

REPUBLIQUE TOGOLAISE

Travail – Liberté – Patrie

MINISTERE DE LA PLANIFICATION
DU DEVELOPPEMENT ET
DE LA COOPERATION



Institut Africain d'Informatique
Représentation du Togo (IAI-TOGO)

Tel: 22 20 47 00

Email: iaitogo@iai-togo.tg

Site web: www.iai-togo.tg

07 BP 22456 Lomé 07 TOGO

RAPPORT DE STAGE PRATIQUE
Stage de programmation

GESTION DES STAGIAIRES: CAS DE TOGO CELLULAIRE

Du 15 Juillet au 19 Septembre 2021

Rédigé et présenté par :

APAFLO Doko Nicole

Etudiante en Deuxième année : Tronc commun

Année Universitaire : 2020-2021

SUPERVISEUR

M. DAVON Essè

Enseignant à IAI-Togo

REMERCIEMENTS

« **Seul on va vite, ensemble on va plus vite** » dit la sagesse africaine. Dans cet état des choses, c'est en toute humilité que je viens remercier tout un chacun sans qui ce travail n'aurait pas pu être possible. Par ordre, j'aimerais d'abord remercier Le Pourvoyeur de toute chose, Dieu Tout Puissant pour la grâce de la vie, pour tous ses bienfaits dans l'accomplissement de ce projet jusqu'à sa finalisation.

Ensuite mes remerciements vont à l'endroit de l'administration de notre institut en commençant par le Directeur Général de l'IAI-TOGO Monsieur AGBETI Kodjo, à Monsieur le Directeur des Etudes Monsieur AMEYIKPO Kossi ainsi qu'à tout le corps professoral et nos différents encadrants.

A mon superviseur, Monsieur DAVON Essè pour sa disponibilité ses précieuses remarques qui nous ont permis de travailler à l'amélioration de notre démarche.

Et enfin à ma famille, mon père, ma mère qui n'ont jamais manqué de s'investir dans tout ce que je fais. A ma sœur qui a toujours su trouver les mots pour m'encourager. Je tiens également à remercier mes collègues, tous ceux et celles qui de près ou de loin ont eu à contribuer veuillez trouver ici l'expression de ma profonde gratitude.

SOMMAIRE

REMERCIEMENTS.....	I
LISTE DES FIGURES.....	III
LISTE DES TABLEAUX.....	IV
INTRODUCTION	5
1 CAHIER DES CHARGES.....	6
1.1 Présentation du sujet	6
1.2 Problématique du sujet	6
1.3 Intérêt du sujet	6
2 RAPPORT DE PREPROGRAMMATION.....	7
2.1 Etude de l'existant	7
2.2 Critique de l'existant	7
2.3 Planning prévisionnel de réalisation	7
2.4 Etude détaillée de la solution	9
3 REALISATION ET MISE EN ŒUVRE	32
3.1 Matériels et logiciels utilisés	32
3.2 Sécurité de l'application	33
3.3 Evaluation financière de la solution	34
3.4 Présentation de l'application	36
CONCLUSION	59
BIBLIOGRAPHIE INDICATIVE	60
DOCUMENTS ANNEXES.....	60
WEBOGRAPHIE INDICATIVE.....	61
TABLE DES MATIERES.....	62

LISTE DES FIGURES

[Figure 1 : Description du processus 2TUP](#)

[Figure 2 : Principes du processus 2TUP](#)

[Figure 3 : Les phases du processus 2TUP](#)

[Figure 4 : Diagramme de cas d'utilisation de l'administrateur](#)

[Figure 5 : Diagramme de cas d'utilisation du stagiaire](#)

[Figure 6 : Diagramme de cas d'utilisation du système](#)

[Figure 7 : Diagramme de séquence de la demande de stage](#)

[Figure 8 : Diagramme d'activité de la demande de stage](#)

[Figure 9 : Diagramme d'activité de l'authentification](#)

[Figure 10 : Diagramme des classes du système](#)

[Figure 11 : Sécurité de l'application](#)

[Figure 12 : Onglet du profil administrateur](#)

[Figure 13 : page d'accueil de l'application](#)

[Figure 14 : page de l'administrateur avec les demandes](#)

[Figure 15 : page d'authentification](#)

[Figure 16 : page de demande de stage](#)

[Figure 17 : la page de l'administrateur avec les statistiques](#)

[Figure 18 : état présentant la liste des demandes de stage](#)

[Figure 19 : état présentant la liste des demandes de stage acceptées](#)

[Figure 20 : état présentant la liste des directions](#)

[Figure 21 : état présentant la liste des divisions](#)

LISTE DES TABLEAUX

Tableau 1 : Planning prévisionnel de réalisation

Tableau 2 : Les différents cas d'utilisation

Tableau 3 : Matériels utilisés

Tableau 4 : Coût matériel

Tableau 5 : Coût de la conception

Tableau 6 : Coût total

Introduction

Dans le cadre du stage de mi-parcours réalisé chaque année à l'Institut Africain d'Informatique, nous avons eu à travailler durant deux (2) mois sur un thème. Ce stage consiste à réaliser une application permettant de répondre à une problématique donnée. Etant donnée la crise sanitaire, notre stage pratique n'a donc pas pu se dérouler en entreprise c'est alors qu'aider du cahier des charges que nous avons reçu nous avons eu à concevoir une application de gestion des stagiaires plus précisément le cas de la gestion des stagiaires au sein de Togo Cellulaire.

Ce document résume ainsi le processus suivi pour aboutir à notre application. Le plan de lecture est orienté en trois différentes parties : le **cahier des charges**, le **rapport de préprogrammation** et enfin la **réalisation et mise en œuvre**.

1 Cahier des charges

1.1 Présentation du sujet

Togocom autorise chaque année des étudiants à effectuer en son sein divers types de stages (Stage de perfectionnement, stage de mémoire, etc.). C'est dans cette optique que des étudiants en informatique sont affectés à la DSI (Division du Système d'Information) avec pour objectif principal de leur permettre de se familiariser avec le monde du travail. Au cours du stage les tâches réalisées par les stagiaires sont difficiles à retracer, les informations les concernant également se perdent et au final le renouvellement n'est facile à faire. C'est dans le but de palier à ces problèmes cités ci-avant que l'entreprise nous demande de réaliser une application dénommée « **GeSta** ».

1.2 Problématique du sujet

Comment l'application que nous développerons pourrait-elle aider à mieux gérer les informations des stagiaires ? Comment se fera le suivi des stagiaires au sein de l'entreprise ? Comment le processus de renouvellement de stage se déroulera t-il ?

1.3 Intérêt du sujet

1.3.1 Objectifs

L'application aura pour but la gestion automatique :

- Des informations liées au stagiaire
- Du suivi du travail du stagiaire à Togo Cellulaire
- Du renouvellement de stage à certains stagiaires sur décision de la Direction Générale

1.3.2 Résultats

L'application que nous aurons à réaliser permettra :

- Un suivi sain et rigoureux de la prise en charge des stagiaires
- De disposer du nombre de stagiaires que la DEPI est disposé à accueillir
- De sortir ou imprimer les différents états relatifs

2 Rapport de préprogrammation

2.1 Etude de l'existant

Le domaine d'étude concernant la gestion des stagiaires à Togo Cellulaire n'est pas informatisé. Et en ce sens la seule information dont dispose la DSI sur le passage du stagiaire était uniquement la note de service signée par la Direction Générale.

2.2 Critique de l'existant

Il est à relever une réelle insuffisance sur la façon dont sont gérées les informations liées au passage d'un stagiaire dans chaque section. La seule note de service ne permet pas de suivre le parcours du stagiaire dès son arrivée jusqu'à son départ. La perte d'information devient donc possible, ce qui rendra difficile par exemple une recherche effectuée sur un stagiaire.

2.3 Planning prévisionnel de réalisation

Le planning prévisionnel permet de répartir les tâches à accomplir sur la période totale de réalisation afin d'atteindre une meilleure productivité et aussi dans l'optique de respecter le cahier des charges.

Période de réalisation du projet : du 15/07 au 19/09 2021

Durée totale de réalisation du projet : 9 semaines soit 56 jours

Tableau 1 : Planning prévisionnel de réalisation

N°	Tâches	Date début	Date fin	Durée en jours
1	-Prise de contact avec le superviseur -Choix des jours de travail -Lecture du cahier des charges	15/07	19/07	5 jours
2	-Première séance de travail avec le superviseur	20/07	25/07	6 jours

	-Rédaction de la première partie du document -Réalisation de la page de garde -Début de la conception de la base de données -Choix de l'outil de modélisation -Début de la modélisation de la base de données -Envoi du rapport hebdomadaire au superviseur et au DAAS			
3	-Début de rédaction de la deuxième partie du document -Choix de la méthode d'analyse -Fin de la conception de la base de données -Choix du SGBD	26/07	01/08	7 jours
4	-Réalisation des différents diagrammes -Fin de la modélisation de la base de données -Implémentation de la base de données	02/08	07/08	7 jours
5	Obtention d'un document provisoire	08/08	11/08	4 jours
6	Phase de programmation et de test	11/08	24/08	14 jours
7	Phase de programmation et de test	25/08	31/08	7 jours
8	-Tests continuels de l'application -Finalisation du document	1/09	12/09	14 jours

2.4 Etude détaillée de la solution

2.4.1 Analyse et conception

2.4.1.1 Présentation de la méthode d'analyse

2.4.1.1.1 Langage de modélisation

Le langage de modélisation unifié, en anglais Unified Modeling Language (UML) est un langage de modélisation à base de pictogrammes (dessins figuratifs stylisés ayant fonction de signe) 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. UML n'étant qu'un langage il faut lui associer un processus pour rendre totale l'approche objet. Les méthodes imposent au langage l'ordre et la démarche, de ce fait il existe une multitude de méthodes de conception objet tels que : 2TUP (Two Track Unified Process), XP (eXtreme Programming), Scrum, RUP (Rational Unified Process).

Pour notre projet nous utiliserons la méthode 2TUP.

L'outil de modélisation que nous avons utilisé pour ce projet est : StarUML.

C'est un logiciel de modélisation agile et concise conçu par la société **MKLabs Co., Ltd**. Il est compatible avec les modèles standards et les diagrammes UML 2.x. Il supporte la modélisation avec les diagrammes SysML, il supporte également la création de modèles entité-relation, les diagrammes de flots. Il permet aussi de créer des API, des extensions il permet également la génération de code en différents langages. Une autre alternative à cet outil est l'outil PowerAMC de la société **Sybase**.

2.4.1.1.2 Le processus de modélisation

2TUP signifie « 2 Track Unified Process ». C'est un processus UP qui répond aux caractéristiques que nous venons de citer. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des chemins « fonctionnels » et

« d'architecture technique », qui correspondent aux deux axes de changement imposés au système informatique.

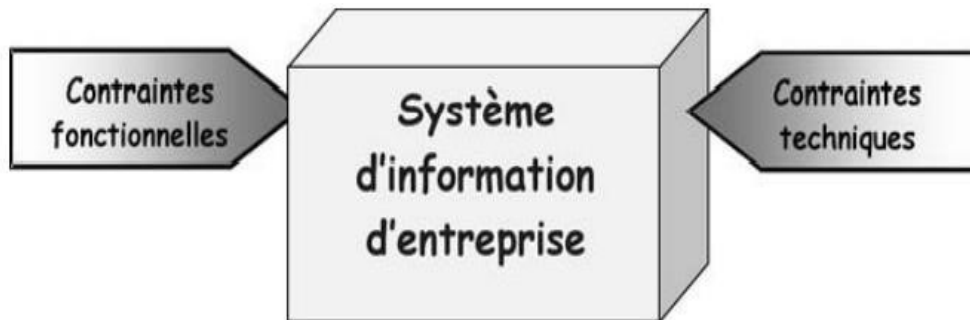


Figure 1 : Description du processus 2TUP

En effet nous avons choisi ce processus à cause de ses principaux principes et concepts et dimensions.

Les principes :

- Basé sur UML
- Est interactif et incrémental
 - UP préconise le principe de développement par itérations
 - Chaque itération constitue une « brique » livrable
 - Le développement se fait par incrément successifs

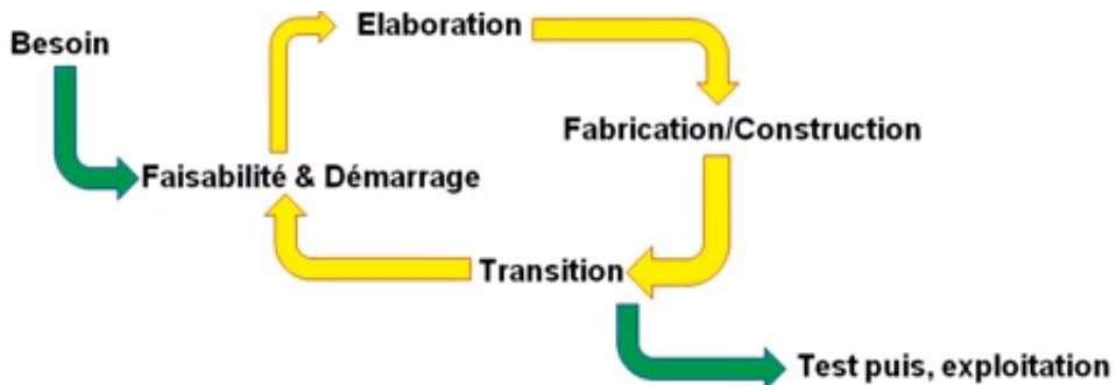


Figure 2 : Principes du processus 2TUP

- Conduit par le cas d'utilisation
 - Le système à construire se définit d'abord avec les utilisateurs
 - Les utilisateurs source d'expressions des besoins
 - La modélisation des besoins se fait grâce au diagramme de cas d'utilisation

- Centré sur l'architecture
 - Le processus UP préconise de définir aussitôt que possible l'architecture type du système
- Orienté par la réduction des risques
 - L'analyse des risques doit être présent à toutes les étapes de développement du système
 - Le processus itératif favorise la réduction des risques

Les concepts

- Le processus unifié permet de répondre aux questions : **qui fait quoi, quand et comment.**
- **Qui ? -----> Rôle** : comportement et responsabilités d'une ressource ou groupe de ressources
- **Comment ? ----->Activité** : travail effectué par une ressource
- **Quoi ? ----->Artefact** : ensemble d'informations produits, utilisé ou modifié par un processus. Les artefacts sont les produits effectifs d'un projet.
- **Quand ? ----->workflow** : une séquence d'activité qui produit un résultat mesurable et montre les interactions entre les ressources.

Les schémas d'ensemble d'UP

UP est décrit suivant deux dimensions :

- Un axe horizontal représentant le temps : aspect dynamique du processus. Sur cet axe, le processus est organisé en phases et itérations (**Inception** "Lancement/Création", **Elaboration**, **Construction** et **transition**)
- Un axe vertical : aspect statique du processus. Sur cet axe, le processus est organisé en activités et workflow (**Expressions des besoins**, **Analyse**, **Conception**, **Implémentation**, **Tests**).

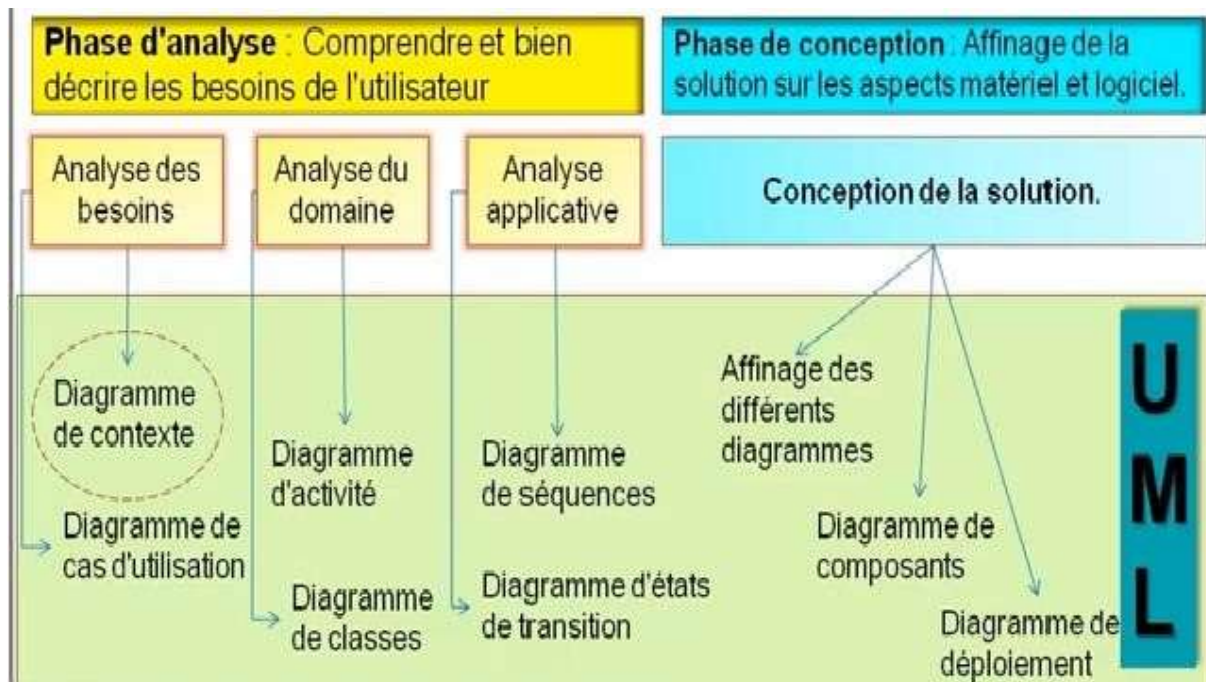


Figure 3 : Les phases du processus 2TUP

Appliqué à notre thème, le processus est divisé en quatre (4) étapes :

- La modélisation métier

Elle est décrite par :

- Un diagramme d'activités
- Un diagramme de cas d'utilisation niveau entreprise

- La capture initiale des besoins

Elle est traduite par :

- Un diagramme de cas d'utilisation niveau utilisateur
- La déduction des IHM et le diagramme de classe métier

- L'analyse

Elle est réalisée grâce :

- A un de cas d'utilisation en fonction de chaque cas de la capture initiale des besoins
- A des fiches descriptives des sous fonctionnalités trouvées

- A un diagramme de séquence de déduite à partir des IHM décrivant chaque sous fonctionnalité afin d'obtenir le diagramme de classe d'analyse de la sous fonctionnalité
- La conception

Elle consiste à associer au diagramme de classe d'analyse un langage de programmation pour obtenir le diagramme de classe de conception.

2.4.2 Implémentation

Dans cette partie, nous explorerons l'application d'un point de vue interne et externe. Les éléments des spécifications fonctionnelles y seront détaillés afin d'avoir une vision claire et sans ambiguë de l'application. Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système

- **Vue de cas d'utilisation**
 - Diagramme de cas d'utilisation
 - Définition du cas d'utilisation

Un diagramme de cas d'utilisation traduit tout ce que l'utilisateur exprime comme action sur le logiciel ou le système à modéliser. C'est une représentation faisant intervenir les acteurs et les cas d'utilisation. Il traduit les besoins des utilisateurs vis-à-vis du système développé.

- Éléments de modélisation de cas d'utilisation

Les acteurs :

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie.

Le diagramme des cas d'utilisation d'UML distingue deux acteurs :

- Les acteurs principaux (qui modifient l'état du système ou qui consultent cet état)
- Les acteurs secondaires (acteurs auxquels le système fait appel pour répondre aux sollicitations d'un acteur principal).

Dans notre projet, nous avons décelé uniquement des acteurs principaux qui sont :

- Utilisateur (est soit un stagiaire, ou un superviseur ou un maître de stage)
- Administrateur (est un agent de la Direction Générale)

○ Le cas d'utilisation

Le cas d'utilisation (ou use case) correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs. L'ensemble des cas d'utilisation décrit les objectifs (le but) du système.

- Les relations entre acteur et cas d'utilisation

Les relations sont utilisées pour lier des acteurs avec des cas d'utilisation. Elles indiquent qu'un acteur participe au cas d'utilisation sous une forme quelconque. Les relations sont représentées par une ligne reliant l'acteur et le cas d'utilisation.

Le résumé des cas d'utilisations de notre projet est consigné dans le tableau suivant :

Tableau 2 : Les différents cas d'utilisation

Cas d'utilisation			Acteurs
S'authentifier			Administrateur Utilisateur (Stagiaire, Superviseur, Maître de stage)
Gestion des demandes de stage	Accepter demande Refuser demande		Administrateur
Gestion des utilisateurs	Créer un utilisateur		Administrateur

	Modifier un utilisateur Supprimer un utilisateur	
Gestion des droits	Créer un droit Modifier un droit Supprimer un droit	Administrateur
Gestion des formations	Créer une formation Modifier une formation Supprimer une formation	Administrateur
Gestions des directions	Créer une direction Modifier une direction Supprimer une direction	Administrateur
Gestion des divisions	Créer une division Modifier une division Supprimer une division	Administrateur
Gestion des sections	Créer une section Modifier une section Supprimer une section	Administrateur

Gestion des maîtres de stage	Ajouter un maître de stage Modifier un maître de stage Supprimer un maître de stage	Administrateur
Gestion des stagiaires	Ajouter un stagiaire Modifier un stagiaire Supprimer un stagiaire	Administrateur
Gestion des superviseurs	Ajouter un superviseur Modifier un superviseur Supprimer un superviseur	Administrateur
Gestion des thèmes	Créer un thème Modifier un thème Supprimer un thème	Administrateur
Attribuer un thème		Maître de stage
Noter un rapport		Maître de stage
Gestion des notations	Ajouter une notation Modifier une notation Supprimer une notation	Administrateur
Gestion des rapports	Ajouter un rapport	Administrateur

	Modifier un rapport Supprimer un rapport	
Gestion des rémunérations	Ajouter une rémunération Modifier une rémunération Supprimer une rémunération	Administrateur
Gestion des matériels	Ajouter un matériel Modifier un matériel Supprimer un matériel	Administrateur
Utiliser un matériel		Stagiaire
Superviser un stagiaire		Superviseur
Suivre une formation		Stagiaire
Déposer un rapport		Stagiaire
Postuler pour un stage	Envoyer une demande de stage	Stagiaire
Demander un renouvellement	Envoyer une demande de renouvellement	Stagiaire
Avoir un thème		Stagiaire
Avoir une notation		Stagiaire
Avoir un maître de stage		Stagiaire
Gestion des renouvellements	Accorder un renouvellement	Administrateur

	Refuser un renouvellement	
--	------------------------------	--

Diagramme de cas d'utilisation par acteur. Il est représenté ici les diagrammes de cas d'utilisation par module identifiés plus haut.

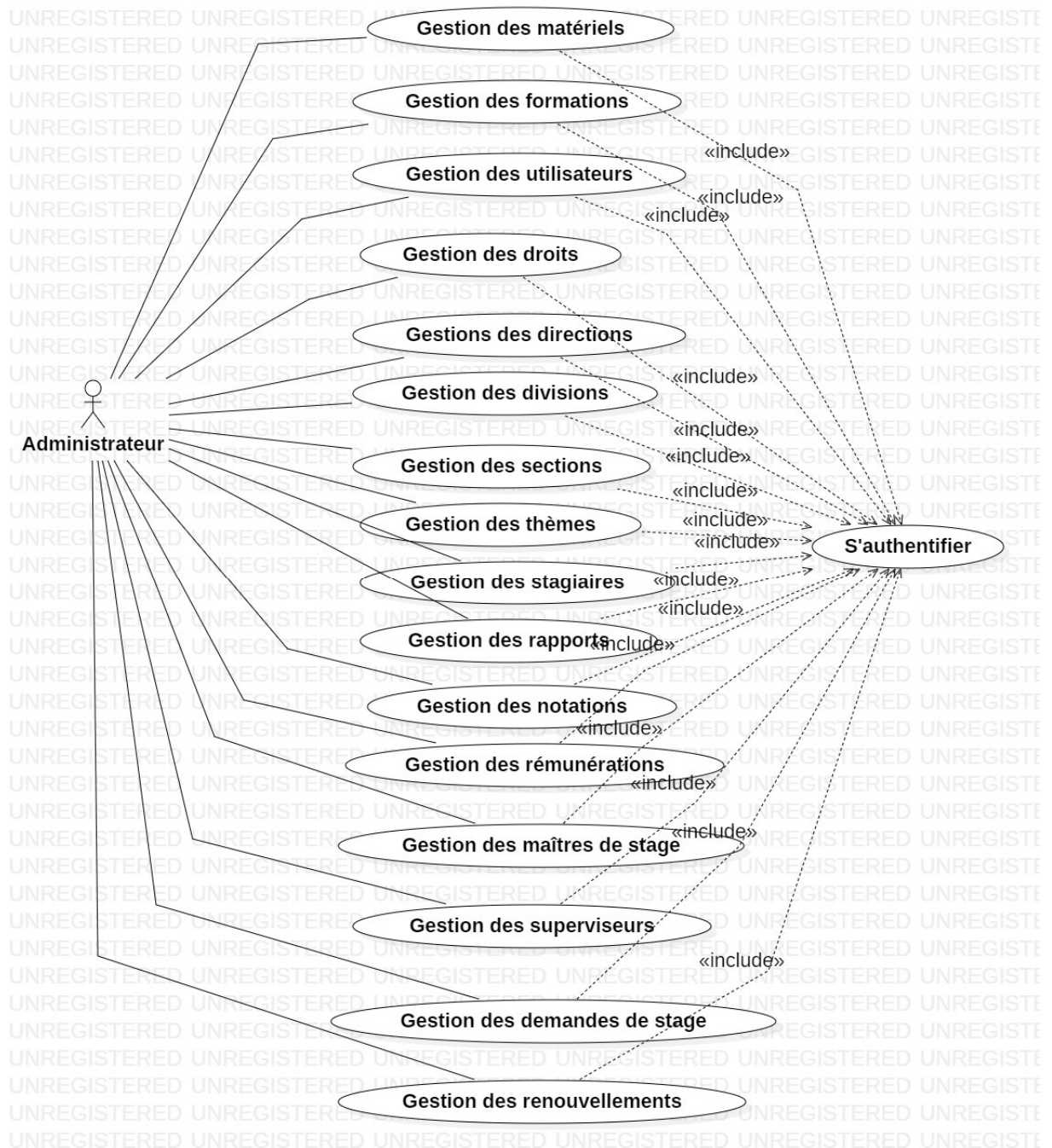


Figure 4 : Diagramme de cas d'utilisation de l'administrateur

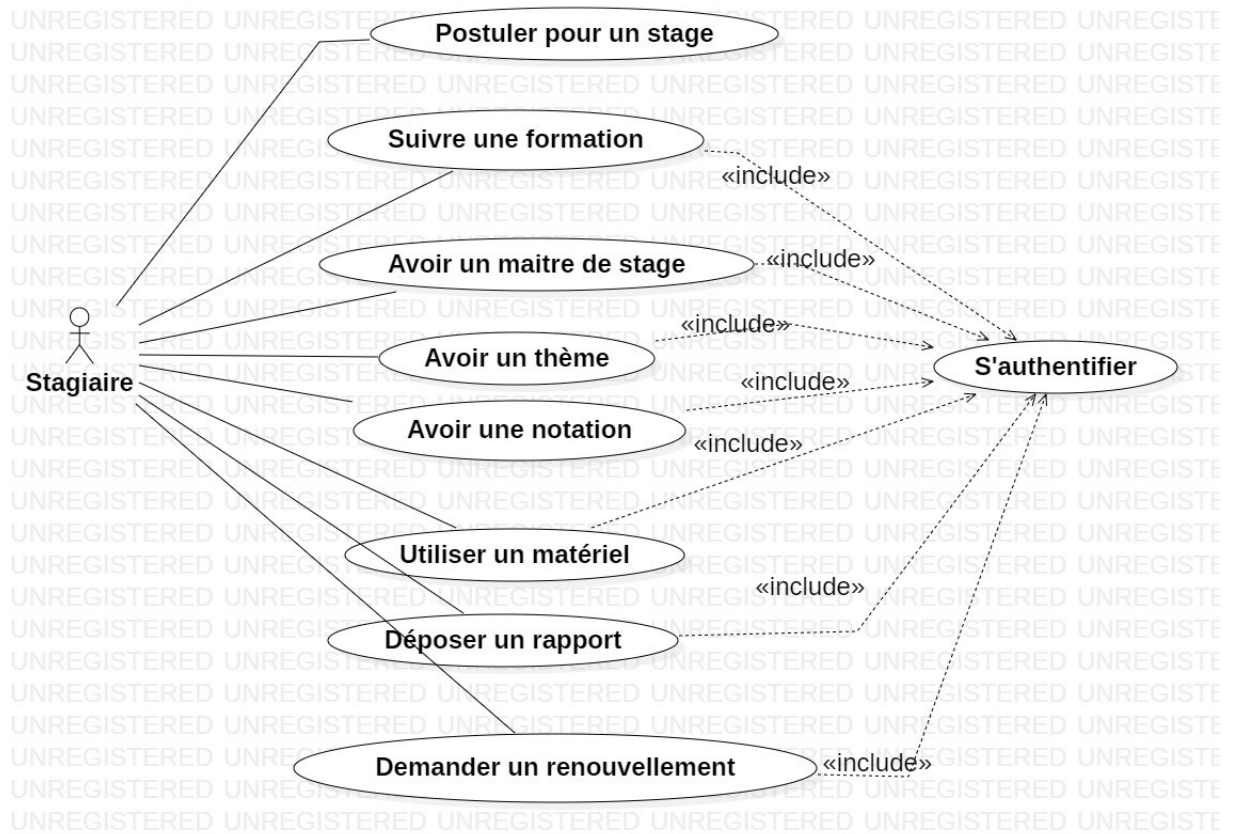


Figure 5 : Diagramme de cas d'utilisation du stagiaire

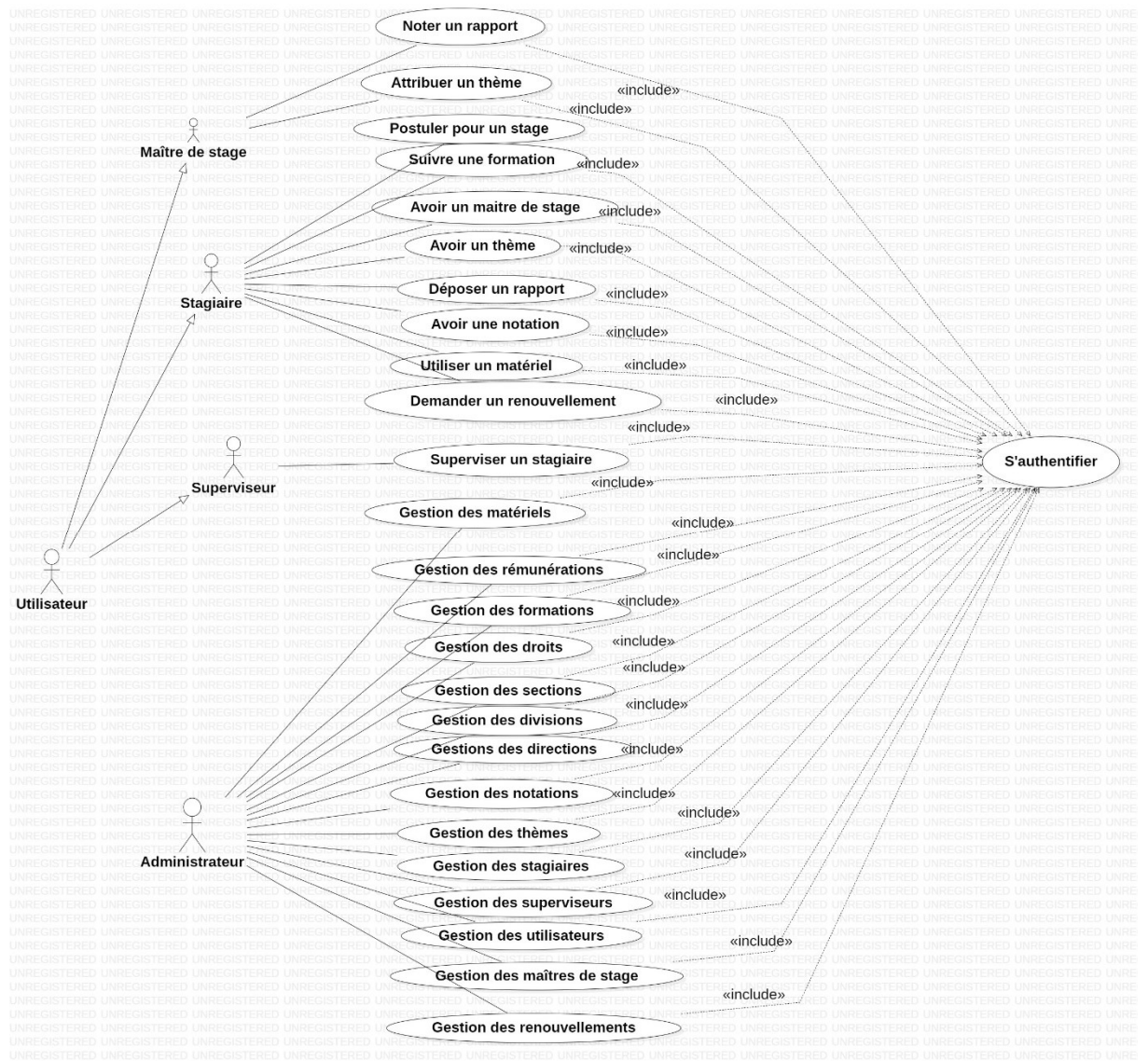


Figure 6 : Diagramme de cas d'utilisation du système

- **Vues de processus**

Description textuelle de quelques cas d'utilisation

La description textuelle d'un cas d'utilisation permet de :

- Clarifier le déroulement de la fonctionnalité ;
- Décrire la chronologie des actions qui devront être réalisées ;
- S'authentifier

Titre : s'authentifier

Acteurs : administrateur, utilisateur

Résumé : accorde l'accès du système à l'utilisateur

Responsable : APAFLO Nicole

Version : 1.0

Date de création : 10 Août 2021

Date de modification : 20 Août 2021

Description des scénarii :

Préconditions :

- Le système est fonctionnel
- La page d'authentification est accessible
- Être enregistré comme utilisateur

Scénario nominal :

L'utilisateur saisit les paramètres de connexion et valide (E1)

Le système vérifie la validité des paramètres saisis par l'utilisateur (A1)

Le système affiche la page d'accueil correspondant aux droits de l'utilisateur

Scénario alternatif :

A1 : les paramètres de connexion ne sont pas conformes

Le système affiche un message d'erreur indiquant que les identifiants sont incorrects : le scénario reprend au point 1 du scénario nominal.

Scénario d'exception :

E1 : l'utilisateur annule la connexion

Le scénario nominal est interrompu

Post conditions :

L'utilisateur est authentifié

Le système est toujours fonctionnel

- Créer un utilisateur

Titre : créer un utilisateur

Acteur : administrateur

Résumé : permet à l'administrateur d'ajouter un nouvel utilisateur au système

Responsable : APAFLO Nicole

Version :1.0

Date de création : 10 Août 2021

Date de modification :23 Août 2021

Description des scénarii :

Préconditions :

- Le système est fonctionnel
- L'administrateur s'est authentifié

Scénario nominal :

1. L'administrateur remplit le formulaire d'enregistrement d'un utilisateur (E1)

2. L'administrateur attribut un profil à l'utilisateur en cours de création
3. Le système vérifie les informations (A1)
4. L'administrateur valide le formulaire
5. Le système enregistre les informations
6. Le système affiche un message de confirmation

Scénario alternatif :

A1 : l'administrateur est invité à saisir des informations correctes

Scénario d'exception :

E1 : le scénario nominal est interrompu

Post conditions :

- Nouvel utilisateur créé
- La liste des utilisateurs est mise à jour
- Le système est toujours fonctionnel

- Postuler pour un stage

Titre : postuler pour un stage

Acteur : stagiaire

Résumé : permet à un stagiaire potentiel de postuler pour un stage

Responsable : APAFLO Nicole

Version :1.0

Date de création : 23 Août 2021

Date de modification :25 Août 2021

Description des scénarii :

Préconditions :

- Le système est fonctionnel

- Le formulaire de postulation est accessible

Scénario nominal :

1. Le stagiaire remplit le formulaire
2. Le stagiaire valide le formulaire(E1)
3. Le système vérifie les valeurs saisies(A1)
4. Le système enregistre la demande
5. Le système envoie un mail de confirmation de réception au stagiaire

Scénario d'exception :

E1 : le scénario nominal est interrompu, la validation du formulaire est annulée

Scénario alternatif :

A1 : le système vérifie l'existence des informations entrées

Post conditions :

- La demande envoyée est étudiée

○ Gestion des demandes

Titre : gestion des demandes de stage

Acteur : administrateur

Résumé : permet à l'administrateur de traiter les demandes envoyées

Responsable : APAFLO Nicole

Version :1.0

Date de création : 23 Août 2021

Date de modification :06 Septembre 2021

Description des scénarii :

Préconditions :

- Le système est fonctionnel

- Les demandes ont été enregistrées

Scénario nominal :

1. Consultation des demandes(E1)
2. Vérification des récentes offres de stage
3. Accord du stage(A1)
4. Refus du stage(A2)

Scénario d'exception

E1 : interruption du scénario nominal : quitter la liste des demandes

Scénario alternatif

A1 : profil du demandeur satisfaisant : accord du stage

A2 : profil non satisfaisant : refus du stage

Post conditions

- Le système est toujours fonctionnel
- Un mail d'acceptation est envoyé au demandeur dont la demande est accordée
- Un mail de refus est envoyé au demandeur dont la demande est refusée

Diagramme de séquence

Les diagrammes de séquence sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Ils permettent de monter les interactions d'objets dans le cadre d'un scénario d'un diagramme de cas d'utilisation.

Quelques diagrammes de séquence :

- Postuler pour un stage

Le diagramme suivant montre le fonctionnement de la demande de stage

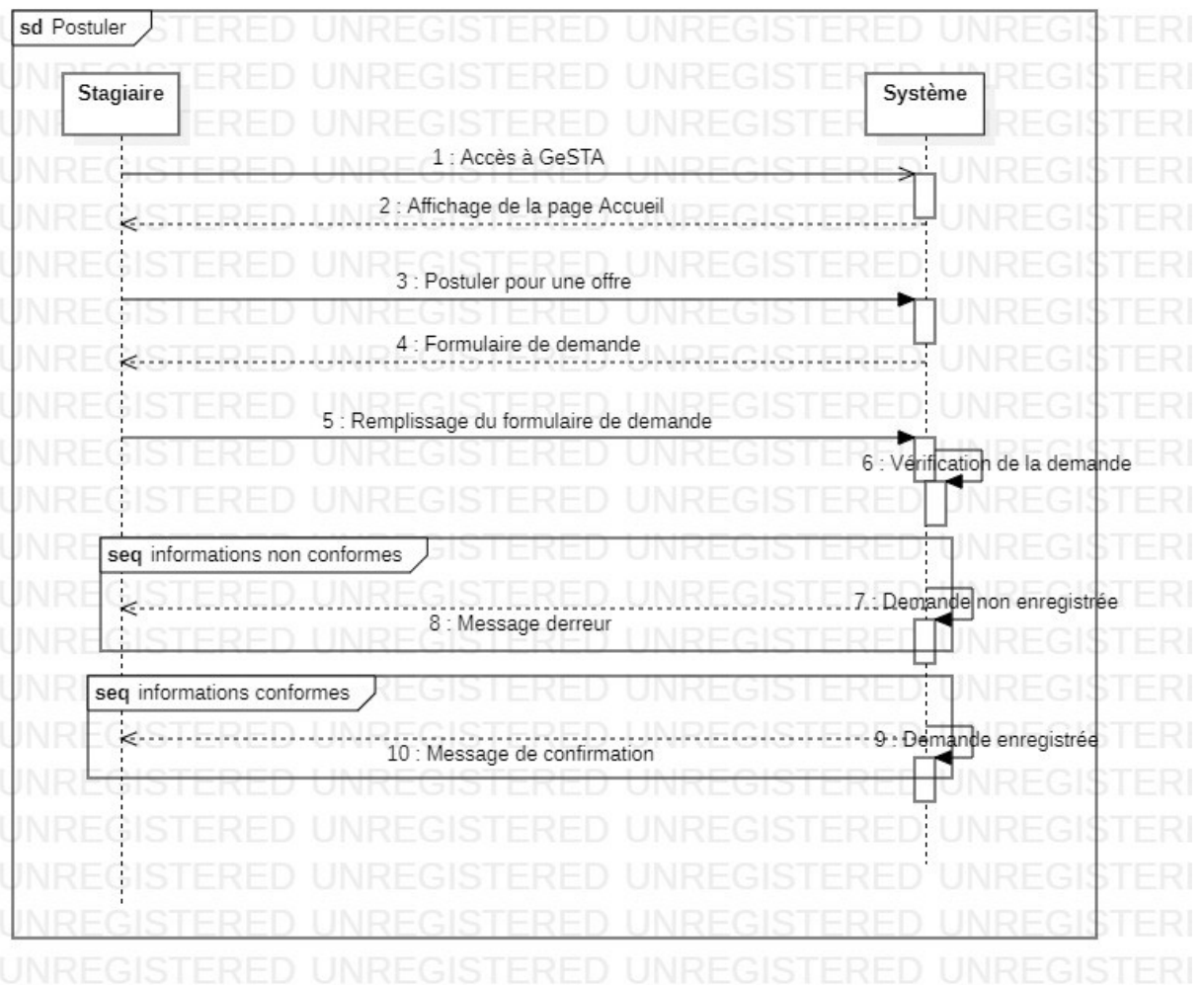


Figure7 : Diagramme de séquence de la demande de stage

Diagramme d'activité

Ils permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils représentent graphiquement le comportement d'une méthode d'un cas d'utilisation.

- Postuler pour un stage

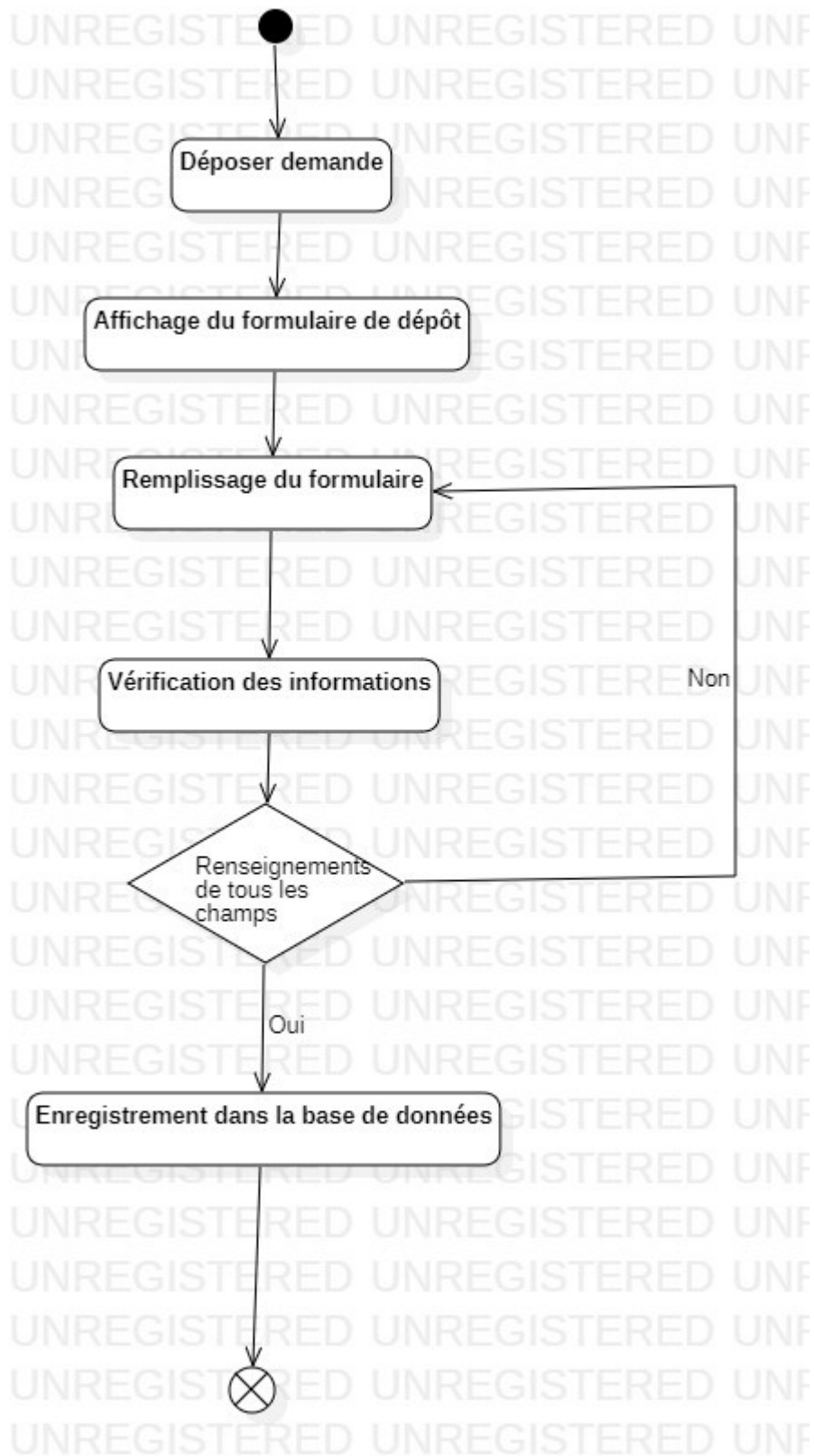


Figure 8 : Diagramme d'activité de la demande de stage

- S'authentifier

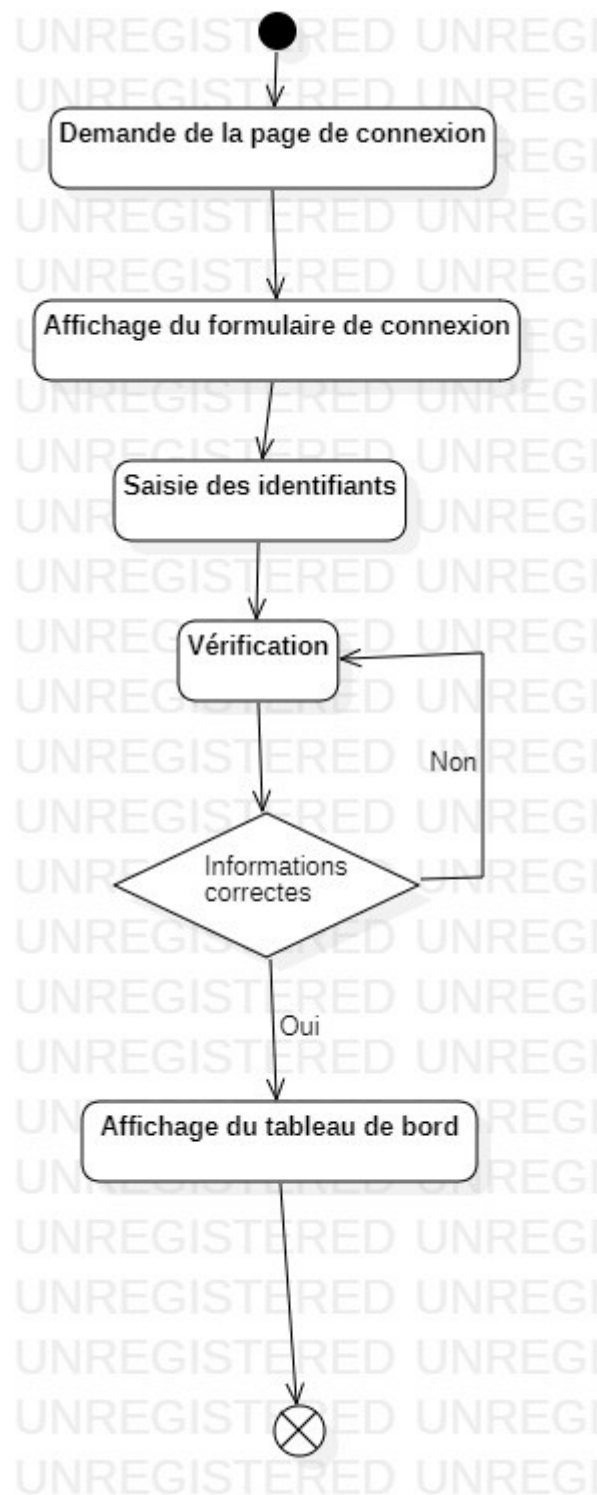


Figure 9 : Diagramme d'activité de l'authentification

- **Vue logique**

Diagramme de classe-entité

Généralités

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Le diagramme des classes comporte les concepts suivants :

- Classe
- Attribut
- Identifiant
- Opération
- Relation
- Généralisation

Diagramme des classes de notre système

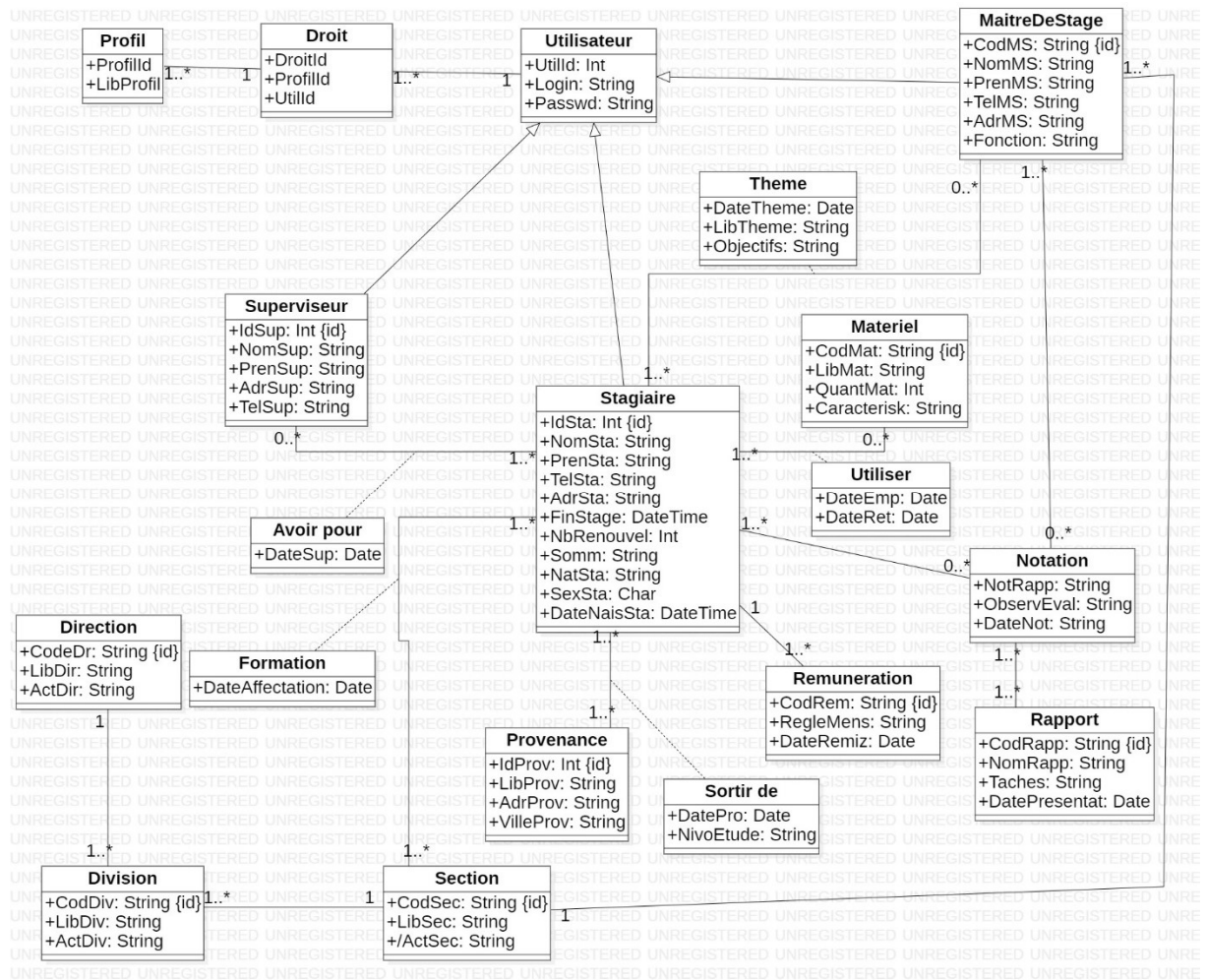


Figure 10 : Diagramme des classes du système

3 Réalisation et mise en œuvre

3.1 Matériels et logiciels utilisés

3.1.1 -Matériels

Tableau 3 : Matériels utilisés

Marque	Modèle	Processeur	Mémoire RAM	Disque dur	Système d'exploitation
HP	HP 250 G6 Notebook	Intel® Core™i3	4Go	256Go	Windows 10 Professionnel

3.1.2 Logiciels

- Microsoft Visual Studio 2019

C'est un EDI (Environnement de développement intégré) conçu par Microsoft et qui permet de développer des applications pour Android, iOS, Mac, Windows, le web et le cloud. Notre choix s'est porté sur ce logiciel parce qu'il intégrait déjà le .Net. Nous aurions pu utiliser également comme EDI, Visual Studio Code de la même famille que notre EDI sélectionné sauf qu'il nous aurait fallu intégrer des extensions nous permettant de réaliser notre solution.

- Google Chrome

C'est un navigateur web développé par Google.

- Microsoft Edge

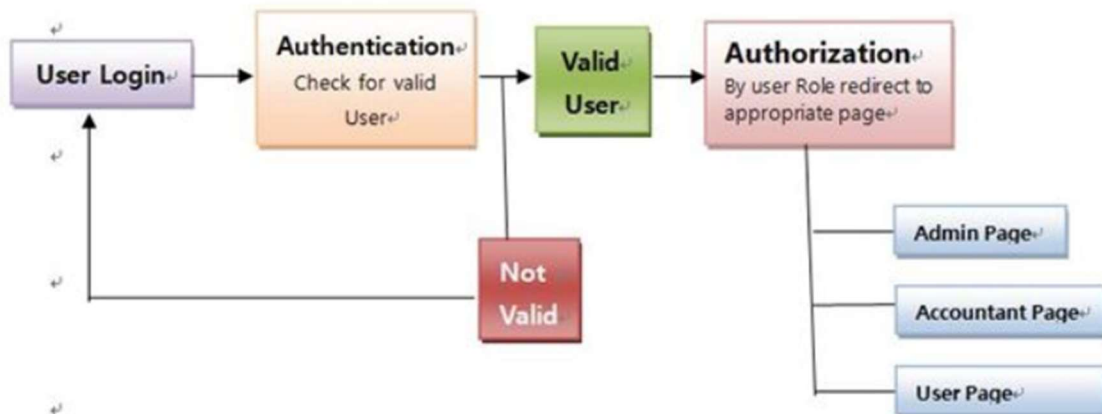
C'est un navigateur web développé par Microsoft.

- Microsoft SQL Server 2017

Est un SGBD (Système de Gestion de Base de Données).

3.2 Sécurité de l'application

La sécurité de l'application est gérée par deux filtres : un filtre d'authentification et un filtre d'autorisation. Un utilisateur possédant un compte utilisateur se connecte avec les identifiants de son compte et suivant le type d'autorisation qu'il a il est redirigé vers son tableau de bord.



Source : www.c-sharpcorner.com/uploadfile/shivprasadk/Asp-Net-authentication-and-authorization/

Figure 11 : Sécurité de l'application

3.3 Evaluation financière de la solution

➤ Coût matériel

Tableau 4 : Coût matériel

LIBELLE	PRIX UNITAIRE	QUANTITE	MONTANT	Source
Serveur de base de données	800.000	01	300.000	Mémoire de M. WOLF Kodjo Adelphe Daniel : « Gestion automatisée des commandes et des livraisons de sachets d'eau : cas de SHINE WATER » (2019-2020)
Ordinateur	300.000	01	300.000	
LIBELLE	PRIX UNITAIRE	QUANTITE	MONTANT	
Onduleur	58.385	02	116.770	
Imprimante	80.000	01	80.000	
TOTAL			796.770	

➤ Coût de la conception

Tableau 5 : Coût de la conception

FONCTION	PRIX/HEURE	NOMBRE DE DEVELOPPEURS	NOMBRE D'HEURES	COÛT TOTAL (FCFA)	Source
Développeur	4.000	01	360	1.440.400	Mémoire de M. WOLF Kodjo Adelphe Daniel : « Gestion automatisée des commandes et des livraisons de sachets d'eau : cas de SHINE WATER » (2019-2020)
TOTAL			1.440.000		

➤ **Coût total**

Tableau 6 : Coût total

Critères	Montant	Source
Coût d'acquisition/conception	1.440.000	Mémoire de M. WOLF Kodjo Adelphe Daniel : « Gestion automatisée des commandes et des livraisons de sachets d'eau : cas de SHINE WATER » (2019-2020)
Coût d'hébergement	18.500	
Coût de formation	25.000	
Coût de maintenance	20.000	
Coût du matériel obligatoire	796.770	
Coût total	2.300.270	

Après évaluation des différents coûts ; le coût total s'élève à : **2.300.270 FCFA**.

3.4 Présentation de l'application

3.4.1 Mise en place de la base de données

Script de création de la base de données :

USE [master]

GO

CREATE DATABASE [GesSta]

CONTAINMENT = NONE

ON PRIMARY

**(NAME = N'GesSta', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\GesSta.mdf', SIZE = 8192KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB)**

LOG ON

**(NAME = N'GesSta_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\GesSta_log.ldf' , SIZE =
8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB)**

WITH CATALOG_COLLATION = DATABASE_DEFAULT

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [GesSta].[dbo].[sp_fulltext_database] @action = 'enable'

end

GO

ALTER DATABASE [GesSta] SET ANSI_NULL_DEFAULT OFF

GO

ALTER DATABASE [GesSta] SET ANSI_NULLS OFF

GO

ALTER DATABASE [GesSta] SET ANSI_PADDING OFF

GO

ALTER DATABASE [GesSta] SET ANSI_WARNINGS OFF

GO

ALTER DATABASE [GesSta] SET ARITHABORT OFF

GO

ALTER DATABASE [GesSta] SET AUTO_CLOSE OFF

GO

ALTER DATABASE [GesSta] SET AUTO_SHRINK OFF

GO

ALTER DATABASE [GesSta] SET AUTO_UPDATE_STATISTICS ON

GO

ALTER DATABASE [GesSta] SET CURSOR_CLOSE_ON_COMMIT OFF

GO

ALTER DATABASE [GesSta] SET CURSOR_DEFAULT GLOBAL

GO

ALTER DATABASE [GesSta] SET CONCAT_NULL_YIELDS_NULL OFF

GO

ALTER DATABASE [GesSta] SET NUMERIC_ROUNDABORT OFF

GO

ALTER DATABASE [GesSta] SET QUOTED_IDENTIFIER OFF

GO

ALTER DATABASE [GesSta] SET RECURSIVE_TRIGGERS OFF

GO

ALTER DATABASE [GesSta] SET ENABLE_BROKER

GO

ALTER DATABASE [GesSta] SET AUTO_UPDATE_STATISTICS_ASYNC OFF

GO

ALTER DATABASE [GesSta] SET DATE_CORRELATION_OPTIMIZATION OFF

GO

ALTER DATABASE [GesSta] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [GesSta] SET ALLOW_SNAPSHOT_ISOLATION OFF

GO

ALTER DATABASE [GesSta] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [GesSta] SET READ_COMMITTED_SNAPSHOT ON

GO

ALTER DATABASE [GesSta] SET HONOR_BROKER_PRIORITY OFF

GO

ALTER DATABASE [GesSta] SET RECOVERY FULL

GO

ALTER DATABASE [GesSta] SET MULTI_USER

GO

ALTER DATABASE [GesSta] SET PAGE_VERIFY CHECKSUM

GO

ALTER DATABASE [GesSta] SET DB_CHAINING OFF

GO

***ALTER DATABASE [GesSta] SET FILESTREAM(NON_TRANSACTED_ACCESS
= OFF)***

GO

ALTER DATABASE [GesSta] SET TARGET_RECOVERY_TIME = 60 SECONDS

GO

ALTER DATABASE [GesSta] SET DELAYED_DURABILITY = DISABLED

G

***ALTER DATABASE [GesSta] SET ACCELERATED_DATABASE_RECOVERY =
OFF***

GO

ALTER DATABASE [GesSta] SET QUERY_STORE = OFF

GO

ALTER DATABASE [GesSta] SET READ_WRITE

GO

3.4.2 Plan de navigation

Onglet du profil « administrateur »

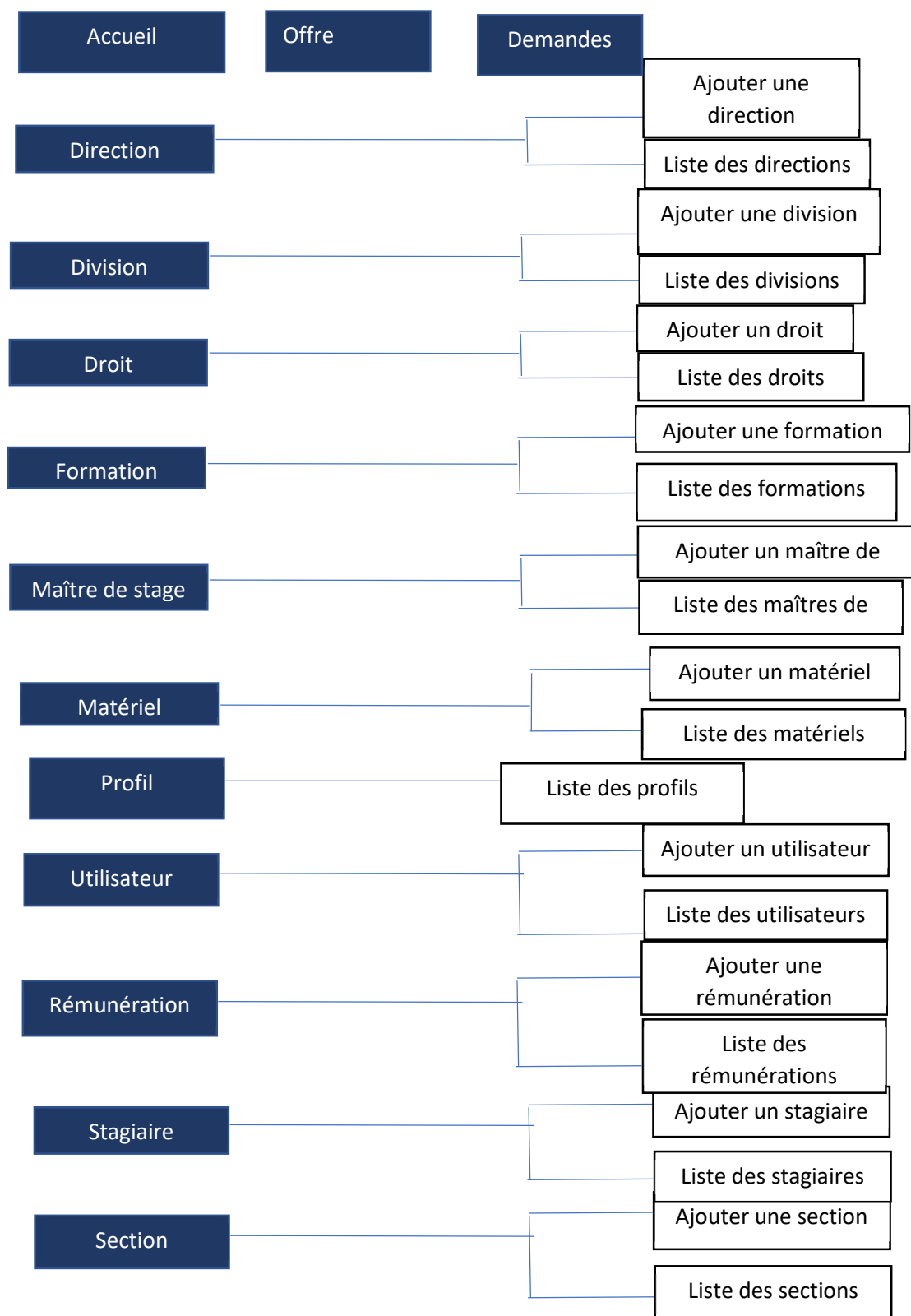


Figure 12 : Onglet du profil administrateur

3.4.3 Quelques masques de saisie

Page d'accueil

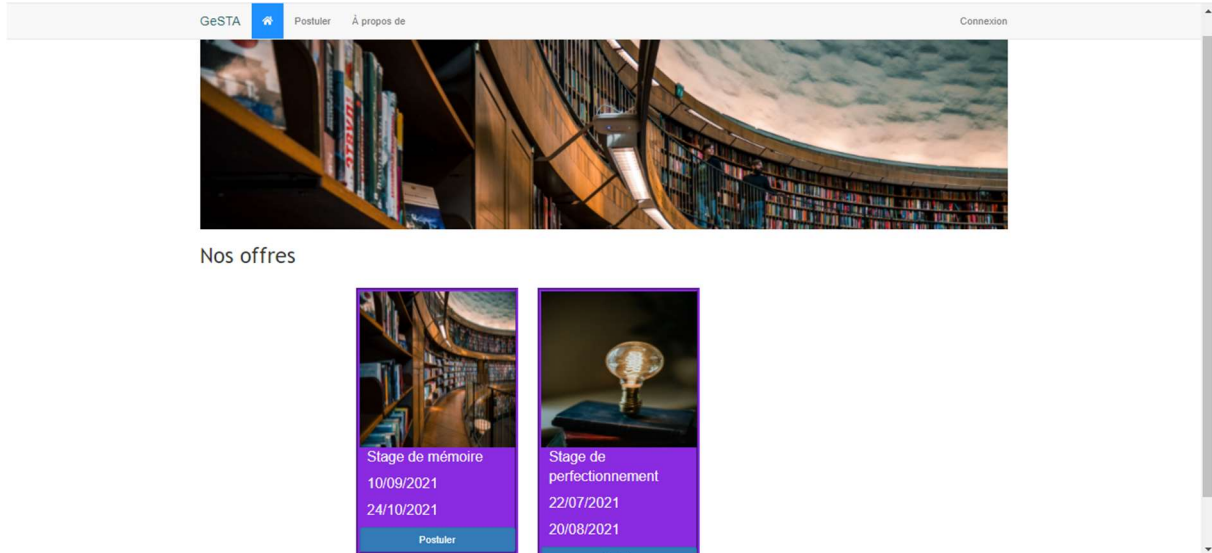


Figure 13 : page d'accueil de l'application

Code source de la page d'accueil

```
@model IEnumerable<GesStaDemo.Models.Entities.Offre>

@{
    ViewBag.Title = "Accueil";
    Layout = "~/Views/Home/_LayoutPage.cshtml";
    GesStaDbContext db = new GesStaDbContext();
}

<style>
    .container-fluid {
        width: 800px;
        height: 100%;
    }
    .carousel slide {
        height: 300px;
    }

    .item {
        width: 100%; height: 300px;
    }

    h2{
```

```

        font-
family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Ari
al, sans-serif;
    }
    .card-body {
        padding-left: 1px;
        height: 100%;
    }

    .card-img-top {
        width: 220px;
        height: 220px;
        background-position: center;
        background-repeat: no-repeat;
        background-size: cover;
    }

    .card-text {
        padding-left: 10px;
        color: whitesmoke;
        text-align: left;
        font-size: 20px;
    }

    }

    .card {
        margin: 20px;
        background-color: blueviolet;
        display: grid;
        border: 5px groove blueviolet;
        width: 229px;
        /*height: 400px;*/
        height:100%;
    }
</style>
<div class="carousel slide" data-ride="carousel">
    <div class="carousel-inner">
        <div class="item active">
            
        </div>

        <div class="item">
            
        </div>
        <div class="item">
            
        </div>
        <div class="item">

```

```

        
    </div>
    <div class="item">
        
    </div>
</div>

<h2>Nos offres</h2>
@using(db)
{
<div class="container-fluid">
    @foreach (var item in Model)
    {
        if (item.OffreId == "Stage de mémoire")
        {

            <div class="col-md-4">
                <div class="card">
                    
                    <div class="card-body">
                        <p class="card-text">
                            @item.OffreId
                        </p>
                        <p class="card-
text">@item.DebutStage.ToShortDateString()</p>
                        <p class="card-
text">@item.FinStage.ToShortDateString()</p>

                    </div>
                    @Html.ActionLink(
                        linkText: "Postuler",
                        actionName: "Create",
                        controllerName: "Demande",
                        routeValues: new { @id = item.Demandes },
                        htmlAttributes: new { @class = "btn btn-primary" }
                    )

                    <!--<a href="@Url.Action("Create", "Demande")" class="btn btn-
primary">Postuler</a-->
                </div>
            </div>

        }
        else
        {
            <div class="col-md-4">
                <div class="card">

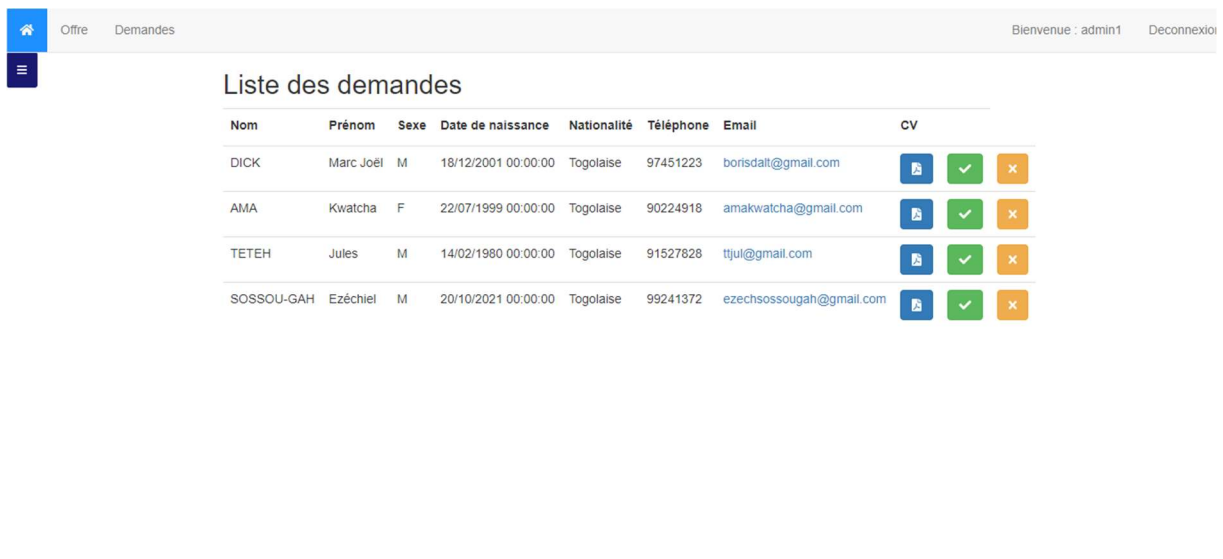
```

```

        
        <div class="card-body">
            <p class="card-text">
                @item.OffreId
            </p>
            <p class="card-
text">@item.DebutStage.ToShortDateString()</p>
            <p class="card-
text">@item.FinStage.ToShortDateString()</p>
        </div>
        @Html.ActionLink(
            linkText: "Postuler",
            actionName: "Create",
            controllerName: "Demande",
            routeValues: new { @id = item.Demandes },
            htmlAttributes: new { @class = "btn btn-primary" }
        )
        <!--<a href="@Url.Action("Create", "Demande")" class="btn btn-
primary" style="">Postuler</a-->
    </div>
}
}
</div>
}

```

La vue de l'administrateur montrant les demandes de stage



The screenshot shows a web application interface for an administrator. At the top, there is a navigation bar with a home icon, a menu icon, and links for 'Offre' and 'Demandes'. On the right, it says 'Bienvenue : admin1' and 'Deconnexion'. Below the navigation bar, the title 'Liste des demandes' is displayed. A table lists four internship requests with columns for Nom, Prénom, Sexe, Date de naissance, Nationalité, Téléphone, Email, and CV. Each row has three action buttons: a blue document icon, a green checkmark, and an orange X.













Nom	Prénom	Sexe	Date de naissance	Nationalité	Téléphone	Email	CV
DICK	Marc Joël	M	18/12/2001 00:00:00	Togolaise	97451223	borisdatt@gmail.com	  
AMA	Kwatcha	F	22/07/1999 00:00:00	Togolaise	90224918	amakwatcha@gmail.com	  
TETEH	Jules	M	14/02/1980 00:00:00	Togolaise	91527828	ttjul@gmail.com	  
SOSSOU-GAH	Ezéchiél	M	20/10/2021 00:00:00	Togolaise	99241372	ezechsossougah@gmail.com	  

Figure 14 : page de l'administrateur avec les demandes

Code source de la page

```
@model IEnumerable<GesStaDemo.Models.Entities.Demande>

@{
    ViewBag.Title = "Demandes";
    Layout = "~/Views/Admin/_AdminLayout.cshtml";
}

<link href="~/Content/admin.css" type="text/css"/>
<div id="main" class="main">

    <h2>Liste des demandes</h2>

    <table class="table">
        <tr>
            <th>
                @Html.DisplayName("Nom")
            </th>
            <th>
                @Html.DisplayName("Prénom")
            </th>
            <th>
                @Html.DisplayName("Sexe")
            </th>
            <th>
                @Html.DisplayName("Date de naissance")
```

```

        </th>
        <th>
            @Html.DisplayName("Nationalité")
        </th>
        <th>
            @Html.DisplayName("Téléphone")
        </th>
        <th>
            @Html.DisplayName("Email")
        </th>
        <th>
            @Html.Label("CV")
        </th>
    </th></th>
</tr>

@foreach (var item in Model)
{
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Nom)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Prenom)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Sexe)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.DateNaissance)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Nationalite)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Telephone)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Email)
        </td>
        <td>
            <span id="popuptext1" style="display:none;">Voir CV</span>
            <a href="@Url.Action("Voir","Admin",new { id = item.IdDem })"
class="btn btn-primary" onmouseover="myFunction()">
                <i class="fa fa-file-pdf"></i>
            </a>
        </td>
        <td>
            <span id="popuptext2" style="display:none;"> Accepter</span>

```

```

        <a href="@Url.Action("Valider", "Admin", new { id = item.IdDem
    })" class="btn btn-success" onmouseover="myFunction()">
        <i class="fa fa-check"></i>
    </a>
</td>
<td>
        <span id="popuptext3" style="display:none;">Refuser</span>
        <a href="@Url.Action("Refuser", "Admin", new { id = item.IdDem
    })" class="btn btn-warning" onmouseover="myFunction()">
        <i class="fa fa fa-times"></i>
    </a>
</td>
</tr>

    }

</table>
</div>
<!--<script>
    function myFunction() {
        var popup = document.getElementById("popuptext1");
        popup.classList.toggle("show");
        var popup1 = document.getElementById("popuptext2");
        popup1.classList.toggle("show");
        var popup3 = document.getElementById("popuptext3");
        popup3 = document.getElementById("show");
    }
</script>-->

```


La page d'authentification

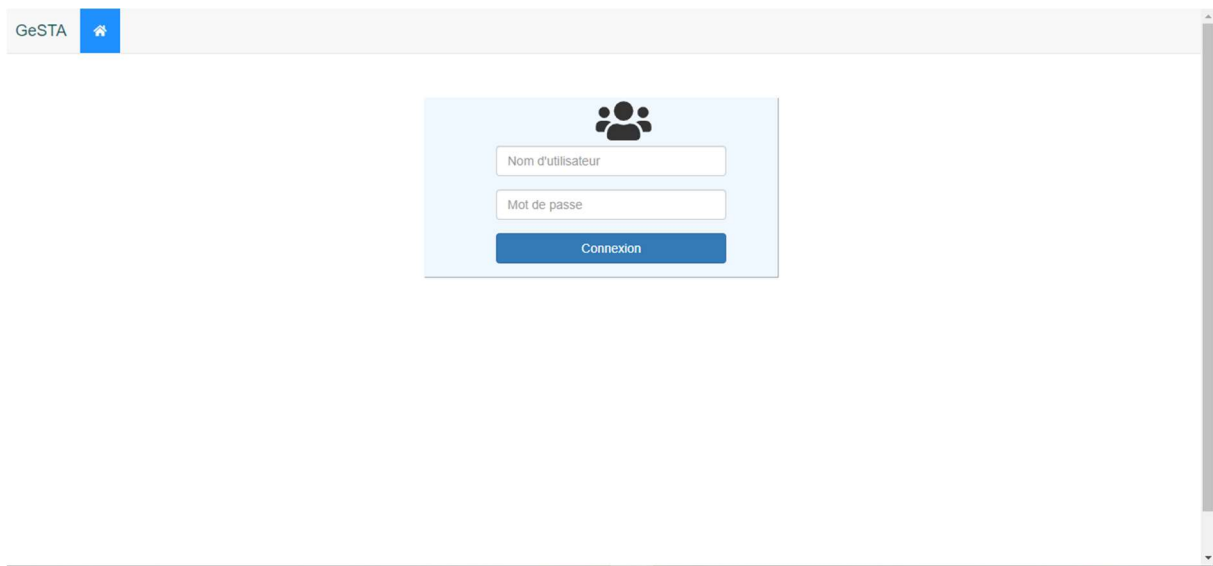


Figure 15 : page d'authentification

Code source de la page

```
@model GesStaDemo.Models.Entities.Utilisateur
@{
    ViewBag.Title = "Page de connexion";
    Layout = "~/Views/Login/_LoginLayout.cshtml";
}
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-
horizontal" style="height:100%; width:35%;position:center; pointer-
events:none; margin-top:50px; margin: auto;padding-left:50px; background-
color: aliceblue;border: 1px outset aliceblue;">

        <i class="fa fa-users" style="            width: 100%;
            position: center;
            height: 50px;
            margin: auto;
            "></i>

        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group"style="            height: 100%;
            width: 100%;
            position: center;
            ">

            <div class="col-md-10">
```

```

        <input type="text" name="Login" class="form-
control" placeholder="Nom d'utilisateur" style="width: 100%;margin-left: 30px;" />
        @Html.ValidationMessageFor(model => model.Login, "", new { @class = "t
ext-danger" })
    </div>
</div>

<div class="form-group" style="          height: 100%;
width: 100%;
position: center;
">
    <div class="col-md-10">
        <input type="password" name="Passwd" class="form-
control" placeholder="Mot de passe" style="width:100%; margin-left:30px;" />
        @Html.ValidationMessageFor(model => model.Passwd, "", new { @class = "
text-danger" })
    </div>
</div>

<div class="form-group" style="          height: 100%;
width: 100%;
position: center;
">
    <div class="col-md-10">
        <input type="submit" value="Connexion" class="btn btn-
primary" style="width: 100%;margin-left: 30px;" />
    </div>
</div>
</div>
}

```

Page de demande de stage

GeSTA | Postuler | À propos de | Connexion

Demande

Nom:

Prénom:

Sexe: ☐ F ☐ M ☐ O

Date de naissance:

Nationalité:

Numero de téléphone:

Email:

Type de stage:

CV: Aucun fichi...sélectionné

© 2021 - GeSTA

Figure 16 : page de demande de stage

Code source de la page

```
@model GesStaDemo.Models.Entities.Demande

@{
    var db = new GesStaDbContext();
    ViewBag.TypeDeStage = new SelectList(db.Offres, "OffreId", "OffreId");
    ViewBag.Title = "Dépôt de demande";
    Layout = "~/Views/Home/_LayoutPage.cshtml";
}

<h2>Demande</h2>

@using (Html.BeginForm("Create", "Demande", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
```

```

        @Html.LabelFor(model => model.Nom, htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Nom, new { htmlAttributes = new { @class = "form-control", placeholder = "AMA" } })
            @Html.ValidationMessageFor(model => model.Nom, "", new { @class = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.Label("Prénom", htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Prenom, new { htmlAttributes = new { @class = "form-control", placeholder = "Kwatcha" } })
            @Html.ValidationMessageFor(model => model.Prenom, "", new { @class = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Sexe, htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            <div class="col-md-2">
                @Html.Label("F", htmlAttributes: new { @class = "checkbox-label" })
                <input type="radio" name="sexe" value="F" />
            </div>
            @Html.Label("M", htmlAttributes: new { @class = "checkbox-label" })
            <input type="radio" name="sexe" value="M" />
        </div>
    </div>

    <div class="form-group">
        @Html.Label("Date de naissance", htmlAttributes: new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.DateNaissance, new { htmlAttributes = new { @class = "form-control", type = "date" } })
            @Html.ValidationMessageFor(model => model.DateNaissance, "", new { @class = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.Label("Nationalité", htmlAttributes: new { @class = "control-label col-md-2" })

```

```

        <div class="col-md-10">
            @Html.EditorFor(model => model.Nationalite, new { htmlAttributes = new
            { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Nationalite, "", new { @clas
            s = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.Label("Numero de téléphone", htmlAttributes: new { @class = "control
        -label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Telephone, new { htmlAttributes = new {
            @class = "form-control", type = "tel", placeholder = "90280728" } })
            @Html.ValidationMessageFor(model => model.Telephone, "", new { @class
            = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "contr
        ol-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Email, new { htmlAttributes = new { @cl
            ass = "form-control", type = "email", placeholder = "amakwatcha@gmail.com" } })
            @Html.ValidationMessageFor(model => model.Email, "", new { @class = "t
            ext-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.TypeDeStage, "Type de stage", htmlAttributes
        : new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("TypeDeStage", null, htmlAttributes: new { @class =
            "form-control" })
            @Html.ValidationMessageFor(model => model.TypeDeStage, "", new { @clas
            s = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Curriculum, htmlAttributes: new { @class = "
        control-label col-md-2" })
        <div class="col-md-10">
            <input type="file" name="CV" accept=".pdf" />
            @Html.ValidationMessage("Le CV doit être un pdf", new { @class = "text
            -danger" })
        </div>
    </div>

```

```

<div class="form-group" hidden>
    @Html.LabelFor(model => model.Accepter, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10" hidden>
        <div class="checkbox">
            @Html.EditorFor(model => model.Accepter)
            @Html.ValidationMessageFor(model => model.Accepter, "", new { @class = "text-danger" })
        </div>
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Valider" onclick="" class="btn btn-primary" />
        <input type="reset" value="Annuler" class="btn btn-default" />
    </div>
</div>
</div>
}

```

3.4.4 Quelques états et statistiques

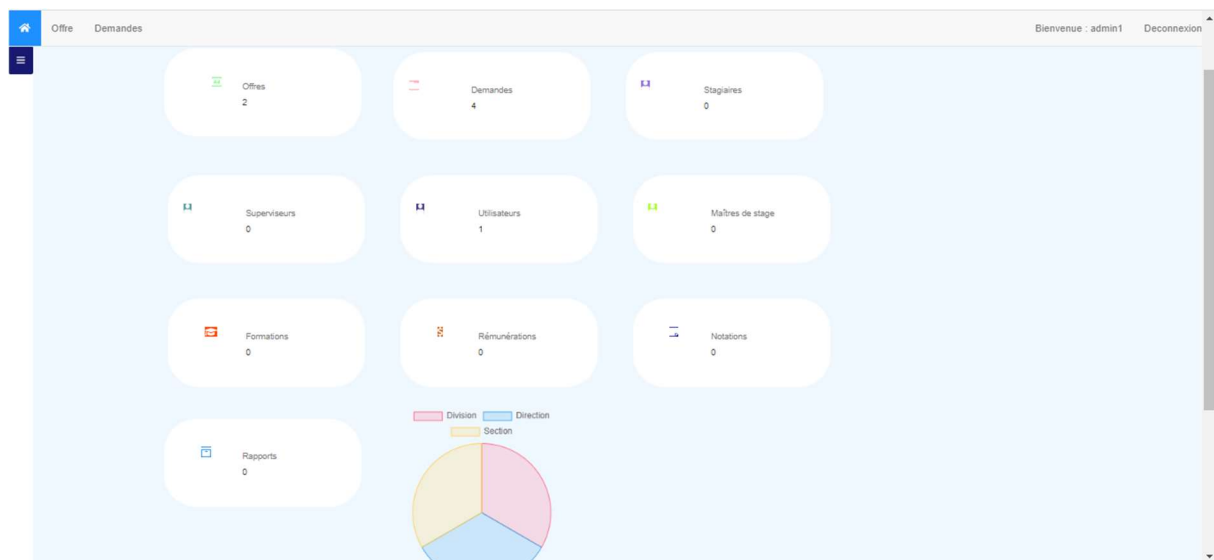


Figure 17 : la page de l'administrateur avec les statistiques

15/09/2021							
<u>Nom</u>	<u>Prenom</u>	<u>Sex</u>	<u>DateNaissance</u>	<u>Nationalite</u>	<u>Telephon</u>	<u>Email</u>	<u>TypeDeStage</u>
DICK	Marc Joël	M	18/12/2001 00:	Togolaise	9745122:	borisdalt@gmail.com	Stage de mémoire
SOSSOU-GAH	Ezéchiél	M	20/10/2021 00:	Togolaise	9924137:	ezechsossougah@gmail.com	Stage de perfectionnement
AMA	Kwatcha	F	22/07/1999 00:	Togolaise	9022491:	amakwatcha@gmail.com	Stage de mémoire
TETEH	Jules	M	14/02/1980 00:	Togolaise	9152782:	ttjul@gmail.com	Stage de mémoire
EDOH	Komi	M	19/03/2000 00:	Togolaise	9234105:	komiedoh@gmail.com	Stage de mémoire

Figure 18 : état présentant la liste des demandes de stage

15/09/2021

<u>Nom</u>	<u>Prenom</u>	<u>Sex</u>	<u>DateNaissance</u>	<u>Nationalite</u>	<u>Telephon</u>	<u>Email</u>	<u>TypeDeStage</u>
DICK	Marc Joël	M	18/12/2001 00:00	Togolaise	9745122	borisdalt@gmail.com	Stage de mémoire
EDOH	Komi	M	19/03/2000 00:00	Togolaise	9234105	komiedoh@gmail.com	Stage de mémoire

Figure 19 : état présentant la liste des demandes de stage acceptées

16/09/2021

DCom
DirG
DTech

Direction commerciale
Direction Générale
Direction technique

Gestion commerciale
Administration générale
Gestion technique

Figure 20 : état présentant la liste des directions

-

16/09/2021

DvMar
DvTec
R&D

Division Marketing
Division technique
Recherche et développement

Gestion de la stratégie marketing
Etablis les besoins du SI
Veille informationnelle

Figure 21 : état présentant la liste des divisions

Conclusion

Ce stage nous aura permis de travailler de A à Z sur un projet avec un document scientifique à l'appui. C'est un travail qui nous a obligé à nous conformer à un canevas préétabli, tout en nous donnant la largesse de pouvoir organiser notre travail comme bon nous le semblait tout en ayant également une rigueur vis à vis de notre planning prévisionnel. Nous avons intégré au fur et à mesure les apports de notre superviseur pour pouvoir atteindre nos différents objectifs, ainsi nous avons pu implémenter les trois différentes fonctionnalités de l'application, et ajouter de nouvelles. Grâce à ce stage, nous avons appris à nous documenter, à faire des recherches efficaces en vue de la prise de décision pour l'avancement d'un projet ce qui nous a permis d'être dans les temps.

Bibliographie indicative

**UML 2 en action « De l'analyse des besoins à la conception » Pascal Roques, Franck Vallée
Cours d'UML dispensé par M. KETOGLO, 2020-2021**

Documents annexes

Mémoire de Mlle NIKABOU Siba Fidèle : « Plateforme web d'achat et de produits en ligne » (2017-2018)

**Mémoire de M. KPAKPASSIM Esso-wédéou Nikabou Barthélémy :
« Etude et mise en œuvre d'une application de gestion électronique de documents : module d'archivage électronique des contrats et courriers » (2018-2019)**

Mémoire de M. WOLF Kodjo Adelphe Daniel : « Gestion automatisée des commandes et des livraisons de sachets d'eau : cas de SHINE WATER » (2019-2020)

Mémoire de M. OGOU Muriel Stéphane : « Développement d'application web de vente en ligne et de livraison » (2016-2017)

Mémoire de M. AWI Assima : « Plateforme de commande de produits agricoles en ligne » (2016-2017)

Mémoire de M. AGOTSI Komi Gédéon Cyrille : « Mise en place d'un système de sauvegarde et restauration de base de données » (2016-2017)

Mémoire de M. WOAGOU Yendouboamé Aimé : « Gestion des concours de bourse et des séances d'orientation » (2019-2020)

Webographie indicative

<https://www.c-sharpcorner.com/> (tout au long du projet)

<https://docs.microsoft.com/en-us/dotnet/> (tout au long du projet)

<https://www.codeproject.com/> (tout au long du projet)

<https://canvasjs.com/> (tout au long du projet)

<https://docs.gleamtech.com/> (tout au long du projet)

<https://www.chartjs.org/> (tout au long du projet)

<https://askcodez.com/> (tout au long du projet)

<https://www.w3schools.com/> (tout au long du projet)

<https://stackoverflow.com/> (tout au long du projet)

<https://www.it-swarm-fr.com/> (tout au long du projet)

<https://github.com/lukencode/FluentEmail/> (tout au long du projet)

<https://www.geeksforgeeks.org/> (tout au long du projet)

<https://getbootstrap.com/docs/4.5/> (tout au long du projet)

<https://www.codegrepper.com/> (tout au long du projet)

Table des matières

REMERCIEMENTS.....	I
SOMMAIRE	II
LISTE DES FIGURES.....	III
LISTE DES TABLEAUX.....	IV
INTRODUCTION	5
1 CAHIER DES CHARGES.....	6
1.1 Présentation du sujet	6
1.2 Problématique du sujet	6
1.3 Intérêt du sujet	6
1.3.1 Objectifs.....	6
1.3.2 Résultats	6
2 RAPPORT DE PREPROGRAMMATION.....	7
2.1 Etude de l'existant	7
2.2 Critique de l'existant	7
2.3 Planning prévisionnel de réalisation	7
2.4 Etude détaillée de la solution	9
2.4.1 Analyse et conception	9
2.4.1.1 Présentation de la méthode d'analyse	9
2.4.2 Implémentation	13
3 REALISATION ET MISE EN ŒUVRE	32
3.1 Matériels et logiciels utilisés	32
3.1.1 Matériels.....	32
3.1.2 Logiciels	32
3.2 Sécurité de l'application	33
3.3 Evaluation financière de la solution	34
3.4 Présentation de l'application	36
3.4.1 Mise en place de la base de données	36
3.4.2 Plan de navigation	40
3.4.3 Quelques masques de saisie.....	41
3.4.4 Quelques états et statistiques	54
CONCLUSION	59
BIBLIOGRAPHIE INDICATIVE	60
DOCUMENTS ANNEXES.....	60
WEBOGRAPHIE INDICATIVE.....	61
TABLE DES MATIERES.....	62