|  |  |  |
| --- | --- | --- |
|  | **Université Mohammed Premier**  **Faculté des Sciences d’Oujda**  **Département d’Informatique** |  |

**Rapport de**

**Projet de Fin d’Études**

**Licence fondamentale**

**Sciences Mathématiques et Informatique**

**Conception et réalisation d’une application Web pour l’apprentissage de l’algorithmique et du langage C**

**Présenté par :**  **Encadré par :**

**Reda** **NAIM** **Mme. Khadija LAAROUSSI**

**Safae LAGHMOUCHI**

**Soutenu le : 24/06/2021, devant le jury :**

**Pr. Khadija LAAROUSSI**

**Pr. Mohammed MOUSSI**

**Année universitaire : 2020/2021**

# Dédicace

Premièrement nous remercions Dieu tout puissant de nous avoir aidé à mener à terme ce projet de fin d’études.

Nous dédions ce travail à nos parents qui sont pour nous une source de soutien et d’encouragement car sans leurs sacrifices, leurs tendresses et leur affection nous ne serions arriver jusqu’au bout. Nous nous réjouissons de cet amour filial.

Que Dieu vous garde afin que votre regard puisse suivre notre destinée.

A nos frères et nos sœurs qui ont été toujours présents à côté de nous.

A nos amis avec qui nous avons partagé les moments les plus agréables.

# Remerciements

*Je tiens tout d’abord à remercier Dieu le tout puissant et miséricordieux, qui m’a donné la force et la patience d’accomplir ce modeste travail.*

*En second lieu je présente notre profonde reconnaissance et nos sincères remerciements au Professeur Madame LAAROUSSI KHADIJA pour avoir accueillir et encadrer tout au long de ce travail et aussi pour la confiance totale, la patience ainsi que son temps et son aide qui a bien voulu nous accorder.*

*Nos vifs remerciements vont également aux membres du jury pour l’intérêt qu’ils ont porté à notre recherche en acceptant d’examiner notre travail et de l’enrichir par leurs propositions.*

*Nous remercions, également, tous nos professeurs pour leurs qualités scientifiques, leurs disponibilités et le savoir qu’ils nous ont transmis.*

*Merci à nos chers parents, à nos amis et à notre famille.*

*Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.*

# Résumé

Faisant partie de notre travail global pour cette année, notre document de recherche vise à présenter une synthèse compréhensive de notre application **“SOFT-ALGO”**.

**“SOFT-ALGO”** est une application web dont le but principal est de fournir des solutions aux utilisateurs, notamment les étudiants et les chercheurs dans le domaine de l’informatique, afin qu’ils puissent trouver aisément des formations en ligne centrées sur **l’algorithmique** et le langage de **programmation C**. En outre, l’application garantie la réduction de temps et des efforts en évitant au maximum les recherches inutiles et mal-organisées.

Le travail que nous présentons vise à expliquer les étapes que nous avons traversé ainsi que les résultats que nous avons obtenus.

**Mots-clés :** Application web sécurisée, algorithmique, langage C, formation en ligne, tests de niveau, compilation, exécution.

# Abstract

As part of our graduation project, the research paper is meant to present a comprehensive synthesis of our application **“SOFT-ALGO”**.

SOFT-ALGO: is a simplified application whose main objective is to provide solutions to users, especially students and researchers in the computer domain, so that they can easily find online formation centered around **Algorithmic** and **Programming on C** language. Furthermore, this application will grant an opportunity to reduce the time and efforts by avoiding useless or unorganized research.

The work we are presenting aims at explaining the stages we went through, and the results we have come up with.

**Keywords:** Secureweb application, algorithmic, C language, online formation, level tests, compilation, execution.

**Table des matières**

[Dédicace 2](#_Toc75151707)

[Remerciements 3](#_Toc75151708)

[Résumé 4](#_Toc75151709)

[Abstract 5](#_Toc75151710)

[Liste des figures 7](#_Toc75151711)

[Introduction générale 9](#_Toc75151712)

[Chapitre 1 : 11](#_Toc75151713)

[Contexte général du projet 11](#_Toc75151714)

[1.1 Introduction 12](#_Toc75151715)

[1.2 Présentation du contexte du projet 12](#_Toc75151716)

[1.3 Analyse et spécification des besoins 13](#_Toc75151717)

[1.3.1 Besoins fonctionnels 13](#_Toc75151718)

[1.3.2 Besoins non fonctionnels 14](#_Toc75151719)

[1.4 Conclusion 14](#_Toc75151720)

[Chapitre 2 : 15](#_Toc75151721)

[Étude Conceptuelle 15](#_Toc75151722)

[2.1 Introduction 16](#_Toc75151723)

[2.2 Langage UML 16](#_Toc75151724)

[2.3 Diagramme de cas d’utilisation 17](#_Toc75151725)

[2.4 Diagramme de séquence 19](#_Toc75151726)

[2.4.1 Authentification 19](#_Toc75151727)

[2.4.2. Création d’un nouveau compte 20](#_Toc75151728)

[2.4.3 Passer le test 21](#_Toc75151729)

[2.4.4 Gérer les questions 22](#_Toc75151730)

[2.4.5 Compiler et exécuter le code : 23](#_Toc75151731)

[2.5 Diagramme de classes 24](#_Toc75151732)

[2.6 Conclusion 25](#_Toc75151733)

[Chapitre 3 : Outils utilisés 26](#_Toc75151734)

[3.1 Introduction 27](#_Toc75151735)

[3.2 Langages de programmation 27](#_Toc75151736)

[3.2.1 HTML 27](#_Toc75151737)

[3.2.2 CSS 27](#_Toc75151738)

[3.2.3 PHP 7.4.9 28](#_Toc75151739)

[3.2.4 MySQL 28](#_Toc75151740)

[3.3 Logiciels utilisés 29](#_Toc75151741)

[3.3.1 WampServer 29](#_Toc75151742)

[3.3.2 Editeur de programmes 29](#_Toc75151743)

[3.3.3 Logiciel de modélisation 30](#_Toc75151744)

[3.4 Conclusion 30](#_Toc75151745)

[Chapitre 4 : Réalisation 31](#_Toc75151746)

[4.1 Introduction 32](#_Toc75151747)

[4.2 Description architecturale 32](#_Toc75151748)

[4.3 Présentation des interfaces de l’application 33](#_Toc75151749)

[4.3.1 Interface d’accueil 33](#_Toc75151750)

[4.3.2 Interface d’inscription 34](#_Toc75151751)

[4.3.3 Interface d’authentification 34](#_Toc75151752)

[4.3.4 Interface récupération du mot de passe 35](#_Toc75151753)

[4.3.5 Interface test préliminaire 36](#_Toc75151754)

[4.3.6 Interface cours d’apprentissage 37](#_Toc75151755)

[4.3.7 Interface test niveau 38](#_Toc75151756)

[4.3.8 Interface question 40](#_Toc75151757)

[4.3.9 Interface notification 40](#_Toc75151758)

[4.3.10 Interface compilation et exécution 41](#_Toc75151759)

[4.3.11 Côté administrateur 41](#_Toc75151760)

[4.4 Conclusion 42](#_Toc75151761)

[Conclusion générale 43](#_Toc75151762)

[Bibliographie 44](#_Toc75151763)

[Webographie 45](#_Toc75151764)

# 

# **Liste des figures**

[**Figure 1:** Diagramme de cas d'utilisation 18](#_Toc75099908)

[**Figure 2 :** Diagramme de séquence d'authentification. 19](#_Toc75099909)

[**Figure 3:** Diagramme de séquence pour la création d’un nouveau compte. 20](#_Toc75099910)

[**Figure 4:** Digramme de séquence pour passer les tests. 21](#_Toc75099911)

[**Figure 5:** Diagramme de séquence pour gérer les questions. 22](#_Toc75099912)

[**Figure 6:** Diagramme de séquence pour compiler et exécuter le code 23](#_Toc75099913)

[**Figure 7:** Diagramme de classe. 24](#_Toc75099914)

[**Figure 8:** Logo de HTML 26](#_Toc75099915)

[**Figure 9:** Logo de CSS 27](#_Toc75099916)

[**Figure 10:** Fonctionnement du modèle client/serveur 27](#_Toc75099917)

[**Figure 11:** Logo de WAMPSERVER 28](#_Toc75099918)

[**Figure 12:** Logo de Notepad 29](#_Toc75099919)

[**Figure 13:** Logo de Sublime 29](#_Toc75099920)

[**Figure 14:** Logo de EDraw Max 29](file:///C:\Users\Safae\Desktop\PFE-Profa\PFE_Finale%20(1).docx#_Toc75099921)

[**Figure 15:** Description architecturale (MVC) 32](#_Toc75099922)

[**Figure 16:** Interface d’accueil. 33](#_Toc75099923)

[**Figure 17:** Interface d'inscription. 33](#_Toc75099924)

[**Figure 18**:Interface d'authentification. 34](#_Toc75099925)

[**Figure 19:** Interface récupération du mot de passe. 34](#_Toc75099926)

[**Figure 20:** Interface mot de passe récupéré. 35](#_Toc75099927)

[**Figure 21:** Interface test préliminaire. 36](#_Toc75099928)

[**Figure 22:** Interface cours d'apprentissage de l'algorithmique. 36](#_Toc75099929)

[**Figure 23:** Interface cours d'apprentissage pour la programmation en C. 37](#_Toc75099930)

[**Figure 24:** Interface passage du cours vers le test. 37](#_Toc75099931)

[**Figure 25:** Interface test niveau pour l'algorithmique. 38](#_Toc75099932)

[**Figure 26:** Interface test niveau pour la programmation en C. 38](#_Toc75099933)

[**Figure 27:** Interface question. 39](#_Toc75099934)

[**Figure 28:** Interface notification. 39](#_Toc75099935)

[**Figure 29:** Interface compilation et exécution. 40](#_Toc75099936)

[**Figure 30:** Interface liste des utilisateurs. 41](#_Toc75099937)

[**Figure 31:** Interface liste des questions. 41](#_Toc75099938)

# **Introduction générale**

Le 21ème siècle est l’ère de la technologie par excellence. Tous les aspects de la vie humaine sur notre planète sont directement impactés par les technologies modernes. Aussi beaucoup de gens intègrent ce domaine de plus en plus, et pour d’innombrables raisons.

Parmi ces gens, il y’a ceux qui intègrent le domaine de l’informatique, ce qui très sensé si l’on tient compte des différentes débouchées de cette spécialité comme la création de site web, les applications mobiles ou encore le e-commerce.

Afin de bien se former dans ce domaine, il faut d’abord gravir les premières marches vers l’univers de la programmation. Ce qui nous amené à parler de la science des algorithmes et la technologie des langages de programmation (par exemple C le sujet en question).

Parmi les problèmes qui confrontent les informaticiens contre la meilleure métrisation des langages de programmation, on cite la perte du temps dans le procès de la recherche des formations précises qui enchainent l’algorithmique et le langage de programmation C, la nécessite de l’existence d’une fenêtre à partir d’elle on peut visualiser l’exécution des programmes.

L’algorithmique est sans doute la roue centrale qui fait tourner les langages de programmation au sein du monde du digital. Notre travail consiste donc à concevoir et mettre en œuvre une application web pour apprendre l’algorithmique avec ces interprétations en langage de programmation C, qui tient compte de tous les obstacles qui peuvent faire face à l’apprentissage d’un apprenant.

Nous avons conçu une application web sécurisée intitulée **« SOFT-ALGO »** dans le but de rendre la tâche plus légère avec l’auto-formation. À l’aide de cette application, l’apprenant aura la possibilité de faire une formation en ligne quand il veut et où il veut ainsi qu’il peut examiner son niveau à chaque processus d’enseignement. D’ailleurs pour les utilisateurs inscrits dans notre plateforme on a mis à leur disposition une page dont ils peuvent compiler leur propre code en langage C dans le but de rendre le processus plus pratique et plus compréhensif. De plus, ils peuvent poser leurs questions et leurs remarques dans une page séparée afin de former un échange d’informations entre les utilisateurs.

Ce mémoire est organisé en quatre chapitres comme suit :

Le **premier chapitre** présenteune étude préalable qui va permettre de placer le projet dans son cadre général, une description du projet et des spécifications des besoins.

Le **deuxième chapitre** est consacré à la présentation des outils utilisés pour la modélisation et la conception en se basant sur le langage de modélisation UML dont nous présenterons les diagrammes (cas d'utilisation, classes, séquences) avec leurs définitions et explications.

Le **troisième chapitre** décrit les outils logiciels nécessaires pour la réalisation de notre application web.

Le **quatrième et le dernier chapitre** est ciblé pour la représentation architecturale de notre application web. Ensuite, on va mettre en place un guide qui va expliquer l’utilisation de notre plateforme à travers quelques interfaces.

À la fin, on récapitule avec une conclusion générale qui déterminera quelques résultats souhaitant d’être achevés par l’application « **SOFT-ALGO »**.

|  |
| --- |
| Chapitre 1 :Contexte général du projet |

### 1.1 Introduction

Une étape essentielle de tout cycle de développement logiciel ou conceptuel consiste à effectuer une étude préalable. Le but de cette phase est de comprendre le contexte du système. Il s’agit d’éclaircir au mieux les besoins fonctionnels et non fonctionnels, apparaître les acteurs et identifier les cas d’utilisation.

Tout au long de ce chapitre, nous commencerons par la présentation du contexte de notre projet puis on va définir les besoins fonctionnels et les besoins non fonctionnels de notre projet.

### 1.2 Présentation du contexte du projet

L’application « SOFT-ALGO » est une simple application dont le but principal est d'aider les gens notamment les étudiants et les chercheurs dans le domaine d’informatique à trouver une formation en ligne facilement à comprendre en ce qui concerne l'apprentissage de l'algorithmique et le langage de programmation C tout en minimisant l’effort de faire de nombreuses recherches désordonnées.

Notre vision est de créer une application facile à utiliser qui pourra révolutionner le domaine de l’informatique au même temps la compléter dans le futur proche par d’autres outils, d’autres filiales, d’autres modules comme le développement web, et d’autres langages deprogrammation comme JavaScript qui appartenant au même secteur d’apprentissage. De plus on peut ajouter les réseaux informatiques et les bases de données.

Ce genre des applications réduisent le temps de recherche et augmentent l'assiduité au sein des cours virtuelles précises, en fait, l’utilisateur n’aura plus besoin d’une formation qui coûte plus chère pour trouver un professeur qui va lui apporter l'enseignement.

Cette application offrira à l’éduqué le pouvoir de trouver une classe en ligne à proximité facilement à comprendre, une diversité de cours pour l’algorithmique et le langage C, et la possibilité de se mettre en contact facilement avec un système éducatif complet contient aussi les tests pour examiner le niveau scolaire de chaque utilisateur.

L’application permettra aussi à l’éduqué de compiler et d’exécuter des exercices pratiques via le langage C, augmenter le taux de maîtrise de la programmation en C, l’intégration facile des débutants en algorithmique, et la possibilité de développer un niveau assez riche pour mettre en œuvre des programmes bien structurés qui répondent au besoin de notre vie quotidienne.

L’utilité de notre application web réside dans le fait qu’elle fusionne le côté théorique (algorithmique) et le côté pratique (programmation en c), de plus elle fournit une plateforme de discussion entre les utilisateurs afin de gérer leurs compétences et éliminer l’ambiguïté autour d’un sujet de l’application.

### 1.3 Analyse et spécification des besoins

La phase d’analyse et spécification des besoins présente une étape primordiale dans le cycle de développement d’un projet dans laquelle nous allons identifier les besoins de notre application.

Nous allons mettre l'accent sur les besoins fonctionnels qui présentent les fonctionnalités attendues de notre application ainsi que les besoins non fonctionnels pour éviter le développement d’une application non satisfaisante.

#### 1.3.1 Besoins fonctionnels

Les besoins fonctionnels expriment les principales fonctionnalités de l'application sans se préoccuper de la façon de l'implémentation. Ces besoins nous conduisent à l'élaboration des modèles de cas d’utilisation.

L’application web que nous envisageons développer permettre à son utilisateur de :

* **S’inscrire *:*** Créer un compte et saisir les informations personnelles.
* **S’authentifier *:*** Se connecter par son email et un mot de passe.
* **Faire le test préliminaire *:*** l'utilisateur qui visite l'application pour la première fois doitexaminer son niveau.
* **Choisir le cours**: l'utilisateur peut choisir le cours qui va avec son niveau.
* **Compiler et exécuter les codes :** chaque utilisateur peut entrer son propre code en langage C pour être compilé et exécuté.
* **Tester le niveau :** après la lecture du cours l'utilisateur doit tester le niveau actuel qui a eu pendant la phase d'acquisition et d’apprentissage.
* **Poster les questions :** l’utilisateur peut poser des questions autour de l’algorithmique ou bien du langage C, en parallèle les autres utilisateurs peuvent répondre au commentaire en question.
* **Recevoir les notifications :** quand l’utilisateur poste une question les autres utilisateurs reçoivent des notifications.
* **Recherche :** dans la page de questions chaque utilisateur peut rechercher sur des anciennes questions posées par d’autres utilisateurs dans un contexte précis.

#### 1.3.2 Besoins non fonctionnels

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement, ses exigences techniques sont souvent exprimées sous forme d’objectifs spécifiques que doit atteindre le système.

Nous décrivons les besoins non fonctionnels de notre application comme suit :

* **La disponibilité :** l’application doit être disponible pour être utilisé par n’importe quel utilisateur.
* **La sécurité de l’accès aux informations critiques :**L’application devra être hautement sécurisée, les informations ne devront pas être accessibles à tout le monde, c'est-à-dire que l'application est accessible par un identifiant (email) et un mot de passe attribué à chaque utilisateur de l’application.
* **La fiabilité :** les données fournies par l’application doivent être fiables.
* **La convivialité de l’interface graphique :** l’application doit fournir une interface conviviale et simple pour tout type d’utilisateur car elle présente le premier contact de l’utilisateur avec l’application et par le biais de celle-ci on découvrira ses fonctionnalités.
* **Le design :** l’apparence qui représente virtuellement l’ambiance, l’atmosphère et la qualité de la formation comme s’il s’agissait d’une formation réelle. Il faut accorder une grande importance à la phase de design en essayant d’utiliser un design professionnel et attrayant.
* **Une solution ouverte et évoluée :** l’application peut être améliorée par l’ajout d’autres modules pour garantir la souplesse, l’évolutivité et l’ouverture de la solution.

### 1.4 Conclusion

Dans ce premier chapitre, nous avons essayé de donner une vision générale sur notre projet. Nous avons défini les acteurs qui interagissent avec l’application, les besoins fonctionnels et les besoins non-fonctionnels afin qu’on puisse passer à la modélisation et la conception de notre application.

|  |
| --- |
| Chapitre 2 :Étude Conceptuelle |

### 2.1 Introduction

Nous présenterons dans ce chapitre une étude conceptuelle où nous commencerons par la définition de quelques notions de langages de modélisation, ensuite nous identifierons les acteurs du système et nous détaillerons les diagrammes des cas d'utilisation, de classes et de séquences.

### 2.2 Langage UML

Le langage **UML** « Langage de Modélisation Unifié », partant du terme anglais « Unified Modeling Language », est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. Il est utilisé pour spécifier et concevoir un logiciel.

Dans la spécification, 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 via des classes, qui n’intéressent pas l’utilisateur du logiciel mais seulement ses concepteurs. Il comporte des modèles des classes, des états et d’interaction. UML est également utilisé dans les phases terminales du développement avec des modèles de réalisation [1]. Il permet de définir un modèle à partir des diagrammes, les trois principaux diagrammes sont :

* **Diagramme de cas d’utilisation (use case)**

Les diagrammes de cas d’utilisation permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Généralement, dans le diagramme de cas d’utilisation on définit les différents utilisateurs (acteurs) et les différentes interactions. Il faut répondre aux questions suivantes :

* Quels sont les utilisateurs du système ?
* Quelles sont leurs interactions avec le système ?
* **Diagramme de classes**

Le diagramme de classes exprime la structure statique du système en termes de classes et des relations entre ces classes. L’intérêt du diagramme de classe est de modéliser les entités du système d’information, et il permet de représenter l’ensemble des informations analysées qui sont gérées par le domaine. Ces informations sont structurées, c’est-à-dire qu’elles sont regroupées sous forme des classes.

* **Diagramme de séquences**

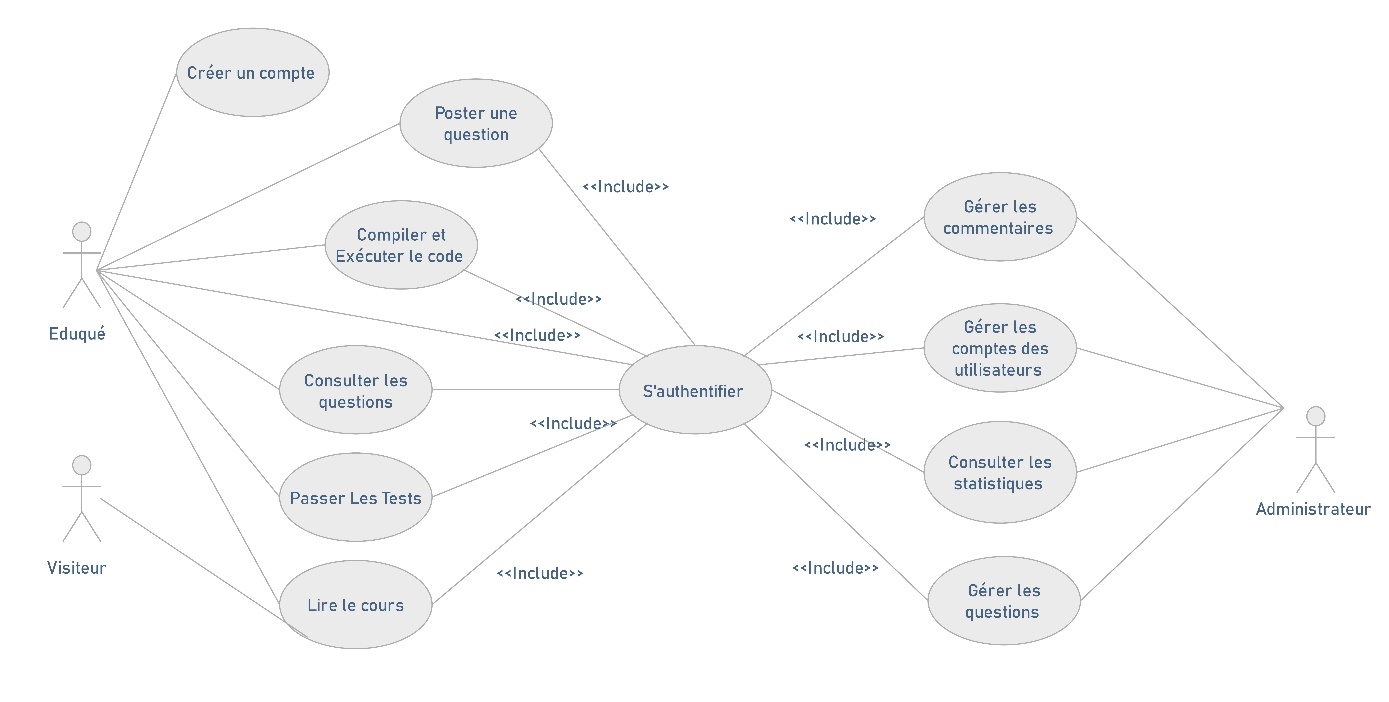
Le diagramme de séquence permet de visualiser les messages par une lecture de haut en bas. L’axe vertical représente le temps, l’axe horizontal les objets convenables. Une ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie.

### 2.3 Diagramme de cas d’utilisation

Les acteurs sont des entités externes qui interagissent directement avec l'application web et communiquent avec cette dernière en envoyant et en recevant des messages, les acteurs qui interagissent avec le système sont :

* **Administrateur :** Une personne ayant des privilèges d’administration, il possède un compte administrateur par lequel il peut ajouter des cours ou des tests dans le système d’information. Les fonctions principales de l’administrateur sont :
* **Gérer les comptes des utilisateurs :** activer, désactiver, supprimer, éditer et modifier les comptes.
* **Gérer les questions :** vérifier si une question respecte les conditions de l’application SOFT-ALGO.
* **Consulter les statistiques :** les statistiques présentent la liste des utilisateurs authentifiés avec leurs informations (nom, prénom, email…), et les questions qui sont encore de traitement afin d’être publié sur la page des questions.
* **Gérer les commentaires** **:** les commentaires qui ne respectent pas les conditions seront supprimés par l’admin.
* **Visiteur :** une personne qui n’est pas membre du site ou qu’il souhaite le devenir. La seule fonction d’un visiteur est :
* **Accéder aux cours** :le visiteur peut visualiser les différents cours disponibles dans le site, mais il ne peut pas passer les tests.
* **Utilisateur authentifié** : une personne qui possède un compte utilisateur dans le système d’information, il bénéfice de tous les privilèges d’un visiteur, en plus il peut passer des tests, poster des questions, reçoive des notifications en plus compiler et exécuter un code source. Les fonctionsprincipales de l’utilisateur sont :
* **Créer un compte** : chaque utilisateur pour qu’il puisse passer les tests il doit créer un compte. La création d’un compte se fait par le remplissage des champs demandés et après la création du compte l’utilisateur devient un utilisateur authentifié**.**
* **S’authentifier :** Cliquer sur le bouton “Se connecter”, l’application va nous rediriger vers une nouvelle page qui demande à l’utilisateur l’email ou nom complet et mot de passe, pour identifier la connexion, après il va se rediriger vers sa page personnelle.
* **Passer les tests** : chaque utilisateur peut passer un test qui s’appelle Test de Niveau, ce test s’effectue si l’utilisateur lit le cours correspondant au niveau en question. Si l’utilisateur réponde à toutes les questions correctement (Note Obtenue 75%), donc on indique le Niveau est validé, puis le niveau suivant va être accessible par l’utilisateur, (dans ce cas l’utilisateur peut accéder aux niveaux validés pour récupérer les infos). Dans le cas contraire, si la Note est inférieure à 75%, la lecture du cours sera mise en œuvre une deuxième fois pour répéter le test.
* **Poster une question :** Parfois les utilisateurs désirent poser des questions en ce qui concerne notre sujet en question “Algorithmique et programmation en C”. Donc pour faire face à ce problème on met à la disposition de notre utilisateur une interface ou ils peuvent ajouter des commentaires ou des recommandations.
* **Compiler et exécuter le code** : chaque utilisateur inscrit dans la plateforme peut entrer son propre code et le compiler après l’exécuter.
* **Consulter les questions :** comme l’utilisateur peut poser des questions sous forme de commentaires, il peut aussi accéder à la page des questions, et répondre à des questions qui sont déjà posées par d’autres utilisateurs.

La figure 1 présente le diagramme de cas d’utilisation de notre application :



**Figure 1:** Diagramme de cas d'utilisation.

### 2.4 Diagramme de séquence

#### 2.4.1 Authentification

L’authentification permet à l’utilisateur authentifié d’accéder à son compte en suivant les étapes suivantes :

* Cliquez sur le bouton connexion.
* Ouvrir la page de connexion.
* Remplir les champs demandés.
* Vérifier si les données sont correctes (SGBD).
* Données correctes : Accéder à la page principale.
* Données incorrectes : Rediriger vers la page de connexion jusqu'à l’utilisateur entre les données correctes.

La figure 2 présente le diagramme de séquence pour l’authentification :

Une image contenant table

Description générée automatiquement

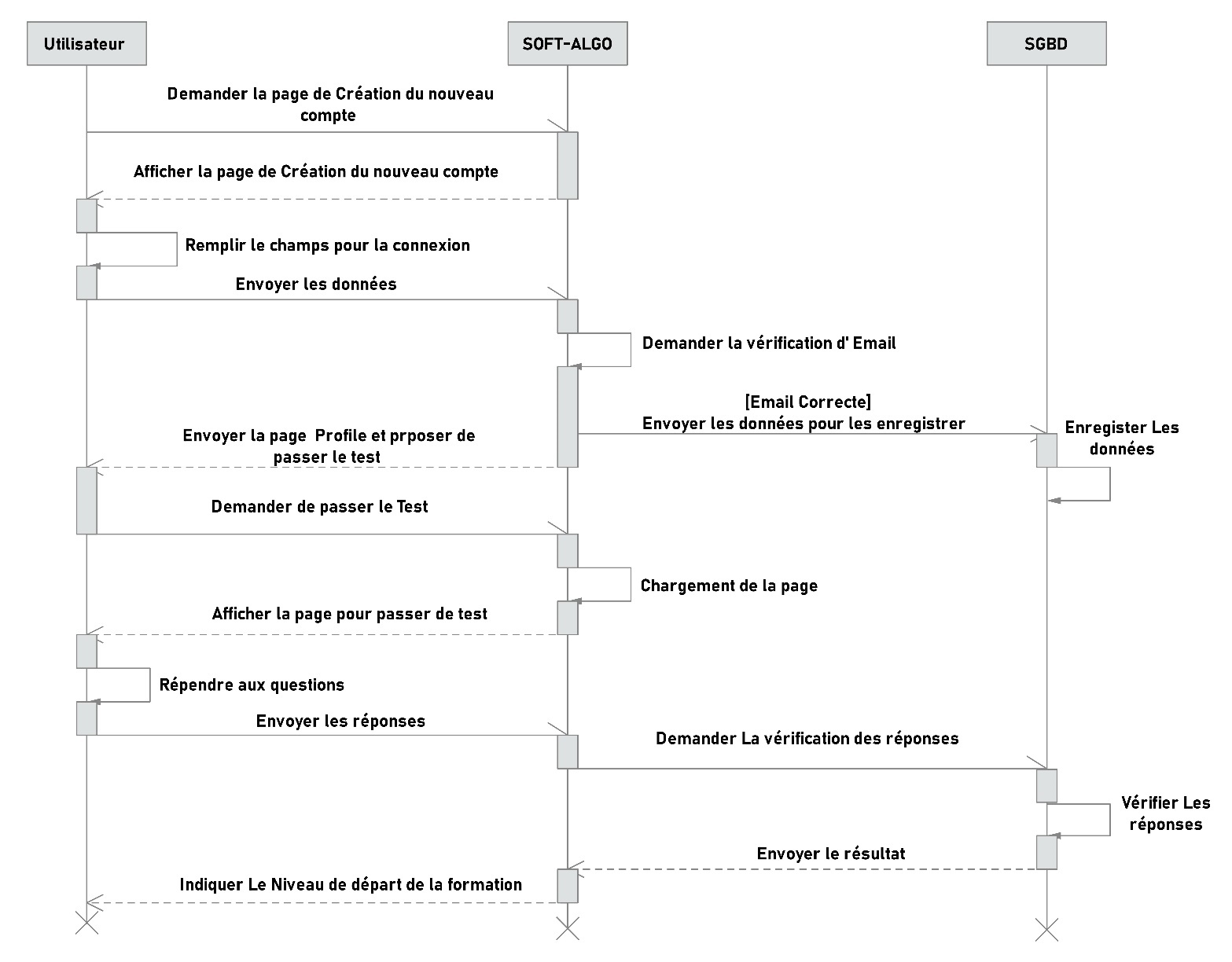
**Figure 2 :** Diagramme de séquence d'authentification.

#### 2.4.2. Création d’un nouveau compte

L’utilisateur qui veut s’inscrire dans l’application, il doit créer un nouveau compte d’après le cursus suivant :

* Cliquer sur bouton créer nouveau compte.
* Ouvrir la page de création de Compte.
* Remplir les champs demandés.
* Vérifier l’email.
* Ouvrir la page Profile.
* Commencer l’apprentissage.

La figure 3 présente le diagramme de séquence pour la création d’un nouveau compte :



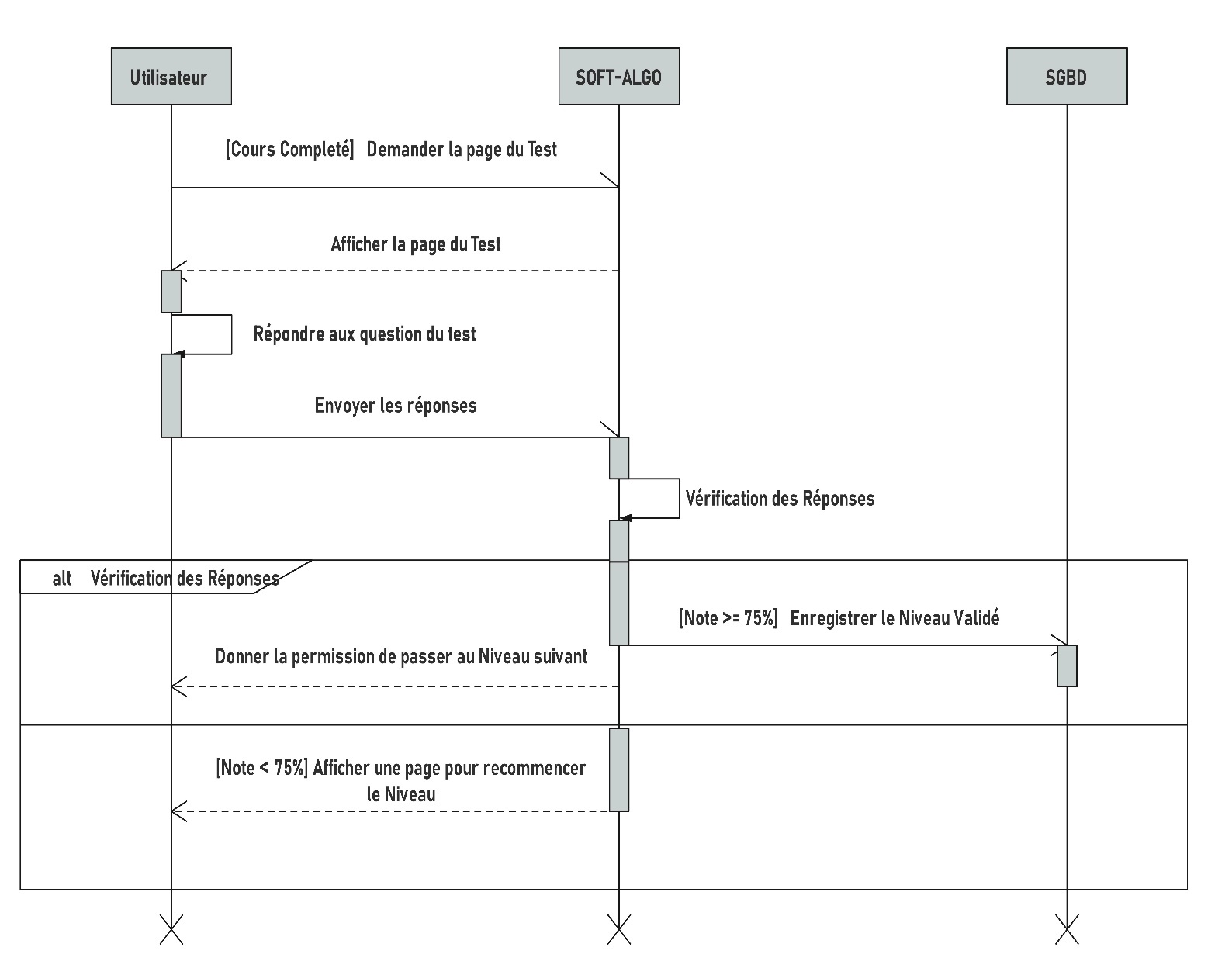
**Figure 3:** Diagramme de séquence pour la création d’un nouveau compte.

#### 2.4.3 Passer le test

L’utilisateur qui fait déjà l’inscription, il peut passer les tests en suivant les étapes ci-dessous :

* Cliquer sur le bouton Passer Test.
* Répondre à chaque question, choisir la réponse correcte.
* Cliquer sur le bouton Consulter pour vérification les réponses.
* Valider les réponses :
* Si les réponses sont correctes, ouvrir la page du Niveau suivant.
* Si plusieurs réponses sont incorrectes, refaire le test.

La figure 4 montre le diagramme de séquence pour passer les tests :



**Figure 4:** Digramme de séquence pour passer les tests.

#### 2.4.4 Gérer les questions

L’utilisateur authentifié a le droit de poser des questions en ce qui concerne le sujet en question de l’application selon les critères suivants :

* Cliquer sur le bouton Questions.
* Ouvrir la page de questions.
* Publier des questions :
* Cliquer sur le bouton Ajouter une question.
* Remplir les champs demandés.
* Lancer les questions.
* Consulter les questions précédentes à l’aide d’une recherche manuelle sur la page des questions.

La figure 5 présente le diagramme de séquence pour gérer les questions :

Une image contenant table

Description générée automatiquement

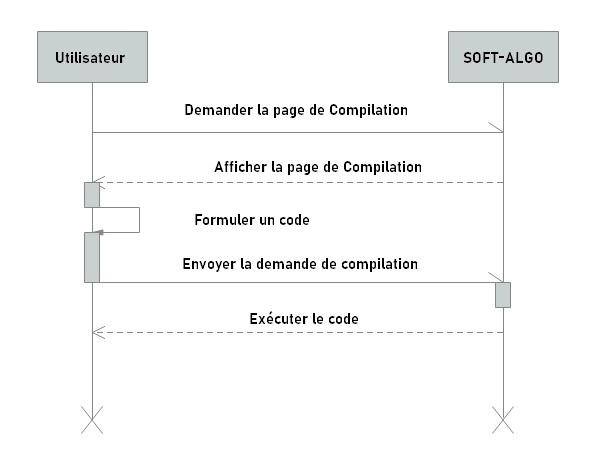
**Figure 5:** Diagramme de séquence pour gérer les questions.

#### 2.4.5 Compiler et exécuter le code :

Les fonctions compiler et exécuter sert à faire la compilation et l’exécution du code source selon les étapes suivantes :

* Cliquer sur le bouton Compilation et exécution.
* Ouvrir la page de compilation et exécution.
* Taper les instructions du programme.
* Cliquer sur le bouton Run.

La figure 6 montre le diagramme de séquence pour compiler et exécuter le code :



**Figure 6:** Diagramme de séquence pour compiler et exécuter le code.

### 2.5 Diagramme de classes

Les diagrammes de classes sont des schémas utilisés en génie logiciel pour afficher les classes et les interfaces du système

Dans notre diagramme de classe, on a :

**Administrateur :**

Attributs : id - MotDePass.

**Utilisateur :**

Attributs : Id -MotDePass \_ Nom – Prenom – Email – Tel – Niveau Scolaire – DateNaissance – NiveauActuel.

**Question :**

Attributs : IdPub – idQuestion – NombreReponse – Titre – Contenu – DatePublier

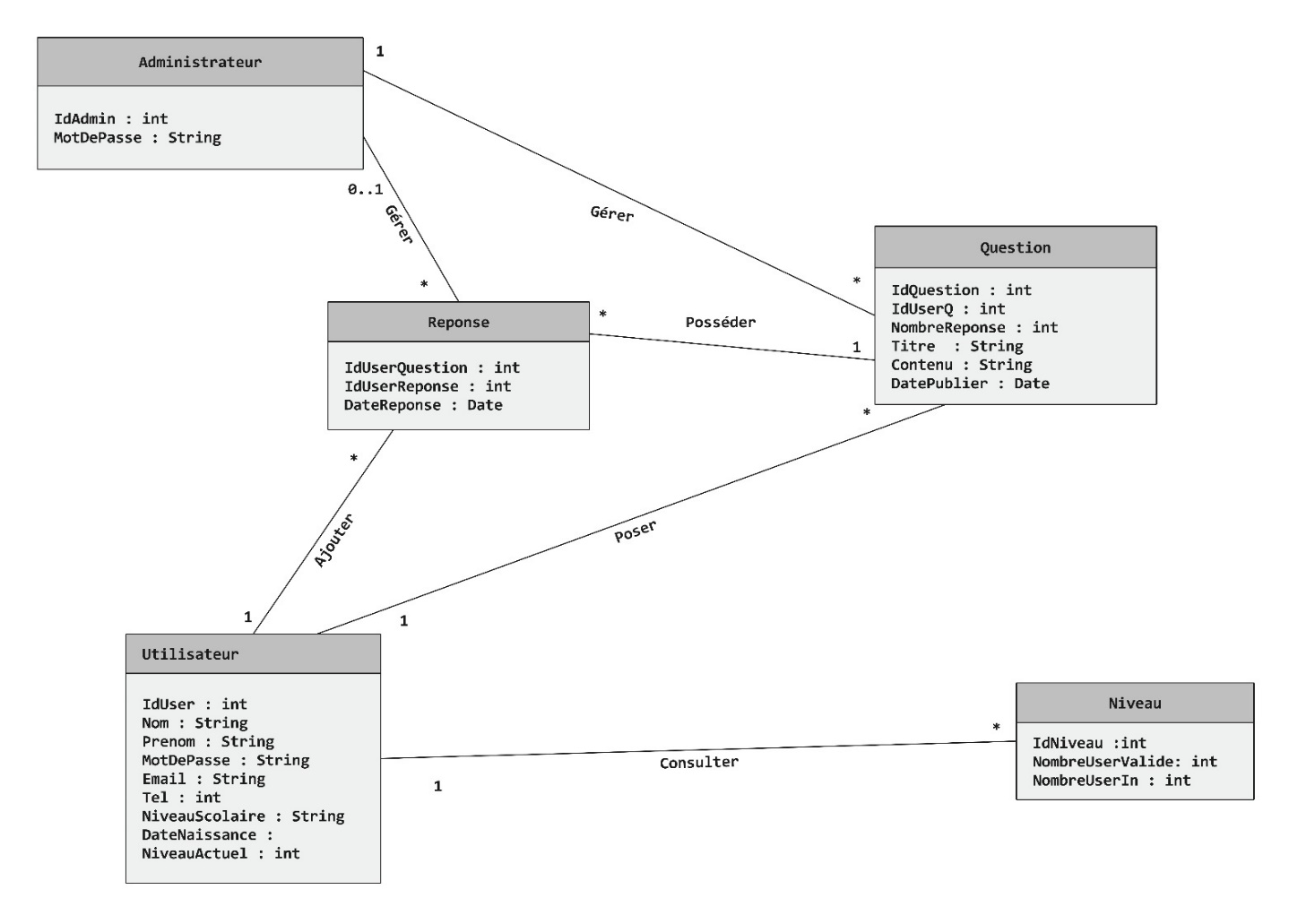
**Commentaire :**

Attributs : IdUser – Date-Commenter.

**Niveau :**

Attributs : IdNiveau – NombreUserValide – NombreUserIn.

La figure 7 présente le diagramme de classe :



**Figure 7:** Diagramme de classe.

### 2.6 Conclusion

La phase de conception avait pour intérêt de présenter les différentes étapes de conception de l’application tout en évoluant au niveau des détails, doit par conséquent aboutir immédiatement à l’implémentation avec une vision claire des aspects fonctionnels ainsi que organisationnels de l’application.

|  |
| --- |
| Chapitre 3 : Outils utilisés |

### 3.1 Introduction

Le **Web** est le terme communément employé pour parler du World Wide Web, ou WWW, traduit en français par la toile d'araignée mondiale. Il fait référence au système [hypertexte](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203435-lien-hypertexte-definition-traduction/) fonctionnant sur le réseau informatique mondial Internet. Par abus de langage, le Web désigne de façon plus large tout ce qui se rapproche à cet univers internet. On ne fait plus aujourd'hui la distinction technique entre ce que définit le Web et ce que définit Internet. Dans ce chapitre, nous allons citer les technologies utilisées pour la réalisation de notre application web.

### 3.2 Langages de programmation

#### 3.2.1 HTML

Le langage HTML « HyperText Markup Language » généralement abrège HTML, est le format de données conçu pour représenter les pages web. C’est un langage de balisage permettant décrire de l’hypertexte, d’où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l’accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS) [2].



**Figure 8:** Logo de HTML.

#### 3.2.2 CSS

Le langage CSSpour« Cascading Style Sheets » est un langage de feuille de style utilisé afin de décrire la présentation d’un document écrit en HTML ou en XML.CSS décrit la façon dont les éléments doivent être afficher, à l’écran, sur du papier ou sur autre support.CSS est l’un des langages principaux du Web et a été standardise par le W3C.Ce standard évolue sous forme de niveaux(Levels), CSS1 est désormais considéré comme obsolète, CSS2.1 correspond à la recommandation et CSS3,qui est découpe en module plus petits est en voie de standardisation [3].

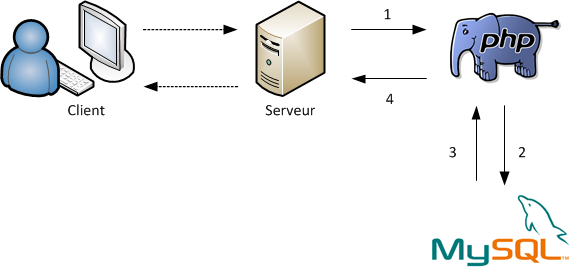


**Figure 9:** Logo de CSS

#### 3.2.3 PHP 7.4.9

PHP (officiellement, ce sigle est un acronyme récursif pour HyperText Préprocesseur) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d’applications web. Il peut être intègre facilement au HTML. Ce qui distingue PHP des langages de script comme le JavaScript, est que le code est exécuté sur le serveur, génèrent ainsi le HTML, qui sera ensuite envoyé au client. Le client ne reçoit que le résultat du script, sans aucun moyen d’avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu’il analyse tous vos fichiers HTML colle des fichiers PHP. Ainsi, il n’y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques [4].

#### 3.2.4 MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) basé sur SQL (Structures Query Language). MySQL fonctionne sur pratiquement toutes les plates-formes, y compris Linux, Unix et Windows. Il est entièrement multithread avec un noyau de threads, et fournit des API (Application Programming Interface) pour de nombreux langages de programmation, notamment C, C++, Eiffel, Java, Perl, PHP, Python, et Tcl. MySQL est utilisé dans une large gamme d’applications, le commerce électronique, les bases de données Web… [5]. La figure 10 présente le fonctionnement entre le client et le serveur.

**Figure 10:** Fonctionnement du modèle client/serveur.

Les étapes suivantes expliquent comment le fonctionnement se déroule entre le client et le serveur :

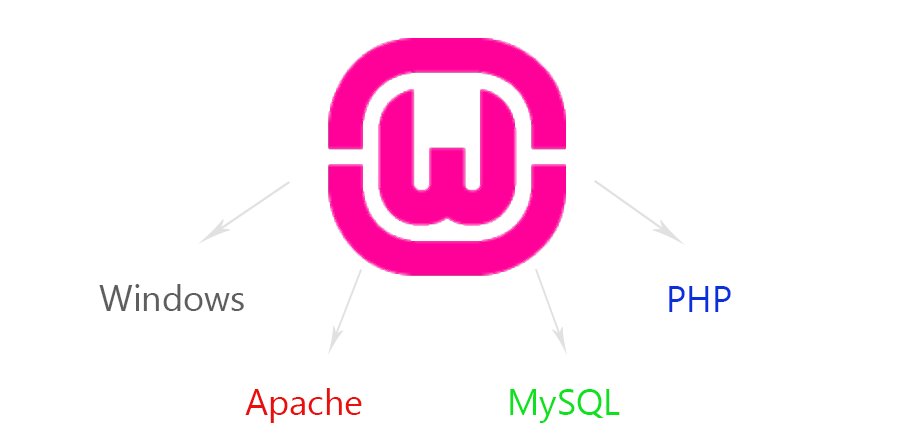
* Envoi du script PHP.
* Traitement de requête SQL par MYSQL.
* Renvoi le résultat de la requête à PHP.
* Transmission et traitement PHP du résultat.

### 3.3 Logiciels utilisés

#### 3.3.1 WampServer

**WampServer** est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l’aide du serveur Apache2, du langage de scripts PHP et d’une base de données MYSQL. Il possède également PHPMYADMIN pour gérer plus facilement les bases de données.

WampServer réside dans la possibilité d’y installer et d’utiliser n’importe quelle version de PHP, Apache ou MySQL.



**Figure 11:** Logo de WAMPSERVER.

#### 3.3.2 Editeur de programmes

Nous avons travaillé avec les éditeurs de texte **Notepad++** et **Sublime texte**. Ce sont des programmes conçus pour l’édition de code source et ils sont compatibles avec plusieurs langages de programmation.

|  |  |
| --- | --- |
|  | Sublime Text 3 License Key (2020) | Free Source Code |
| **Figure 12:** Logo de Notepad. | **Figure 13:** Logo de Sublime. |

### 3.3.3 Logiciel de modélisation

 **EDraw Max** est une solution logicielle permettant de réaliser des diagrammes, des graphiques et des dessins professionnels pour de nombreux besoins. Les développeurs, commerciaux et ingénieurs pourront l'utiliser pour concevoir des diagrammes de flux, des cartes mentales détaillées, des dessins électriques ou encore des plans pour les architectes.

**Figure 14:** Logo de EDraw Max.

### 3.4 Conclusion

Dans ce chapitre, nous visons à braquer le projecteur sur les langages de programmation et les logiciels utilisés pour développer notre application web. Le choix de ces technologies est venu de fait que notre formation est basée sur la plupart de ces langages, de plus la recherche d’information pour leur manipulation est pertinente à partir du web.

|  |
| --- |
| Chapitre 4 : Réalisation |

### 4.1 Introduction

Ce chapitre constitue le dernier volet de ce rapport. Il a pour objectif d’exposer le travail réalisé, nous présentons, en premier lieu la structure architecturale de notre site. En second lieu, nous élaborons une présentation des différentes interfaces développées dans notre application.

### 4.2 Description architecturale

**<<Le pattern modèle-vue-contrôleur** (en abrégé **MVC**), une architecture orientée autour de trois pôles : le modèle, la vue et le contrôleur. L'architecture MVC, très répandue dans les Framework du web. Son objectif est de séparer les responsabilités de chaque pôle afin que chacun se concentre sur ses tâches**.>>**

Cette architecture adopte les définitions de **Modèle, Vue** et **Contrôleur** :

* **La couche Modèle :**

Elle représente la partie qu’est responsable de récupérer les données, et de les convertir selon les concepts chargés de notre application, tels que le traitement, la validation…L’objet Model peut être vu comme la première couche d’interaction avec n’importe quelle base de données que nous pourrions utiliser pour notre application.

* **La couche Vue :**

Elle représente l’interface utilisateur. Elle se concentre simplement sur l’affichage des données fournit par le modèle sous format HTML, XHTML…

* **La couche Contrôleur :**

Elle permet de faire le lien entre la vue et le modèle lorsqu’une action de l’utilisateur intervienne sur la vue. Le contrôleur on peut le considérer comme un manager qui gère toutes les ressources souhaitées pour accomplir une tâche. Il attend les requêtes des clients, vérifie leur validité selon l’authentification, les règles d’autorisation, et délèguent les données traitées par le model.

La figure suivante explique l’interaction entre les différentes couches du model MVC :



**Figure 15:** Description architecturale (MVC).

### 4.3 Présentation des interfaces de l’application

Dans cette partie, nous allons présenter quelques cas d’utilisations de notre application, sous forme d’un guide utilisateur.

#### 4.3.1 Interface d’accueil

Pour utiliser cette application, il faut d’abord s’inscrire, si vous êtes un nouvel utilisateur ou authentifié, si vous êtes un membre, ou bien accéder au cours direct si vous êtes un utilisateur non inscrit.

Cette interface vous permet de choisir le bouton qui convient à votre cas nouveau utilisateur (Inscription) ou utilisateur membre (Connexion) ou bien utilisateur non inscrit (Visiteur). La figure 16 présente l’interface d’accueil de l’application.



**Figure 16:** Interface d’accueil.

#### 4.3.2 Interface d’inscription

C’est l’interface responsable de l’inscription des nouveaux utilisateurs, l’utilisateur doit remplir ses informations puis cliquer sur « **Enregistrer** », si toutes les coordonnées sont correctes le système passe au menu principal de l’application.

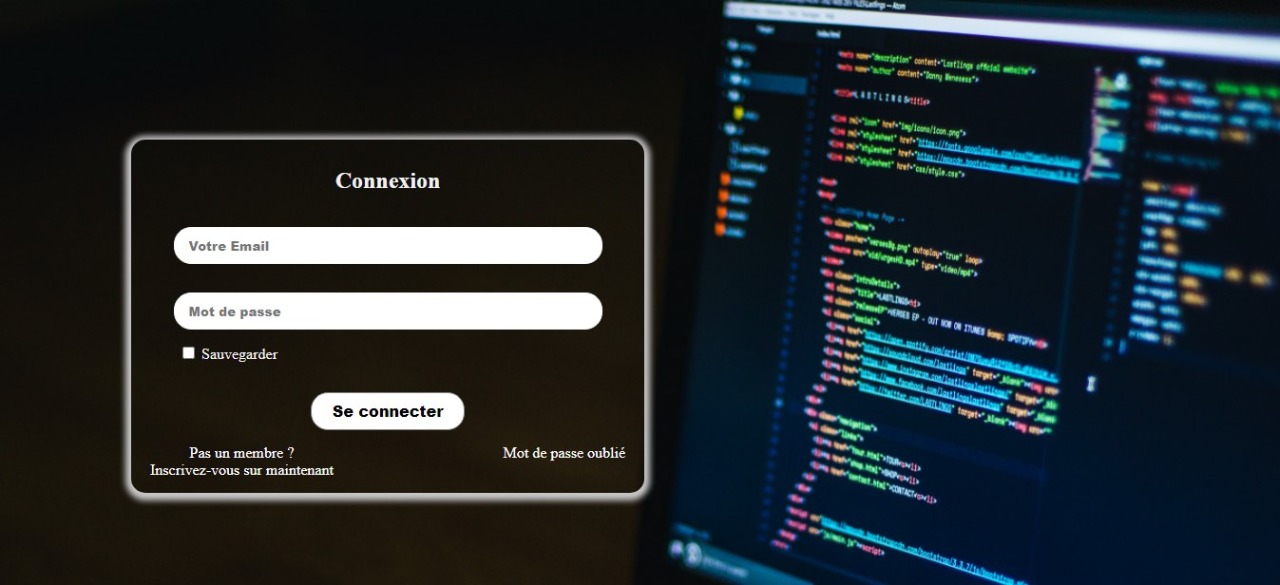
Une image contenant texte, moniteur, capture d’écran

Description générée automatiquement

**Figure 17:** Interface d'inscription.

#### 4.3.3 Interface d’authentification

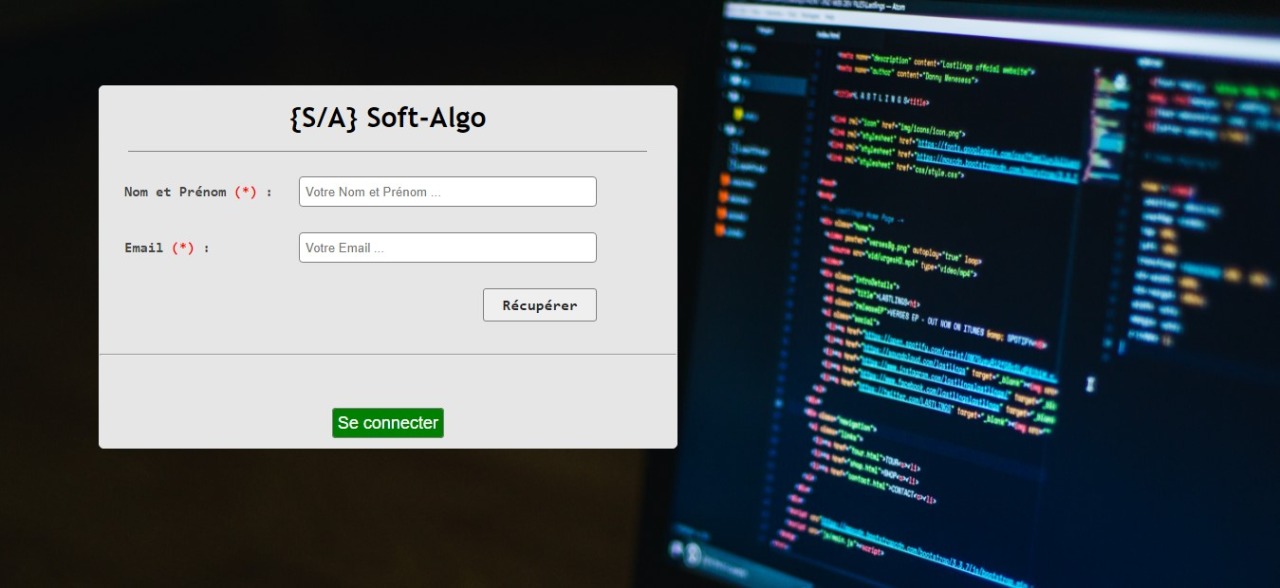
L’utilisateur doit introduire son email et son mot de passe dans les champs correspondants et cliquer sur le bouton « Se connecter », le système vérifie les données entrées. En cas d’échec, il réaffiche un message d’erreur. Si le Login et le mot de passe sont valides, le système affiche le menu de l’application, dont on trouve la liste des cours avec une barre des tâches qui contient les cases suivantes « Question », « Compilation et exécution » et « Notification ». Si l’utilisateur oublie son mot de passe, il doit cliquer sur l’icône « mot de passe oublié » pour récupérer son mot de passe. Sinon, il va être rediriger vers la page d’inscription via l’icône « Pas un membre ? Inscrivez-vous maintenant ».



**Figure 18**:Interface d'authentification.

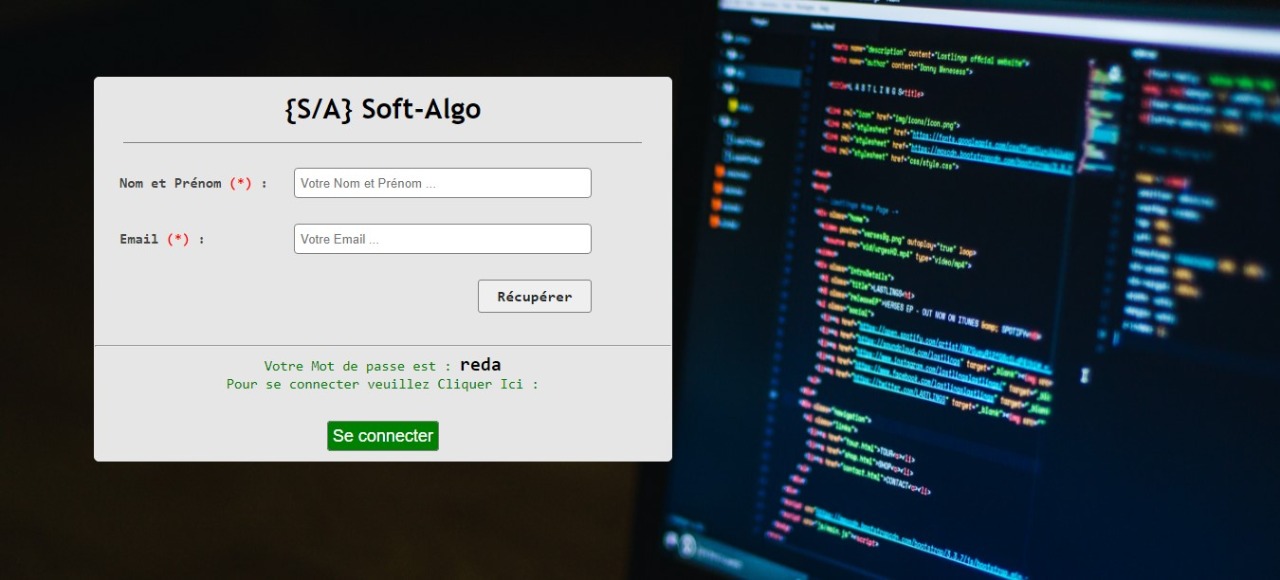
#### 4.3.4 Interface récupération du mot de passe

L’utilisateur authentifié peut récupérer son mot de passe s’il l’oublie lors de la tentation d’accéder à son compte par remplissage des champs qui convient avec le nom, le prénom et l’email de l’utilisateur.



**Figure 19:** Interface récupération du mot de passe.

Quand l’utilisateur entre ses informations d’une manière correcte l’admin traite la demande, ensuite, il envoie le mot de passe sous forme d’un petit message au-dessous de la page de récupération.



**Figure 20:** Interface mot de passe récupéré.

#### 4.3.5 Interface test préliminaire

L’utilisateur qui veut entamer la formation en ligne doit passer le test niveau pour se diriger vers le cours qui va avec ses compétences, sinon l’utilisateur doit cliquer sur le bouton

« Formation complète » pour accéder à la formation dès le premier niveau.

Pour valider le test préliminaire, l’utilisateur est mené à répondre aux questionnaires qui se compose d’un ensemble de 18 questions selon l’ordre suivant :

* Deux questions sur la syntaxe de l’algorithme.
* Trois questions sur la déclaration des variables.
* Trois questions sur les entrées-sorties.
* Trois questions sur les conditions.
* Trois questions sur les boucles.
* Trois questions sur les tableaux.



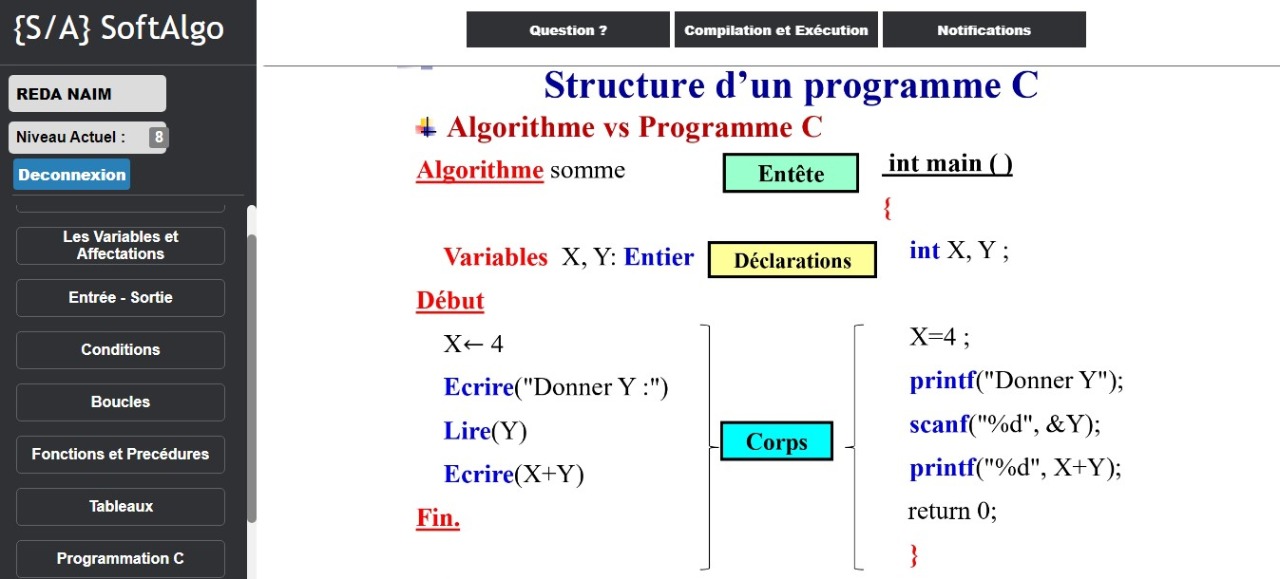
**Figure 21:** Interface test préliminaire.

#### 4.3.6 Interface cours d’apprentissage

Une image contenant texte

Description générée automatiquementPour chaque utilisateur inscrit doit accéder au cours (sous forme des images) qui convient avec le résultat fourni par le test préliminaire, afin de terminer la phase d’apprentissage pour passer au test, sinon pour les utilisateurs non-inscrits peuvent juste lire le cours sans faire aucun test.

**Figure 22:** Interface cours d'apprentissage de l'algorithmique*.*



**Figure 23:** Interface cours d'apprentissage pour la programmation en C.

#### 4.3.7 Interface test niveau

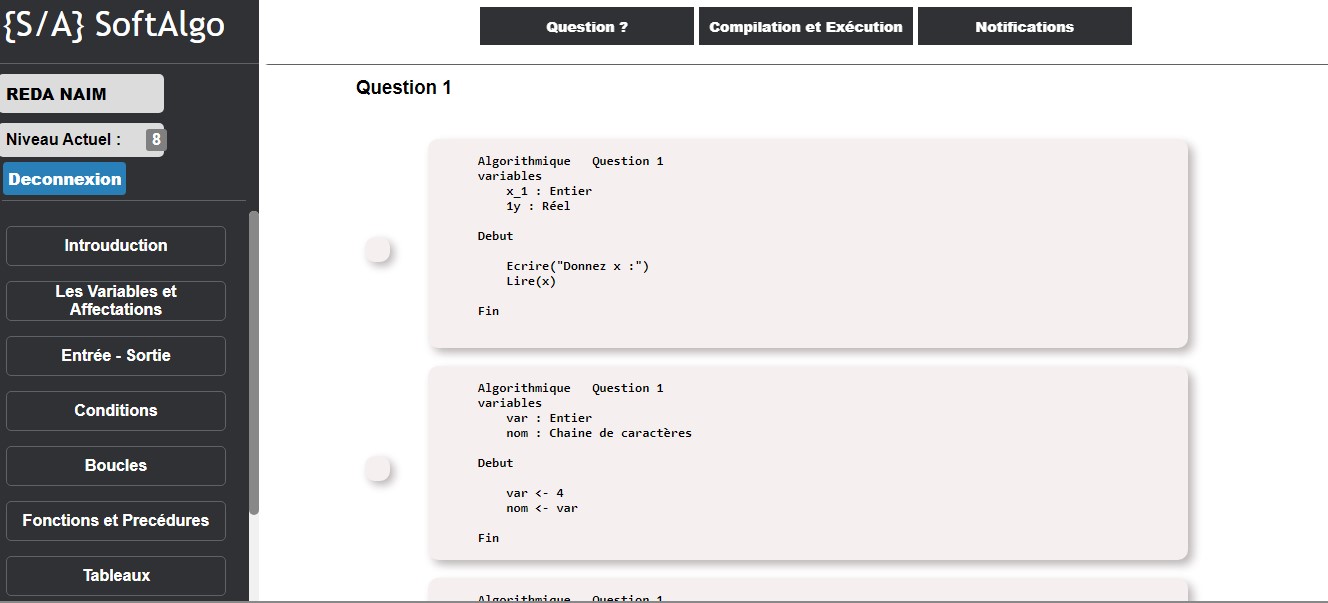
Les utilisateurs inscrits peuvent répondre aux questions de test niveau sous forme d’un QCM. Si un pourcentage de 75% des réponses est correcte, l’utilisateur doit accéder au cours du niveau suivant sinon il va refaire le test.

Quand l’utilisateur authentifié termine la phase de la lecture du cours, il va être rediriger vers la page du test du niveau en question par une simple clique sur le bouton « Passer au test ».

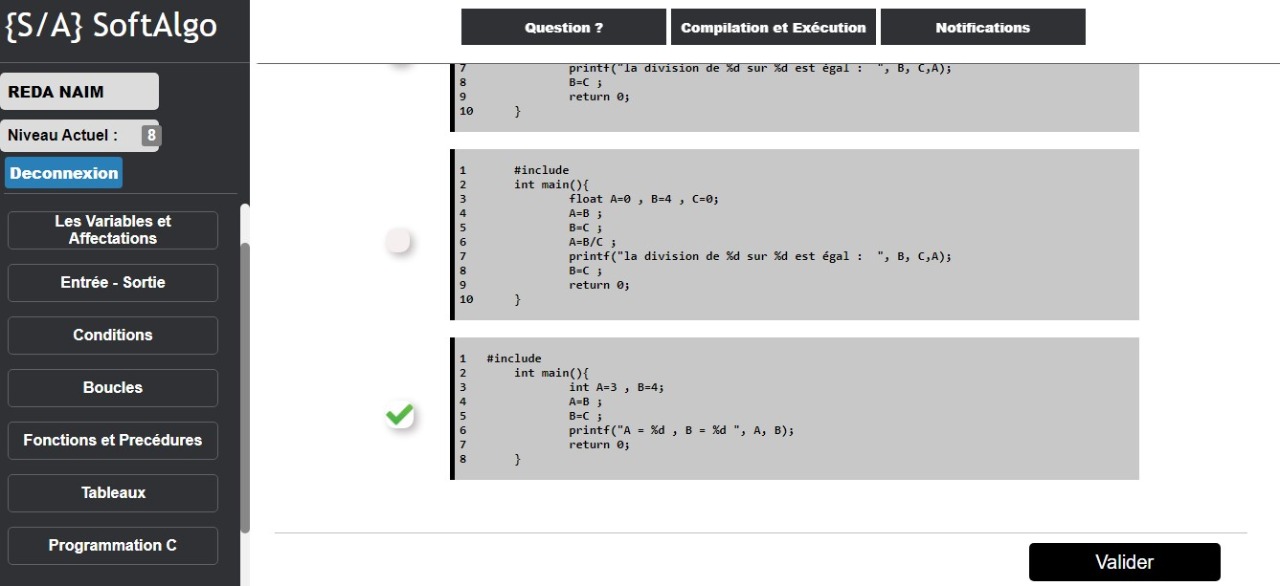
Une image contenant texte

Description générée automatiquement

**Figure 24:** Interface passage du cours vers le test.



**Figure 25:** Interface test niveau pour l'algorithmique.



**Figure 26:** Interface test niveau pour la programmation en C.

#### 4.3.8 Interface question

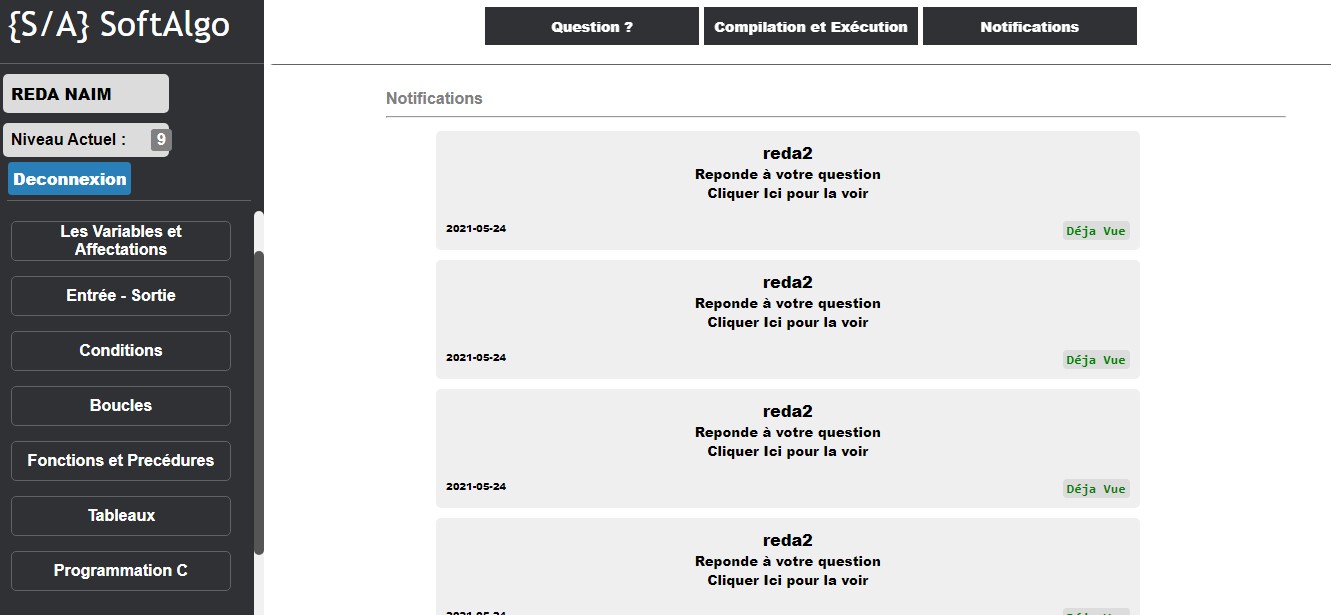
Tout au long de la phase d’apprentissage, si l’utilisateur veut poser une question autour quelque chose qui n’est pas claire ou bien pour demander des informations supplémentaires autour de notre sujet en question, avec cette interface il peut poser ses questions sous forme d’un commentaire qui a un titre qui va être vérifier par l’administrateur de l’application avant d’être publié.

Une image contenant texte

Description générée automatiquement

**Figure 27:** Interface question.

#### 4.3.9 Interface notification

Pour mettre la tâche plus interactive entre les utilisateurs, l’interface notification aide l’utilisateur de trouver directement la réponse de son collègue sur sa question via l’envoie d’un signal qui se situe dans la case des notifications.

**Figure 28:** Interface notification.

#### 4.3.10 Interface compilation et exécution

Pour entamer le côté pratique dans notre application, nous cherchons à mettre en œuvre cette interface qui aide l’utilisateur d’analyser le résultat de l’exécution de son propre code. Quand l’utilisateur entre les instructions du code, il doit cliquer sur le bouton « Run » afin de gérer la compilation via le langage C.

Une image contenant texte

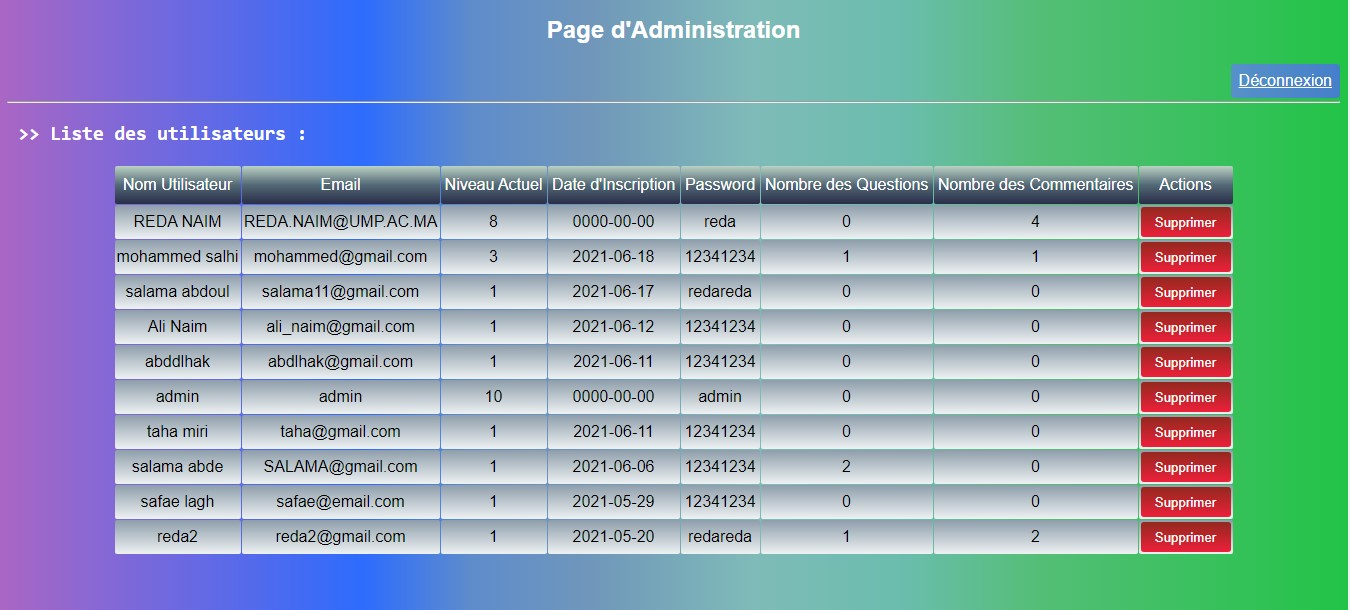
Description générée automatiquement

**Figure 29:** Interface compilation et exécution.

#### 4.3.11 Côté administrateur

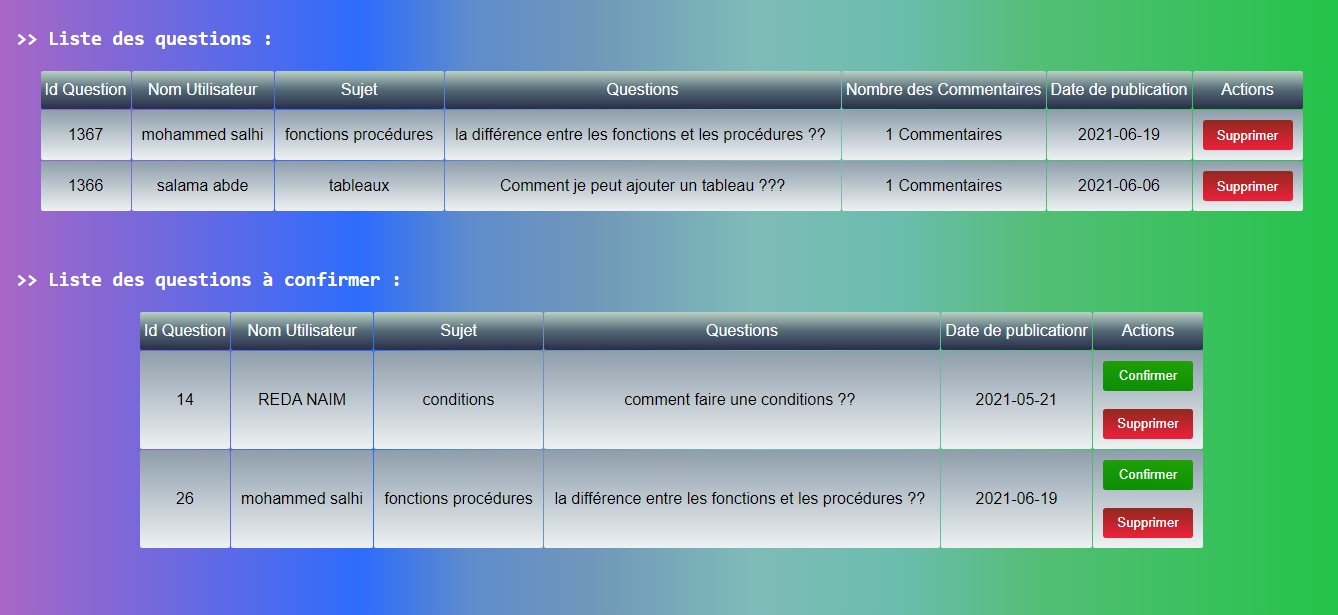
Pour que l’administrateur puisse génère toutes les fonctionnalités de l’application d’abord il doit s’authentifier pour ajouter des cours (sous forme des images) et des tests (sous forme des textes). Afin de gérer les comptes des utilisateurs authentifiés avec leurs informations (email, password …) et leurs activités (nombres des questions...) au sein de l’application.

Si un utilisateur essaye de publier quelque chose qui est contre les lois de l’application l’administrateur a le droit de le supprimer de la liste des utilisateurs.



**Figure 30:** Interface liste des utilisateurs.

Lorsqu’un utilisateur authentifé pose une question, le système envoie une demande de confirmation à l’administrateur, si la question va avec les conditions de l’application, alors elle va être ajoutée à la page des questions pour que les autres utilisateurs peuvent la visualiser.



**Figure 31:** Interface liste des questions.

### 4.4 Conclusion

Après avoir présenté une étude détaillée de l'étude, de l'analyse et de la conception, nous sommes passés à l'étape la plus importante du cycle de vie de l'application Web : Dans la phase de réalisation. Dans ce chapitre, nous avons présenté toutes les interfaces de l’application pour mieux utiliser notre application.

# **Conclusion générale**

À l’issue de la réalisation de ce travail, on peut affirmer que ce projet nous a été d’une grande utilité pour se familiariser avec le travail sur le web.

Ce travail de conception et de développement d’une application web intitulée « SOFT-ALGO » au long du PFE nous a été bénéfique sur plusieurs plans : il nous a permis de perfectionner nos connaissances acquises en programmation et en conception.

Du point de vue technique, ce projet nous a permis de nous adapter avec l’environnement du développement informatique, de même il nous a permis de maîtriser la méthode développement et les technologies de programmation.

Tout au long de l’élaboration du projet, nous avons rencontré plusieurs difficultés tant qu’au niveau conceptuel qu’au niveau de la réalisation. Tout de même, nous avons réussi à les surpasser pour présenter en fin de compte une application opérationnelle.

Comme perspective, nous espérons voir notre application évoluer par une étape d’approfondissement du volet transactionnel avec d’autres langages de programmation, tels que python parce qu’il est très demandé pour la création des algorithmes performants ou des applications puissantes comme : Google, YouTube, Dropbox afin de rendre notre application plus intéressante. Ce qui permettra l’échange des diverses données en utilisant des réseaux et des formats normalisés. En plus, nous insistons de rendre notre application web une application mobile pour assurer la possibilité d’être utilisée par différents utilisateurs accédant par diverses plateformes.

# Bibliographie

|  |  |
| --- | --- |
| [1] | Cours d’informatique de Mme Khadija LAAROUSSI, filière SMC-S4, Faculté des sciences Oujda. |
| [2] | Cours d’algorithmique de Mme Noura OUERDI, filière SMIA-S2, Faculté des sciences Oujda. |
| [3] | Séries des travaux pratiques de Mme Khadija LAAROUSSI, module informatique, filière SMC-S4. |

# Webographie

|  |  |  |
| --- | --- | --- |
| [1] | https://www.uml.org | (Mars 2021) |
| [2] | https://www.w3.org | (Mai 2021) |
| [3] | https://www.css-faciles.com | (Mai 2021) |
| [4] | https://www.php.net | (Mars 2021) |
| [5] | https://www.mysql.com | (Mars 2021) |
| [6] | https://www.php.net/manual/fr/index | (Mars 2021) |
| [7] | https://html-css-js.com/css/generator/box-shadow/ | (Mai 2021) |
| [8] | https://sql.sh/ | (Mai 2021) |
| [9] | https://mrproof.blogspot.com/2010/09/qcm-algorithme- test-logique.html | (Avril2021) |
| [10] | https://www.programiz.com/c-programming/online- compiler/ | (Mai 2021) |
| [11] | https://www.w3schools.com/ | (Mars 2021) |
| [12] | https://css-tricks.com/snippets/css/a-guide-to- flexbox/#background | (Mai 2021) |
| [13] | https://www.pexels.com/ | (Avril 2021) |