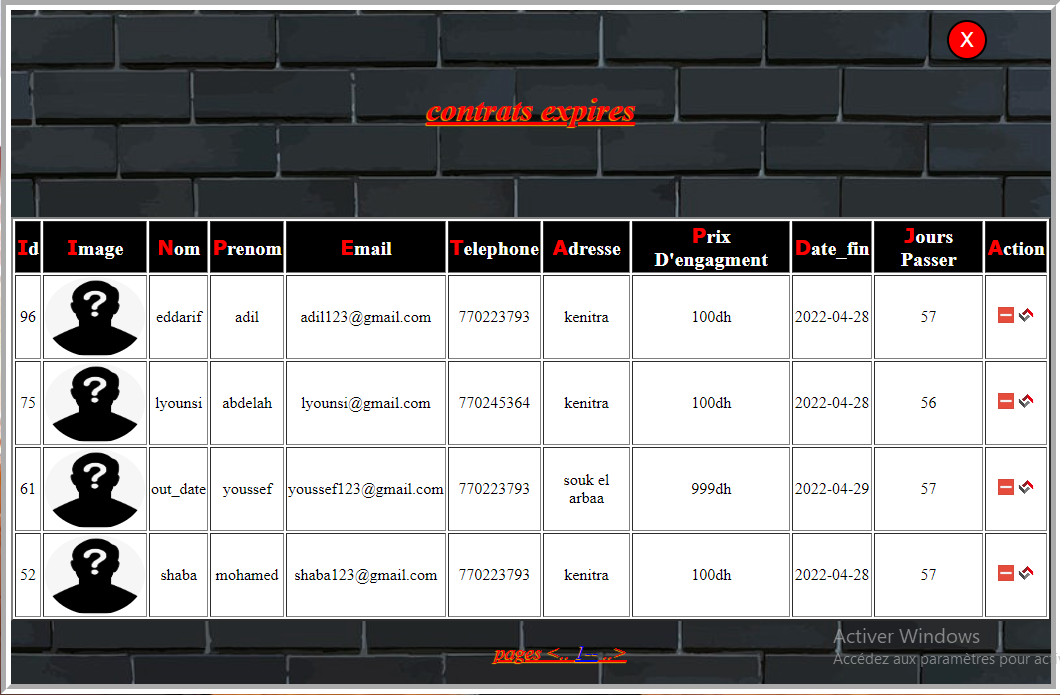
****

### Figure 3. 9 : page d’ajoute un entraineur

La figure 3.9 : Cette page permet d’ajouter un entraineur. Avec la validation des informations saisit par l’administrateur.

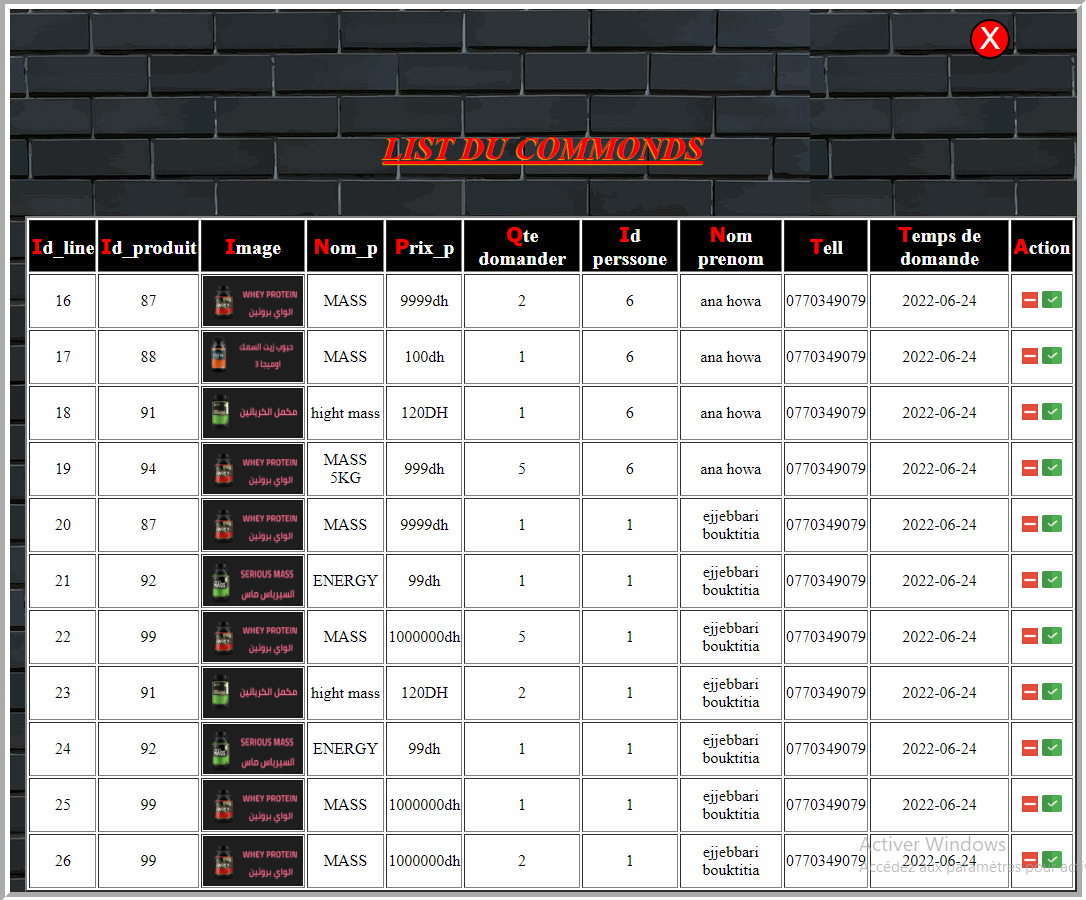
### Figure 3. 10 : page modifier un entraineur

La figure 3.10 : Cette page permet de modifier un entraineur. Avec la validation des informations saisit par l’administrateur.

****

### Figure 3. 11 : page contrats expires

La figure 3.11 : Cette page permet d’afficher les informations des contrats expires

****

### Figure 3. 12 : page liste de commandes

### La figure 3.12 : Cette page permet d’afficher les informations des commandes

.****

### Figure 3. 13 : page modifier contrat obsolète

La figure 3.13 :Cette page permet de modifier les informations d'un m’ombre avec un contrat périmé. et ajouter un nouveau temps de fine de contrat.



### Figure 3. 14 : page ajouter produit

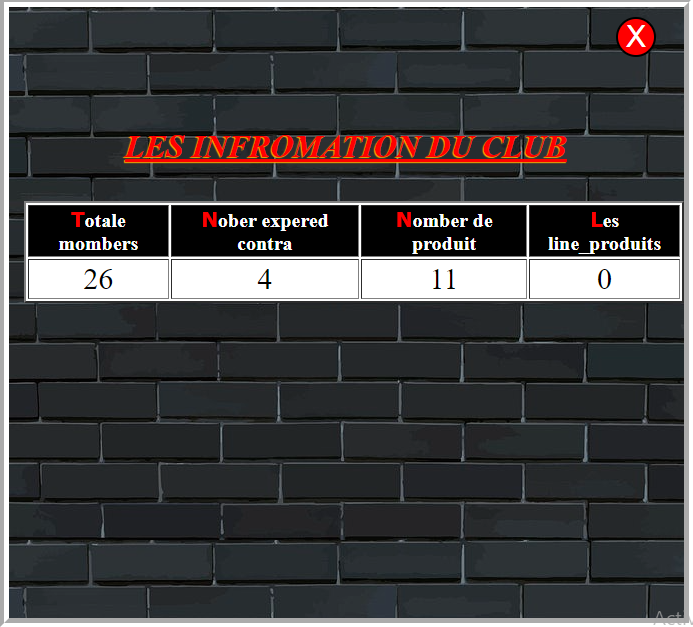
# La figure 3.14 : Cette page permet d’ajouter un produit. Avec la validation des information saisure pare administrateur et renvoie liste des produits.

Une image contenant table

Description générée automatiquement

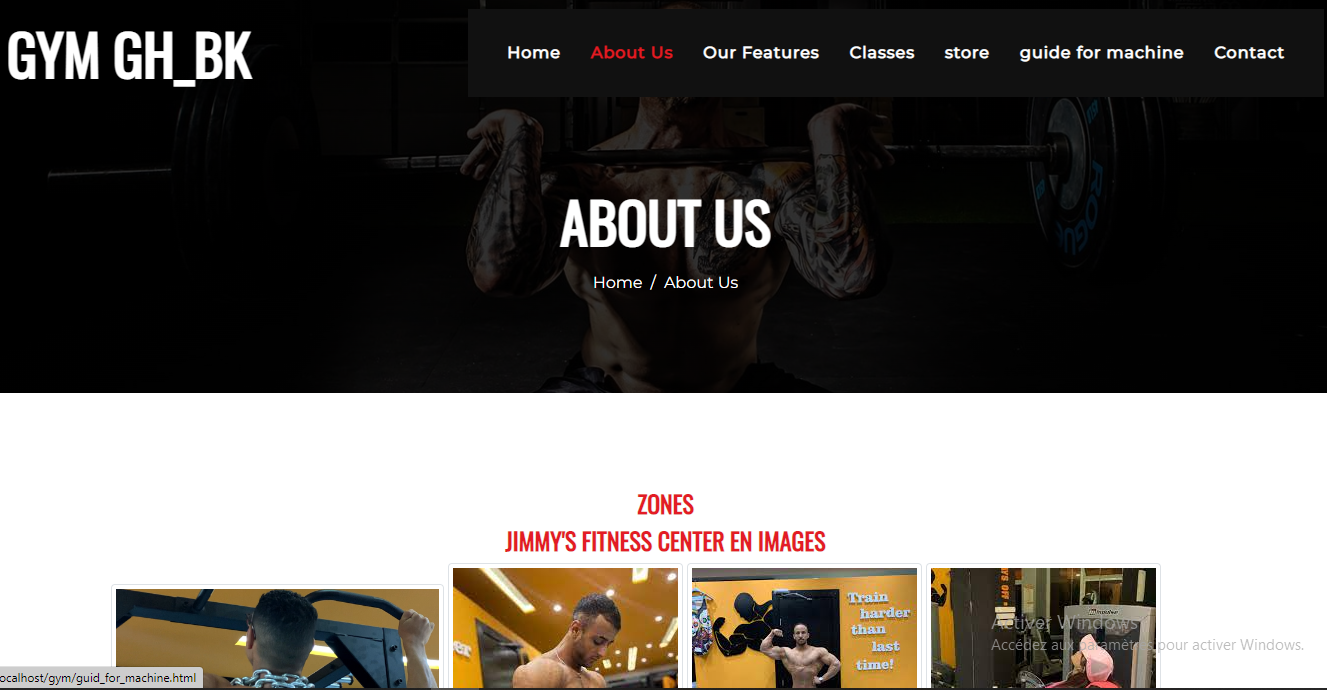
### Figure 3. 15 : page liste produit

La figure 3.15 : Cette page permet d’afficher les informations des produits

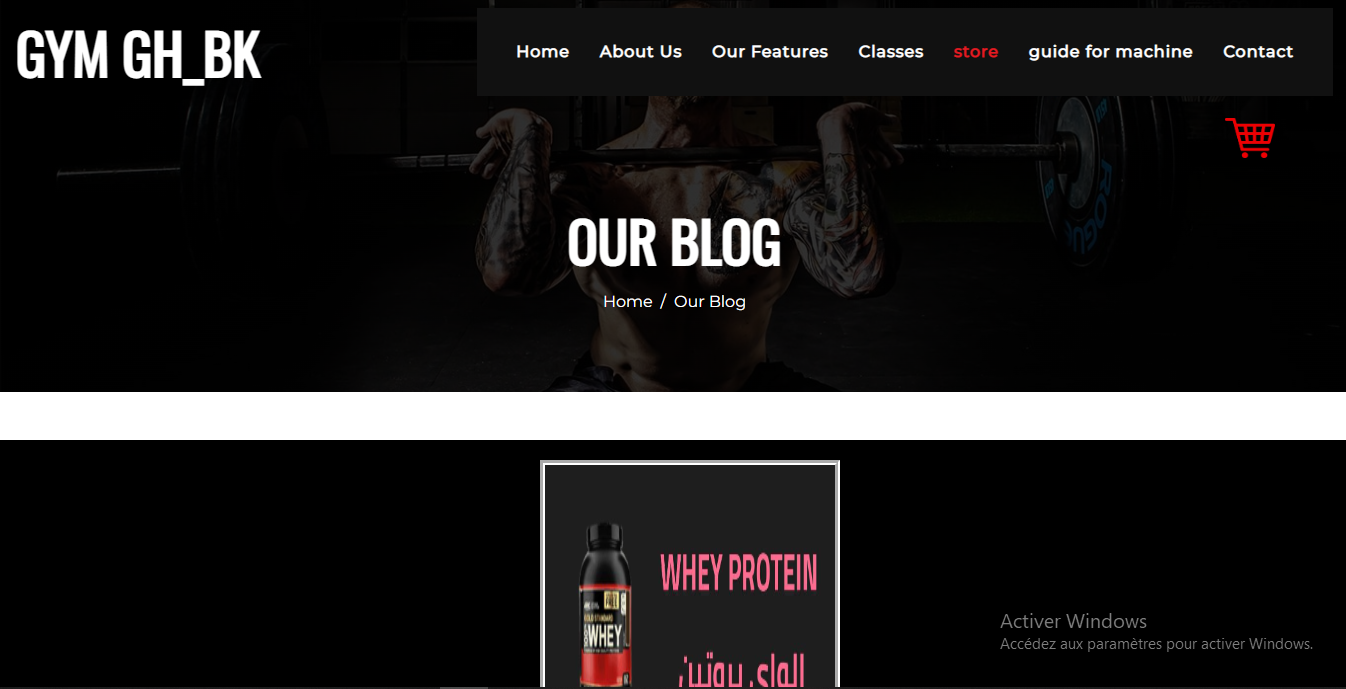
****

### Figure 3. 16 : totale

La figure 3.16 : Cette page permet d’afficher les informations du clubLes interfaces de client

****

### Figure 3. 19 :page des information sur le club

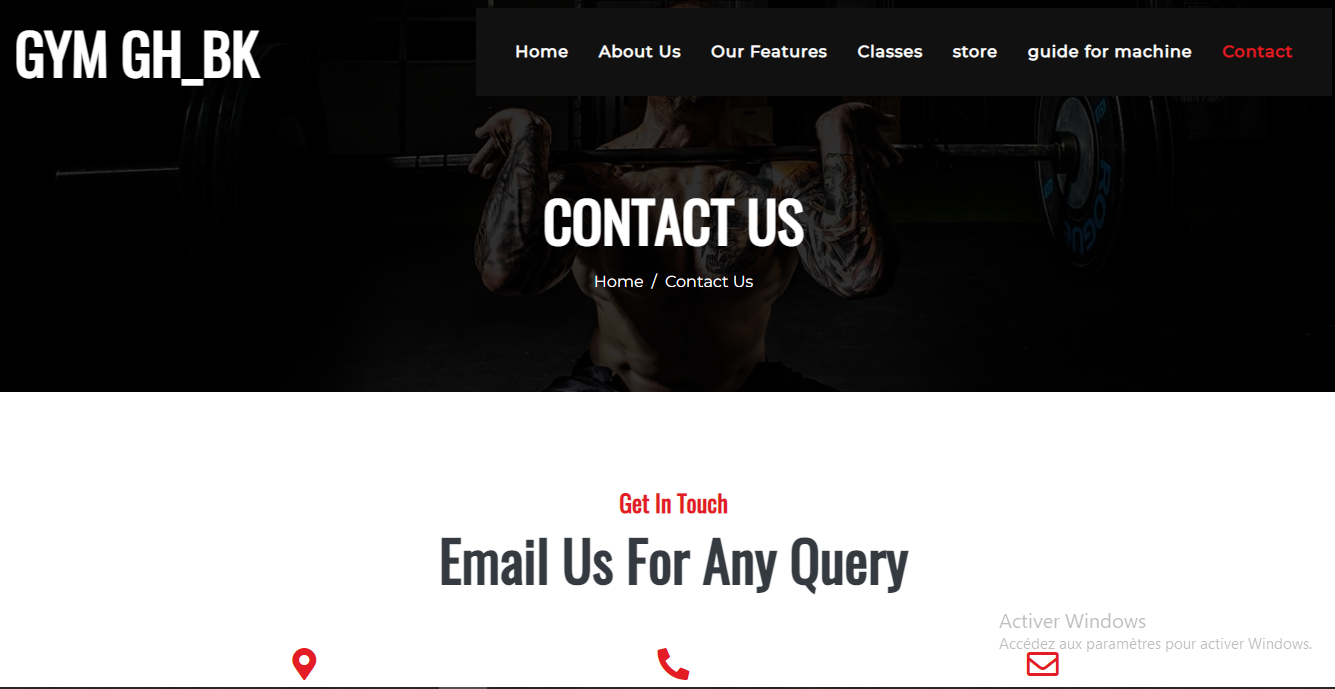
****

### Figure 3. 20 : page d’exposition des produits

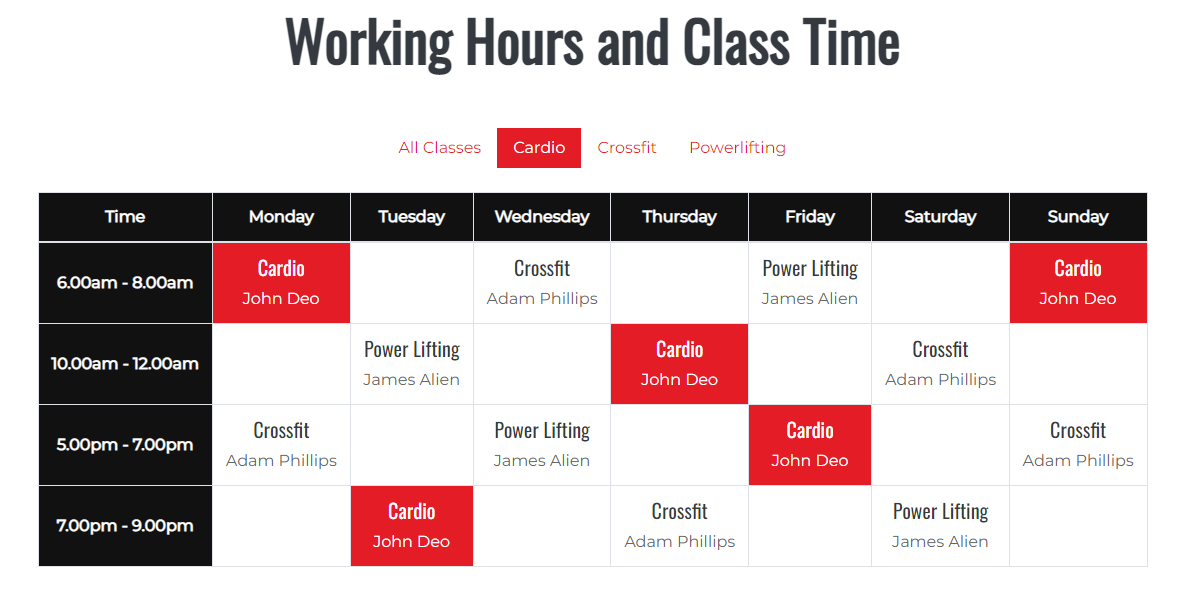


### Figure 3. 21 : les produits

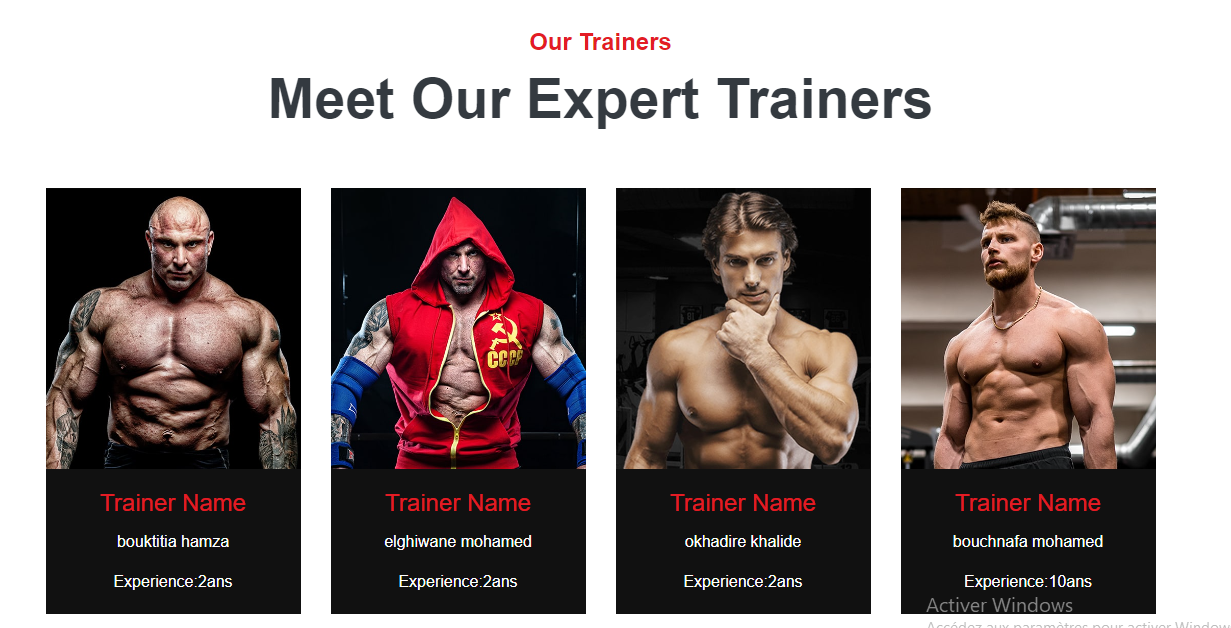
La figure 3.21 :Image des produits avec un bouton et espace peur saisit la quantite.



### Figure 3. 22 : contact



### Figure 3. 23: les calandrers



### Figure 3. 24: les entraineur

figure 3.24 : Cette page permet d’afficher les entraineur avec son image et nom et prénom.

**3.5 CONCLUSION**

Dans ce chapitre nous avons présenté en détails le développement de notre système, Nous avons commencé par présenter l’environnement matériel et logiciel, les choix techniques et les outils de travail sur lesquels se base notre application, et enfin nous avons conclu par les scénarios de test et de validation de l’application.

# CONCLUSION

##### En effet, ce projet était une étape très importante dans notre cycle de formation vu qu’il était une occasion très intéressante et bénéfique pour savoir comment appliquer sur le plan pratique des connaissances théoriques déjà acquises et aussi il nous a permis d’acquérir de nouvelles connaissances techniques.

Au même temps, nous avons appris l’importance de la programmation et du codage pour l’exécution d’un site informatique. Ainsi que l’importance de la recherche et de la planification des tâches pour le bon déroulement des travaux. Et grâce à un environnement favorable pour le travail et la coordination d’efforts nous avons pu réaliser le projet demandé par notre faculté.

##### C’est vrai que ce travail peut s’étendre encore plus, mais le fait d’être arrivés à ce stade dans le projet nous donne plus de confiance en nous- même et nous encourage à continuer, vu les problèmes qu’on a confrontés pour apprendre de nouveaux langages et outils de travail.

# BIBLIOGRAPHIE

[1]. <http://glossaire.infowebmaster.fr/html/>

[2]. <http://www.futura-sciences.com/>

[3]. Livre « PHP et MySQL Par Luke Welling, Laura Thomson »

[4]. <http://www.afci.fr/plan-cours.php?formation=javascript>

[5]. <https://fr.wikipedia.org/wiki/MySQL>

[6]. <https://openclassrooms.com/courses>

TABLE DES TABLEAUX

**Nom du tableau Page**

Tableau 1 : Ajouter un utilisateur 26

Tableau 2 : modifier un utilisateur 26

Tableau 3 : Supprimer un utilisateur 27

Tableau 4 : Consulter liste utilisateurs 27

Tableau 5 : Ajouter une classe 29

Tableau 6 : Modifier une classe 29

Tableau 7 : Supprimer une classe 30

Tableau 8 : Consulter une classe 30

Tableau 9 : Ajouter une spécialité 32

Tableau 10 : Modifier une spécialité 32

Tableau 11 : Supprimer une spécialité 33

Tableau 12 : Consulter une spécialité 34

Tableau 13 : Ajouter un module 35

Tableau 14 : Modifier un module 35

Tableau 15 : Supprimer un module 36

Tableau 16 : Consulter un module 36

Tableau 17 : Ajouter une matière 37

Tableau 18 : Ajouter une matière 38

Tableau 19 : Supprimer une matière 39

Tableau 20 : Consulter une matière 39

Tableau 21 : Ajouter un étudiant 40

Tableau 22 : Modifier un étudiant 41

Tableau 23 : Supprimer un étudiant 41

Tableau 24 : Consulter un étudiant 42

Tableau 25 : Ajouter une note 43

Tableau 26 : Modifier une note 44

Tableau 27 : Supprimer une note 44

Tableau 28 : Consulter une note 45

Tableau 29 : Enregistrer un résultat 46

Tableau 30 : Imprimer un résultat 47